Advancing Differentially Private Optimization: Efficiency, Utility, and Applications

Daogao Liu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Yin Tat Lee, Chair

Simon Du

Jerry Li

Yen-Chi Chen

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Advancing Differentially Private Optimization:
Efficiency, Utility, and Applications

Daogao Liu

Chair of the Supervisory Committee:
Yin Tat Lee
Computer Science & Engineering

With the rapid development and widespread application of modern machine learning and artificial intelligence—particularly following the emergence of large language models—privacy has become a critical concern. Differential privacy, a rigorous mathematical framework for defining privacy, has emerged as the de facto standard.

This thesis addresses two fundamental problems in privacy-preserving machine learning: differentially private empirical risk minimization (DP-ERM) and differentially private stochastic (convex) optimization (DP-SCO). Our goal is to design more efficient algorithms for these problems while achieving better and optimal privacy-utility trade-offs.

The thesis is structured into five parts:

- Part I focuses on improving and achieving near-optimal gradient or function value computation complexity.

- Part II extends the analysis under alternative geometries and norms beyond the classic Euclidean spaces.

- Part III investigates non-convex functions, which are increasingly common in practice and gaining significant attention.

- Part IV examines the user-level differential privacy setting, a practical scenario where

users contribute multiple items, as opposed to the classical item-level DP assumption of a single item per user.

- Part V explores additional settings, including online optimization, heavy-tailed distributions, and low-rank structures.

This work comprehensively explores these challenges, proposing innovative methods to enhance algorithmic efficiency and optimize the privacy-utility trade-off.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

ACKNOWLEDGMENTS

When I began writing this, I suddenly realized how quickly over four years had passed. Many people imagine that pursuing a PhD is difficult and stressful, but looking back, it was not as challenging as expected—perhaps I could have even pushed myself harder. This relatively smooth journey would not have been possible without the support of my advisor, Yin Tat Lee. Yin Tat is incredibly quick-witted, knowledgeable, and patient when addressing even the simplest of questions. What truly stands out is his unwavering enthusiasm and curiosity for research. One of his most memorable pieces of advice was to choose projects that genuinely excited me—and to "trick" my mind into believing I could tackle them. He also generously provided financial support for attending various academic conferences and events.

I am grateful to my committee members, Simon Du, Yen-Chi Chen, Jerry Li, and Sewoong Oh, for their support and guidance since my qualifying exam. Over the years, the theory group has created a rich academic environment through activities such as theory lunches and seminars, as well as hosting many enjoyable and memorable events.

I would also like to thank Janardhan Kulkarni for his guidance during my PhD, particularly in improving my academic writing and identifying promising research directions and problems. Additionally, I spent two wonderful summers at Apple MLR working with Kunal Talwar, Vitaly Feldman, and Hilal Asi, where we often sat on the couch, discussing problems at length in front of a whiteboard. Their curiosity-driven approach and willingness to think deeply about problems for extended periods left a lasting impression on me.

I want to express my gratitude to all the collaborators and academic friends I worked with during my PhD: Hilal Asi, Gavin Brown, Yair Carmon, Lynn Chua, Krishnamurthy Dvijotham, Georgina Evans, Hu Fu, Badih Ghazi, Sivakanth Gopi, Yangsibo Huang, Arun Jambulapati, Haotian Jiang, Yaonan Jin, Yujia Jin, Pritish Kamath, Tomer Koren, Guy

Kornowski, Janardhan Kulkarni, Yin Tat Lee, Jian Li, Jiawei Li, Xuechen Li, Andrew Lowy, Jiuyao Lu, Zhou Lu, Pasin Manurangsi, Ruoqi Shen, Weijia Shi, Aaron Sidford, Sahil Singla, Adam Smith, Zhao Song, Kunal Talwar, Abhradeep Guha Thakurta, Kevin Tian, Menthou Xia, Ziqi Wang, and Chiyuan Zhang. Although I have not collaborated with some of them for a while, recalling our time together brings back many joyful memories. They not only imparted invaluable lessons but also offered steadfast support throughout my journey.

I would like to express my deepest gratitude to all my friends and colleagues who have made these experiences unforgettable: Yingkang Cao, Jinhai Chen, Tuochao Chen, Xin Chen, Yifang Chen, Qixin Feng, Chenguang Guan, Xiaomeng Hu, Linxing Jiang, Shanshan Li, Jialiang Liu, Qinghua Liu, Zongnan Liu, Sirui Lu, Xin Lyu, Jingwei Ma, Yijiao Qin, Vinod Raman, Skyler Seto, Mengyi Shan, Xia Su, Jiyu Tao, Ke Wang, Chulin Xie, Haoyu Xiong, Zhihan Xiong, Guanqun Yang, Yuanyuan Yang, Jiayuan Ye, Feng Zeng, Dinghua Zhang, Junyang Zhang, Maochuan Zhang, Chenxingyu Zhao, Runlong Zhou, and Zidong Zhou. The countless moments we shared—lifting weights in the gym, snowboarding, hiking, savoring delicious meals, and playing video and board games—will forever remain cherished memories. I would also like to thank ChatGPT, whose assistance greatly facilitated the writing process for this thesis.

Finally, I would like to express my deepest gratitude to my family, who have supported me throughout my entire life. My grandfather, who passed away this year, was a hardworking and humble man. During my senior year of high school, he cared for me in an unfamiliar place, enduring boredom and loneliness so that I could focus on my studies. I will forever cherish his memory. I would also like to thank my partner, Hua. It is a wonderful thing to acknowledge you in both my undergraduate and doctoral dissertations. Despite we had frequent arguments during this PhD journey, your presence and support have meant a great deal to me.

INTRODUCTION

Privacy has become an important consideration for learning algorithms dealing with sensitive data. Over the past decade, differential privacy, introduced in the seminal work of [DMNS06], has established itself as the defacto notion of privacy for machine learning problems. Formally, we say a randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if for any event $\mathcal{O} \in \text{Range}(\mathcal{M})$ and for any neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, we have

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta.$$

The Empirical Risk Minimization (ERM) and Stochastic (Convex) Optimization (SCO) problems are the most important and straightforward problems in statistics and machine learning. We study these problems in the DP settings. In the ERM problem, we are given a family of functions $\{f(\cdot; z\}_{z \in [\mathcal{D}]}$, where $\mathcal{D} = \{z_i, \cdots, z_n\}$ is a dataset drawn from some unknown distribution $\mathcal{P}$ and the objective is to

$$\min F_{\mathcal{D}}(x) := \frac{1}{n} \sum_{z \in \mathcal{D}} f(x; z),$$

and the objective of SCO is to

$$\min F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{z \sim \mathcal{P}} f(x; z).$$

We study these problems thoroughly, focusing on improving algorithmic efficiency and enhancing the privacy-utility trade-off. This includes exploring different geometric settings and relaxing the convexity assumption. Below, we briefly discuss the results of each chapter. Each chapter is self-contained and can be read in any order.

Chapter 1: Under appropriate smoothness assumption, the seminar work [FKT20] shows one can solve DP-SCO optimally with only n gradient queries. However, despite the consid-

erable progress in the DP and optimization communities, achieving optimal excess population loss for DP-SCO still requires $O(n^2)$ gradient queries. Indeed, [BFGT20] writes that " Proving that quadratic running time is necessary for general non-smooth DP-SCO is a very interesting open problem...". In Chapter 1, we get a (nearly) optimal bound on the excess empirical risk and excess population loss with subquadratic gradient complexity. More precisely, our differentially private algorithm requires $O(\frac{n^{3/2}}{d^{1/8}} + \frac{n^2}{d})$ gradient queries for optimal excess empirical risk, which is achieved with the help of subsampling and smoothing the function via convolution. This is the first subquadratic algorithm for the non-smooth case when d is super constant. As a direct application, using the iterative localization approach of Feldman et al. [FKT20], we achieve the optimal excess population loss for the stochastic convex optimization problem, with $O(\min\{n^{5/4}d^{1/8}, \frac{n^{3/2}}{d^{1/8}}\})$ gradient queries. Our work progresses towards resolving a question raised by Bassily et al. [BFGT20], giving the first algorithms for private ERM and SCO with subquadratic steps.

Chapter 2: We introduce a new tool for SCO: a Reweighted Stochastic Query (ReSQue) estimator for the gradient of a function convolved with a (Gaussian) probability density. Combining ReSQue with recent advances in *ball oracle acceleration* [CJJ+20, ACJ+21], we develop algorithms achieving state-of-the-art complexities for SCO in parallel and private settings. We obtain the following results for a SCO objective constrained to the unit ball in $\mathbb{R}^d$ (up to polylogarithmic factors).

Given n samples of Lipschitz loss functions, prior works [BFTGT19, BFGT20, AFKT21, KLL21] established that if $n \gtrsim d\varepsilon_{\text{dp}}^{-2}$, $(\varepsilon_{\text{dp}}, \delta)$-differential privacy is attained at no asymptotic cost to the SCO utility. However, these prior works all required a superlinear number of gradient queries. We close this gap for sufficiently large $n \gtrsim d^2\varepsilon_{\text{dp}}^{-3}$, by using ReSQue to design an algorithm with near-linear gradient query complexity in this regime.

Moreover, based on the same framework, we give a parallel algorithm obtaining optimization error ε_{opt} with $d^{1/3}\varepsilon_{\text{opt}}^{-2/3}$ gradient oracle query depth and $d^{1/3}\varepsilon_{\text{opt}}^{-2/3} + \varepsilon_{\text{opt}}^{-2}$ gradient queries in total, assuming access to a bounded-variance stochastic gradient estimator. For $\varepsilon_{\text{opt}} \in [d^{-1}, d^{-1/4}]$, our algorithm matches the state-of-the-art oracle depth of [BJL+19] while maintaining the optimal total work of stochastic gradient descent.

Chapter 3: We show that modifying the exponential mechanism by adding an ℓ_2^2 regularizer to $F(x)$ and sampling from $\pi(x) \propto \exp(-k(F(x) + \mu\|x\|_2^2/2))$ recovers both the known optimal empirical risk and population loss under (ε, δ)-DP. Furthermore, we show how to implement this mechanism using $\widetilde{O}(n \min(d, n))$ queries to $f_i(x)$ for the DP-SCO where n is the number of samples/users and d is the ambient dimension. We also give a (nearly) matching lower bound $\widetilde{\Omega}(n \min(d, n))$ on the number of evaluation queries.

Our results utilize the following tools that are of independent interest:

- We prove Gaussian Differential Privacy (GDP) of the exponential mechanism if the loss function is strongly convex and the perturbation is Lipschitz. Our privacy bound is *optimal* as it includes the privacy of the Gaussian mechanism as a special case and is proved using the isoperimetric inequality for strongly log-concave measures.

- We show how to sample from $\exp(-F(x) - \mu\|x\|_2^2/2)$ for G-Lipschitz F with η error in total variation (TV) distance using $\widetilde{O}((G^2/\mu)\log^2(d/\eta))$ unbiased queries to $F(x)$. This is the first sampler whose query complexity has *polylogarithmic dependence* on both dimension d and accuracy η.

Chapter 4: We propose a new framework for differentially private optimization of convex functions which are Lipschitz in an arbitrary norm $\|\cdot\|_{\mathcal{X}}$. Our algorithms are based on a regularized exponential mechanism which samples from the density $\propto \exp(-k(F + \mu r))$ where F is the empirical loss, and r is a regularizer which is strongly convex with respect to $\|\cdot\|_{\mathcal{X}}$, generalizing a recent work of [GLL22] to non-Euclidean settings. We show that this mechanism satisfies Gaussian differential privacy and solves both DP-ERM (empirical risk minimization) and DP-SCO (stochastic convex optimization) by using localization tools from convex geometry. Our framework is the first to apply to private convex optimization in general normed spaces and directly recovers non-private SCO rates achieved by mirror descent as the privacy parameter $\varepsilon \to \infty$. As applications, for Lipschitz optimization in ℓ_p norms for all $p \in (1, 2)$, we obtain the first optimal privacy-utility tradeoffs; for $p = 1$, we improve tradeoffs obtained by the recent works [AFKT21, BGN21] by at least a logarithmic

factor. Our ℓ_p norm and Schatten-p norm optimization frameworks are complemented with polynomial-time samplers whose query complexity we explicitly bound.

Chapter 5: The development of efficient sampling algorithms catering to non-Euclidean geometries has been a challenging endeavor, as discretization techniques that succeed in the Euclidean setting do not readily carry over to more general settings. We develop a non-Euclidean analog of the recent proximal sampler of [LST21b], which naturally induces regularization by an object known as the log-Laplace transform (LLT) of a density. We prove new mathematical properties (with an algorithmic flavor) of the LLT, such as strong convexity-smoothness duality and an isoperimetric inequality, which are used to prove a mixing time on our proximal sampler matching [LST21b] under a warm start. As our main application, we show our warm-started sampler improves the value oracle complexity of differentially private convex optimization in ℓ_p and Schatten-p norms for $p \in [1, 2]$ to match the Euclidean setting [GLL22], while retaining state-of-the-art excess risk bounds [GLL+23]. Our investigation of the LLT is a promising proof-of-concept of its utility as a tool for designing samplers. We outline directions for future exploration.

Chapter 6: We consider the problem of minimizing a non-convex objective while preserving the privacy of the examples in the training data. Building upon the previous variance-reduced algorithm SpiderBoost, we introduce a new framework that utilizes two different kinds of gradient oracles. The first kind of oracles can estimate the gradient of one point, and the second kind of oracles, less precise and more cost-effective, can estimate the gradient difference between two points. SpiderBoost uses the first kind periodically, once every few steps, while our framework proposes using the first oracle whenever the total drift has become large and relies on the second oracle otherwise. This new framework ensures the gradient estimations remain accurate all the time, resulting in improved rates for finding second-order stationary points.

Moreover, we address a more challenging task of finding the global minima of a non-convex objective using the exponential mechanism. Our findings indicate that the regularized exponential mechanism can closely match previous empirical and population risk

bounds, without requiring smoothness assumptions for algorithms with polynomial running time. Furthermore, by disregarding running time considerations, we show that the exponential mechanism can achieve a good population risk bound and provide a nearly matching lower bound.

Chapter 7: There is a gap between finding a first-order stationary point (FOSP) and a second-order stationary point (SOSP) under differential privacy constraints, and it remains unclear whether privately finding an SOSP is more challenging than finding an FOSP. Specifically, Ganesh et al. (2023) demonstrated that an α-SOSP can be found with $\alpha = \tilde{O}(\frac{1}{n^{1/3}} + (\frac{\sqrt{d}}{n\varepsilon})^{3/7})$, where n is the dataset size, d is the dimension, and ε is the differential privacy parameter. Building on the SpiderBoost algorithm framework, we propose a new approach that uses adaptive batch sizes and incorporates the binary tree mechanism. Our method improves the results for privately finding an SOSP, achieving $\alpha = \tilde{O}(\frac{1}{n^{1/3}} + (\frac{\sqrt{d}}{n\varepsilon})^{1/2})$. This improved bound matches the state-of-the-art for finding an FOSP, suggesting that privately finding an SOSP may be achievable at no additional cost.

Chapter 8: We study differentially private (DP) optimization algorithms for stochastic and empirical objectives, which are neither smooth nor convex, and propose methods that return a Goldstein-stationary point with sample complexity bounds that improve on existing works. We start by providing a single-pass (ε, δ)-DP algorithm that returns an (α, β)-stationary point as long as the dataset is of size $\widetilde{\Omega}\left(1/\alpha\beta^3 + d/\varepsilon\alpha\beta^2 + d^{3/4}/\varepsilon^{1/2}\alpha\beta^{5/2}\right)$, which is $\Omega(\sqrt{d})$ times smaller than the algorithm of [ZTC24] for this task, where d is the dimension. We then provide a multi-pass polynomial time algorithm which further improves the sample complexity to $\widetilde{\Omega}\left(d/\beta^2 + d^{3/4}/\varepsilon\alpha^{1/2}\beta^{3/2}\right)$, by designing a sample-efficient ERM algorithm, and prove that Goldstein's Stationary points generalize from empirical loss to population loss.

Chapter 9: We study differentially private stochastic convex optimization (DP-SCO) under user-level privacy, where each user may hold multiple data items. Existing work for user-level DP-SCO either requires super-polynomial runtime [GKK+23b] or requires

the number of users to grow polynomially with the dimensionality of the problem with additional strict assumptions [BS23]. We develop new algorithms for user-level DP-SCO that obtain optimal rates for both convex and strongly convex functions in polynomial time and require the number of users to grow only logarithmically in the dimension. Moreover, our algorithms are the first to obtain optimal rates for non-smooth functions in polynomial time. These algorithms are based on multiple-pass DP-SGD, combined with a novel private mean estimation procedure for concentrated data, which applies an outlier removal step before estimating the mean of the gradients.

Chapter 10: We study private stochastic convex optimization (SCO) under the constraint of user-level differential privacy (DP). In this setting, there are n users, each possessing m data items, and we need to protect the privacy of each user's entire collection of data items. Existing algorithms for user-level DP SCO are impractical in many large-scale machine learning scenarios because: (i) they make restrictive assumptions on the smoothness parameter of the loss function and require the number of users to grow polynomially with the dimension of the parameter space; or (ii) they are prohibitively slow, requiring at least $(mn)^{3/2}$ gradient computations for smooth losses and $(mn)^3$ computations for non-smooth losses. To address these limitations, we provide novel user-level DP algorithms with state-of-the-art excess risk and runtime guarantees, without the stringent assumptions. First, we develop a *linear-time* algorithm with state-of-the-art excess risk (for a linear-time algorithm) under a mild smoothness assumption. Our second algorithm achieves *optimal excess risk* in $(mn)^{9/8}$ gradient computations under a mild smoothness assumption. Third, for *non-smooth* loss functions, we obtain *optimal excess risk* in $(mn)^{11/8}$ gradient computations. Our algorithms do not require the number of users to grow polynomially with the dimension.

Chapter 11: Large pretrained models can be fine-tuned with differential privacy to achieve performance approaching that of non-private models. A common theme in these results is the surprising observation that high-dimensional models can achieve favorable privacy-utility trade-offs. This seemingly contradicts known results on the model-size de-

pendence of differentially private convex learning and raises the following research question: When does the performance of differentially private learning not degrade with increasing model size? We identify that the magnitudes of gradients projected onto subspaces is a key factor that determines performance. To precisely characterize this for private convex learning, we introduce a condition on the objective that we term *restricted Lipschitz continuity* and derive improved bounds for the excess empirical and population risks that are dimension-independent under additional conditions. We empirically show that in private fine-tuning of large language models, gradients obtained during fine-tuning are mostly controlled by a few principal components. This behavior is similar to conditions under which we obtain dimension-independent bounds in convex settings. Our theoretical and empirical results together provide a possible explanation for the recent success of large-scale private fine-tuning.

Chapter 12: We study the problem of differentially private stochastic convex optimization (DP-SCO) with heavy-tailed gradients, where we assume a k^{th}-moment bound on the Lipschitz constants of sample functions, rather than a uniform bound. We propose a new reduction-based approach that enables us to obtain the first optimal rates (up to logarithmic factors) in the heavy-tailed setting, achieving error $G_2 \cdot \frac{1}{\sqrt{n}} + G_k \cdot (\frac{\sqrt{d}}{n\varepsilon})^{1-\frac{1}{k}}$ under (ε, δ)-approximate differential privacy, up to a mild polylog($\frac{\log n}{\delta}$) factor, where G_2^2 and G_k^k are the 2^{nd} and k^{th} moment bounds on sample Lipschitz constants, nearly-matching a lower bound of [LR23].

We then give a suite of private algorithms in the heavy-tailed setting, which improve upon our basic result under additional assumptions, including an optimal algorithm under a known Lipschitz constant assumption, a near-linear time algorithm for smooth functions, and an optimal linear time algorithm for smooth generalized linear models.

Chapter 13: We study the problem of private online learning, specifically, online prediction from experts (OPE) and online convex optimization (OCO). We propose a new transformation that transforms lazy online learning algorithms into private algorithms. We apply our transformation for differentially private OPE and OCO using existing lazy algorithms

for these problems. Our final algorithms obtain regret which significantly improves the regret in the high privacy regime $\varepsilon \ll 1$, obtaining $\sqrt{T \log d} + T^{1/3} \log(d)/\varepsilon^{2/3}$ for DP-OPE and $\sqrt{T} + T^{1/3}\sqrt{d}/\varepsilon^{2/3}$ for DP-OCO. We also complement our results with a lower bound for DP-OPE, showing that these rates are optimal for a natural family of low-switching private algorithms

Part I

IMPROVING ORACLE EFFICIENCY

Chapter 1

BREAKING QUADRATIC BARRIER

This chapter is based on [KLL21], with Janardhan Kulkarni and Yin Tat Lee.

1.1 Introduction

Privacy has become an important consideration for learning algorithms dealing with sensitive data. Over the past decade, differential privacy, introduced in the seminal work of [DMNS06], has established itself as the defacto notion of privacy for machine learning problems. In this paper, we revisit Empirical Risk Minimization (ERM) and Stochastic Convex Optimization (SCO) problem, which are one of the most important and simplest problems in statistics and machine learning, in differential privacy setting. In the ERM problem, we are given a family of convex functions $\{f(\cdot, x)\}_{x\in\Xi}$ over a closed convex set $\mathcal{K} \subset \mathbb{R}^d$, a data set $S = \{x_1, \cdots, x_N\}$ drawn from some unknown distribution $\mathcal{P}$ over the universe Ξ, and the objective is to

$$\text{minimize} \quad \widehat{F}(\omega) := \frac{1}{N}\sum_{x_i\in S} f(\omega, x_i) \quad \text{over} \quad \omega \in \mathcal{K},$$

while in the SCO the objective is to

$$\text{minimize} \quad F(\omega) := \mathop{\mathbb{E}}_{x\sim\mathcal{P}} f(\omega, x) \quad \text{over} \quad \omega \in \mathcal{K},$$

Differentially private convex optimization has been studied extensively for over a decade now [CM08, RBHT12, CMS11, KST12, JT14, TTZ14, BST14, TTZ15, KJ16, WLK+17, FTS17, ZZMW17, WYX17, INS+19]. Most of the previous results are focus on DP-ERM and roughly speaking, there are three major approaches in DP-ERM: output perturbation, objective perturbation, and gradient perturbation. Output perturbation approach is based on the sensitivity method proposed by [DMNS06] and adds noise to the final output to

the standard ERM problem [CM08, RBHT12, CMS11, ZZMW17]. Objective perturbation [CM08, CMS11, KST12, TTZ14] means to perturb the objective function we want to minimize. In the gradient perturbation approach, we add noise to the first order information using optimization algorithms such as Stochastic Gradient Descent (SGD). This approach was first proposed in [BST14] and was later extended by [TTZ14, WYX17], and has lead to the state-of-the-art theoretical bounds for DP-ERM. For an experimental comparison of various approaches to solving DP-ERM we refer the readers to [RBHT12, INS+19].

DP-ERM for smooth convex functions is well understood in the sense that we know (near) linear time algorithms that achieve optimal excess empirical risk. We refer the readers to [WYX17] for more details. However, for the more general non-smooth convex loss functions our understanding is not yet complete, which is the focus of this paper. A summary of the state-of-the-art results and our contributions for the non-smooth convex loss functions is given in Table 1.1 (General Convex) and Table 1.2 (Strongly Convex). We will discuss the concurrent work [AFKT21] separately at the end of the introduction, and the following discussion are only limited to the previous work.

[KST12] designed a DP-algorithm with $O(\frac{GD\sqrt{d}\log(1/\delta)}{\sqrt{N}\varepsilon})$ excess empirical risk by using the objective perturbation method. This result was improved significantly by [BST14], who first showed a lower bound of $\Omega(\min\{GD, \frac{GD\sqrt{d}}{N\varepsilon}\})$ on the excess empirical risk for DP-ERM. Further, they gave an algorithm with excess empirical risk $O(\frac{GD\log^{\frac{3}{2}}(N/\delta)\sqrt{d\log(1/\delta)}}{N\varepsilon})$, which is sub-optimal by a factor of $\log^{\frac{3}{2}}(N/\delta)$. Their algorithm is based on a modification of SGD by adding Gaussian noise to the gradients to make it DP. The privacy analysis proceeds via amplification by sampling and the strong composition theorem. Roughly speaking, the logarithmic blowup in the excess empirical risk is due to two reasons: 1) the strong composition theorem requires that at each step one needs to add Gaussian noise with a larger variance; 2) They used sub-optimal convergence rate $O(\log T/\sqrt{T})$ for T-step SGD.

However, getting the optimal bounds with small gradient complexity for non-smooth case turns out to be a more difficult problem. This was noted by [WYX17], who raised it as an important open problem. This question was answered in [BFTGT19], who gave an algorithm with almost optimal excess empirical risk. To achieve this, [BFTGT19] first consider the smooth case, and give an improved privacy analysis via the Moments Accountant

technique proposed by [ACG+16]. They extend their result to non-smooth case by applying Moreau-Yosida envelope technique (a.k.a. Moreau envelope smoothing) [Nes05] to make the function smooth. However, this technique is computationally inefficient and leads to $O(N^{4.5})$-gradient computations for the whole algorithm. This limitation was overcome in a recent work of [BFGT20] who gave the optimal excess empirical risk guarantee with $O(N^2)$-gradient computations. The privacy analysis of this result also used Moments Accountant method, and they used the standard online-to-batch conversion technique [CBCG04] to prove the high-probability bound on the excess empirical error of SGD, which leads to the near optimal bound in expectation. We remark that all the papers [BFTGT19, BFGT20] above not only study the ERM problem, but also consider more general DP-SCO settings and uniform stability, and in some cases, results on ERM are byproducts of the more general results.

As we can see from Table 1.1 and Table 1.2, all the previously known results (except the concurrent work [AFKT21]) achieving near optimal excess empirical risk bounds require at least $O(N^2)$-gradient computations. It is natural to ask if there are lower bounds to rule out algorithms with subquadratic gradient complexity that can match the error bounds of the above results.

As Table 1.3 and Table 1.4 show, a similar situation arises in Stochastic Convex Optimization (SCO), which is a closely related problem compared to ERM. In the SCO problem, we want to minimize the objective function $F(\omega) = \mathbb{E}_{x\sim\mathcal{P}}[f(\omega, x)]$ for some unknown distribution $\mathcal{P}$ over the universe Ξ. Many results for SCO [BST14, BFTGT19, BFGT20] are directly based on ERM; that is, solving the ERM and analyzing the generalization error. The first non-trivial result for general convex loss functions achieving excess population loss of $O\left(GD(\frac{d^{1/4}}{\sqrt{N}} + \frac{\sqrt{d}}{N\varepsilon})\right)$ was given by [BST14], who showed the result by first solving the ERM problem and bounding the generalization error. They used the result on universal convergence directly, namely, bounding $\sup_{\omega\in\mathcal{K}} \mathbb{E}[F(\omega) - \widehat{F}(\omega)]$. But this method has its limitations; For example, [Fel16] showed that lower bound of universal convergence is $\Omega(\sqrt{d/N})$ for some (not necessarily convex) loss functions. Later, [BFTGT19], [FKT20] and

	Excess Empirical Risk	Gradient Complexity
[KST12]	$\frac{GD\sqrt{d}\log(1/\delta)}{\sqrt{N}\varepsilon}$	N/A
[BST14]	$\frac{GD\log^{\frac{3}{2}}(N/\delta)\sqrt{d\log(1/\delta)}}{N\varepsilon}$	N^2
[BFTGT19]	$\frac{GD\sqrt{d\log(1/\delta)}}{N\varepsilon}$	$N^{4.5}$
[BFGT20]	$\frac{GD\sqrt{d\log(1/\delta)}}{N\varepsilon}$	N^2
[AFKT21]	$\frac{GD\sqrt{d\log(1/\delta)}}{N\varepsilon}$	$N^2/\sqrt{d}$
Ours	$\frac{GD\sqrt{d\log(1/\delta)}}{N\varepsilon}$	$\frac{N^{3/2}}{d^{1/8}}+\frac{N^2}{d}$

Table 1.1: Comparisons with previous (ε,δ)-differential private algorithms when objective function is G-Lipschitz and convex over a convex set $\mathcal{K}\subset\mathbb{R}^d$ of diameter D. The results are stated asymptotically and the big O notation is hidden for simplicity. The lower bound is $\Omega(\min\{GD,\frac{GD\sqrt{d}}{N\varepsilon}\})$ [BST14].

	Excess Empirical Risk	Gradient Complexity
[KST12]	$\frac{G^2d\log(1/\delta)}{\mu N^{3/2}\varepsilon^2}$	N/A
[BST14]	$\frac{G^2\log^2(N/\delta)d\log(1/\delta)}{\mu N^2\varepsilon^2}$	N^2
[BFTGT19]	$\frac{G^2d\log(1/\delta)}{\mu N^2\varepsilon^2}$	$N^{4.5}$
[BFGT20]	$\frac{G^2d\log(1/\delta)}{\mu N^2\varepsilon^2}$	N^2
Ours	$\frac{G^2d\log(1/\delta)}{\mu N^2\varepsilon^2}$	$\frac{N^{3/2}}{d^{1/8}}+\frac{N^2}{d}$

Table 1.2: Comparisons with previous (ε,δ)-differential private algorithms when objective function is G-Lipschitz and μ-strongly convex over a convex set $\mathcal{K}\subset\mathbb{R}^d$. The results are stated asymptotically and the big O notation is hidden for simplicity. The lower bound is $\Omega(\min\{\frac{G^2}{\mu},\frac{G^2d}{\mu N^2\varepsilon^2}\})$ [BST14].

[BFGT20] obtained near optimal excess population loss with significantly better running times (gradient complexity). The privacy analysis in these papers relied on recent advances in the privacy techniques such as the Moments Accountant method [ACG+16], Rényi differential privacy (RDP) [Mir17] and the Privacy Amplification by Iteration [FMTT18] and other fast stochastic convex optimization algorithms such as [JNN19]. The excess popula-

tion loss bound in most of these works followed by solving a (phased) convex (regularized) ERM problem and then appealing to the uniform stability property [HRS16] or the iterative localization approach [FKT20] to do the generalization error analysis.

	Excess Population Loss	Gradient Complexity
[BST14]	$GD(\frac{d^{1/4}\log(n/\delta)}{\sqrt{N}} + \frac{d^{1/2}\log^2(n/\delta))}{N\varepsilon})$	N^2
[BFTGT19]	$GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon})$	$N^{4.5}$
[FKT20]	$GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon})$	$N^2\log(1/\delta)$
[BFGT20]	$GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon})$	N^2
[AFKT21]	$GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon})$	$\min\{N^{3/2}, N^2/\sqrt{d}\}$
Ours	$GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon})$	$\min\{N^{5/4}d^{1/8}, N^{3/2}/d^{1/8}\}$

Table 1.3: Comparisons with previous (ε, δ)-differential private algorithms when objective function is G-Lipschitz and convex over a convex set $\mathcal{K} \subset \mathbb{R}^d$. The results are stated asymptotically and the big O notation is hidden for simplicity. The lower bound is $\Omega(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d}}{N\varepsilon}))$ [BST14].

	Excess Population Loss	Gradient Complexity
[BFTGT19]	$\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2})$	$N^{4.5}$
[FKT20]	$\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2})$	$N^2\log(1/\delta)$
[BFGT20]	$\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2})$	N^2
[AFKT21]	$\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2})$	$\min\{N^{3/2}, N^2/\sqrt{d}\}$
Ours	$\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2})$	$\min\{N^{5/4}d^{1/8}, N^{3/2}/d^{1/8}\}$

Table 1.4: Comparisons with previous (ε, δ)-differential private algorithms when objective function is G-Lipschitz and μ-strongly convex over a convex set $\mathcal{K} \subset \mathbb{R}^d$. The results are stated asymptotically and the big O notation is hidden for simplicity. The lower bound is $\Omega(\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2}))$ [BST14].

Despite these impressive improvements, as the Table 1.3 and Table 1.4 suggest, the previous algorithms (except the concurrent work [AFKT21]) which achieve the optimal

excess population loss still require $O(N^2)$-gradient computations. Indeed, [BFGT20] write that

> *" Proving that quadratic running time is necessary for general non-smooth DP-SCO is a very interesting open problem... "*

Understanding if the lower bound is the right answer to the above questions or one can design algorithms with subquadratic gradient complexity is the main motivation that spurred our work.

1.1.1 Our Contributions

Given the close connections between the ERM and SCO problems and the bottleneck on gradient complexity of all known algorithms, it is natural to ask if the open question raised in [BFGT20] also holds for the ERM problem. As noted earlier, the state-of-art algorithms for DP-ERM achieving optimal excess empirical risk bounds require $O(N^2)$-gradient computations.

The main contribution of this paper is to show that we can obtain subquadratic gradient complexity bound for ERM when the dimension is super constant. In particular, for the important regime of over-parameterization ($d \geq N$), we achieve a bound of $N^{1+3/8}$. Combining our private ERM algorithm and the iterative localization approach proposed in [FKT20], we can achieve optimal excess population loss with gradient complexity $O\big(N + \min\{\sqrt{\varepsilon}N^{5/4}d^{1/8}, \frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)}\}\big)$.

Let $\mathcal{K}_r = \{y \mid y = \omega + z, \omega \in \mathcal{K}, z \in \mathbb{R}^d, \|z\|_r \leq r\}$. We now state the main technical contributions of this paper formally.

Theorem 1.1.1 (DP-ERM)**.** *Suppose $\mathcal{K} \subset \mathbb{R}^d$ is a closed convex set of diameter D and $\{f(\cdot, x)\}_{x\in\Xi}$ is a family of G-Lipschitz and convex functions over $\mathcal{K}_r$, where $r = \frac{D\sqrt{d\log(1/\delta)}}{\varepsilon N}$ [1]. For $\varepsilon, \delta \leq 1/2$, given any sample set S consists of N samples from Ξ and arbitrary initial*

[1] We only need consider the non-trivial case when $\frac{\sqrt{d\log(1/\delta)}}{\varepsilon N} \leq 1$, or any feasible solution is good enough. This means that $r = O(D)$, which is a mild assumption.

point $\omega_0 \in \mathcal{K}$, we have a (ε, δ)-differentially private algorithm $\mathcal{A}$ which takes

$$O\left(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)}\right)$$

gradient queries and outputs ω_T such that

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right),$$

where $D = \|\omega^* - \omega_0\|_2, \widehat{F}(\omega) = \frac{1}{N}\sum_{x_i \in S} f(\omega, x_i), \widehat{F}^* = \min_{\omega\in\mathcal{K}} \widehat{F}(\omega)$, and the expectation is taken over the randomness of the algorithm.

Moreover, if $\{f(\cdot, x)\}_{x\in\Xi}$ is also μ-strongly convex functions over $\mathcal{K}_r$, we have an (ε, δ)-differentially private algorithm which takes the same bound of gradient queries and outputs ω_T such that

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2}\right).$$

As we have mentioned, combining our private ERM algorithm with the iterative localization technique, we can also give the first algorithm achieving optimal excess population loss with (strictly) sub-quadratic steps for all dimensions:

Theorem 1.1.2 (DP-SCO)**.** *Suppose $\varepsilon, \delta \leq 1/2$ and sample set S consists of N samples drawn i.i.d from a distribution $\mathcal{P}$ over Ξ. Let $\{f(\cdot, x)\}_{x\in\Xi}$ is convex and G-Lipschitz with respect to ℓ_2 norm and convex over $\mathcal{K}_r$, where $r = \frac{D\sqrt{d\log(1/\delta)}}{\varepsilon N}$, there is an (ε, δ)-differentially private algorithm which takes*

$$O(N + \min\{\sqrt{\varepsilon}N^{5/4}d^{1/8}, \frac{\varepsilon N^{3/2}}{d^{1/8}\log^{1/4}(1/\delta)}\})$$

gradient queries to get a solution ω_T

$$\mathbb{E}[F(\omega_T) - F(\omega^*)] = O(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon}).$$

Moreover, if $\{f(\cdot, x)\}_{x\in\Xi}$ is also μ-strongly convex over $\mathcal{K}_r$, we can meet the same gradient

query complexity and get a solution ω_T such that:

$$\mathbb{E}[F(\omega_T) - F(\omega^*)] = O\left(\frac{G^2}{\mu}\left(\frac{d\log(1/\delta)}{\varepsilon^2 N^2} + \frac{1}{N}\right)\right).$$

Finally, we note that our results can also capture the regularized ERM and SCO, which shows up often in the previous work such as [RBHT12, KST12, WYX17, INS+19]. Briefly, in the regularized problem, there is one more simple (and convex) function $h(\omega)$ added to the objective function to encourage certain solutions with better structure. The objective function then takes the form $\frac{1}{N}\sum_{x_i \in S} f(\omega, x_i) + h(\omega)$. We get asymptotically same results for the regularized ERM/SCO problem with straightforward modifications.

1.1.2 Our Techniques

Most of the previous works [BST14, BFTGT19, BFGT20] that achieve near optimal bounds for ERM and SCO are based on adaptations of SGD to make it differentially private. The information theoretic lower bound of $\Omega(1/\sqrt{T})$ for T-step SGD may be one of the important reasons why we can not get subquadratic gradient complexity for non-smooth convex ERM easily. Consider the algorithm in [BFGT20] as an example. It needs to add Gaussian noise $v \sim \mathbb{N}(0, \sigma^2 I_{d\times d})$ with $\sigma^2 = \frac{G^2\log(1/\delta)}{\varepsilon^2}$ to each gradient. By a standard analysis of SGD, we can only show an excess empirical risk of $\Theta(\frac{D\sqrt{d\sigma^2}}{\sqrt{T}})$, which requires us to set $T = \Omega(N^2)$ to get ideal bound, thus hitting the quadratic barrier.

We deviate from the above approaches for designing private algorithms for non-smooth functions. First notice that the gradient complexity $O(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)})$ in Theorem 1.1.1 is the same for both strongly convex and general non-smooth functions; same holds for DP-SCO. This is not a coincidence; If we can achieve optimal empirical risk (population loss) for one case, then we can achieve optimal empirical risk (population loss) for another with the same privacy guarantee and gradient complexity. In fact, the Figure 1.1 shows the relationship among these different problems.

Our result for the general convex non-smooth case is obtained by providing a reduction to the strongly convex non-smooth case. Thus, our task becomes designing better algorithms for the strongly convex non-smooth functions. Rather than using SGD, we let the objective

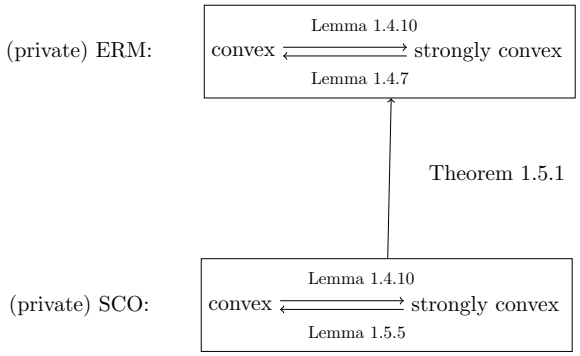


Figure 1.1: Reductions between ERM and SCO for general convex and strongly convex cases. As the lower bound of excess population loss is $\Omega(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d}}{N\varepsilon}))$ while the lower bound of empirical risk is $\Omega(\frac{GD\sqrt{d}}{N\varepsilon})$, we do not know how to reduce from ERM to SCO.

function take convolution with a sphere kernel to make it smooth. We then use the accelerated stochastic approximation algorithm in [GL12] for solving strongly convex stochastic optimization problems. However, this is not enough, as the required noise that needs to be added to the gradients to make the algorithm private is too large to get subquadratic gradient complexity, even if we use the tighter Moments Accountant technique [ACG+16]. We overcome this by increasing the batch size to an appropriate value. Combining these ideas together, we show that the amount of noise we add can be reduced to achieve the optimal excess empirical loss, and we get the gradient complexity of $O(\max\{N^{3/2}/d^{1/8}, N^2/d\})$.

For SCO, we get the gradient complexity of $O(\min\{N^{5/4}d^{1/8}, N^{3/2}/d^{1/8}\})$ via a direct application of the iterative localization approach of Feldman et al [FKT20]. The intuition behind iterative localization is using private ERM to solve regularized objective functions which have low sensitivity, iteration by iteration. Each iteration reduces the distance to an approximate minimizer by a multiplicative factor, so after logarithmic number of phases we are done.

1.1.3 Concurrent and Independent Work

In an independent and concurrent work, [AFKT21] give a new analysis of private regularized mirror descent to do the private ERM. Then they combine the iterative localization approach

to achieve the optimal excess population loss for SCO. Their result also achieves subquaratic gradient complexity. More formally, they get $O\left(\log N \cdot \min\left(N^{3/2}\sqrt{\log d}, N^2/\sqrt{d}\right)\right)$ for SCO in query complexity. We compare their gradient complexity with ours in Figure 1.2. Finally, we remark that the main motivation of [AFKT21] was to study SCO problem in more general ℓ_p norms as much of the literature has focussed on the ℓ_2-norm. They also give news results in ℓ_p-bounded domain together with another concurrent work [BGN21].

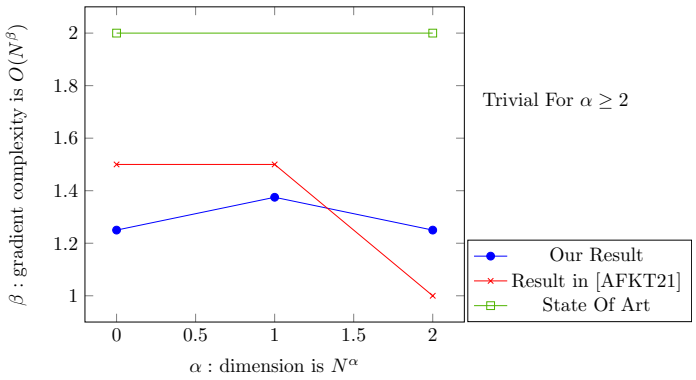


Figure 1.2: Comparison among our results, the recent result in [AFKT21] and the previous best one for the non-trivial regime ($d \leq N^2$). Suppose ε, δ are small constants. Our result is faster for the important case $d \leq N^{1+1/3}$.

Road map

We will give some basic definitions and theorems about convex optimization and differential privacy in Section 1.2. In Section 1.3, we give a general algorithm framework for private convex optimization. The results of DP-ERM are given in Section 1.4 and the results of DP-SCO are shown in Section 1.5. Some technical proofs are left in Appendix 1.6.

1.2 Preliminaries

In this section, we briefly recall some of the main definitions we use from the convex optimization theory and differential privacy. We refer the readers to excellent books [Nes05, DR14] for more details on these topics.

1.2.1 Convex Optimization

Definition 1.2.1 (Empirical risk minimization, Stochastic Convex Optimization)**.** Let $\mathcal{K} \subset \mathbb{R}^d$ be a closed convex set of diameter D. Given a family of convex loss functions $\{f(\omega, x)\}_{x\in\Xi}$ of ω over $\mathcal{K}$ and a set of samples $S = \{x_1, \cdots, x_N\}$ over the universe Ξ, the objective of Empirical Risk Minimization (ERM) is to minimize

$$\widehat{F}(\omega) = \frac{1}{N}\sum_{x_i\in S} f(\omega, x_i).$$

The excess empirical loss with respect to a solution ω is defined by $\widehat{F}(\omega) - \widehat{F}^*$, where $\widehat{F}^* = \min_{\omega\in\mathcal{K}} \widehat{F}(\omega)$.

Stochastic Convex Optimization (SCO) wants to output a solution ω to minimize the expected loss (also referred to *population loss*) $F(\omega) - F^*$ where $F(\omega) = \mathbb{E}[x \sim \mathcal{P}]f(\omega, x)$ and $F^* = \min_{\omega\in\mathcal{K}} F(\omega)$.

Definition 1.2.2 (L-Lipschitz Continuity)**.** A function $f : \mathcal{K} \to \mathbb{R}$ is L-Lipschitz continuous over the domain $\mathcal{K} \subset \mathbb{R}^d$ if the following holds for all $\omega, \omega' \in \mathcal{K} : |f(\omega) - f(\omega')| \leq L\|\omega - \omega'\|_2$.

Definition 1.2.3 (β-Smoothness)**.** A function $f : \mathcal{K} \to \mathbb{R}$ is β-smooth over the domain $\mathcal{K} \subset \mathbb{R}^d$ if for all $\omega, \omega' \in \mathcal{K}$, $\|\nabla f(\omega) - \nabla f(\omega')\|_2 \leq \beta\|\omega - \omega'\|_2$.

Definition 1.2.4 (μ-Strongly convex)**.** A differentiable function $f : \mathcal{K} \to \mathbb{R}$ is called strongly convex with parameter $\mu > 0$ if the following inequality holds for all points $\omega, \omega' \in \mathcal{K}$,

$$\langle \nabla f(\omega) - \nabla f(\omega'), \omega - \omega' \rangle \geq \mu\|\omega - \omega'\|_2^2.$$

Equivalently,

$$f(\omega') \geq f(\omega) + \nabla f(\omega)^\top(\omega' - \omega) + \frac{\mu}{2}\|\omega' - \omega\|_2^2.$$

1.2.2 Differential Privacy

Definition 1.2.5 (Differential privacy)**.** A randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if for any event $\mathcal{O} \in \text{Range}(\mathcal{M})$ and for any neighboring databases that differ in a

single data element, one has

$$\Pr[\mathcal{M}(S) \in \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(S') \in \mathcal{O}] + \delta.$$

Lemma 1.2.6 (Proposition 2.1 in [DR14]). *(Post-Processing) Let $\mathcal{M} : \mathbb{N}^{|\Xi|} \to R$ be a randomized algorithm that is (ε, δ)-differentially private. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\Xi|} \to R'$ is (ε, δ)-differentally private.*

Theorem 1.2.7 (Basic Composition). *Let $\mathcal{M}_i : \mathbb{N}^{|\Xi|} \to R_i$ be $(\varepsilon_i, \delta_i)$-differentially private. Then if mechanism $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \prod_{i=1}^{k} \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.*

1.3 A Meta Algorithm for DP Convex Optimization

Many convex optimization algorithms with noisy first-order information have the following simple format.

Algorithm 1: Meta Algorithm META

Input: The objective convex function $F(\omega)$ we want to minimize, an initial point ω_0.

Process: for *phases* $t = 1, \cdots$, **do**

  Get the noisy gradient $G_t \approx \nabla F(\omega_{t-1})$;

  Update the result by some sub-procedure: $\omega_t \leftarrow$ Sub-procedure(ω_{t-1}, G_t);

end

Output: Some function of $\{\omega_i\}_{i \geq 1}$.

We can use the above algorithmic framework to solve ERM privately. Specifically, we make two simple modifications to make it private. First, we compute gradients over a uniform sample of some size B. Next, we add a carefully calibrated Gaussian noise to these gradients and take average, before updating our results. This gives us a meta differentially private algorithm for convex optimization problems, and is described in Algorithm 2. The DP analysis then follows from a careful accounting of the privacy budget lost in each iteration, and the bound on excess empirical risk comes from the property of the optimization algorithm.

Algorithm 2: Private Meta Algorithm $\mathsf{META}_{\mathsf{DP}}$

Input: Sample set $S = \{x_1, \cdots, x_N\}$, the objective convex function $F(\omega)$ we want to minimize, the initial point ω_0, and privacy parameter ε, δ;

Process: for *phases* $t = 1, \cdots, T$ **do**

  Select a random sample set S_t from the uniform distribution over all subsets of S of size B;

  Let $G_t = (\sum_{x_i \in S_t} \nabla f(\omega_{t-1}, x_i) + v)/B$, where $v \sim \mathbb{N}(0, \sigma^2 I_{d\times d})$;

  Update the result by some sub-procedure $\omega_t \leftarrow$ Sub-procedure(ω_{t-1}, G_t);

end

Output: Some function of $\{\omega_i\}_{i\geq 1}$.

The above framework is a sub-sampled Gaussian mechanism, for which we can use tCDP proposed in [BDRS18] to analyze its privacy guarantee. As this is a direct application of the main result in [BDRS18], we leave the proof of the following theorem in the Appendix.

Theorem 1.3.1. *Suppose* $\{f(\cdot, x)\}_{x\in\Xi}$ *is a family of* G*-Lipschitz and convex functions over* $\mathcal{K}$*, for* $\varepsilon < c_1 B^2 T/N^2, B \leq N/10$ *and* $1/2 \geq \delta > 0$*, by setting* $\sigma = \frac{c_2 GB\sqrt{T\log(1/\delta)}}{\varepsilon N}$ *for some constant* c_1 *and* c_2*,* $\mathsf{META}_{\mathsf{DP}}$ *is* (ε, δ)*-differential private.*

1.4 Differentially Private ERM

In this section, we present private algorithms achieving the optimal excess empirical loss with subquadratic gradient complexity when the dimension is super constant. We consider non-smooth strongly-convex functions first, and then show how to reduce the general non-smooth case to the strongly-convex case in the last subsection.

1.4.1 Non-smooth Strongly-convex Functions

We use the framework introduced in Section 1.3 and give a faster private algorithm. Specifically, we modify a stochastic convex optimization algorithm in [GL12] to fit into our framework. First we recall some properties of that algorithm.

Suppose $f : \mathcal{K} \to \mathbb{R}$ is a convex function, and the objective is to get

$$\Psi^* := \min_{\omega \in \mathcal{K}} \{\Psi(\omega) = f(\omega) + h(\omega)\},$$

where $\mathcal{K}$ is a closed convex set and $h(\omega)$ is a simple convex function with known structure.

Theorem 1.4.1 (Proposition 9 in [GL12]). *If the following conditions are met:*

- *For some $L \geq 0, M \geq 0$ and $\mu > 0$,*

$$\frac{\mu}{2}\|y - \omega\|_2^2 \leq f(y) - f(\omega) - \langle g(\omega), y - \omega \rangle \leq \frac{L}{2}\|y - \omega\|_2^2 + M\|y - \omega\|_2, \quad \forall \omega, y \in \mathcal{K},$$

  where $g(\omega) \in \partial f(\omega)$ and $\partial f(\omega)$ denotes the sub-differential of f at ω.

- *For each call of the stochastic oracle $\mathcal{G}$ with the input $\omega_t \in \mathcal{K}$, the stochastic oracle $\mathcal{G}$ can output an independent vector $\mathcal{G}(\omega_t)$ such that $\mathbb{E}[\mathcal{G}(\omega_t)] \in \partial f(\omega_t)$.*

- *For any $t \geq 1$ and $\omega_t \in \mathcal{K}$, $\mathbb{E}[\|\mathcal{G}(\omega_t) - g(\omega_t)\|_2^2] \leq V$.*

Then after T iterations, Algorithm 3 given below outputs ω_T such that

$$\mathbb{E}[\Psi(\omega_T) - \Psi^*] \leq O\left(\frac{L\|\omega_0 - \omega^*\|_2^2}{T^2} + \frac{M^2 + V}{\mu T}\right),$$

where $\omega^* = \arg\min_{\omega \in \mathcal{K}} \Psi(\omega)$ and $\Psi^* = \Psi(\omega^*)$.

Algorithm 3: Accelerated stochastic approximation (AC-SA) algorithm

Input: Initial point $\omega_0 \in \mathcal{K}$.

Initialization: Set the initial point $\omega_0^{ag} = \omega_0$;

Set the step-size parameters $\alpha_t = \frac{2}{t+2}$ and $\gamma_t = \frac{4L}{t(t+1)}$;

Process:

for $t = 1, \cdots, T$ **do**

  Let $\omega_t^{md} = \frac{(1-\alpha_t)(\mu+\gamma_t)}{\gamma_t+(1-\alpha_t^2)\mu}\omega_{t-1}^{ag} + \frac{\alpha_t[(1-\alpha_t)\mu+\gamma_t]}{\gamma_t+(1-\alpha_t^2)\mu}\omega_{t-1}$;

  Query Oracle $\mathcal{G}_t \equiv \mathcal{G}(\omega_t^{md})$;

  Let

  $\omega_t = \arg\min_{\omega\in\mathcal{K}}\{\alpha_t[\langle \mathcal{G}_t, \omega\rangle + h(\omega) + \mu\|\omega_t^{md} - \omega\|_2^2] + [(1-\alpha_t)\mu + \gamma_t]\|\omega_{t-1} - \omega\|_2^2\}$;

  $\omega_t^{ag} = \alpha_t\omega_t + (1-\alpha_t)\omega_{t-1}^{ag}$;

end

Return: ω_T^{ag}.

Smoothing function

From the statement of Theorem 1.4.1, it is clear that the Algorithm 3 gives much better convergence rates for smooth functions. As we are considering non-smooth functions, we need an efficient way to smooth the objective function without introducing too much error. In the next few paragraphs, we show how to achieve that.

Recall that D denotes the diameter of the closed convex set $\mathcal{K} \subset \mathbb{R}^d$. Suppose $\{f(\cdot, x)\}_{x\in\Xi}$ is a family of G-Lipschitz and μ-strongly convex functions over $\mathcal{K}$. This implies that for any sample set S, the empirical loss function $\widehat{F}(\omega)$ we consider is G-Lipschitz and μ-strongly convex over the domain $\mathcal{K}$.

We do a convolution on $f(\cdot, x)$, which is denoted by $f(\cdot, x) * n_r$. The objective function after the convolution step becomes $\widehat{F}_{n_r}(\omega) = \frac{1}{N}\sum_{x_i\in S}\mathbb{E}_{y\sim n_r} f(\omega + y, x_i)$, where n_r is the uniform density on the ℓ_2 ball of radius r. By Lemma 7 and Lemma 8 in [YNS12] while the forth result on forth item was supplemented by Lemma E.2 in [DBW12], we know the claim below.

Claim 1.4.2. *Suppose $\{f(\cdot, x)\}_{x\in\Xi}$ is convex and G-Lipschitz over $\mathcal{K} + B_2(0, r)$. For $\omega \in \mathcal{K}$,*

$\widehat{F}_{n_r}(\omega)$ has following properties:

- $\widehat{F}(\omega) \leq \widehat{F}_{n_r}(\omega) \leq \widehat{F}(\omega) + Gr$;
- $\widehat{F}_{n_r}(\omega)$ *is* G*-Lipschitz and* μ*-strongly convex;*
- $\widehat{F}_{n_r}(\omega)$ *is* $\frac{G\sqrt{d}}{r}$*-Smooth;*
- *For random variables* $y \sim n_r$ *and* x *uniformly from* S*, one has*

$$\mathbb{E}[\nabla f(\omega + y, x)] = \nabla \widehat{F}_{n_r}(\omega)$$

and

$$\mathbb{E}[\|\nabla \widehat{F}_{n_r}(\omega) - \nabla f(\omega + y, x)\|_2^2] \leq G^2.$$

Furthermore, the convolution operation preserves strong convexity, which implies the fact below.

Fact 1.4.3. *Let* n_r *be the uniform density on the* ℓ_2 *ball of radius* r*, and* $f : \mathcal{K}_r \to \mathbb{R}$ *be a* μ*-strongly convex function over* $\mathcal{K}_r$*. Then* $\mathbb{E}_{y \sim n_r} f(y + \cdot)$ *is* μ*-strongly convex over* $\mathcal{K}$*.*

Algorithm

Now we state the our modifications to make AC−SA private and prove its properties. Recall that $y \sim n_r$ is a d-dimension vector drawn from the uniform density on the ℓ_2 ball of radius r. We start with the description of our algorithm.

Algorithm 4: Private AC−SA

Input: A convex set $\mathcal{K}$ with diameter D, a family $\{f(\cdot, x_i)\}_{i\in[N]}$ of G-Lipschitz and μ-strongly convex functions over $\mathcal{K}$, an initial point $\omega_0 \in \mathcal{K}$, privacy parameters ε, δ, the batch size B, and the number of steps T.

Set $r \leftarrow \frac{D}{Td^{1/4}}$ and $\sigma \leftarrow \Theta(\frac{GB\sqrt{T\log(1/\delta)}}{\varepsilon N})$;

Run the AC−SA with the Oracle $\mathcal{G}$ defined below;

Return: The output of AC−SA

Oracle $\mathcal{G}(\omega)$**:**

Select a random sample set S_t from the uniform distribution over all subsets of S of size B.

Return: $\big(\sum_{x_i\in S_t} \partial f(\omega + y_i, x_i) + v)/B$, where $y_i \sim n_r$ and $v \sim \mathbb{N}(0, \sigma^2\mathbf{I}_{d\times d})$.

Utility and Privacy

It is not hard to show that Private AC−SA (Algorithm 4) is an instance of $\mathsf{META}_{\mathsf{DP}}$ (see Section 1.3), so we have the following guarantee directly by Theorem 1.3.1.

Lemma 1.4.4. *For $\varepsilon \le c_1B^2T/N^2, \delta \le 1/2, B \le N/10$ and $\sigma = \frac{c_2GB\sqrt{T\log(1/\delta)}}{\varepsilon N}$ where $c_1 \le 1, c_2 \ge 1$ are constants, Private* AC−SA *is (ε, δ)-DP.*

Now, we consider the accuracy of Private AC−SA. We need a technical lemma to argue about the variance of the gradient $\mathcal{G}_t$ returned at t-th step by the oracle.

Lemma 1.4.5. *Under the assumptions defined in Algorithm Private* AC−SA*, after T iterations, it outputs ω_T such that*

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2/B + \sigma^2 d/B^2}{\mu T} + \frac{GDd^{1/4}}{T}\right),$$

where $\omega^* = \arg\min_{\omega\in\mathcal{K}} \widehat{F}(\omega)$, and $\widehat{F}^* = \min_\omega \widehat{F}(\omega)$.

Proof. By Claim 1.4.2, we know that $\widehat{F}_{n_r}$ is G-Lipschitz and $\frac{G\sqrt{d}}{r}$-smooth. Furthermore, by Fact 1.4.3, we know that $\widehat{F}_{n_r}$ is μ-strongly convex. For any tth iteration, one has that

$\mathbb{E}[\mathcal{G}_t] = \nabla \widehat{F}_{n_r}(\omega_t^{md})$ and $\mathbb{E}[\|\mathcal{G}_t - \nabla \widehat{F}_{n_r}(\omega_t^{md})\|_*^2] \le G^2/B + \sigma^2 d/B^2$. Then by Theorem 1.4.1 with $M = 0, L = \frac{G\sqrt{d}}{r}, V = G^2/B + \sigma^2 d/B^2$, we get

$$\mathbb{E}[\widehat{F}_{n_r}(\omega_T) - \min_{\omega} \widehat{F}_{n_r}(\omega)] = O\left(\frac{G^2/B + \sigma^2 d/B^2}{\mu T} + \frac{GD^2\sqrt{d}}{T^2 r}\right).$$

Next, by the first bullet of Claim 1.4.2, we know that $\widehat{F}(\omega) \le \widehat{F}_{n_r}(\omega) \le \widehat{F}(\omega) + Gr$ for any ω. Combining these together, we get

$$\begin{aligned}
&\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}(\omega^*)] \\
=&\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}_{n_r}(\omega_T)] + \mathbb{E}[\widehat{F}_{n_r}(\omega_T) - \min_{\omega} \widehat{F}_{n_r}(\omega)] + \min_{\omega} \widehat{F}_{n_r}(\omega) - \widehat{F}(\omega^*) \\
\le& 2Gr + O(\frac{G^2/B + \sigma^2 d/B^2}{\mu T} + \frac{GD^2\sqrt{d}}{T^2 r}).
\end{aligned}$$

By setting $r = \frac{Dd^{1/4}}{T}$, we completes the proof. □

Before stating the main result of this section, we prove two technical lemmas that can remove the dependence on the diameter term. Lemma 1.4.6 below is used to prove Lemma 1.4.7.

Lemma 1.4.6. *Consider a sequence $x_1, x_2, \cdots$. Suppose $0 \le x_1 \le n$ and $0 \le x_{i+1} \le \sqrt{x_i} + 1$, then for $k \ge \lceil \log\log n \rceil$, one has that $x_k \le 16$.*

Proof. Without loss of generality, let $x_{i+1} = \sqrt{x_i} + 1$.

We construct another sequence $y_1, \cdots, y_k$ such that $y_1 = x_1$ and $y_{i+1} = 2\sqrt{y_i}$. Then by induction, it is easy to prove that for each $i \in [k]$, $y_i \ge x_i$. So we only need to prove that $y_k \le 16$.

Let $z_i = \log_2 y_i$, then one has $z_{i+1} = z_i/2 + 1$. Obviously, we know that $z_i = 2^{-i+1}(z_1 - 2) + 2$ and $z_k \le 4$, which means that $x_k \le y_k \le 16$. □

Recall that the lower bound of strongly convex case is $\Omega(\frac{G^2}{\mu} + \frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2})$ while for the general case is $\Omega(GD + \frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N})$. Therefore, we only need to think about the case when $\frac{d\log(1/\delta)}{\varepsilon^2 N^2} \le 1$, or the bound will be trivial. The following lemma says if we can achieve sum of these two lower bounds for strongly-convex case, then we can achieve the optimal

bound for the strongly-convex case, which implies we can reduce the Strongly-Convex Case to General Convex Case.

Lemma 1.4.7 (Reduction to General Convex Case)**.** *Given $\widehat{F}$ is G-Lipschitz and μ-strongly convex. Suppose for any $\varepsilon, \delta < 1/2$, we have an (ε, δ)-differentially private algorithm $\mathcal{A}$ which takes ω_0 as the initial start point and outputs a solution ω_T such that*

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2 d \log(1/\delta)}{\mu \varepsilon^2 N^2} + \frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right),$$

where $\omega^* = \arg\min_{\omega \in \mathcal{K}} \widehat{F}(\omega)$ and $D = \|\omega_0 - \omega^*\|_2$. Then by taking $\mathcal{A}$ as sub-procedure with some modifications on parameters, we can get an (ε, δ)-differentially private solution with excess empirical loss at most

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2 d \log(1/\delta)}{\mu \varepsilon^2 N^2}\right).$$

Furthermore, if $\mathcal{A}$ uses $g(N, \varepsilon, \delta)$ many gradients, the new algorithm uses $\sum_{i \geq 1} g(N, \varepsilon/2^i, \delta/2^i)$ many gradients.

Remark 1.4.8. All algorithms in this paper uses less gradients if ε and δ are smaller. So, the new algorithm uses essentially as much as the given algorithm.

Proof. Repeat the private algorithm $\mathcal{A}$ for $k = \lceil \log \log N^3 \rceil$ times. For the ith repetition, we start from the output of the last repetition and use $\mathcal{A}$ as a sub-procedure with privacy parameter $\varepsilon_i = \varepsilon/2^{k+1-i}$ and $\delta_i = \delta/2^{k+1-i}$. (Note that the noise is decreasing so that the last step gives the best solution). We show that the last output has excess empirical risk at most $O\left(\frac{G^2 d \log(1/\delta)}{\mu \varepsilon^2 N^2}\right)$.

More specifically, let ω_i be the output of the ith repetition, $\Delta_i = \mathbb{E}[\widehat{F}(\omega_i) - \widehat{F}^*]$ and $D_i^2 = \mathbb{E}[\|\omega_i - \omega^*\|^2]$. As the objective function $\widehat{F}$ is μ-strongly convex, we know that $\frac{1}{2}\mu D_i^2 \leq \Delta_i$ for all $i \geq 0$.

By the guarantee of the algorithm, there exists some constant $c \geq 1$ such that

$$\Delta_{i+1} = \mathbb{E}[\widehat{F}(\omega_{i+1}) - \widehat{F}^*]$$

$$\leq c\frac{GD_i\sqrt{d\log(1/\delta_i)}}{\varepsilon_i N} + c\frac{G^2 d\log(1/\delta_i)}{\mu\varepsilon_i^2 N^2}$$
$$\leq c\frac{G\sqrt{d\log(1/\delta_i)}}{\varepsilon_i N}\sqrt{\frac{2\Delta_i}{\mu}} + \frac{E_i}{c},$$

where we define $E_i = 2c^2\frac{G^2 d\log(1/\delta_i)}{\mu\varepsilon_i^2 N^2}$.

As $E_i/E_{i+1} = \frac{\varepsilon_{i+1}^2\log(1/\delta_i)}{\varepsilon_i^2\log(1/\delta_{i+1})} \leq 8$, we can rearrange the above function and get

$$\begin{aligned}\frac{\Delta_{i+1}}{64E_{i+1}} \leq& \frac{\sqrt{\Delta_i E_i} + \frac{E_i}{c}}{64E_{i+1}}\\ \leq& \frac{E_i}{64E_{i+1}}(\sqrt{\frac{\Delta_i}{E_i}} + \frac{4}{c})\\ \leq& \sqrt{\frac{\Delta_i}{64E_i}} + 1.\end{aligned}$$

By strong convexity one has that $\Delta_1 \leq G^2/\mu$, and $E_1 = \Omega(G^2\log^3 N/(\mu N^2)) = \Omega(G^2/(\mu N^3))$ by the definition, so $\Delta_1/E_1 \leq N^3$. Then by Lemma 1.4.6, after $k = \lceil\log\log N^3\rceil$ repetitions, we get $\frac{\Delta_k}{64E_k} \leq 16$. This further implies that there is a solution with expected error

$$\mathbb{E}[\widehat{F}(\omega_k) - \widehat{F}^*] = O(\frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2}).$$

The privacy guarantee comes directly from the basic composition theorem (See Theorem 1.2.7). □

We did not optimize constants in the calculations above. Now we are ready to state the main result for the strongly-convex case.

Theorem 1.4.9 (Strongly Convex Case for Theorem 1.1.1)**.** *Suppose $\mathcal{K} \subset \mathbb{R}^d$ is a closed convex set of diameter D and $\{f(\cdot, x)\}_{x\in\Xi}$ is a family of G-Lipschitz and μ-strongly convex functions over $\mathcal{K}_r$ where $r = \frac{D\sqrt{d\log(1/\delta)}}{\varepsilon N}$. For $\varepsilon, \delta \leq 1/2$, given any sample set S consists of N samples from Ξ and arbitrary initial point $\omega_0 \in \mathcal{K}$, we have an (ε, δ)-differentially private algorithm $\mathcal{A}$ which takes $O(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)})$ gradient queries and outputs ω_T such*

that

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2}\right),$$

where $\widehat{F}(\omega) = \frac{1}{N}\sum_{x_i\in S} f(\omega, x_i)$, $\widehat{F}^* = \min_{\omega\in\mathcal{K}} \widehat{F}(\omega)$, *and the expectation is taken over the randomness of the algorithm itself.*

Proof. By Lemma 1.4.5, the output ω of Private AC−SA satisfies

$$\mathbb{E}[\widehat{F}(\omega) - \widehat{F}^*] = O\left(\frac{\frac{G^2}{B} + \frac{\sigma^2 d}{B^2}}{\mu T} + \frac{GDd^{1/4}}{T}\right).$$

By setting $\sigma = \frac{c_2 GB\sqrt{T\log(1/\delta)}}{\varepsilon N}$ and $T = \lceil \frac{100\varepsilon N}{c_1 d^{1/4}\sqrt{\log(1/\delta)}} \rceil$ (c_1, c_2 are defined in Lemma 1.4.4), one has

$$\begin{aligned}\mathbb{E}[\widehat{F}(\omega) - \widehat{F}^*] &= O\left(\frac{G^2}{\mu BT} + \frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2} + \frac{GDd^{1/4}}{T}\right)\\ &= O\left(\frac{G^2}{\mu BT} + \frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2} + \frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right)\end{aligned}$$

To ensure that Private AC−SA is (ε, δ)-DP, we set $B = \lceil \sqrt{\frac{\varepsilon N^2}{c_1 T}} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)T} \rceil$. By our choice of T, we have $B \le N/10$ and $\varepsilon \le c_1 B^2 T/N^2$. Hence, we can apply Lemma 1.4.4 to conclude the guarantee of (ε, δ) differential privacy.

Furthermore, we get a solution ω such that

$$\mathbb{E}[\widehat{F}(\omega) - \widehat{F}^*] = O\left(\frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2} + \frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right).$$

As for the total gradient complexity of our algorithm, we are under the assumption that $\frac{d\log(1/\delta)}{\varepsilon^2 N^2} \le 1$, which means that $\frac{\varepsilon N}{d^{1/4}\sqrt{\log(1/\delta)}} \ge d^{1/4}$, and $T = \lceil \frac{100\varepsilon N}{c_1 d^{1/4}\sqrt{\log(1/\delta)}} \rceil = \Theta(\frac{\varepsilon N}{d^{1/4}\sqrt{\log(1/\delta)}})$. As for the batch size, we know $\sqrt{\frac{\varepsilon N^2}{T}} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)T} = \omega(1)$ and thus $B =$

$\lceil\sqrt{\frac{\varepsilon N^2}{T}}+\frac{\varepsilon^2N^2}{d\log(1/\delta)T}\rceil=\Theta(\sqrt{\frac{\varepsilon N^2}{T}}+\frac{\varepsilon^2N^2}{d\log(1/\delta)T})$, from which we get the gradient complexity is

$$BT=\Theta\left(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)}+\frac{\varepsilon^2N^2}{d\log(1/\delta)}\right).$$

By Lemma 1.4.7 we can adjust Private AC−SA and get a final solution ω_T such that

$$\mathbb{E}[\widehat{F}(\omega_T)-\widehat{F}^*]=O\left(\frac{G^2d\log(1/\delta)}{\mu\varepsilon^2N^2}\right),$$

with gradient complexity $\Theta(\sum_{i=1}^{\log\log N^3}\frac{(\varepsilon/2^i)N^{3/2}}{d^{1/8}\log^{1/4}(2^i/\delta)}+\frac{(\varepsilon/2^i)^2N^2}{d\log(2^i/\delta)})=\Theta(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)}+\frac{\varepsilon^2N^2}{d\log(1/\delta)})$, which completes the proof. □

1.4.2 General Non-smooth Convex Functions

In the general non-smooth case, we only assume that the family of functions $\{f(\cdot,x)\}_{x\in\Xi}$ is G-Lipschitz and convex over $\mathcal{K}$. We now give a reduction from this case to the strongly-convex case, which completes our second main result.

Lemma 1.4.10. *Suppose $\mathcal{K}\subset\mathbb{R}^d$ is a convex set of diameter D and let $\{f(\cdot,x)\}_{x\in\Xi}$ be a family of convex functions over $\mathcal{K}$, which are G-Lipschitz and μ-strongly convex. Given any sample set S consists of N samples from Ξ and other necessary inputs, suppose we have a (ε,δ)-DP algorithm $\mathcal{A}$ which can output a solution ω_T such that*

$$\mathbb{E}[\widehat{F}(\omega_T)-\widehat{F}^*]=O\left(\frac{G^2d\log(1/\delta)}{\mu\varepsilon^2N^2}\right),$$

where $\widehat{F}^*=\min_{\omega\in\mathcal{K}}\widehat{F}(\omega)$.

Then when $\{h(\cdot,x)\}_{x\in\Xi}$ is only G-Lipschitz and convex with necessary inputs, for any sample set S of size N, we also have a (ε,δ)-DP algorithm $\mathcal{A}'$ which can get a solution ω_T such that

$$\mathbb{E}[\widehat{H}(\omega_T)-\widehat{H}^*]=O\left(\frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right).$$

where $\widehat{H}(\omega)=\frac{1}{N}\sum_{x_i\in S}h(\omega,x_i)$, $\widehat{H}^*=\min_{\omega\in\mathcal{K}}H(\omega)$. The gradient complexity and privacy

guarantee of $\mathcal{A}$ and $\mathcal{A}'$ are the same.

Moreover, the reduction also holds in the context of SCO.

Proof. We only consider this lemma in the context of ERM, as we can use the nearly the same argument for SCO.

The proof of this reduction is rather simple: After getting $\{h(\cdot, x_i)\}_{x_i \in S}$, we only need to consider $h_u(\omega, x) = h(\omega, x) + u\|\omega\|^2$. Then $h_u(\cdot, x)$ is u-strongly convex and $O(G + uD)$-Lipschitz for any x with $\|x\|_2 \leq 2D$.

For the case $uD \leq G$, we run $\mathcal{A}$ on $\{h_u(\cdot, x_i)\}_{x_i \in S}$ to get a solution ω_T with loss

$$\mathbb{E}[H_u(\omega_T) - H_u^*] = O\left(\frac{G^2 d \log(1/\delta)}{u\varepsilon^2 N^2}\right),$$

where $H_u(\omega) = \frac{1}{N}\sum_{x_i \in S} h(\omega, x_i) + u\|\omega\|^2$ and $H_u^* = \min_{\omega \in \mathcal{K}} H_u(\omega)$. Now by setting $u = \Theta\left(\frac{G\sqrt{d\log(1/\delta)}}{D\varepsilon N}\right)$, one has

$$\begin{aligned}\mathbb{E}[\widehat{H}(\omega_T) - \widehat{H}^*] =& O\left(\frac{G^2 d \log(1/\delta)}{u\varepsilon^2 N^2} + uD^2\right) \\ =& O\left(\frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right).\end{aligned}$$

For the case $uD \geq G$, we have $\frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N} \geq GD$ and hence we can simply output the initial point ω_0 as the solution with a loss no more than GD. □

The above reduction completes the main result of this subsection.

Theorem 1.4.11 (General Convex Case for Theorem 1.1.1)**.** *Suppose $\mathcal{K} \subset \mathbb{R}^d$ is a convex set of diameter D and $\{f(\cdot, x)\}_{x \in \Xi}$ is a family of G-Lipschitz and convex functions over $\mathcal{K}_r$ where $r = \frac{D\sqrt{d\log(1/\delta)}}{\varepsilon N}$. For $\varepsilon, \delta \leq 1/2$, given any sample set S consists of N samples from Ξ and arbitrary initial point $\omega_0 \in \mathcal{K}$, we have a (ε, δ)-differentially private algorithm $\mathcal{A}$ which takes $O\left(\frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)} + \frac{\varepsilon^2 N^2}{d\log(1/\delta)}\right)$ gradient queries and outputs ω_T such that*

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{GD\sqrt{d\log(1/\delta)}}{\varepsilon N}\right),$$

where $\widehat{F}^* = \min_{\omega\in\mathcal{K}} \widehat{F}(\omega)$*, and the expectation is taken over the randomness of the algorithm.*

1.5 Differentially Private SCO

In this section we study SCO. We can use the iterative localization technique in [FKT20] to reduce the SCO problem to an ERM problem. More specifically, if we can solve private ERM and get a (nearly) optimal empirical loss, then we can solve private SCO with (nearly) optimal excess population loss with the following algorithm framework (Algorithm 5). See Theorem 1.5.1 for the corresponding formal statement.

Algorithm 5: Iterative Localized Algorithm Framework $\mathcal{A}'$

Input: A family of G-Lipschitz and μ-strongly convex function $f : \mathcal{K} \times \Xi \to \mathbb{R}$, initial point $\omega_0 \in \mathcal{K}$ and privacy parameter ε, δ.
Process: Set $k = \lceil \log N \rceil$;
for $i = 1, \cdots, k$ **do**
  Set $\varepsilon_i = \varepsilon/2^i, N_i = N/2^i, \eta_i = \eta/2^{5i}$;
  Apply $(\varepsilon_i, \delta_i)$-DP ERM algorithm $\mathcal{A}_{\varepsilon_i,\delta_i}$ over $\mathcal{K}_i = \{\omega \in \mathcal{K} : \|\omega - \omega_{i-1}\|_2 \le 2G\eta_i N_i\}$ with the function $\widehat{F}_i(\omega) = \frac{1}{N_i}\sum_{j\in S_i} f(\omega, x_j) + \frac{1}{\eta_i N_i}\|\omega - \omega_{i-1}\|^2$ where S_i consists of N_i samples with replacement from $\mathcal{P}$;
  Let ω_i be the output of the ERM algorithm;
end
Return: The final iterate ω_k;

Theorem 1.5.1. *Suppose we have an algorithm $\mathcal{A}$ which can solve ERM under strongly convex case and gets a solution with excess empirical loss $O(\frac{G^2}{\mu N})$ by using $g(N)$ many gradients, then we have an algorithm $\mathcal{A}'$ which can solve SCO under general case and gets a solution with excess population loss $O(\frac{GD}{\sqrt{N}})$ by using $\sum_{i=1}^{\lceil \log N\rceil} g(N/2^i)$ many gradients.*

Moreover, for $\varepsilon, \delta \le 1/2$, if $\mathcal{A}_{\varepsilon,\delta}$ is (ε, δ)-differentially private with excess empirical loss $O(\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{\varepsilon^2 N^2}))$ under the strongly convex case by using $g(N, \varepsilon, \delta)$ many gradients, then we can get (ε, δ)-differentially private $\mathcal{A}'$ with excess population loss $O(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon N}))$ by querying gradients at most $\sum_{i=1}^{\lceil \log N\rceil} g(N/2^i, \varepsilon/2^i, \delta/2^i)$ times.

We only prove the bound with privacy guarantee, as the (non-private) bound can be proved with similar argument. Two technical lemmas will be proved at first, after which we will complete the proof.

Lemma 1.5.2. *Let* $\widehat{\omega}_i = \arg\min_{\omega\in\mathcal{K}} \widehat{F}_i(\omega)$, *then*

$$\mathbb{E}[\|\omega_i - \widehat{\omega}_i\|_2^2] \leq O(\frac{G^2\eta_i^2 d\log(1/\delta_i)}{\varepsilon_i^2} + G^2\eta_i^2 N_i).$$

Proof. At first, we prove that $\widehat{\omega}_i \in \mathcal{K}_i$. The definition of $\widehat{\omega}_i$ implies that

$$\frac{1}{N_i}\sum_{j=1}^{N_i} f\left(\widehat{\omega}_i, x_j\right) + \frac{1}{\eta_i N_i}\left\|\widehat{\omega}_i - \omega_{i-1}\right\|_2^2 \leq \frac{1}{N_i}\sum_{j=1}^{N_i} f\left(\omega_{i-1}, x_j\right).$$

Then we know that

$$\frac{1}{\eta_i N_i}\|\widehat{\omega}_i - \omega_{i-1}\|_2^2 \leq G\|\widehat{\omega}_i - \omega_{i-1}\|_2,$$

which implies $\widehat{\omega}_i \in \mathcal{K}_i$.

Next, note that $\widehat{F}_i$ is $\lambda_i = \frac{1}{\eta_i N_i}$-strongly convex, by the guarantee of our ERM algorithm, we know that

$$\begin{aligned}\frac{\lambda_i}{2}\mathbb{E}[\|\widehat{\omega}_i - \omega_i\|_2^2] \leq& \mathbb{E}[\widehat{F}_i(\widehat{\omega}_i) - \widehat{F}_i(\omega_i)]\\ \leq& O(\frac{G^2 d\log(1/\delta_i)}{\lambda_i\varepsilon_i^2 N_i^2} + \frac{G^2}{\lambda_i N_i})\\ =& O(\frac{G^2\eta_i d\log(1/\delta_i)}{\varepsilon_i^2 N_i} + G^2\eta_i),\end{aligned}$$

which implies

$$\mathbb{E}[\|\widehat{\omega}_i - \omega_i\|_2^2] \leq O(\frac{G^2\eta_i^2 d\log(1/\delta_i)}{\varepsilon_i^2} + G^2\eta_i^2 N_i).$$

□

Lemma 1.5.3. *For any $y \in \mathcal{K}$, we know that*

$$\mathbb{E}[F(\widehat{\omega}_i) - F(y)] \leq \frac{\mathbb{E}[\|\omega_{i-1} - y\|_2^2]}{\eta_i N_i} + O(G^2\eta_i).$$

Proof. Let $r(\omega, x) = f(\omega, x) + \frac{1}{\eta_i N_i}\|\omega - \omega_{i-1}\|_2^2$, $R(\omega) = \mathbb{E}_{x\sim\mathcal{P}}\, r(\omega, x)$ and $y^* = \arg\min_{\omega\in\mathcal{K}} R(\omega)$. By Theorem 6 in [SSSSS09], one has that

$$\begin{aligned}\mathbb{E}[R(\widehat{\omega}_i) - R(y)] =& \mathbb{E}[F(\widehat{\omega}_i) + \frac{1}{\eta_i N_i}\|\widehat{\omega}_i - \omega_{i-1}\|_2^2 - F(y) - \frac{1}{\eta_i N_i}\|y - \omega_{i-1}\|_2^2] \\ \leq& \mathbb{E}[R(\widehat{\omega}_i) - R(y^*)] \\ \leq& O(G^2\eta_i),\end{aligned}$$

which implies that

$$\begin{aligned}\mathbb{E}[F(\widehat{\omega}_i) - F(y)] \leq& O(G^2\eta_i) - \frac{1}{\eta_i N_i}\mathbb{E}[\|\widehat{\omega}_i - \omega_{i-1}\|_2^2] + \frac{1}{\eta_i N_i}\mathbb{E}[\|y - \omega_{i-1}\|_2^2] \\ \leq& O(G^2\eta_i) + \frac{1}{\eta_i N_i}\mathbb{E}[\|y - \omega_{i-1}\|_2^2].\end{aligned}$$

□

Having these two lemmas, we can begin the proof.

Proof of Theorem 1.5.1. The privacy guarantee comes directly from the basic composition theorem (See Theorem 1.2.7).

Let $S_i = \{x_j\}_{N-N/2^{i-1}\leq j\leq N-N/2^i}$. Let $N_i = N/2^i, \varepsilon_i = \varepsilon/2^i, \delta_i = \delta/2^i$ and $\eta_i = \eta/2^{5i}$ where η will be defined soon. For $i \in [k]$, let $\widehat{F}_i(\omega) = \sum_{x_j\in S_i} f(\omega, x_j) + \frac{1}{\eta_i N_i}\|\omega - \omega_{i-1}\|_2^2$.

Let $\widehat{\omega}_0 = \omega^*$, we have

$$\mathbb{E}[F(\omega_k)] - F(\omega^*) = \sum_{i=1}^{k}\mathbb{E}[F(\widehat{\omega}_i) - F(\widehat{\omega}_{i-1})] + \mathbb{E}[F(\omega_k) - F(\widehat{\omega}_k)].$$

First, Lemma 1.5.2 implies that

$$\mathbb{E}[F(\omega_k) - F(\widehat{\omega}_k)] \leq O(G\sqrt{\mathbb{E}[\|\omega_k - \widehat{\omega}_k\|_2^2]})$$

$$\leq O(\frac{G^2\eta_k\sqrt{d\log(1/\delta_k)}}{\varepsilon_k} + G^2\eta_k\sqrt{N_k})$$
$$= O(\frac{G^2\eta\sqrt{d\log(N/\delta)}}{\varepsilon N^3} + \frac{G^2\eta}{N^4}),$$

which is negligible.

Then one has

$$\sum_{i=1}^{k}\mathbb{E}[F(\widehat{\omega}_i) - F(\widehat{\omega}_{i-1})] \leq \sum_{i=1}^{k}\frac{\mathbb{E}[\|\widehat{\omega}_{i-1} - \omega_{i-1}\|_2^2]}{\eta_i N_i} + O(G^2\eta_i)$$
$$\leq O(\frac{D^2}{\eta N} + \eta G^2 + \sum_{i=2}^{k}(\frac{G^2\eta_i d\log(1/\delta_i)}{\varepsilon_i^2 N_i} + G^2\eta_i))$$
$$\leq O(\frac{D^2}{\eta N} + \eta G^2 + \frac{G^2\eta d\log(1/\delta)}{\varepsilon^2 N}).$$

By setting $\eta = \frac{D}{G}\cdot\min\{\frac{1}{\sqrt{N}}, \frac{\varepsilon}{\sqrt{d\log(1/\delta)}}\}$, we get the excess population loss:

$$\mathbb{E}[F(\widehat{\omega}_k) - F(\omega^*)] = O(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon}).$$

As for the gradient complexity, as we use $g(N_i, \varepsilon_i, \delta_i)$ queries of gradients in i-th iteration, the total gradient complexity is $\sum_{i=1}^{k} g(N_i, \varepsilon_i, \delta_i)$ as claimed. □

Note that Theorem 1.5.1 allows the ERM algorithm has an extra $G^2/(\mu N)$ loss. This allows us to design a faster ERM algorithm compared Theorem 1.4.9 by choosing a different set of parameters.

Lemma 1.5.4. *Under the assumption defined in Algorithm Private* AC−SA*, with*

$$O\big(N + \min\{\sqrt{\varepsilon}N^{5/4}d^{1/8}, \frac{\varepsilon N^{\frac{3}{2}}}{d^{1/8}\log^{1/4}(1/\delta)}\}\big)$$

gradient complexity, one can get a solution ω_T such that

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O(\frac{G^2}{\mu}(\frac{1}{N} + \frac{d\log(1/\delta)}{\varepsilon^2 N^2})).$$

Proof. By Lemma 1.4.5, one has

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2/B + \sigma^2 d/B^2}{\mu T} + \frac{GDd^{1/4}}{T}\right).$$

Again, setting $\sigma = \frac{c_2 GB\sqrt{T\log(1/\delta)}}{\varepsilon N}$ one has

$$\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] = O\left(\frac{G^2}{\mu BT} + \frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2} + \frac{GDd^{1/4}}{T}\right).$$

Taking $T = 400\lceil\min\{N^{1/2}d^{1/4}, \frac{N\varepsilon}{d^{1/4}\sqrt{\log(1/\delta)}}\}\rceil$ and using $BT \geq N$ (which we will ensure), we have

$$\begin{aligned}\mathbb{E}[\widehat{F}(\omega_T) - \widehat{F}^*] &= O\left(\frac{G^2}{\mu N} + \frac{G^2 d\log(1/\delta)}{\mu\varepsilon^2 N^2} + \frac{GD}{\sqrt{N}} + \frac{GD\sqrt{d\log(1/\delta)}}{\mu\varepsilon N}\right) \\ &= O\left(\frac{G^2}{\mu}\zeta + GD\sqrt{\zeta}\right).\end{aligned}$$

where $\zeta = \frac{1}{N} + \frac{d\log(1/\delta)}{\varepsilon^2 N^2}$.

To ensure that Private AC−SA is (ε, δ)-DP, we set $B = \lceil N/T + N\sqrt{\varepsilon/T}\rceil = \Theta(N/T + N\sqrt{\varepsilon/T})$. By our choice of T, we have $B \leq N/10$ and $\varepsilon \leq c_1 B^2 T/N^2$. Hence, we can apply Lemma 1.4.4 to conclude (ε, δ)-DP.

Hence, we have a (ε, δ)-DP for ERM with loss $O\left(\frac{G^2}{\mu}\zeta + GD\sqrt{\zeta}\right)$ with $\zeta = \frac{1}{N} + \frac{d\log(1/\delta)}{\varepsilon^2 N^2}$. Note however that Theorem 1.5.1 requires us to have a DP-ERM algorithm with loss $O(\frac{G^2}{\mu}\zeta)$, namely, we have the extra term $O(GD\sqrt{\zeta})$. To remove this term, we follow the reduction in Lemma 1.4.7.

We note that the exact same proof as Lemma 1.4.7 shows that for any $\zeta > 0$, if we can solve strongly ERM with loss $O(G^2\zeta^2/\mu + GD\zeta)$, then we can solve strongly ERM with loss $O(G^2\zeta^2/\mu)$ by using the same number of gradient. This completes the proof.

□

Before stating our result on SCO, we need the following variant of Lemma 1.4.7. The proof is essentially the same, we state it for future reference.

Lemma 1.5.5 (Reduction to General Convex Case)**.** *Given F is G-Lipschitz and μ-strongly convex. Suppose for any $\varepsilon, \delta < 1/2$, we have an (ε, δ)-differentially private algorithm $\mathcal{A}$ which takes ω_0 as the initial start point and N samples i.i.d drawn from some distribution $\mathcal{P}$, and outputs a solution ω_T such that*

$$\mathbb{E}[F(\omega_T) - F^*] = O\left(\frac{G^2}{\mu}\left(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2}\right) + GD\left(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon}\right)\right),$$

where $\omega^* = \arg\min_{\omega\in\mathcal{K}} F(\omega)$ and $D = \|\omega_0 - \omega^*\|$. Then by taking $\mathcal{A}$ as sub-procedure with some modifications on parameters, we can get an (ε, δ)-differentially private solution with excess population loss at most

$$\mathbb{E}[F(\omega_T) - F^*] = O\left(\frac{G^2}{\mu}\left(\frac{1}{N} + \frac{d\log(1/\delta)}{N^2\varepsilon^2}\right)\right).$$

If $\mathcal{A}$ uses $g(N, \varepsilon, \delta)$ many gradients, the new algorithm uses $\sum_{i\geq 1} g(N/2^i, \varepsilon/2^i, \delta/2^i)$ many gradients.

Proof. The only difference to Lemma 1.4.7 is that this algorithm takes $N/2^{k+1-i}$ samples instead of N samples in the i-th step for $k = \lceil \log\log N^3 \rceil$, so it may have less gradient complexity. The rest of the proof is identical. □

Now, we can get the result for general convex case by Theorem 1.5.1 and Lemma 1.5.4, then extend it to strongly convex case by Lemma 1.5.5.

Theorem 1.5.6 (DP-SCO, Theorem 1.1.2 restated)**.** *Suppose $\varepsilon, \delta \leq 1/2$. Let $\{f(\cdot, x)\}_{x\in\Xi}$ is convex and G-Lipschitz with respect to ℓ_2 norm and convex over $\mathcal{K}_r$, where $r = \frac{D\sqrt{d\log(1/\delta)}}{\varepsilon N}$, there is an (ε, δ)-differentially private algorithm which takes*

$$O(N + \min\{\sqrt{\varepsilon}N^{5/4}d^{1/8}, \frac{\varepsilon N^{3/2}}{d^{1/8}\log^{1/4}(1/\delta)}\})$$

gradient queries to get a solution ω_T

$$\mathbb{E}[F(\omega_T) - F(\omega^*)] = O(GD(\frac{1}{\sqrt{N}} + \frac{\sqrt{d\log(1/\delta)}}{N\varepsilon}).$$

Moreover, if $\{f(\cdot, x)\}_{x\in\Xi}$ *is also* μ*-strongly convex over* $\mathcal{K}_r$*, we can use the same gradient complexity and get a solution* ω *such that:*

$$\mathbb{E}[F(\omega_T) - F(\omega^*)] = O\left(\frac{G^2}{\mu}(\frac{d\log(1/\delta)}{\varepsilon^2 N^2} + \frac{1}{N})\right).$$

1.6 Proof of Theorem 1.3.1

As mentioned before, we can use the result in [BDRS18] to give a formal proof of our result. Before we start, let us define something necessary.

Definition 1.6.1 (Truncated CDP)**.** Let $\rho > 0$ and $\omega > 1$. A randomized algorithm $\mathcal{M} : \mathbb{N}^{|\Xi|} \to R$ satisfies ω-truncated ρ-concentrated differential privacy (or (ρ, ω)-tCDP) if for all neighboring S, S' that differ in a single entry,

$$\forall\alpha \in (1, \omega), \mathrm{D}_\alpha\left(\mathcal{M}(S) \| \mathcal{M}\left(S'\right)\right) \leq \rho\alpha,$$

where $D_\alpha(\cdot\|\cdot)$ denotes the Rényi divergence [Rén61] of order α (in nats, rather than bits).

Similar to classic differential privacy, tCDP also enjoys a property of composition:

Lemma 1.6.2 (Composition of tCDP)**.** *Let* $\mathcal{M}_1 : \mathbb{N}^{|\Xi|} \to R_1$ *satisfy* (ρ, ω)*-tCDP and let* $\mathcal{M}_2 : \mathbb{N}^{|\Xi|} \times R_1 \to R_2$ *satisfy* (ρ', ω')*-tCDP for all* $y \in R_1$*. Difine* $\mathcal{M} : \mathbb{N}^{|\Xi|} \to R_3$ *by* $\mathcal{M}(S) = \mathcal{M}_2(S, \mathcal{M}_1(S))$*. Then* $\mathcal{M}$ *satisfies* $(\rho + \rho', \min\{\omega, \omega'\})$*-tCDP.*

Now we state the main result of [BDRS18]:

Theorem 1.6.3 (Privacy Amplification By Subsampling)**.** *Let* $\rho, s \in (0, 0.1]$ *and* $B, N \in \mathbb{N}$ *with* $q = B/N$ *and* $\log(1/q) \geq 3\rho(2 + \log_2(1/\rho))$*. Let* $\mathcal{M} : \mathbb{N}^{|\Xi|} \to R$ *satisfy* (ρ, ω')*-tCDP for* $\omega' \geq \frac{\log(1/q)}{2\rho} \geq 3$*. Define the mechanism* $\mathcal{M}_q : \mathbb{N}^{|\Xi|} \to R$ *by* $\mathcal{M}_q(S) = \mathcal{M}(S_q)$ *where* $S_q \in \mathbb{N}^{|\Xi|}$ *is the restriction of* S *to the entries specified by a uniformly ransom subset of size* B*.*

The algorithm $\mathcal{M}_q$ *satisfies* $(13q^2\rho, \omega)$*-tCDP for*

$$\omega = \frac{\log(1/q)}{4\rho}.$$

This theorem can apply to our algorithm $\mathsf{META}_{\mathsf{DP}}$ directly, as we are using subsampling without replacement. More specifically, we are using subsampling Gaussian Mechanism, and for Gaussian Mechanism we have the following fact:

Fact 1.6.4. *Let $P = \mathbb{N}(1, 1/2\rho)$ and $Q = \mathbb{N}(0, 1/2\rho)$. Then $D_\alpha(P \mid Q) = \rho\alpha$ for all $\alpha \in (1, \infty)$. In other word, the Gaussian Mechanism with sensitive 1 satisfies (ρ, ∞)-tCDP.*

Now we can start our proof.

Proof of Theorem 1.3.1. For the t-th phase of $\mathsf{META}_{\mathsf{DP}}$, let $\mathcal{M}(S) = \sum_{x \in S} \nabla f(\omega_{t-1}, x) + v$ where $v \sim \mathbb{N}(0, \sigma^2 I_{d\times d})$. As we are considering G-Lipschitz function f, then we know that $\|\nabla(f)\|_2 \leq G$, which means that $\mathcal{M}$ is (ρ, ∞)-tCDP where $\rho = G^2/(2\sigma^2)$.

Assume our parameters satisfy the precondition of Theorem 1.6.3 first, then we know that the t-th phase of $\mathsf{META}_{\mathsf{DP}}$ is $(13q^2\rho, 1/\rho)$-tCDP. By the composition property (Lemma 1.6.2), we know that $\mathsf{META}_{\mathsf{DP}}$ is $(13Tq^2\rho, 1/\rho)$-tCDP.

When $Tq^2\rho \cdot \frac{\log(1/\delta)}{\varepsilon} \leq O(\varepsilon)$ and $\frac{\log(1/\delta)}{\varepsilon} \leq O(\frac{1}{\rho})$, we know that $\mathsf{META}_{\mathsf{DP}}$ is (ε, δ)-differentially private [BDRS18].

By setting $\sigma = \frac{c_2 GB\sqrt{T\log(1/\delta)}}{\varepsilon N}$, we have that $\rho = \frac{\varepsilon^2 N^2}{2c_2^2 B^2 T \log(1/\delta)}$. Together with the assumption $\varepsilon \leq c_1 B^2 T/N^2$, we have both $Tq^2\rho \cdot \frac{\log(1/\delta)}{\varepsilon} \leq O(\varepsilon)$ and $\frac{\log(1/\delta)}{\varepsilon} \leq O(\frac{1}{\rho})$ as claimed. This completes the proof.

□

Chapter 2

RESQUEING PARALLEL AND PRIVATE STOCHASTIC CONVEX OPTIMIZATION

2.1 Introduction

Stochastic convex optimization (SCO) is a foundational problem in optimization theory, machine learning, theoretical computer science, and modern data science. Variants of the problem underpin a wide variety of applications in machine learning, statistical inference, operations research, signal processing, and control and systems engineering [Sha07, SB14]. Moreover, it provides a fertile ground for the design and analysis of scalable optimization algorithms such as the celebrated stochastic gradient descent (SGD), which is ubiquitous in machine learning practice [Bot12].

SGD approximately minimizes a function $f : \mathbb{R}^d \to \mathbb{R}$ by iterating $x_{t+1} \leftarrow x_t - \eta g(x_t)$, where $g(x_t)$ is an unbiased estimator to a (sub)gradient of f at iterate x_t. When f is convex, $\mathbb{E}\,\|g(x)\|^2 \le 1$ for all x and f is minimized at $x^\star$ in the unit ball, SGD finds an ε_{opt}-optimal point (i.e. x satisfying $\mathbb{E}\, f(x) \le f(x^\star) + \varepsilon_{\text{opt}}$) using $O(\varepsilon_{\text{opt}}^{-2})$ stochastic gradient evaluations [Bub15]. This complexity is unimprovable without further assumptions [Duc18]; for sufficiently large d, this complexity is optimal even if g is an exact subgradient of f [DG19].

Although SGD is widely-used and theoretically optimal in this simple setting, the algorithm in its basic form has natural limitations. For example, when parallel computational resources are given (i.e. multiple stochastic gradients can be queried in batch), SGD has suboptimal sequential depth in certain regimes [DBW12, BJL+19]. Further, standard SGD is not differentially private, and existing private[1] SCO algorithms are not as efficient as SGD in terms of gradient evaluation complexity [BST14, BFTGT19, FKT20, BFGT20, AFKT21,

[1]Throughout this paper, when we use the description "private" without further description we always refer to differential privacy [DR14]. For formal definitions of differential privacy, see Section 2.4.1.

KLL21]. Despite substantial advances in both the parallel and private settings, the optimal complexity of each SCO problem remains open (see Sections 2.1.1 and 2.1.2 for more precise definitions of problem settings and the state-of-the-art rates, and Section 2.1.3 for a broader discussion of related work).

Though seemingly disparate at first glance, in spirit parallelism and privacy impose similar constraints on effective algorithms. Parallel algorithms must find a way to query the oracle multiple times (possibly at multiple points) without using the oracle's output at these points to determine where they were queried. In other words, they cannot be too reliant on a particular outcome to adaptively choose the next query. Likewise, private algorithms must make optimization progress without over-relying on any individual sample to determine the optimization trajectory. In both cases, oracle queries must be suitably robust to preceding oracle outputs.

In this paper, we provide a new stochastic gradient estimation tool which we call *Reweighted Stochastic Query (ReSQue) estimators* (defined more precisely in Section 2.1.4). ReSQue is essentially an efficient parallel method for computing an unbiased estimate of the gradient of a convolution of f with a continuous (e.g. Gaussian) kernel. These estimators are particularly well-suited for optimizing a convolved function over small Euclidean balls, as they enjoy improved stability properties over these regions. In particular, these local stability properties facilitate tighter control over the stability of SGD-like procedures. We show that careful applications of ReSQue in conjunction with recent advances in accelerated ball-constrained optimization [CJJ+20, ACJ+21] yield complexity improvements for both parallel and private SCO.

Paper organization. In Sections 2.1.1 and 2.1.2 respectively, we formally describe the problems of parallel and private SCO we study, stating our results and contextualizing them in the prior literature. We then cover additional related work in Section 2.1.3 and, in Section 2.1.4, give an overview of our approach to obtaining these results. In Section 2.1.5, we describe the notation we use throughout.

In Section 2.2.1 we introduce our ReSQue estimator and prove some of its fundamental properties. In Section 2.2.2 we describe our adaptation of the ball acceleration frameworks

of [ACJ+21, CH22], reducing SCO to minimizing the objective over small Euclidean balls, subproblems which are suitable for ReSQue-based stochastic gradient methods. Finally, in Sections 2.3 and 2.4, we prove our main results for parallel and private SCO (deferring problem statements to Problem 2.3.1 and Problem 2.4.1), respectively, by providing suitable implementations of our ReSQue ball acceleration framework.

2.1.1 Parallelism

In Section 2.3 we consider the following formulation of the SCO problem, simplified for the purposes of the introduction. We assume there is a convex function $f : \mathbb{R}^d \to \mathbb{R}$ which can be queried through a *stochastic gradient oracle* g, satisfying $\mathbb{E}\, g \in \partial f$ and $\mathbb{E}\, \|g\|^2 \leq 1$. We wish to minimize the restriction of f to the unit Euclidean ball to expected additive error ε_{opt}. In the standard sequential setting, SGD achieves this goal using roughly $\varepsilon_{\text{opt}}^{-2}$ queries to g; as previously mentioned, this complexity is optimal. A generalization of this formulation is restated in Problem 2.3.1 with a variance bound L^2 and a radius bound R, which are both set to 1 here.

In settings where multiple machines can be queried simultaneously, the parallel complexity of an SCO algorithm is a further important measure for consideration. In [Nem94], this problem was formalized in the setting of oracle-based convex optimization, where the goal is to develop iterative methods with a number of parallel query batches to g. In each batch, the algorithm can submit polynomially many queries to g in parallel, and then perform computations (which do not use g) on the results. The *query depth* of a parallel algorithm in the [Nem94] model is the number of parallel rounds used to query g, and was later considered in stochastic algorithms [DBW12]. Ideally, a parallel SCO algorithm will also have bounded *total queries* (the number of overall queries to g), and bounded *computational depth*, i.e. the parallel depth used by the algorithm outside of oracle queries. We discuss these three complexity measures more formally in Section 2.3.1.

In the low-accuracy regime $\varepsilon_{\text{opt}} \geq d^{-1/4}$, recent work [BJL+19] showed that SGD indeed achieves the optimal oracle query depth among parallel algorithms.[2] Moreover, in the

[2]We omit logarithmic factors when discussing parameter regimes throughout the introduction.

Method	g query depth	computational depth	# g queries
SGD [Nes18]	ε^{-2}	ε^{-2}	ε^{-2}
[DBW12]	$d^{\frac{1}{4}}\varepsilon^{-1}$	$d^{\frac{1}{4}}\varepsilon^{-1}$	$d^{\frac{1}{4}}\varepsilon^{-1} + \varepsilon^{-2}$
[BJL+19]	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}}$	$d^{\frac{4}{3}}\varepsilon^{-\frac{8}{3}}$	$d^{\frac{4}{3}}\varepsilon^{-\frac{8}{3}}$
CPM [KTE88]	d	d	d
BallAccel + EpochSGD (Theorem 2.3.2)	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}}$	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}} + \varepsilon^{-2}$	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}} + \varepsilon^{-2}$
BallAccel + AC-SA (Theorem 2.3.3)	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}}$	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}} + d^{\frac{1}{4}}\varepsilon^{-1}$	$d^{\frac{1}{3}}\varepsilon^{-\frac{2}{3}} + \varepsilon^{-2}$

Table 2.1: **Comparison of parallel SCO results.** The complexity of finding a point with expected error $\varepsilon := \varepsilon_{\text{opt}}$ in Problem 2.3.1, where $L = R = 1$. We hide polylogarithmic factors in d and ε^{-1}.

high-accuracy regime $\varepsilon_{\text{opt}} \le d^{-1}$, cutting plane methods (CPMs) by e.g. [KTE88] (see [JLSW20] for an updated overview) achieve the state-of-the-art oracle query depth of d, up to logarithmic factors in $d, \varepsilon_{\text{opt}}$.

In the intermediate regime $\varepsilon_{\text{opt}} \in [d^{-1}, d^{-1/4}]$, [DBW12, BJL+19] designed algorithms with oracle query depths that improved upon SGD, as summarized in Table 2.1. In particular, [BJL+19] obtained an algorithm with query depth $\widetilde{O}(d^{1/3}\varepsilon_{\text{opt}}^{-2/3})$, which they conjectured is optimal for intermediate ε_{opt}. However, the total oracle query complexity of [BJL+19] is $\widetilde{O}(d^{4/3}\varepsilon_{\text{opt}}^{-8/3})$, a (fairly large) polynomial factor worse than SGD.

Our results. The main result of Section 2.3 is a pair of improved parallel algorithms in the setting of Problem 2.3.1. Both of our algorithms achieve the "best of both worlds" between the [BJL+19] parallel algorithm and SGD, in that their oracle query depth is bounded by $\widetilde{O}(d^{1/3}\varepsilon_{\text{opt}}^{-2/3})$ (as in [BJL+19]), but their total query complexity matches SGD's in the regime $\varepsilon_{\text{opt}} \le d^{-1/4}$. We note that $\varepsilon_{\text{opt}} \le d^{-1/4}$ is the regime where a depth of $\widetilde{O}(d^{1/3}\varepsilon_{\text{opt}}^{-2/3})$ improves upon [DBW12] and SGD. Our guarantees are formally stated in Theorems 2.3.2 and 2.3.3, and summarized in Table 2.1.

Our first algorithm (Theorem 2.3.2) is based on a batched SGD using our ReSQue es-

timators, within the "ball acceleration" framework of [ACJ+21] (see Section 2.1.4). By replacing SGD with an accelerated counterpart [GL12], we obtain a further improved *computational depth* in Theorem 2.3.3. Theorem 2.3.3 simultaneously achieves the query depth of [BJL+19], the computational depth of [DBW12], and the total query complexity of SGD in the intermediate regime $\varepsilon_{\text{opt}} \in [d^{-1}, d^{-1/4}]$.

2.1.2 Differential privacy

Differential privacy (DP) is a mathematical quantification for privacy risks in algorithms involving data. When performing stochastic convex optimization with respect to a sampled dataset from a population, privacy is frequently a natural practical desideratum [BST14, EPK14, Abo16, App17]. For example, the practitioner may want to privately learn a linear classifier or estimate a regression model or a statistical parameter from measurements.

In this paper, we obtain improved rates for private SCO in the following model, which is standard in the literature and restated in Problem 2.4.1 in full generality. Symmetrically to the previous section, in the introduction, we only discuss the specialization of Problem 2.4.1 with $L = R = 1$, where L is a Lipschitz parameter and R is a domain size bound. We assume there is a distribution dist over a population $\mathcal{S}$, and we obtain independent samples $\{s_i\}_{i\in[n]} \sim \text{dist}$. Every element $s \in \mathcal{S}$ induces a 1-Lipschitz convex function $f(\cdot; s)$, and the goal of SCO is to approximately optimize the population loss $F_{\mathcal{P}} := \mathbb{E}_{s\sim\text{dist}}[f(\cdot; s)]$. The setting of Problem 2.4.1 can be viewed as a specialization of Problem 2.3.1 which is more compatible with the notion of DP, discussed in more detail in Section 2.4.1.

The cost of achieving approximate DP with privacy loss parameter ε_{dp} (see Section 2.4.1 for definitions) has been studied by a long line of work, starting with [BST14]. The optimal error (i.e. excess population loss) given n samples scales as (omitting logarithmic factors)

$$\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\varepsilon_{\text{dp}}}, \tag{2.1}$$

with matching lower and upper bounds given by [BST14] and [BFTGT19], respectively. The $n^{-1/2}$ term is achieved (without privacy considerations) by simple one-pass SGD, i.e. treating sample gradients as unbiased for the population loss, and discarding samples after

we query their gradients. Hence, the term $\sqrt{d} \cdot (n\varepsilon_{\text{dp}})^{-1}$ can be viewed as the "cost of privacy" in SCO. Assuming that we have access to $n \geq d\varepsilon_{\text{dp}}^{-2}$ samples is then natural, as this is the setting where privacy comes at no asymptotic cost from the perspective of the bound (2.1). Moreover, many real-world problems in data analysis have low intrinsic dimension, meaning that the effective number of degrees of freedom in the optimization problem is much smaller than the ambient dimension [SSTT21, LLH+22], which can be captured via a dimension-reducing preprocessing step. For these reasons, we primarily focus on the regime when the number of samples n is sufficiently large compared to d.

An unfortunate property of private SCO algorithms achieving error (2.1) is they all query substantially more than n sample gradients without additional smoothness assumptions [BST14, BFTGT19, FKT20, BFGT20, AFKT21, KLL21], which can be viewed as a statistical-computational gap. For example, analyses of simple perturbed SGD variants result in query bounds of $\approx n^2$ [BFGT20]. In fact, [BFGT20] conjectured this quadratic complexity was necessary, which was disproven by [AFKT21, KLL21]. The problem of obtaining the optimal error (2.1) using n gradient queries has been repeatedly highlighted as an important open problem by the private optimization community, as discussed in [BFGT20, AFKT21, KLL21, ACJ+21] as well as the recent research overview [Tal22].

Qualitatively, optimality of the bound (2.1) shows that there is no statistical cost of privacy when the number of samples n is large enough, as the solver relies less on any specific sample. A natural first step towards developing optimal private SCO algorithms is to ask a similar qualitative question regarding their computational guarantees. Concretely, given enough samples n, can we develop statistically-optimal SCO algorithms which only query $\approx n$ sample gradients?

Our results. In Section 2.4, we develop the first private SCO algorithm with this aforementioned computational guarantee. Our algorithm achieves the error bound (2.1) up to logarithmic factors, as well as a new gradient query complexity. Our result is formally stated in Theorem 2.4.22 and summarized in Table 2.2 and Figure 2.1. Up to logarithmic factors,

our gradient query complexity is

$$\min\left(n, \frac{n^2\varepsilon_{\mathrm{dp}}^2}{d}\right) + \min\left(\frac{(nd)^{\frac{2}{3}}}{\varepsilon_{\mathrm{dp}}}, n^{\frac{4}{3}}\varepsilon_{\mathrm{dp}}^{\frac{1}{3}}\right).$$

Theorem 2.4.22 improves upon the prior state-of-the-art gradient query complexity by polynomial factors whenever $d \ll n^{4/3}$ (omitting $\varepsilon_{\mathrm{dp}}$ dependencies for simplicity). As with prior recent SCO advancements, our result has the appealing property that it achieves the optimal $n^{-1/2}$ error for SCO when $n \gtrsim d\varepsilon_{\mathrm{dp}}^{-2}$. Moreover, given $n \gtrsim d^2\varepsilon_{\mathrm{dp}}^{-3}$ samples, the gradient query complexity of Theorem 2.4.22 improves to $\widetilde{O}(n)$, the first near-linear query complexity for a statistically-optimal private SCO algorithm in any regime. In Table 2.2 and Figure 2.1, we compare our bounds with the prior art.

While there remains a gap between the sample complexity at which our algorithm is statistically optimal, and that at which it is computationally (nearly)-optimal, we find it promising that our result comes within logarithmic factors of achieving the best-of-both-worlds for sufficiently large n. This is a key step towards optimal algorithms for the fundamental problem of private SCO. It is an interesting open question to refine current algorithmic techniques for private SCO to remove this gap, and we are optimistic that the tools developed in this paper will be fruitful in this endeavor.

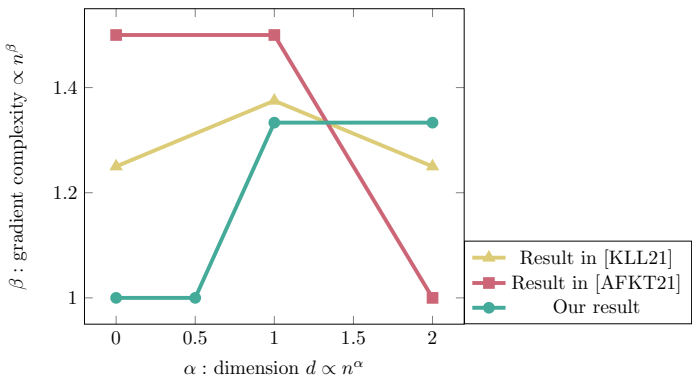


Figure 2.1: Comparison among our gradient complexity and previous results in [AFKT21, KLL21] for the non-trivial regime $d \leq n^2$. We omit dependencies on $\varepsilon_{\mathrm{dp}}$ (treated as $\Theta(1)$ in this figure) and logarithmic terms for simplicity.

Method	excess $F_{\mathcal{P}}$ loss	# gradient queries to samples
[BST14]	$\frac{\sqrt[4]{d}\log\frac{n}{\delta}}{\sqrt{n}} + \frac{\sqrt{d}\log^2\frac{n}{\delta}}{n\varepsilon}$	n^2
[BFTGT19]	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}$	$n^{\frac{9}{2}}$
[FKT20]	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}$	n^2
[BFGT20]	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}$	n^2
[AFKT21]	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}$	$\min\left(n^{\frac{3}{2}}, \frac{n^2\varepsilon}{\sqrt{d}}\right)$
[KLL21]	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}$	$\min\left(n^{\frac{5}{4}}d^{\frac{1}{8}}\sqrt{\varepsilon}, \frac{n^{\frac{3}{2}}\varepsilon}{d^{\frac{1}{8}}}\right)$
Theorem 2.4.22	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}\log n\log^{1.5}\frac{n}{\delta}}{n\varepsilon}$	$\min\left(n, \frac{n^2\varepsilon^2}{d}\right) + \min\left(\frac{(nd)^{\frac{2}{3}}}{\varepsilon}, n^{\frac{4}{3}}\varepsilon^{\frac{1}{3}}\right)$

Table 2.2: **Comparison of private SCO results.** The excess loss and gradient complexity of $(\varepsilon := \varepsilon_{\text{dp}}, \delta)$-DP in Problem 2.4.1, where $L = R = 1$. We hide polylogarithmic factors in $d, n, \delta^{-1}, \varepsilon^{-1}$ in the third column. The optimal loss [BST14, SU15] is achieved by rows 2-6.

2.1.3 Related work

Stochastic convex optimization. Convex optimization is a fundamental task with numerous applications in computer science, operations research, and statistics [BV14, Bub15, Nes18], and has been the focus of extensive research over the past several decades. This paper's primary setting of interest is non-smooth (Lipschitz) stochastic convex optimization in private and parallel computational models. Previously, [Gol64] gave a gradient method that used $O(\varepsilon^{-2})$ gradient queries to compute a point achieving ε error for Lipschitz convex minimization. This rate was shown to be optimal in an information-theoretic sense in [NY83]. The stochastic gradient descent method extends [Gol64] to tolerate randomized, unbiased gradient oracles with bounded second moment: this yields algorithms for Problem 2.3.1 and Problem 2.4.1 (when privacy is not a consideration).

Acceleration. Since the first proposal of accelerated (momentum-based) methods [Pol64, Nes83, Nes03], acceleration has become a central topic in optimization. This work builds

on the seminal Monteiro-Svaiter acceleration technique [MS13] and its higher-order variants [GDG+19, BJL+19]. More specifically, our work follows recent developments in accelerated ball optimization [CJJ+20, CJJS21, ACJ+21], which can be viewed as a limiting case of high-order methods. Our algorithms directly leverage error-robust variants of this framework developed by [ACJ+21, CH22].

Parallel SCO. Recently, parallel optimization has received increasing interest in the context of large-scale machine learning. Speeding up SGD by averaging stochastic gradients across mini-batches is extremely common in practice, and optimal in certain distributed optimization settings; see e.g. [DGBSX12, DRY18, WBSS21]. Related to the setting we study are the distributed optimization methods proposed in [SBB+18], which also leverage convolution-based randomized smoothing and apply to both stochastic and deterministic gradient-based methods (but do not focus on parallel depth in the sense of [Nem94]). Finally, lower bounds against the oracle query depth of parallel SCO algorithms in the setting we consider have been an active area of study, e.g. [Nem94, BS18, DG19, BJL+19].

Private SCO. Both the private stochastic convex optimization problem (DP-SCO) and the private empirical risk minimization problem (DP-ERM) are well-studied by the DP community [CM08, RBHT12, CMS11, JT14, BST14, KJ16, FTS17, ZZMW17, Wan18, INS+19, BFTGT19, FKT20]. In particular, [BST14] shows that the exponential mechanism and noisy stochastic gradient descent achieve the optimal loss for DP-ERM for $(\varepsilon_{\text{dp}}, 0)$-DP and $(\varepsilon_{\text{dp}}, \delta)$-DP. In follow-up works, [BFTGT19, FKT20] show that one can achieve the optimal loss for DP-SCO as well, by a suitable modification of noisy stochastic gradient descent. However, these algorithms suffer from large (at least quadratic in n) gradient complexities. Under an additional assumption that the loss functions are sufficiently smooth (i.e. have Lipschitz gradient), [FKT20] remedies this issue by obtaining optimal loss and optimal *gradient complexity* under differential privacy. In a different modification of Problem 2.4.1's setting (where sample function access is modeled through value oracle queries instead of subgradients), [GLL22] designs an exponential mechanism-based method that uses the optimal value oracle complexity to obtain the optimal SCO loss for non-smooth functions.

Most directly related to our approach are the recent works [KLL21] and [ACJ+21]. Both propose methods improving upon the quadratic gradient complexity achieved by noisy SGD, by using variants of smoothing via Gaussian convolution. The former proposes an algorithm that uses noisy accelerated gradient descent for private SCO with subquadratic gradient complexity. The latter suggests a ball acceleration framework to solve private SCO with linear gradient queries, under a hypothetical algorithm to estimate subproblem solutions. Our work can be viewed as a formalization of the connection between ball acceleration strategies and private SCO as suggested in [ACJ+21], by way of ReSQue estimators, which we use to obtain improved query complexities.

2.1.4 Our approach

Here we give an overview of our approach towards obtaining the results outlined in Section 2.1.1 and Section 2.1.2. To illustrate and situate our approach, we first briefly discuss prior approaches, their insights that we leverage, and obstacles that we overcome. Then we discuss a common framework based on a new stochastic gradient estimation tool we introduce and call *Reweighted Stochastic Query (ReSQue) estimators* which enables our results on parallel and private SCO. Our new tool is naturally compatible with ball-constrained optimization frameworks, where an optimization problem is localized to a sequence of constrained subproblems (solved to sufficient accuracy), whose solutions are then stitched together. We exploit this synergy, as well as the local stability properties of our ReSQue estimators, to design our SCO algorithms. We discuss the different instantiations of our framework for parallel and private SCO at the end of this section.

Convolutions and prior approaches. All new results on parallel and private SCO in this paper use the convolution of a function of interest $f : \mathbb{R}^d \to \mathbb{R}$ with a Gaussian density γ_ρ (with covariance $\rho^2 \mathbf{I}_d$), which we denote by $\widehat{f}_\rho$. Such *Gaussian convolutions* have a longer history of facilitating algorithmic advances for SCO. All previous advances on parallel SCO and Lipschitz convex function minimization used Gaussian convolutions, i.e. [DBW12, BJL+19], as did a state-of-the-art (in some regimes) private SCO algorithm [KLL21]. Each of [DBW12, KLL21] leverage that $\widehat{f}_\rho$ is a smooth, additive approximation

to f, and [BJL+19] further used that the higher derivatives of $\widehat{f}_\rho$ are bounded, as well as the fact that its gradients can be well-approximated within small balls.

As one of our motivating problems, we seek to move beyond the reliance on (high-order) smoothness properties of $\widehat{f}_\rho$, and achieve total work bounds improving upon [BJL+19]. Unfortunately, doing so while following the strategy of [BJL+19] poses an immediate challenge. Though [BJL+19] achieves improved parallel depth bounds for Lipschitz convex optimization, it comes at a cost. Their approach, which relies on the p^{th}-order Lipschitzness of $\widehat{f}_\rho$, would naively involve computing p^{th} derivatives of the objective, and their approach to gradient approximation involves estimating the gradient everywhere inside a ball of sufficient radius. Naively, either of these approaches would involve making $\Omega(d)$ queries per parallel step. Removing this cost is one of our main contributions to parallel SCO, and our corresponding development is key to enabling our private SCO results.

ReSQue estimators and ball acceleration. To overcome this bottleneck to prior approaches, we introduce a new tool that capitalizes upon a different property of Gaussian convolutions: the fact that the Gaussian density is locally stable in a small ball around its center. This property is arguably closely related to how [BJL+19] are able to prove that they can approximate the gradients of $\widehat{f}_\rho$ inside a ball. However, rather than building such a complete model of $\widehat{f}_\rho$, we instead use only use this property to suitably implement independent stochastic gradient queries to $\widehat{f}_\rho$.

Given a reference point $\overline{x}$ and a query point x, our proposed estimator for $\nabla \widehat{f}_\rho(x)$ is outputting

$$\frac{\gamma_\rho(x - \overline{x} - \xi)}{\gamma_\rho(\xi)} g(\overline{x} + \xi), \tag{2.2}$$

where $\xi \sim \mathcal{N}(0, \rho^2 \mathbf{I}_d)$, and $g(z)$ is an unbiased estimate for a subgradient of f, i.e., $\mathbb{E}\, g(z) \in \partial f(z)$. That is, to estimate the gradient of $\widehat{f}_\rho$, we simply reweight (stochastic) gradients of f that were queried at random perturbations of reference point $\overline{x}$. This reweighted stochastic query (ReSQue) estimator is unbiased for $\nabla \widehat{f}_\rho(x)$, regardless of $\overline{x}$. However, when $\|x - \overline{x}\| \ll \rho$, i.e. x is contained in a small ball around $\overline{x}$, the reweighting factor $\frac{\gamma_\rho(x-\overline{x}-\xi)}{\gamma_\rho(\xi)}$ is likely to be close to 1. As a result, when g is bounded, and x is near $\overline{x}$, the estimator

(2.2) enjoys regularity properties such as moment bounds. Crucially, the stochastic gradient queries performed by ReSQue (at points of the form $\overline{x} + \xi$) *do not depend* on the point x at which we eventually estimate the gradient.

We develop this theory in Section 2.2, but mention one additional property here, which can be thought of as a "relative smoothness" property. We show that when $\|x - x'\|$ is sufficiently smaller than ρ, the *difference* of estimators of the form (2.2) has many bounded moments, where bounds scale as a function of $\|x - x'\|$. When we couple a sequence of stochastic gradient updates by the randomness used in defining (2.2), we can use this property to bound how far sequences drift apart. In particular, initially nearby points are likely to stay close. We exploit this property when analyzing the stability of private stochastic gradient descent algorithms later in the paper.

To effectively use these local stability properties of (2.2), we combine them with an optimization framework called *ball-constrained optimization* [CJJ+20]. It is motivated by the question: given parameters $0 < r < R$, and an oracle which minimizes $f : \mathbb{R}^d$ in a ball of radius r around an input point, how many oracles must we query to optimize f in a ball of larger radius R? It is not hard to show that simply iterating calls to the oracle gives a good solution in roughly $\frac{R}{r}$ queries. In recent work, [CJJ+20] demonstrated that the optimal number of calls scales (up to logarithmic factors) as $(\frac{R}{r})^{2/3}$, and [ACJ+21] gave an approximation-tolerant variant of the [CJJ+20] algorithm. We refer to these algorithms as *ball acceleration*. Roughly, [ACJ+21] shows that running stochastic gradient methods on $\approx (\frac{R}{r})^{2/3}$ subproblems constrained to balls of radius r obtains total gradient query complexity comparable to directly running SGD on the global function of domain radius R.

Importantly, in many structured cases, we have dramatically more freedom in solving these subproblems, compared to the original optimization problem, since we are only required to optimize over a small radius. One natural form of complexity gain from ball acceleration is when there is a much cheaper gradient estimator, which is only locally defined, compared to a global estimator. This was the original motivation for combining ball acceleration with stochastic gradient methods in [CJJS21], which exploited local smoothness of the softmax function; the form of our ReSQue estimator (2.2) is motivated by the [CJJS21] estimator. In this work, we show that using ReSQue with reference point $\overline{x}$ signif-

icantly improves the parallel and private complexity of minimizing the convolution $\widehat{f}_\rho$ inside a ball of radius $r \approx \rho$ centered at $\overline{x}$.

Parallel subproblem solvers. A key property of the ReSQue estimator (2.2) is that its estimate of $\nabla \widehat{f}_\rho(x)$ is a scalar reweighting of $g(\overline{x} + \xi)$, where $\xi \sim N(0, \rho^2 \mathbf{I}_d)$ and $\overline{x}$ is a fixed reference point. Hence, in each ball subproblem (assuming $r = \rho$), we can make *all* the stochastic gradient queries in parallel, and use the resulting pool of vectors to perform standard (ball-constrained) stochastic optimization using ReSQue. Thus, we solve each ball subproblem with a single parallel stochastic gradient query, and — using ball acceleration — minimize $\widehat{f}_\rho$ with query depth of roughly $\rho^{-2/3}$. To ensure that $\widehat{f}_\rho$ is a uniform ε_{opt}-approximation of the original f, we must set ρ to be roughly $\varepsilon_{\text{opt}}/\sqrt{d}$, leading to the claimed $d^{1/3}\varepsilon_{\text{opt}}^{-2/3}$ depth bound. Furthermore, the ball acceleration framework guarantees that we require no more than roughly $\rho^{-2/3} + \varepsilon_{\text{opt}}^{-2}$ stochastic gradient computations throughout the optimization, yielding the claimed total query bound. However, the computational depth of the algorithm described thus far is roughly $\varepsilon_{\text{opt}}^{-2}$, which is no better than SGD. In Section 2.3 we combine our approach with the randomized smoothing algorithm of [DBW12] by using an accelerated mini-batched method [GL12] for the ball-constrained stochastic optimization, leading to improved computational depth as summarized in Table 2.1. Our parallel SCO results use the ReSQue/ball acceleration technique in a simpler manner than our private SCO results described next and in Section 2.4, so we chose to present them first.

Private subproblem solvers. To motivate our improved private SCO solvers, we make the following connection. First, it is straightforward to show that the convolved function $\widehat{f}_\rho$ is $\frac{1}{\rho}$-smooth whenever the underlying function f is Lipschitz. Further, recently [FKT20] obtained a linear gradient query complexity for SCO, under the stronger assumption that each sample function (see Problem 2.4.1) is $\lesssim \sqrt{n}$-smooth (for $L = R = 1$ in Problem 2.4.1). This bound is satisfied by the result of Gaussian convolution with radius $\frac{1}{\sqrt{n}}$; however, two difficulties arise. First, to preserve the function value approximately up to ε_{opt}, we must take a Gaussian convolution of radius $\rho \approx \frac{\varepsilon_{\text{opt}}}{\sqrt{d}}$. For ε_{opt} in (2.1), this is much smaller than $\frac{1}{\sqrt{n}}$ in many regimes. Second, we cannot access the exact gradients of the convolved

sampled functions. Hence, it is natural to ask: is there a way to simulate the smoothness of the convolved function, under stochastic query access?

Taking a step back, the primary way in which [FKT20] used the smoothness assumption was through the fact that gradient steps on a sufficiently smooth function are *contractive*. This observation is formalized as follows: if $x' \leftarrow x - \eta \nabla f(x)$ and $y' \leftarrow y - \eta \nabla f(y)$, when f is $O(\frac{1}{\eta})$-smooth, then $\|x' - y'\| \leq \|x - y\|$. As alluded to earlier, we show that ReSQue estimators (2.2) allow us to simulate this contractivity up to polylogarithmic factors. We show that by coupling the randomness ξ in the estimator (2.2), the drift growth in two-point sequences updated with (2.2) is predictable. We give a careful potential-based argument (see Lemma 2.4.7) to bound higher moments of our drift after a sequence of updates using ReSQue estimators, when they are used in an SGD subroutine over a ball of radius $\ll \rho$. This allows for the use of "iterative localization" strategies introduced by [FKT20], based on iterate perturbation via the Gaussian mechanism.

We have not yet dealt with the fact that while this "smoothness simulation" strategy allows us to privately solve *one* constrained ball subproblem, we still need to solve $K \approx (\frac{1}{r})^{2/3}$ ball subproblems to optimize our original function, where $r \ll \rho$ is the radius of each subproblem. Here we rely on arguments based on amplification by subsampling, a common strategy in the private SCO literature [ACG+16, BBG18]. We set our privacy budget for each ball subproblem to be approximately $(\varepsilon_{\mathrm{dp}}, \delta)$ (our final overall budget), before subsampling. We then use solvers by suitably combining the [FKT20] framework and our estimator (2.2) to solve these ball subproblems using $\approx n \cdot K^{-1/2}$ gradient queries each. Finally, our algorithm obtains the desired query complexity:

$$\approx \underbrace{\frac{n}{\sqrt{K}}}_{\text{gradient queries per subproblem}} \cdot \underbrace{K}_{\text{number of subproblems}} = n\sqrt{K},$$

and privacy:

$$\approx \underbrace{\varepsilon_{\mathrm{dp}}}_{\text{privacy budget per subproblem}} \cdot \underbrace{\frac{1}{\sqrt{K}}}_{\text{subsampling}} \cdot \underbrace{\sqrt{K}}_{\text{advanced composition}} = \varepsilon_{\mathrm{dp}}.$$

Here we used the standard technique of advanced composition (see e.g. Section 3.5.2, [DR14]) to bound the privacy loss over K consecutive ball subproblems.

Let us briefly derive the resulting complexity bound and explain the bottleneck for improving it further. First, the ball radius r must be set to $\approx \rho$ (the smoothing parameter) for our ReSQue estimators to be well-behaved. Moreover, we have to set $\rho \approx \frac{\varepsilon_{\text{opt}}}{\sqrt{d}}$, otherwise the effect of the convolution begins to dominate the optimization error. For $\varepsilon_{\text{opt}} \approx \frac{1}{\sqrt{n}} + \sqrt{d}(n\varepsilon_{\text{dp}})^{-1}$ (see (2.1)), this results in $\frac{1}{r} \approx \min(\sqrt{nd}, n\varepsilon_{\text{dp}})$. Next, $K \approx (\frac{1}{r})^{2/3}$ is known to be essentially tight for ball acceleration with $R = 1$ [CJJ+20]. For the subproblem accuracies required by the [ACJ+21] ball acceleration framework,[3] known lower bounds on private empirical risk minimization imply that $\approx \frac{n}{\sqrt{K}}$ gradients are necessary for each subproblem to preserve a privacy budget of ε_{dp} [BST14]. As subsampling requires the privacy loss before amplification to already be small (see discussion in [Smi09, BBG18]), all of these parameter choices are optimized, leading to a gradient complexity of $n\sqrt{K}$. For our lower bound on $\frac{1}{r}$, this scales as $\approx \min(n^{4/3}, (nd)^{2/3})$ as we derive in Theorem 2.4.22.[4] To go beyond the strategies we employ, it is natural to look towards other privacy amplification arguments (for aggregating ball subproblems) beyond subsampling, which we defer to future work.

Our final algorithm is analyzed through the machinery of Rényi differential privacy (RDP) [Mir17], which allows for more fine-grained control of the effects of composition and subsampling. We modify the standard RDP machinery in two main ways. We define an approximate relaxation and control the failure probability of our relaxation using high moment bounds on our drift (see Section 2.4.2). We also provide an analysis of amplification under subsampling with replacement by modifying the truncated CDP (concentrated DP) tools introduced by [BDRS18], who analyzed subsampling without replacement. Sampling with replacement is crucial in order to guarantee that our ReSQue estimators are unbiased for the empirical risks we minimize when employing a known reduction [FKT20, KLL21] from private SCO to private regularized empirical risk minimization.

---

[3] These subproblem accuracy requirements cannot be lowered in general, because combined they recover the optimal gradient complexities of SGD over the entire problem domain.

[4] In the low-dimensional regime $d \le n\varepsilon_{\text{dp}}^2$, the gradient queries used per subproblem improves to $\frac{\sqrt{nd}}{\varepsilon_{\text{dp}}\sqrt{K}}$.

2.1.5 Notation

Throughout $\widetilde{O}$ hides polylogarithmic factors in problem parameters. For $n \in \mathbb{N}$, we let $[n] := \{i \mid 1 \le i \le n\}$. For $x \in \mathbb{R}^d$ we let $\|x\|$ denote the Euclidean norm of x, and let $\mathbb{B}_x(r) := \{x' \in \mathbb{R}^d \mid \|x' - x\| \le r\}$ denote a Euclidean ball of radius r centered at x; when x is unspecified we take it to be the origin, i.e., $\mathbb{B}(r) := \{x' \in \mathbb{R}^d \mid \|x'\| \le r\}$. We let $\mathcal{N}(\mu, \mathbf{\Sigma})$ denote a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$, and $\mathbf{I}_d$ is the identity matrix in $\mathbb{R}^{d \times d}$. For $\mathcal{K} \subseteq \mathbb{R}^d$, we define the Euclidean projection onto $\mathcal{K}$ by $\Pi_{\mathcal{K}}(x) := \operatorname{argmin}_{x' \in \mathcal{K}} \|x - x'\|$. For $p \in [0, 1]$, we let $\mathrm{Geom}(p)$ denote the geometric distribution with parameter p.

Optimization. We say a function $f : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz if for all $x, x' \in \mathbb{R}^d$ we have $|f(x) - f(x')| \le L \|x - x'\|$. We say f is λ-strongly convex if for all $x, x' \in \mathbb{R}^d$ and $t \in [0, 1]$ we have

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) - \frac{\lambda t(1-t)}{2} \left\|x - x'\right\|^2.$$

We denote the subdifferential (i.e., set of all subgradients) of a convex function $f : \mathbb{R}^d \to \mathbb{R}$ at $x \in \mathbb{R}^d$ by $\partial f(x)$. Overloading notation, when clear from the context we will write $\partial f(x)$ to denote an arbitrary subgradient.

Probability. Let μ, ν be two probability densities μ, ν on the same probability space Ω. We let $D_{\mathrm{TV}}(\mu, \nu) := \frac{1}{2} \int |\mu(\omega) - \nu(\omega)| \mathrm{d}\omega$ denote the total variation distance. The following fact is straightforward to see and will be frequently used.

Fact 2.1.1. *Let $\mathcal{E}$ be any event that occurs with probability at least $1 - \delta$ under the density μ. Then $D_{\mathrm{TV}}(\mu, \mu \mid \mathcal{E}) \le \delta$, where $\mu \mid \mathcal{E}$ denotes the conditional distribution of μ under $\mathcal{E}$.*

For two densities μ, ν, we say that a joint distribution $\Gamma(\mu, \nu)$ over the product space of outcomes is a coupling of μ, ν if for $(x, x') \sim \Gamma(\mu, \nu)$, the marginals of x and x' are μ and ν, respectively. When μ is absolutely continuous with respect to ν, and $\alpha > 1$, we define the α-Rényi divergence by

$$D_\alpha(\mu \| \nu) := \frac{1}{\alpha - 1} \log \left(\int \left(\frac{\mu(\omega)}{\nu(\omega)} \right)^\alpha \mathrm{d}\nu(\omega) \right). \tag{2.3}$$

D_α is quasiconvex in its arguments, i.e. if $\mu = \mathbb{E}_\xi \mu_\xi$ and $\nu = \mathbb{E}_\xi \nu_\xi$ (where ξ is a random variable, and μ_ξ, ν_ξ are distribution families indexed by ξ), then $D_\alpha(\mu\|\nu) \le \max_\xi D_\alpha(\mu_\xi\|\nu_\xi)$.

2.2 Framework

We now outline our primary technical innovation, a new gradient estimator for stochastic convex optimization (ReSQue). We define this estimator in Section 2.2.1 and prove that it satisfies several local stability properties in a small ball around a "centerpoint" used for its definition. In Section 2.2.2, we then give preliminaries on a "ball acceleration" framework developed in [CJJ+20, ACJ+21]. This framework aggregates solutions to proximal subproblems defined on small (Euclidean) balls, and uses these subproblem solutions to efficiently solve an optimization problem on a larger domain. Our algorithms in Sections 2.3 and 2.4 instantiate the framework of Section 2.2.2 with new subproblem solvers enjoying improved parallelism or privacy, based on our new ReSQue estimator.

2.2.1 ReSQue estimators

Throughout we use $\gamma_\rho : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ to denote the probability density function of $\mathcal{N}(0, \rho^2 \mathbf{I}_d)$, i.e. $\gamma_\rho(x) = (2\pi\rho)^{-\frac{d}{2}} \exp(-\frac{1}{2\rho^2}\|x\|^2)$. We first define the Gaussian convolution operation.

Definition 2.2.1 (Gaussian convolution)**.** For a function $f : \mathbb{R}^d \to \mathbb{R}$ we denote its convolution with a Gaussian of covariance $\rho^2 \mathbf{I}_d$ by $\widehat{f}_\rho := f * \gamma_\rho$, i.e.

$$\widehat{f}_\rho(x) := \mathop{\mathbb{E}}_{y \sim \mathcal{N}(0, \rho^2 \mathbf{I}_d)} f(x + y) = \int_{y \in \mathbb{R}^n} f(x - y)\gamma_\rho(y)\mathrm{d}y. \tag{2.4}$$

Three well-known properties of $\widehat{f}_\rho$ are that it is differentiable, that if f is L-Lipschitz, so is $\widehat{f}_\rho$ for any ρ, and that $|\widehat{f}_\rho - f| \le L\rho\sqrt{d}$ pointwise (Lemma 8, [BJL+19]). Next, given a centerpoint $\overline{x}$ and a smoothing radius ρ, we define the associated reweighted stochastic query (ReSQue) estimator.

Definition 2.2.2 (ReSQue estimator)**.** Let $\overline{x} \in \mathbb{R}^d$ and let $f : \mathbb{R}^d \to \mathbb{R}$ be convex. Suppose we have a gradient estimator $g : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\mathbb{E}\, g \in \partial f$. We define the *ReSQue*

estimator of radius ρ as the random vector

$$\widetilde{\nabla}^{g}_{\overline{x}}\widehat{f}_{\rho}(x) := \frac{\gamma_{\rho}(x-\overline{x}-\xi)}{\gamma_{\rho}(\xi)}g(\overline{x}+\xi) \text{ where } \xi \sim \mathcal{N}(0,\rho^2\mathbf{I}_d),$$

where we first sample ξ, and then independently query g at $\overline{x}+\xi$. When g is deterministically an element of ∂f, we drop the superscript and denote the estimator by $\widetilde{\nabla}_{\overline{x}}\widehat{f}_{\rho}$.

When g is unbiased for ∂f and enjoys a variance bound, the corresponding ReSQue estimator is unbiased for the convolved function, and inherits a similar variance bound.

Lemma 2.2.3. *The estimator in Definition 2.2.2 satisfies the following properties, where expectations are taken over both the randomness in* ξ *and the randomness in* g*.*

1. *Unbiased:* $\mathbb{E}\,\widetilde{\nabla}^{g}_{\overline{x}}\widehat{f}_{\rho}(x) = \nabla\widehat{f}_{\rho}(x)$.

2. *Bounded variance: If* $\mathbb{E}\,\|g\|^2 \le L^2$ *everywhere, and* $x \in \mathbb{B}_{\overline{x}}(\rho)$*, then* $\mathbb{E}\,\|\widetilde{\nabla}^{g}_{\overline{x}}\widehat{f}_{\rho}(x)\|^2 \le 3L^2$.

Proof. The first statement follows by expanding the expectation over ξ and g:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{g}\int \frac{\gamma_{\rho}(x-\overline{x}-\xi)}{\gamma_{\rho}(\xi)}g(\overline{x}+\xi)\gamma_{\rho}(\xi)\mathrm{d}\xi \\
&= \int \frac{\gamma_{\rho}(x-\overline{x}-\xi)}{\gamma_{\rho}(\xi)}\partial f(\overline{x}+\xi)\gamma_{\rho}(\xi)\mathrm{d}\xi \\
&= \int \partial f(\overline{x}+\xi)\gamma_{\rho}(x-\overline{x}-\xi)\mathrm{d}\xi = \nabla\widehat{f}_{\rho}(x).
\end{aligned}$$

The last equality used that the integral is a subgradient of $\widehat{f}_{\rho}$, and $\widehat{f}_{\rho}$ is differentiable.

For the second statement, denote $v := x-\overline{x}$ for simplicity. Since f is L-Lipschitz,

$$\begin{aligned}
\mathbb{E}\,\|\widetilde{\nabla}^{g}_{\overline{x}}\widehat{f}_{\rho}(x)\|^2 &= \mathop{\mathbb{E}}_{g}\int \frac{(\gamma_{\rho}(v-\xi))^2}{\gamma_{\rho}(\xi)}\|g(\overline{x}+\xi)\|^2\,\mathrm{d}\xi \\
&\le L^2(2\pi\rho)^{-\frac{d}{2}}\int \exp\left(-\frac{\|v-\xi\|^2}{\rho^2}+\frac{\|\xi\|^2}{2\rho^2}\right)\mathrm{d}\xi.
\end{aligned}$$

Next, a standard calculation for Gaussian integrals shows

$$\begin{aligned}
&\int \exp\left(\frac{2\langle v,\xi\rangle - \|\xi\|^2}{2\rho^2}\right) \mathrm{d}\xi \\
&= \exp\left(\frac{\|v\|^2}{2\rho^2}\right) \int \exp\left(-\frac{\|\xi - v\|^2}{2\rho^2}\right) \mathrm{d}\xi \\
&= \exp\left(\frac{\|v\|^2}{2\rho^2}\right) (2\pi\rho)^{\frac{d}{2}}.
\end{aligned} \tag{2.5}$$

The statement then follows from (2.5), which yields

$$\begin{aligned}
&\int \exp\left(-\frac{\|v-\xi\|^2}{\rho^2} + \frac{\|\xi\|^2}{2\rho^2}\right) \mathrm{d}\xi \\
&= \exp\left(-\frac{\|v\|^2}{\rho^2}\right) \int \exp\left(\frac{4\langle v,\xi\rangle - \|\xi\|^2}{2\rho^2}\right) \mathrm{d}\xi \\
&= (2\pi\rho)^{\frac{d}{2}} \exp\left(\frac{2\|v\|^2}{\rho^2}\right) \le 3\cdot(2\pi\rho)^{\frac{d}{2}}
\end{aligned} \tag{2.6}$$

and completes the proof of the second statement. □

When the gradient estimator g is deterministically a subgradient of a Lipschitz function, we can show additional properties about ReSQue. The following lemma will be used in Section 2.4 both to obtain higher moment bounds on ReSQue, as well as higher moment bounds on the difference of ReSQue estimators at nearby points, where the bound scales with the distance between the points.

Lemma 2.2.4. *If $x, x' \in \mathbb{B}_{\overline{x}}(\frac{\rho}{p})$ for $p \ge 2$ then*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\xi\sim\mathcal{N}(0,\rho^2\mathbf{I}_d)} \left[\left(\frac{\gamma_\rho(x-\overline{x}-\xi)}{\gamma_\rho(\xi)}\right)^p\right] \le 2, \\
&\mathop{\mathbb{E}}_{\xi\sim\mathcal{N}(0,\rho^2\mathbf{I}_d)} \left[\left|\frac{\gamma_\rho(x-\overline{x}-\xi) - \gamma_\rho(x'-\overline{x}-\xi)}{\gamma_\rho(\xi)}\right|^p\right] \\
&\qquad \le \left(\frac{24p\,\|x-x'\|}{\rho}\right)^p.
\end{aligned}$$

We defer a proof to Appendix 2.5, where a helper calculation (Fact 2.5.2) is used to

obtain the result.

2.2.2 Ball acceleration

We summarize the guarantees of a recent "ball acceleration" framework originally proposed by [CJJ+20]. For specified parameters $0 < r < R$, this framework efficiently aggregates (approximate) solutions to constrained optimization problems over Euclidean balls of radius r to optimize a function over a ball of radius R. Here we give an approximation-tolerant variant of the [CJJ+20] algorithm in Proposition 2.2.8, which was developed by [ACJ+21]. Before stating the guarantee, we require the definitions of three types of oracles. In each of the following definitions, for some function $F : \mathbb{R}^d \to \mathbb{R}$, scalars λ, r, and point $\overline{x} \in \mathbb{R}^d$ which are clear from context, we will denote

$$x^\star_{\overline{x},\lambda} := \operatorname*{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \left\{ F(x) + \frac{\lambda}{2} \|x - \overline{x}\|^2 \right\}. \tag{2.7}$$

We mention that in the non-private settings of prior work [ACJ+21, CH22] (and under slightly different oracle access assumptions), it was shown that the implementation of line search oracles (Definition 2.2.5) and stochastic proximal oracles (Definition 2.2.7) can be reduced to ball optimization oracles (Definition 2.2.6). Indeed, such a result is summarized in Proposition 2.2.9 and used in Section 2.3 to obtain our parallel SCO algorithms. To tightly quantify the privacy loss of each oracle for developing our SCO algorithms in Section 2.4 (and to implement these oracles under only the function access afforded by Problem 2.4.1), we separate out the requirements of each oracle definition separately.

Definition 2.2.5 (Line search oracle)**.** We say $\mathcal{O}_{\text{ls}}$ is a (Δ, λ)-*line search oracle* for $F : \mathbb{R}^d \to \mathbb{R}$ if given $\overline{x} \in \mathbb{R}^d$, $\mathcal{O}_{\text{ls}}$ returns $x \in \mathbb{R}^d$ with

$$\left\|x - x^\star_{\overline{x},\lambda}\right\| \le \Delta.$$

Definition 2.2.6 (Ball optimization oracle)**.** We say $\mathcal{O}_{\text{bo}}$ is a (φ, λ)-*ball optimization oracle*

for $F : \mathbb{R}^d \to \mathbb{R}$ if given $\overline{x} \in \mathbb{R}^d$, $\mathcal{O}_{\text{bo}}$ returns $x \in \mathbb{R}^d$ with

$$\mathbb{E}\left[F(x) + \frac{\lambda}{2}\|x - \overline{x}\|^2\right] \le F(x^\star_{\overline{x},\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2 + \varphi.$$

Definition 2.2.7 (Stochastic proximal oracle)**.** We say $\mathcal{O}_{\text{sp}}$ is a $(\Delta, \sigma, \lambda)$-*stochastic proximal oracle* for $F : \mathbb{R}^d \to \mathbb{R}$ if given $\overline{x} \in \mathbb{R}^d$, $\mathcal{O}_{\text{sp}}$ returns $x \in \mathbb{R}^d$ with

$$\left\|\mathbb{E}\, x - x^\star_{\overline{x},\lambda}\right\| \le \frac{\Delta}{\lambda},\ \mathbb{E}\left\|x - x^\star_{\overline{x},\lambda}\right\|^2 \le \frac{\sigma^2}{\lambda^2}.$$

Leveraging Definitions 2.2.5, 2.2.6, and 2.2.7, we state a variant of the main result of [ACJ+21]. Roughly speaking, Proposition 2.2.8 states that to optimize a function F over a ball of radius R, it suffices to query $\approx (\frac{R}{r})^{\frac{2}{3}}$ oracles which approximately optimize a sufficiently regularized variant of F over a ball of radius r. We quantify the types of approximate optimization of such regularized functions in Proposition 2.2.8, and defer a detailed discussion of how to derive this statement from [ACJ+21] in Appendix 2.6, as it is stated slightly differently in the original work.[5]

Proposition 2.2.8. *Let $F : \mathbb{R}^d \to \mathbb{R}$ be L-Lipschitz and convex, and let $x^\star \in \mathbb{B}(R)$. There is an algorithm* BallAccel *taking parameters $r \in [0, R]$ and $\varepsilon_{\text{opt}} \in (0, LR]$ with the following guarantee. Define*

$$\kappa := \frac{LR}{\varepsilon_{\text{opt}}},\ K := \left(\frac{R}{r}\right)^{\frac{2}{3}},\ \lambda_\star := \frac{\varepsilon_{\text{opt}} K^2}{R^2}\log^2\kappa.$$

For a universal constant $C_{\text{ba}} > 0$, BallAccel *runs in at most $C_{\text{ba}} K \log\kappa$ iterations and produces $x \in \mathbb{R}^d$ such that*

$$\mathbb{E}\, F(x) \le F(x^\star) + \varepsilon_{\text{opt}}.$$

Moreover, in each iteration BallAccel *requires the following oracle calls (all for F).*

1. *At most $C_{\text{ba}}\log(\frac{R\kappa}{r})$ calls to a $(\frac{r}{C_{\text{ba}}}, \lambda)$-line search oracle with values of $\lambda \in [\frac{\lambda_\star}{C_{\text{ba}}}, \frac{C_{\text{ba}}L}{\varepsilon_{\text{opt}}}]$.*
2. *A single call to $(\frac{\lambda r^2}{C_{\text{ba}}\log^3\kappa}, \lambda)$-ball optimization oracle with $\lambda \in [\frac{\lambda_\star}{C_{\text{ba}}}, \frac{C_{\text{ba}}L}{\varepsilon_{\text{opt}}}]$.*

[5] In particular, we use an error tolerance for the ball optimization oracles, which is slightly larger than in [ACJ+21], following a tighter error analysis given in Proposition 1 of [CH22].

3. *A single call to* $(\frac{\varepsilon_{\mathrm{opt}}}{C_{\mathrm{ba}}R}, \frac{\varepsilon_{\mathrm{opt}}\sqrt{K}}{C_{\mathrm{ba}}R}, \lambda)$*-stochastic proximal oracle with* $\lambda \in [\frac{\lambda_\star}{C_{\mathrm{ba}}}, \frac{C_{\mathrm{ba}}L}{\varepsilon_{\mathrm{opt}}}]$.

The optimization framework in Proposition 2.2.8 is naturally compatible with our ReSQue estimators, whose stability properties are local in the sense that they hold in balls of radius $\approx \rho$ around the centerpoint $\overline{x}$ (see Lemma 2.2.4). Conveniently, BallAccel reduces an optimization problem over a domain of size R to a sequence of approximate optimization problems on potentially much smaller domains of radius r. In Sections 2.3 and 2.4, by instantiating Proposition 2.2.8 with $r \approx \rho$, we demonstrate how to use the local stability properties of ReSQue estimators (on smaller balls) to solve constrained subproblems, and consequently design improved parallel and private algorithms.

Finally, as mentioned previously, in settings where privacy is not a consideration, Proposition 1 of [CH22] gives a direct implementation of all the line search and stochastic proximal oracles required by Proposition 2.2.8 by reducing them to ball optimization oracles. The statement in [CH22] also assumes access to *function evaluations* in addition to gradient (estimator) queries; however, it is straightforward to use geometric aggregation techniques (see Lemma 2.4.17) to bypass this requirement. We give a slight rephrasing of Proposition 1 in [CH22] without the use of function evaluation oracles, and defer further discussion to Appendix 2.7 where we prove the following.

Proposition 2.2.9. *Let* $F : \mathbb{R}^d \to \mathbb{R}$ *be* L*-Lipschitz and convex, and let* $x^\star \in \mathbb{B}(R)$*. There is an implementation of* BallAccel *(see Proposition 2.2.8) taking parameters* $r \in [0, R]$ *and* $\varepsilon_{\mathrm{opt}} \in (0, LR]$ *with the following guarantee, where we define* $\kappa, K, \lambda_\star$ *as in Proposition 2.2.8. For a universal constant* $C_{\mathrm{ba}} > 0$*,* BallAccel *runs in at most* $C_{\mathrm{ba}} K \log \kappa$ *iterations and produces* $x \in \mathbb{R}^d$ *such that* $\mathbb{E}\, F(x) \le F(x^\star) + \varepsilon_{\mathrm{opt}}$.

1. *Each iteration makes at most* $C_{\mathrm{ba}} \log^2(\frac{R\kappa}{r})$ *calls to* $(\frac{\lambda r^2}{C_{\mathrm{ba}}}, \lambda)$*-ball optimization oracle with values of* $\lambda \in [\frac{\lambda_\star}{C_{\mathrm{ba}}}, \frac{C_{\mathrm{ba}}L}{\varepsilon_{\mathrm{opt}}}]$.

2. *For each* $j \in [\lceil \log_2 K + C_{\mathrm{ba}} \rceil]$*, at most* $C_{\mathrm{ba}}^2 \cdot 2^{-j} K \log(\frac{R\kappa}{r})$ *iterations query a* $(\frac{\lambda r^2}{C_{\mathrm{ba}} 2^j} \cdot \log^{-2}(\frac{R\kappa}{r}), \lambda)$*-ball optimization oracle for some* $\lambda \in [\frac{\lambda_\star}{C_{\mathrm{ba}}}, \frac{C_{\mathrm{ba}}L}{\varepsilon_{\mathrm{opt}}}]$.

2.3 Parallel stochastic convex optimization

In this section, we present our main results on parallel convex optimization with improved computational depth and total work. We present our main results below in Theorems 2.3.2 and 2.3.3, after formally stating our notation and the SCO problem we study in this section.

2.3.1 Preliminaries

In this section, we study the following SCO problem, which models access to an objective only through the stochastic gradient oracle.

Problem 2.3.1. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex. We assume there exists a *stochastic gradient oracle* $g : \mathbb{R}^d \to \mathbb{R}^d$ satisfying for all $x \in \mathbb{R}^d$, $\mathbb{E}\, g(x) \in \partial f(x)$, $\mathbb{E}\, \|g(x)\|^2 \le L^2$. Our goal is to produce $x \in \mathbb{R}^d$ such that $\mathbb{E}\, f(x) \le \min_{x^\star \in \mathbb{B}(R)} f(x^\star) + \varepsilon_{\text{opt}}$. We define parameter

$$\kappa := \frac{LR}{\varepsilon_{\text{opt}}}. \tag{2.8}$$

When discussing a parallel algorithm which queries a stochastic gradient oracle, in the sense of Problem 2.3.1, we separate its complexity into four parameters. The *query depth* is the maximum number of sequential rounds of interaction with the oracle, where queries are submitted in batch. The *total number of queries* is the total number of oracle queries used by the algorithm. The *computational depth and work* are the sequential depth and total amount of computational work done by the algorithm outside of these oracle queries. For simplicity we assume that all d-dimensional vector operations have a cost of d when discussing computation.

2.3.2 Proofs of Theorems 2.3.2 and 2.3.3

Theorem 2.3.2 (Parallel EpochSGD-based solver)**.** BallAccel *(Proposition 2.2.9) using parallel* EpochSGD *(Algorithm 15) as a ball optimization oracle solves Problem 2.3.1 with expected error* ε_{opt}*, with*

$$O\left(d^{\frac{1}{3}} \kappa^{\frac{2}{3}} \log^3(d\kappa)\right)$$

query depth and

$$O\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+\kappa^2\log^4(d\kappa)\right) \text{ total queries,}$$

and an additional computational cost of

$$O\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+\kappa^2\log^4(d\kappa)\right)$$

depth and

$$O\left(\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+\kappa^2\log^4(d\kappa)\right)\cdot d\right) \text{ work.}$$

Theorem 2.3.3 (Parallel AC-SA-based solver)**.** BallAccel *(Proposition 2.2.9) using parallel* AC-SA *(Algorithm 17) as a ball optimization oracle solves Problem 2.3.1 with expected error* ε_{opt}*, with*

$$O\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log\kappa\right) \text{ query depth}$$
$$\text{and } O\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+d^{\frac{1}{4}}\kappa\log^4(d\kappa)+\kappa^2\log^4(d\kappa)\right)$$

total queries, and an additional computational cost of

$$O\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+d^{\frac{1}{4}}\kappa\log^4(d\kappa)\right) \text{ depth and}$$
$$O\left(\left(d^{\frac{1}{3}}\kappa^{\frac{2}{3}}\log^3(d\kappa)+d^{\frac{1}{4}}\kappa\log^4(d\kappa)+\kappa^2\log^4(d\kappa)\right)\cdot d\right)$$

work.

The query depth, total number of queries, and total work for both of our results are the same (up to logarithmic factors). The main difference is that AC-SA attains an improved computational depth for solving SCO, compared to using EpochSGD. Our results build upon the BallAccel framework in Section 2.2.2, combined with careful parallel implementations of the required ball optimization oracles to achieve improved complexities.

We begin by developing our parallel ball optimization oracles using our ReSQue estimator machinery from Section 2.2.1. First, Proposition 2.2.9 reduces Problem 2.3.1 to

implementation of a ball optimization oracle. Recall that a ball optimization oracle (Definition 2.2.6) requires an approximate solution x of a regularized subproblem. In particular, for some accuracy parameter φ, and defining $x^\star_{\overline{x},\lambda}$ as in (2.7), we wish to compute a random $x \in \mathbb{B}_{\overline{x}}(r)$ such that

$$\mathbb{E}\left[\widehat{f}_\rho(x) + \frac{\lambda}{2}\|x - \overline{x}\|^2\right] \leq \widehat{f}_\rho(x^\star_{\overline{x},\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2 + \varphi, \; x \in \mathbb{B}_{\overline{x}}(r).$$

Note that such a ball optimization oracle can satisfy the requirements of Proposition 2.2.9 with $F \leftarrow \widehat{f}_\rho$, $r \leftarrow \rho$. In particular, Lemma 2.2.3 gives a gradient estimator variance bound under the setting $r = \rho$.

EpochSGD. We implement EpochSGD [HK14, ACJ+21], a variant of standard stochastic gradient descent on regularized objective functions, in parallel using the stochastic ReSQue estimator constructed in Definition 2.2.2. Our main observation is that the gradient queries in Definition 2.2.2 can be implemented in parallel at the beginning of the algorithm. We provide the pseudocode of our parallel implementation of EpochSGD in Algorithm 15 and state its guarantees in Proposition 2.3.4.

Proposition 2.3.4 (Proposition 3, [ACJ+21])**.** *Let f, g satisfy the assumptions of Problem 2.3.1. When $\rho = r$, Algorithm 15 is a (φ, λ)-ball optimization oracle for $\widehat{f}_\rho$ which makes $O(\frac{L^2}{\varphi\lambda})$ total queries to g with constant query depth, and an additional computational cost of $O(\frac{L^2}{\varphi\lambda})$ depth and work.*

AC-SA. We can also implement AC-SA [GL12], a variant of accelerated gradient descent under stochastic gradient queries, in parallel using stochastic ReSQue estimators. We provide the pseudocode of our parallel implementation of AC-SA in Algorithm 17 and state its guarantees in Lemma 2.3.5.

Proposition 2.3.5 (Special case of Theorem 1, [GL12])**.** *Let f, g satisfy the assumptions of Problem 2.3.1. When $\rho = r$, Algorithm 17 is a (φ, λ)-ball optimization oracle for $\widehat{f}_\rho$ which*

Algorithm 6: EpochSGD$(f, g, \overline{x}, r, \rho, \lambda, \varphi)$

Input: $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ satisfying the assumptions of Problem 2.3.1, $\overline{x} \in \mathbb{R}^d$, $r, \rho, \lambda, \varphi > 0$
$\eta_1 \leftarrow \frac{1}{4\lambda}$, $T_1 \leftarrow 16$, $T \leftarrow \lceil \frac{48L^2}{\lambda\varphi} \rceil$
Sample $\xi_i \sim \mathcal{N}(0, \rho^2 \mathbf{I}_d)$, $i \in [2T]$ independently
Query $g(\overline{x} + \xi_i)$ for all $i \in [2T]$ (in parallel)
$x_1^0 \leftarrow \overline{x}$, $k \leftarrow 1$
while $\sum_{j \in [k]} T_j \le T$ **do**
  $x_k^1 \leftarrow \operatorname{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \left\{ \frac{\eta_k \lambda}{2} \|x - \overline{x}\|^2 + \frac{1}{2} \|x - x_k^0\|^2 \right\}$
  for $t \in [T_k - 1]$ **do**
    $i \leftarrow \sum_{j \in [k-1]} T_j + t$
    $\widetilde{\nabla}^g_{\overline{x}} \widehat{f}_\rho(x_k^t) \leftarrow \frac{\gamma_\rho(x_k^t - \overline{x} - \xi_i)}{\gamma_\rho(\xi_i)} g(\overline{x} + \xi_i)$
    $x_k^{t+1} \leftarrow \operatorname{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \left\{ \eta_k \langle \widetilde{\nabla}^g_{\overline{x}} \widehat{f}_\rho(x_k^t), x \rangle + \frac{\eta_k \lambda}{2} \|x - \overline{x}\|^2 + \frac{1}{2} \|x - x_k^t\|^2 \right\}$
  end
  $x_{k+1}^0 \leftarrow \frac{1}{T_k} \sum_{t \in [T_k]} x_k^t$, $T_{k+1} \leftarrow 2T_k$, $\eta_{k+1} \leftarrow \frac{\eta_k}{2}$, $k \leftarrow k + 1$
end
return x_k^0

makes

$$O\left(\sqrt{1 + \frac{L}{\rho\lambda}} \log\left(\frac{\lambda r^2}{\varphi}\right) + \frac{L^2}{\lambda\varphi}\right) \text{ total queries}$$

with constant query depth, and an additional computational cost of

$$O\left(\sqrt{1 + \frac{L}{\rho\lambda}} \log\left(\frac{\lambda r^2}{\varphi}\right)\right) \text{ depth and } O\left(\sqrt{1 + \frac{L}{\rho\lambda}} \log\left(\frac{\lambda r^2}{\varphi}\right) + \frac{L^2}{\lambda\varphi}\right) \text{ work.}$$

Because the statement of Proposition 2.3.5 follows from specific parameter choices in the main result in [GL12], we defer a more thorough discussion of how to obtain this result to Appendix 2.8.

Main results. We now use our parallel ball optimization oracles to prove Theorems 2.3.2 and 2.3.3.

Proofs of Theorems 2.3.2 and 2.3.3. We use Proposition 2.2.9 with $r = \rho = \frac{\varepsilon_{\text{opt}}}{\sqrt{d}L}$ on $F \leftarrow \widehat{f}_\rho$, which approximates f to additive ε_{opt}, and $x^\star := \arg\min_{x \in \mathbb{B}(R)} f(x)$. Rescaling ε_{opt} by a

Algorithm 7: AC-SA$(f, \overline{x}, r, \rho, \lambda, \varphi)$

Input: $f : \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$ satisfying the assumptions of Problem 2.3.1, $\overline{x} \in \mathbb{R}^d$, $r, \rho, \lambda, \varphi > 0$

$K \leftarrow \lceil \log_2(\frac{\lambda r^2}{\varphi}) \rceil$, $T \leftarrow \lceil 4\sqrt{\frac{L}{\rho\lambda} + 1} \rceil$, $N_k \leftarrow \left\lceil 48 \cdot 2^k \cdot \frac{L^2}{\lambda^2 r^2 T} \right\rceil$ for $k \in [K]$

Sample $\xi_i \sim \mathcal{N}(0, \rho^2 \mathbf{I}_d)$, $i \in [N]$ independently, for $N = T \cdot (\sum_{k \in [K]} N_k)$

Query $g(\overline{x} + \xi_i)$ for all $i \in [N]$ (in parallel)

$x_0^{\mathsf{ag}} \leftarrow \overline{x}$, $x_0 \leftarrow \overline{x}$

for $k \in [K]$ **do**

  for $t \in [T]$ **do**

    $\alpha_t \leftarrow \frac{2}{t+1}$, $\gamma_t \leftarrow \frac{4(\frac{L}{\rho} + \lambda)}{t(t+1)}$

    $x_t^{\mathsf{md}} \leftarrow \frac{(1-\alpha_t)(\lambda+\gamma_t)}{\gamma_t + (1-\alpha_t^2)\lambda} x_{t-1}^{\mathsf{ag}} + \frac{\alpha_t(1-\alpha_t)(\lambda+\gamma_t)}{\gamma_t + (1-\alpha_t^2)\lambda} x_{t-1}$

    $N_{T,[k-1]} \leftarrow T \cdot \sum_{k' \in [k-1]} N_{k'}$

    $\widehat{\nabla} f(x_t^{\mathsf{md}}) \leftarrow \frac{1}{N_k} \sum_{n \in [N_k]} \frac{\gamma_\rho(x_t^{\mathsf{md}} - \overline{x} - \xi_{N_{T,[k-1]}+n})}{\gamma_\rho(\xi_{N_{T,[k-1]}+n})} g(\overline{x} + \xi_{N_{T,[k-1]}+n})$

    $x_t \leftarrow \mathrm{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \Psi_t(x)$, where $\Psi_t(x) :=$
    $\langle \alpha_t \widehat{\nabla} f(x_t^{\mathsf{md}}) + \lambda(x_t^{\mathsf{md}} - \overline{x}), x - x_t \rangle + \frac{\gamma_t + \lambda(1-\alpha_t)}{2} \|x - x_{t-1}\|^2 + \frac{\lambda \alpha_t}{2} \|x - x_t^{\mathsf{md}}\|^2$

    $x_t^{\mathsf{ag}} \leftarrow \alpha_t x_t + (1 - \alpha_t) x_{t-1}^{\mathsf{ag}}$

  end

  $x_0^{\mathsf{ag}} \leftarrow x_T^{\mathsf{ag}}$, $x_0 \leftarrow x_T^{\mathsf{ag}}$

end

Return: x_T^{ag}

constant from the guarantee of Proposition 2.2.9 gives the error claim. For the oracle query depths, note that each ball optimization oracle (whether implemented using Algorithm 15 or Algorithm 17) has constant query depth, and at most $O(\log^2(d\kappa))$ ball optimization oracles are queried per iteration on average. Note that (see Proposition 2.2.8)

$$\kappa = \frac{LR}{\varepsilon_{\mathrm{opt}}}, \ K = \left(\frac{R}{r}\right)^{\frac{2}{3}} = d^{\frac{1}{3}} \kappa^{\frac{2}{3}}, \ \lambda_\star = \frac{\varepsilon_{\mathrm{opt}} K^2}{R^2} \log^2 \kappa = \frac{\varepsilon_{\mathrm{opt}} d^{\frac{2}{3}} \kappa^{\frac{4}{3}}}{R^2} \log^2 \kappa.$$

For the total oracle queries, computational depth, and work, when implementing each ball optimization oracle with EpochSGD, we have that for $j_{\max} := \lceil \log_2 K + C_{\mathrm{ba}} \rceil$, these are all

$$O\left(K \log(d\kappa) \cdot \left(\sum_{j \in [j_{\max}]} \frac{1}{2^j} \left(\frac{L^2 \cdot 2^j \log^2(d\kappa)}{\lambda_\star^2 r^2} \right) + \left(\frac{L^2}{\lambda_\star^2 r^2} \right) \log^2(d\kappa) \right) \right)$$

$$= O\left(K \log^4(d\kappa) \cdot \frac{L^2}{\lambda_\star^2 r^2}\right) = O\left(\kappa^2 \log^4(d\kappa)\right)$$

due to Proposition 2.3.4. The additional terms in the theorem statement are due to the number of ball oracles needed. For the computational depth when implementing each ball optimization oracle with AC-SA we have that (due to Proposition 2.3.5), it is bounded by

$$O\left(K \log^3(d\kappa) \cdot \sqrt{\frac{L}{r\lambda_\star}} \log(d\kappa)\right) = O\left(K \log^4(d\kappa) \cdot \frac{\sqrt{\kappa}}{K^{\frac{1}{4}}}\right) = O\left(d^{\frac{1}{4}} \kappa \log^4(d\kappa)\right).$$

Finally, for the total oracle queries and work bounds, the bound due to the $\frac{L^2}{\lambda\varepsilon}$ term is as was computed for Theorem 2.3.2, and the bound due to the other term is the same as the above display. □

2.4 Private stochastic convex optimization

We now develop our main result on an improved gradient complexity for private SCO. First, in Section 2.4.1, we introduce several variants of differential privacy including a relaxation of Rényi differential privacy [Mir17], which tolerates a small amount of total variation error. Next, in Sections 2.4.2, 2.4.3, and 2.4.4, we build several private stochastic optimization subroutines which will be used in the ball acceleration framework of Proposition 2.2.8. Finally, in Sections 2.4.5 and 2.4.6, we give our main results on private ERM and SCO respectively, by leveraging the subroutines we develop.

2.4.1 Preliminaries

In this section, we study the following specialization of Problem 2.3.1 naturally compatible with preserving privacy with respect to samples, through the formalism of DP (to be defined shortly).

Problem 2.4.1. Let dist be a distribution over $\mathcal{S}$, and suppose there is a family of functions indexed by $s \in \mathcal{S}$, such that $f(\cdot; s) : \mathbb{R}^d \to \mathbb{R}$ is convex for all $s \in \mathcal{S}$. Let $\mathcal{D} := \{s_i\}_{i \in [n]}$

consist of n i.i.d. draws from dist, and define the *empirical risk* and *population risk* by

$$f^{\text{erm}}(x) := \frac{1}{n}\sum_{i\in[n]} f(x; s_i) \text{ and } F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s\sim\text{dist}} f(x; s).$$

We denote $f^i := f(\cdot; s_i)$ for all $i \in [n]$, and assume that for all $s \in \mathcal{S}$, $f(\cdot; s)$ is L-Lipschitz. We are given $\mathcal{D}$, and can query subgradients of the "sampled functions" f^i. Our goal is to produce $x \in \mathbb{R}^d$ such that $\mathbb{E}\, F_{\mathcal{P}}(x) \le \min_{x^\star\in\mathbb{B}(R)} F_{\mathcal{P}}(x^\star) + \varepsilon_{\text{opt}}$. We again define $\kappa = \frac{LR}{\varepsilon_{\text{opt}}}$ as in (2.8).

In the "one-pass" setting where we only query each ∂f^i a single time, we can treat each ∂f^i as a bounded stochastic gradient of the underlying population risk $F_{\mathcal{P}}$. We note the related problem of *empirical risk minimization*, i.e. optimizing f^{erm} (in the setting of Problem 2.4.1), can also be viewed as a case of Problem 2.3.1 where we construct g by querying ∂f^i for $i \sim_{\text{unif.}} [n]$. We design $(\varepsilon_{\text{dp}}, \delta)$-DP algorithms for solving Problem 2.4.1 which obtain small optimization error for f^{erm} and $F_{\mathcal{P}}$. To disambiguate, we will always use ε_{opt} to denote an optimization error parameter, and ε_{dp} to denote a privacy parameter. Our private SCO algorithm will require querying ∂f^i multiple times for some $i \in [n]$, and hence incur bias for the population risk gradient. Throughout the rest of the section, following the notation of Problem 2.4.1, we will fix a dataset $\mathcal{D} \in \mathcal{S}^n$ and define the empirical risk f^{erm} and population risk $F_{\mathcal{P}}$ accordingly. We now move on to our privacy definitions.

We say that two datasets $\mathcal{D} = \{s_i\}_{i\in[n]} \in \mathcal{S}^n$ and $\mathcal{D}' = \{s'_i\}_{i\in[n]} \in \mathcal{S}^n$ are *neighboring* if $|\{i \mid s_i \ne s'_i\}| = 1$. We say a mechanism (i.e. a randomized algorithm) $\mathcal{M}$ satisfies $(\varepsilon_{\text{dp}}, \delta)$-differential privacy (DP) if, for its output space Ω and all neighboring $\mathcal{D}$, $\mathcal{D}'$, we have for all $S \subseteq \Omega$,

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \le \exp(\varepsilon_{\text{dp}}) \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta. \tag{2.9}$$

We extensively use the notion of Rényi differential privacy due to its compatibility with the subsampling arguments we will use, as well as an approximate relaxation of its definition which we introduce. We say that a mechanism $\mathcal{M}$ satisfies (α, ε)-Rényi differential privacy

(RDP) if for all neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$, the α-Rényi divergence (2.3) satisfies

$$D_\alpha(\mathcal{M}(\mathcal{D}) \| \mathcal{M}(\mathcal{D}')) \leq \varepsilon. \tag{2.10}$$

RDP has several useful properties which we now summarize.

Proposition 2.4.2 (Propositions 1, 3, and 7, [Mir17])**.** *RDP has the following properties.*

1. *(Composition): Let $\mathcal{M}_1 : \mathcal{S}^n \to \Omega$ satisfy (α, ε_1)-RDP and $\mathcal{M}_2 : \mathcal{S}^n \times \Omega \to \Omega'$ satisfy (α, ε_2)-RDP for any input in Ω. Then the composition of $\mathcal{M}_2$ and $\mathcal{M}_1$, defined as $\mathcal{M}_2(\mathcal{D}, \mathcal{M}_1(\mathcal{D}))$ satisfies $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.*

2. *(Gaussian mechanism): For $\mu, \mu' \in \mathbb{R}^d$, $D_\alpha(\mathcal{N}(\mu, \sigma^2 \mathbf{I}_d) \| \mathcal{N}(\mu', \sigma^2 \mathbf{I}_d)) \leq \frac{\alpha}{2\sigma^2} \|\mu - \mu'\|^2$.*

3. *(Standard DP): If $\mathcal{M}$ satisfies (α, ε)-RDP, then for all $\delta \in (0, 1)$, $\mathcal{M}$ satisfies $(\varepsilon + \frac{1}{\alpha - 1} \log \frac{1}{\delta}, \delta)$-DP.*

We also use the following definition of approximate Rényi divergence:

$$D_{\alpha,\delta}(\mu \| \nu) := \min_{D_{\mathrm{TV}}(\mu', \mu) \leq \delta, D_{\mathrm{TV}}(\nu', \nu) \leq \delta} D_\alpha(\mu' \| \nu'). \tag{2.11}$$

We relax the definition (2.10) and say that $\mathcal{M}$ satisfies $(\alpha, \varepsilon, \delta)$-RDP if for all neighboring $\mathcal{D}$, $\mathcal{D}' \in \mathcal{S}^n$, recalling definition (2.11),

$$D_{\alpha,\delta}(\mathcal{M}(\mathcal{D}) \| \mathcal{M}(\mathcal{D}')) \leq \varepsilon.$$

The following is then immediate from Proposition 2.4.2, and our definition of approximate RDP, by coupling the output distributions with the distributions realizing the minimum (2.11).

Corollary 2.4.3. *If $\mathcal{M}$ satisfies $(\alpha, \varepsilon, \delta)$-RDP, then for all $\delta' \in (0, 1)$, $\mathcal{M}$ satisfies $(\varepsilon_{\mathrm{dp}}, \delta' + (1 + \exp(\varepsilon_{\mathrm{dp}}))\delta)$-DP for $\varepsilon_{\mathrm{dp}} := \varepsilon + \frac{1}{\alpha - 1} \log \frac{1}{\delta'}$.*

Proof. Let μ, ν be within total variation δ of $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$, such that $D_\alpha(\mu\|\nu) \leq \varepsilon$ and hence for any event S,

$$\Pr_{\omega\sim\mu}[\omega \in S] \leq \exp(\varepsilon_{\mathrm{dp}}) \Pr_{\omega\sim\nu}[\omega \in S] + \delta'.$$

Combining the above with

$$\Pr_{\omega\sim\mathcal{M}(\mathcal{D})}[\omega \in S] - \delta \leq \Pr_{\omega\sim\mu}[\omega \in S],\ \Pr_{\omega\sim\nu}[\omega \in S] \leq \Pr_{\omega\sim\mathcal{M}(\mathcal{D}')}[\omega \in S] + \delta,$$

we have

$$\begin{aligned}\Pr_{\omega\sim\mathcal{M}(\mathcal{D})}[\omega \in S] &\leq \exp(\varepsilon_{\mathrm{dp}}) \Pr_{\omega\sim\nu}[\omega \in S] + \delta' + \delta \\ &\leq \exp(\varepsilon_{\mathrm{dp}}) \Pr_{\omega\sim\mathcal{M}(\mathcal{D}')}[\omega \in S] + \delta' + (1 + \exp(\varepsilon_{\mathrm{dp}}))\delta.\end{aligned}$$

□

Finally, our approximate RDP notion enjoys a composition property similar to standard RDP.

Lemma 2.4.4. *Let $\mathcal{M}_1 : \mathcal{S}^n \to \Omega$ satisfy $(\alpha, \varepsilon_1, \delta_1)$-RDP and $\mathcal{M}_2 : \mathcal{S}^n \times \Omega \to \Omega'$ satisfy $(\alpha, \varepsilon_2, \delta_2)$-RDP for any input in Ω. Then the composition of $\mathcal{M}_2$ and $\mathcal{M}_1$, defined as $\mathcal{M}_2(\mathcal{D}, \mathcal{M}_1(\mathcal{D}))$ satisfies $(\alpha, \varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-RDP.*

Proof. Let $\mathcal{D}, \mathcal{D}'$ be neighboring datasets, and let μ, μ' be distributions within total variation δ_1 of $\mathcal{M}_1(\mathcal{D})$, $\mathcal{M}_1(\mathcal{D}')$ realizing the bound $D_\alpha(\mu\|\mu') \leq \varepsilon_1$. For any $\omega \in \Omega$, similarly let ν_ω, ν'_ω be the distributions within total variation δ_2 of $\mathcal{M}_2(\mathcal{D}, \omega)$ and $\mathcal{M}_2(\mathcal{D}', \omega)$ realizing the bound $D_\alpha(\nu_\omega\|\nu'_\omega) \leq \varepsilon_2$. Finally, let P_1 be the distribution of $\omega \in \Omega$ according to $\mathcal{M}_1(\mathcal{D})$, and Q_1 to be the distribution of $\mathcal{M}_1(\mathcal{D}')$; similarly, let $P_{2,\omega}$, $Q_{2,\omega}$ be the distributions of $\omega' \in \Omega'$ according to $\mathcal{M}_2(\mathcal{D}, \omega)$ and $\mathcal{M}_2(\mathcal{D}', \omega)$. We first note that by a union bound,

$$\begin{aligned}D_{\mathrm{TV}}\left(\int \nu_\omega(\omega')\mu(\omega)d\omega d\omega', \int P_1(\omega)P_{2,\omega}(\omega')d\omega d\omega'\right) &\leq \delta_1 + \delta_2, \\ D_{\mathrm{TV}}\left(\int \nu'_\omega(\omega')\mu'(\omega)d\omega d\omega', \int Q_1(\omega)Q_{2,\omega}(\omega')d\omega d\omega'\right) &\leq \delta_1 + \delta_2.\end{aligned}$$

Finally, by Proposition 1 of [Mir17], we have

$$D_\alpha\left(\int \nu_\omega(\omega')\mu(\omega)d\omega d\omega' \middle\| \int \nu'_\omega(\omega')\mu'(\omega)d\omega d\omega'\right) \le \varepsilon_1 + \varepsilon_2.$$

Combining the above two displays yields the claim. □

2.4.2 Subsampled smoothed ERM solver: the convex case

We give an ERM algorithm that takes as input a dataset $\mathcal{D} \in \mathcal{S}^n$, parameters $T \in \mathbb{N}$ and $r, \rho, \beta > 0$, and a center point $\overline{x} \in \mathbb{R}^d$. Our algorithm is based on a localization approach introduced by [FKT20] which repeatedly decreases a domain size to bound the error due to adding noise for privacy. In particular we will obtain an error bound on $\widehat{f_\rho^{\mathrm{erm}}}$ with respect to the set $\mathbb{B}_{\overline{x}}(r)$, using at most T calls to the ReSQue estimator in Definition 2.2.2 with a deterministic subgradient oracle. Here we recall that f^{erm} is defined as in Problem 2.4.1, and $\widehat{f_\rho^{\mathrm{erm}}}$ is correspondingly defined as in Definition 2.2.1. Importantly, our ERM algorithm developed in this section attains RDP bounds improving with the subsampling parameter $\frac{T}{n}$ when $T \ll n$, due to only querying T random samples in our dataset.

We summarize our optimization and privacy guarantees on Algorithm 13 in the following. The proof follows by combining Lemma 2.4.6 (the utility bound) and Lemma 2.4.10 (the privacy bound).

Proposition 2.4.5. *Let $x^\star_{\overline{x}} \in \operatorname{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \widehat{f_\rho^{\mathrm{erm}}}(x)$. Algorithm 13 uses at most T gradients and produces $x \in \mathbb{B}_{\overline{x}}(r)$ such that, for a universal constant C_{cvx},*

$$\mathbb{E}\left[\widehat{f_\rho^{\mathrm{erm}}}(x)\right] - \widehat{f_\rho^{\mathrm{erm}}}(x^\star_{\overline{x}}) \le C_{\mathrm{cvx}} Lr\left(\frac{\sqrt{d}}{\beta T} + \frac{1}{\sqrt{T}}\right).$$

Moreover, there is a universal constant $C_{\mathrm{priv}} \ge 1$, such that if $\frac{T}{n} \le \frac{1}{C_{\mathrm{priv}}}$, $\beta^2 \log^2(\frac{1}{\delta}) \le \frac{1}{C_{\mathrm{priv}}}$, $\delta \in (0, \frac{1}{6})$, and $\frac{\rho}{r} \ge C_{\mathrm{priv}} \log^2(\frac{\log T}{\delta})$, Algorithm 13 satisfies $(\alpha, \alpha\tau, \delta)$-RDP for

$$\tau := C_{\mathrm{priv}}\left(\beta \log\left(\frac{1}{\delta}\right) \cdot \frac{T}{n}\right)^2 \text{ and } \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}}\beta^2 \log^2(\frac{1}{\delta})}\right).$$

Algorithm 8: Subsampled ReSQued ERM solver, convex case

Input: $\overline{x} \in \mathbb{R}^d$, ball radius, convolution radius, and privacy parameter $r, \rho, \beta > 0$, dataset $\mathcal{D} \in \mathcal{S}^n$, iteration count $T \in \mathbb{N}$
$\widehat{T} \leftarrow 2^{\lfloor \log_2 T \rfloor}$, $k \leftarrow \log_2 \widehat{T}$, $\eta \leftarrow \frac{r}{L} \min(\frac{1}{\sqrt{T}}, \frac{\beta}{\sqrt{d}})$, $x_0 \leftarrow \overline{x}$
for $i \in [k]$ **do**
  $T_i \leftarrow 2^{-i}\widehat{T}$, $\eta_i \leftarrow 4^{-i}\eta$, $\sigma_i \leftarrow \frac{L\eta_i}{\beta}$
  $y_0 \leftarrow x_{i-1}$
  for $j \in [T_i]$ **do**
    $z_{i,j} \sim_{\text{unif.}} [n]$
    $y_j \leftarrow \Pi_{\mathbb{B}_{\overline{x}}(r)}(y_{j-1} - \eta_i \widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho^{z_{i,j}}(y_{j-1}))$; ▷ PSGD step using ReSQue (See Definition 2.2.2) for a subsampled function. Lemma 2.4.7 denotes the random Gaussian sample by $\xi_{i,j}$.
  end
  $\overline{y}_i \leftarrow \frac{1}{T_i} \sum_{j \in [T_i]} y_j$
  $x_i \leftarrow \overline{y}_i + \zeta_i$, for $\zeta_i \sim \mathcal{N}(0, \sigma_i^2 \mathbf{I}_d)$
end
return x_k

Utility analysis. We begin by proving a utility guarantee for Algorithm 13, following [FKT20].

Lemma 2.4.6. *Let* $x_{\overline{x}}^\star := \operatorname{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \widehat{f_\rho^{\text{erm}}}(x)$. *We have, for a universal constant* C_{cvx},

$$\mathbb{E}\left[\widehat{f_\rho^{\text{erm}}}(x_k)\right] - \widehat{f_\rho^{\text{erm}}}(x_{\overline{x}}^\star) \le C_{\text{cvx}} L r \left(\frac{\sqrt{d}}{\beta T} + \frac{1}{\sqrt{T}}\right).$$

Proof. Denote $F := \widehat{f_\rho^{\text{erm}}}$, $\overline{y}_0 := x_{\overline{x}}^\star$, and $\zeta_0 := \overline{x} - x_{\overline{x}}^\star$, where by assumption $\|\zeta_0\| \le r$. We begin by observing that in each run of Line 8, by combining the first property in Lemma 2.2.3 with the definition of f^{erm}, we have that $\mathbb{E}\left[\widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho^{z_{i,j}}(y_{j-1}) \mid y_{j-1}\right] \in \partial F(y_{j-1})$. Moreover, by the second property in Lemma 2.2.3 and the fact that $f^{z_{i,j}}$ is L-Lipschitz,

$$\mathbb{E}\left\|\widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho^{z_{i,j}}(y_{j-1})\right\|^2 \le 3L^2.$$

We thus have

$$\begin{aligned}\mathbb{E}\left[F(x_k)\right] - F(x^\star_{\overline{x}}) &= \sum_{i\in[k]} \mathbb{E}[F(\overline{y}_i) - F(\overline{y}_{i-1})] + \mathbb{E}\left[F(x_k) - F(\overline{y}_k)\right] \\ &\le \sum_{i\in[k]} \left(\frac{\mathbb{E}\left[\left\|x_{i-1} - \overline{y}_{i-1}\right\|^2\right]}{2\eta_i T_i} + \frac{3\eta_i L^2}{2} \right) + L\,\mathbb{E}\left[\|x_k - \overline{y}_k\|\right] \qquad (2.12) \\ &\le \frac{8r^2}{\eta T} + 4\sum_{i\in[k-1]} \frac{\sigma_i^2 d}{\eta_i T_i} + \sum_{i\in[k]} \frac{3\eta_i L^2}{2} + L\sigma_k\sqrt{d}.\end{aligned}$$

In the second line, we used standard regret guarantees on projected stochastic gradient descent, e.g. Lemma 7 of [HK14], where we used that all $\overline{y}_i \in \mathbb{B}_{\overline{x}}(r)$; in the third line, we used

$$\mathbb{E}[\|x_k - \overline{y}_k\|] \le \sqrt{\mathbb{E}\left[\|x_k - \overline{y}_k\|^2\right]} = \sqrt{\mathbb{E}\left[\|\zeta_k\|^2\right]} = \sigma_k\sqrt{d}$$

by Jensen's inequality. Continuing, we have by our choice of parameters that $\frac{\sigma_i^2}{\eta_i T_i} \le 2^{-i}\frac{L^2\eta}{2\beta^2\widehat{T}}$, hence

$$\begin{aligned}\mathbb{E}\left[F(x_k)\right] - F(x^\star_{\overline{x}}) &\le \frac{8r^2}{\eta T} + \frac{4L^2\eta d}{\beta^2\widehat{T}} + \frac{3\eta L^2}{2} + \frac{L^2\eta\sqrt{d}}{\beta}\cdot\frac{1}{\widehat{T}^2} \\ &\le \left(\frac{8Lr}{\sqrt{T}} + \frac{8Lr\sqrt{d}}{\beta T}\right) + \frac{8Lr\sqrt{d}}{\beta T} + \frac{3Lr}{2\sqrt{T}} + \frac{Lr}{\sqrt{T}}.\end{aligned}$$

Here we used that $2\widehat{T} \ge T$ and $\widehat{T}^2 \ge \sqrt{T}$, for all $T \in \mathbb{N}$. □

Privacy analysis. We now show that our algorithm satisfies a strong (approximate) RDP guarantee. Let $\mathcal{D}' = \{s'_i\}_{i\in[n]} \in \mathcal{S}^n$ be such that $\mathcal{D} = \{s_i\}_{i\in[n]}$ and $\mathcal{D}'$ are neighboring, and without loss of generality assume $s'_1 \ne s_1$. Define the multiset

$$\mathcal{I} := \{z_{i,j} \mid i \in [k], j \in [T_i]\} \qquad (2.13)$$

to contain all sampled indices in $[n]$ throughout Algorithm 13. We begin by giving an (approximate) RDP guarantee conditioned on the number of times "1" appears in $\mathcal{I}$. The proof of Lemma 2.4.7 is primarily based on providing a potential-based proof of a "drift bound,"

i.e. how far away iterates produced by two neighboring datasets drift apart (coupling all other randomness used). To carry out this potential proof, we rely on the local stability properties afforded by Lemma 2.2.4.

Lemma 2.4.7. *Define $\mathcal{I}$ as in (2.13) in one call to Algorithm 13. Let $\mathcal{I}$ be deterministic (i.e. this statement is conditioned on the realization of $\mathcal{I}$). Let b be the number of times the index 1 appears in $\mathcal{I}$. Let μ be the distribution of the output of Algorithm 13 run on $\mathcal{D}$, and μ' be the distribution when run on $\mathcal{D}'$, such that $\mathcal{D}$ and $\mathcal{D}'$ are neighboring and differ in the first entry, and the only randomness is in the Gaussian samples used to define ReSQue estimators and on Line 11. Suppose $\frac{\rho}{r} \geq 1728 \log^2(\frac{\log T}{\delta})$. Then we have for any $\alpha > 1$,*

$$D_{\alpha,\delta}(\mu \| \mu') \leq 1500 \alpha \beta^2 b^2.$$

Proof. Throughout this proof we treat $\mathcal{I}$ as fixed with b occurrences of the index 1. Let b_i be the number of times 1 appears in $\mathcal{I}_i := \{z_{i,j} \mid j \in [T_i]\}$, such that $\sum_{i \in [k]} b_i = b$. We first analyze the privacy guarantee of one loop, and then analyze the privacy of the whole algorithm.

We begin by fixing some $i \in [k]$, and analyzing the RDP of the i^{th} outer loop in Algorithm 13, conditioned on the starting point y_0. Consider a particular realization of the T_i Gaussian samples used in implementing Line 8, $\Xi_i := \{\xi_{i,j}\}_{j \in [T_i]}$, where we let $\xi_{i,j} \sim \mathcal{N}(0, \rho^2 \mathbf{I}_d)$ denote the Gaussian sample used to define the update to y_{j-1}. Conditioned on the values of $\mathcal{I}_i$, Ξ_i, the i^{th} outer loop in Algorithm 13 (before adding ζ_i in Line 11) is a deterministic map. For a given realization of $\mathcal{I}_i$ and Ξ_i, we abuse notation and denote $\{y_j\}_{j \in [T_i]}$ to be the iterates of the i^{th} outer loop in Algorithm 13 using the dataset $\mathcal{D}$ starting at y_0, and $\{y'_j\}_{j \in [T_i]}$ similarly using $\mathcal{D}'$. Finally, define

$$\Phi_j := \left\| y_j - y'_j \right\|^2, \; p := \left\lceil 5 \log \left(\frac{\log T}{\delta} \right) \right\rceil.$$

In the following parts of the proof, we will bound for this p the quantity $\mathbb{E}\, \Phi_{T_i}^p$, to show that with high probability it remains small at the end of the loop, regardless of the location of the 1 indices.

Potential growth: iterates with $z_{i,j} \neq 1$. We first bound the potential growth in any iteration $j \in [T_i]$ where $z_{i,j} \neq 1$. Fix y_0, y_0' and $\{\xi_{i,t}\}_{t\in[j-1]}$, so that Φ_{j-1} is deterministic. We have (taking expectations over only $\xi_{i,j}$),

$$\mathop{\mathbb{E}}_{\xi_{i,j}} \Phi_j^p \leq \mathbb{E}\left(\Phi_{j-1} + A_j + B_j\right)^p, \tag{2.14}$$

where

$$\begin{aligned} A_j &:= -2\eta_i Z_j \left\langle \partial f^{z_{i,j}}(\overline{x} + \xi_{i,j}), y_{j-1} - y_{j-1}' \right\rangle, \\ B_j &:= \eta_i^2 Z_j^2 \left\| \partial f^{z_{i,j}}(\overline{x} + \xi_{i,j}) \right\|^2, \quad \text{and} \\ Z_j &:= \frac{\gamma_\rho(y_{j-1} - \overline{x} - \xi_{i,j}) - \gamma_\rho(y_{j-1}' - \overline{x} - \xi_{i,j})}{\gamma_\rho(\xi_{i,j})}. \end{aligned}$$

The inequality in (2.14) follows from expanding the definition of the update to Φ_j before projection, and then using the fact that Euclidean projections onto a convex set only decrease distances. By the second part of Lemma 2.2.4, for all $q \in [2, p]$, if $\sqrt{\Phi_{j-1}} \leq \frac{\rho}{p}$ (which is always satisfied as $\sqrt{\Phi_{j-1}} \leq r$),

$$\mathop{\mathbb{E}}_{\xi_{i,j}} Z_j^q \leq \left(\frac{24q\sqrt{\Phi_{j-1}}}{\rho}\right)^q.$$

By Lipschitzness of $f^{z_{i,j}}$ and Cauchy-Schwarz (on A_j), we thus have

$$\begin{aligned} \mathop{\mathbb{E}}_{\xi_{i,j}} |A_j|^q &\leq \left(\frac{48\eta_i L q \Phi_{j-1}}{\rho}\right)^q \text{ for all } q \in [2, p], \\ \mathop{\mathbb{E}}_{\xi_{i,j}} B_j^q &\leq \left(\frac{48\eta_i L q}{\rho}\right)^{2q} \Phi_{j-1}^q \text{ for all } q \in [1, p]. \end{aligned} \tag{2.15}$$

Next, we perform a Taylor expansion of (2.14), which yields

$$\begin{aligned} \mathop{\mathbb{E}}_{\xi_{i,j}} \Phi_j^p \leq \Phi_{j-1}^p &+ p\Phi_{j-1}^{p-1} \mathop{\mathbb{E}}_{\xi_{i,j}} \left[A_j + B_j\right] \\ &+ p(p-1) \int_0^1 (1-t) \mathop{\mathbb{E}}_{\xi_{i,j}} \left[\left(\Phi_{j-1} + t(A_j + B_j)\right)^{p-2} (A_j + B_j)^2\right] \mathrm{d}t. \end{aligned} \tag{2.16}$$

By monotonicity of convex gradients and the first part of Lemma 2.2.3, we have

$$\mathop{\mathbb{E}}_{\xi_{i,j}}[A_j] = -2\eta_i \left\langle \partial \widehat{f}_\rho^{z_{i,j}}(y_{j-1}) - \partial \widehat{f}_\rho^{z_{i,j}}(y'_{j-1}), y_{j-1} - y'_{j-1} \right\rangle \le 0. \tag{2.17}$$

By applying (2.15), we have

$$p\Phi_{j-1}^{p-1} \mathop{\mathbb{E}}_{\xi_{i,j}} B_j \le p \left(\frac{48\eta_i L}{\rho}\right)^2 \Phi_{j-1}^p. \tag{2.18}$$

Next we bound the second-order terms. For any $t \in [0,1]$ we have denoting $C_j := A_j + B_j$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\xi_{i,j}}\left[(\Phi_{j-1} + tC_j)^{p-2} C_j^2\right] &= \sum_{q=0}^{p-2} \binom{p-2}{q} \Phi_{j-1}^{p-2-q} \mathop{\mathbb{E}}_{\xi_{i,j}}\left[t^{2+q} C_j^{2+q}\right] \\
&\le 4\sum_{q=0}^{p-2} 2^q \binom{p-2}{q} \Phi_{j-1}^{p-2-q} \mathop{\mathbb{E}}_{\xi_{i,j}}\left[|A_j|^{2+q}\right] \\
&+ 4\sum_{q=0}^{p-2} 2^q \binom{p-2}{q} \Phi_{j-1}^{p-2-q} \mathop{\mathbb{E}}_{\xi_{i,j}}\left[B_j^{2+q}\right] \\
&\le 4\Phi_{j-1}^p \left(\frac{48\eta_i Lp}{\rho}\right)^2 \sum_{q=0}^{p-2} 2^q \binom{p-2}{q} \left(\frac{48\eta_i Lq}{\rho}\right)^q \\
&+ 4\Phi_{j-1}^p \left(\frac{48\eta_i Lp}{\rho}\right)^2 \sum_{q=0}^{p-2} 2^q \binom{p-2}{q} \left(\frac{48\eta_i L(2+q)}{\rho}\right)^{2q+2} \\
&\le 8\Phi_{j-1}^p \left(\frac{48\eta_i Lp}{\rho}\right)^2 \left(1 + \frac{96\eta_i Lp}{\rho}\right)^{p-2} \\
&\le 16\Phi_{j-1}^p \left(\frac{48\eta_i Lp}{\rho}\right)^2.
\end{aligned} \tag{2.19}$$

The first inequality used $(a+b)^p \le 2^p(a^p + b^p)$ for any nonnegative a, b and $0 \le t \le 1$, the second inequality used (2.15), and the third and fourth inequalities used

$$\frac{48\eta_i L(2+q)}{\rho} \le \frac{1}{2p}$$

for our choices of $\eta_i L \le \frac{r}{4}$ and ρ. Finally, plugging (2.17), (2.18), and (2.19) into (2.16),

$$\mathop{\mathbb{E}}_{\xi_{i,j}} \Phi_j^p \le \Phi_{j-1}^p \left(1 + 16p^2 \left(\frac{48\eta_i L p}{\rho}\right)^2\right) \le \Phi_{j-1}^p \left(1 + 16p \left(\frac{48\eta_i L p}{\rho}\right)^2\right)^p.$$

Finally, using $(\eta_i L)^2 \le \frac{r^2}{16T} \le \frac{r^2}{16T_i}$ and our assumed bound on $\frac{r}{\rho}$, which implies $\frac{16p}{\rho^2}(48\eta_i L p)^2 \le \frac{1}{T_i}$, taking expectations over $\{\xi_t\}_{t\in[j-1]}$ yields

$$\mathbb{E}\,\Phi_j^p \le \mathbb{E}\,\Phi_{j-1}^p \left(1 + \frac{1}{T_i}\right)^p \text{ when } z_{i,j} \ne 1. \tag{2.20}$$

Potential growth: iterates with $z_{i,j} = 1$. Next, we handle the case where $z_{i,j} = 1$. We have that conditional on fixed values of $\{\xi_{i,t}\}_{t\in[j-1]}$, y_0 and y_0',

$$\begin{aligned} \mathop{\mathbb{E}}_{\xi_{i,j}} \Phi_j^p &\le \mathop{\mathbb{E}}_{\xi_{i,j}} (\Phi_{j-1} + D_j + E_j)^p \\ &\le \mathop{\mathbb{E}}_{\xi_{i,j}} \left(\left(1 + \frac{1}{b_i}\right)\Phi_{j-1} + 2b_i E_j\right)^p, \end{aligned} \tag{2.21}$$

where overloading $f \leftarrow f(\cdot; s_1)$, $h \leftarrow f(\cdot; s_1')$,

$$\begin{aligned} D_j &:= -2\eta_i \left\langle \widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho(y_{j-1}) - \widetilde{\nabla}_{\overline{x}} \widehat{h}_\rho(y'_{j-1}), y_{j-1} - y'_{j-1} \right\rangle, \\ E_j &:= \eta_i^2 \left\| \widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho(y_{j-1}) - \widetilde{\nabla}_{\overline{x}} \widehat{h}_\rho(y'_{j-1}) \right\|^2, \end{aligned}$$

and we use $D_j \le \frac{1}{b_i}\Phi_{j-1} + b_i E_j$ by Cauchy-Schwarz and Young's inequality. Next, convexity of $\|\cdot\|^{2q}$ implies that

$$E_j^q \le \eta_i^{2q} 2^{2q-1} \left(\left\| \widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho(y_{j-1}) \right\|^{2q} + \left\| \widetilde{\nabla}_{\overline{x}} \widehat{h}_\rho(y'_{j-1}) \right\|^{2q} \right).$$

Next, we note that since f is Lipschitz, the first part of Lemma 2.2.4 implies for all $q \le p$,

$$\mathbb{E} \left\| \widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho(y_{j-1}) \right\|^{2q} \le L^{2q}\, \mathbb{E} \left[\left(\frac{\gamma_\rho(y_{j-1} - \overline{x} - \xi)}{\gamma_\rho(\xi)} \right)^{2q} \right] \le 2(L)^{2q},$$

and a similar calculation holds for h. Here we used our assumed bound on $\frac{r}{\rho}$ to check the

requirement in Lemma 2.2.4 is satisfied. By linearity of expectation, we thus have

$$\mathop{\mathbb{E}}_{\xi_{i,j}} E_j^q \le (9\eta_i L)^{2q}\,. \tag{2.22}$$

Finally, expanding (2.21) and plugging in the moment bound (2.22),

$$\begin{aligned}
\mathop{\mathbb{E}}_{\xi_{i,j}} \Phi_j^p &\le \sum_{q=0}^{p} \binom{p}{q} \left(1+\frac{1}{b_i}\right)^q \Phi_{j-1}^q (2b_i)^{p-q} \mathop{\mathbb{E}}_{\xi_{i,j}} \left[E_j^{p-q}\right] \\
&\le \sum_{q=0}^{p} \binom{p}{q} \left(1+\frac{1}{b_i}\right)^q \Phi_{j-1}^q (2b_i)^{p-q} (9\eta_i L)^{2(p-q)} \\
&= \left(\left(1+\frac{1}{b_i}\right)\Phi_{j-1} + 2b_i(9\eta_i L)^2\right)^p .
\end{aligned}$$

Taking expectations over $\{\xi_{i,t}\}_{t\in[j-1]}$, and using Fact 2.5.3 with $Z \leftarrow (1+\frac{1}{b_i})\Phi_{j-1}$ and $C \leftarrow 2b_i(9\eta_i L)^2$,

$$\mathbb{E}\,\Phi_j^p \le \left(\left(1+\frac{1}{b_i}\right)\mathbb{E}\left[\Phi_{j-1}^p\right]^{\frac{1}{p}} + 2b_i(9\eta_i L)^2\right)^p, \text{ when } z_{i,j}=1. \tag{2.23}$$

One loop privacy. We begin by obtaining a high-probability bound on Φ_{T_i}. Define

$$W_j := \mathbb{E}[\Phi_j^p]^{\frac{1}{p}}.$$

By using (2.20) and (2.23), we observe

$$W_j \le \begin{cases} \left(1+\frac{1}{T_i}\right) W_{j-1} & z_{i,j}\neq 1 \\ \left(1+\frac{1}{b_i}\right) W_{j-1} + 2b_i(9\eta_i L)^2 & z_{i,j}=1 \end{cases}.$$

Hence, regardless of the b_i locations of the 1 indices in $\mathcal{I}_i$, we have

$$W_{T_i} \le \left(1+\frac{1}{T_i}\right)^{T_i}\left(1+\frac{1}{b_i}\right)^{b_i}\left(2b_i^2(9\eta_i L)^2\right) \le 1200 b_i^2(\eta_i L)^2.$$

Thus, by Markov's inequality, with probability at least $1-\frac{\delta}{\log T}$ over the randomness of

$\Xi_i = \{\xi_{i,j}\}_{j\in[T_i]}$, we have using our choice of p,

$$\left\|y_{T_i} - y'_{T_i}\right\|^2 \le 1200 b_i^2 (\eta_i L)^2 \cdot \left(\frac{\log T}{\delta}\right)^{\frac{1}{p}} \le 1500 b_i^2 (\eta_i L)^2. \tag{2.24}$$

In the last inequality, we used our choice of p. Call $\mathcal{E}_i$ the event that the sampled Ξ_i admits a deterministic map which yields the bound in (2.24). By the second part of Proposition 2.4.2, the conditional distribution of the output of the i^{th} outer loop under $\mathcal{E}_i$ satisfies $(\alpha, 1500\beta^2 b_i^2)$-RDP, where we use the value of σ_i in Line 4 of Algorithm 13. We conclude via Fact 2.1.1 with $\mathcal{E} \leftarrow \mathcal{E}_i$ that the i^{th} outer loop of Algorithm 13 satisfies

$$\left(\alpha, 1500\alpha\beta^2 b_i^2, \frac{\delta}{\log T}\right)\text{-RDP}.$$

All loops privacy. By applying composition of RDP (the third part of Proposition 2.4.2), for a given realization of $\mathcal{I} = \cup_{i\in[k]}\mathcal{I}_i$ with b occurrences of 1, applying composition over the $\log T$ outer iterations (Lemma 2.4.4), Algorithm 13 satisfies

$$\left(\alpha, 1500\alpha\beta^2 b^2, \delta\right)\text{-RDP}.$$

Here, we used $\sum_{i\in[k]} b_i^2 \le b^2$. This is the desired conclusion. □

We next apply amplification by subsampling to boost the guarantee of Lemma 2.4.7. To do so, we use the following key Proposition 2.4.8, which was proven in [BDRS18]. The use case in [BDRS18] involved subsampling with replacement and was used in a framework they introduced termed truncated CDP, but we will not need the framework except through the following powerful fact.

Proposition 2.4.8 (Theorem 12, [BDRS18])**.** *Let $\tau \le \frac{1}{3}$, $s \in (0, \frac{1}{40})$. Let P, Q, R be three distributions over the same probability space, such that for each pair $P_1, P_2 \in \{P, Q, R\}$, we have $D_\alpha(P_1\|P_2) \le \alpha\tau$ for all $\alpha > 1$. Then for all $\alpha \in (1, \frac{3}{\tau})$,*

$$D_\alpha(sP + (1-s)R\|sQ + (1-s)R) \le 13 s^2 \alpha\tau.$$

We also require a straightforward technical fact about binomial distributions.

Lemma 2.4.9. *Let $m, n \in \mathbb{N}$ satisfy $\frac{m}{n} \leq \frac{1}{60}$. Consider the following partition of the elements $\mathcal{I} \in [n]^m$ with at most b copies of 1:*

$$S_0 := \{\mathcal{I} \in [n]^m \mid \mathcal{I}_i \neq 1 \text{ for all } i \in [m]\},$$
$$S_1 := \{\mathcal{I} \in [n]^m \mid \mathcal{I}_i = 1 \text{ for between 1 and } b \text{ many } i \in [m]\}.$$

Let π_0 and π_1 be the uniform distributions on S_0 and S_1 respectively. Then there exists a coupling $\Gamma(\pi_0, \pi_1)$ such that for all $(\mathcal{I}, \mathcal{I}')$ in the support of Γ,

$$\left|\left\{i \mid \mathcal{I}_i \neq \mathcal{I}'_i\right\}\right| \leq b.$$

Proof. Define a probability distribution p on elements of $[b]$ such that

$$p_a := \frac{\binom{m}{a}(n-1)^{m-a}}{\sum_{a \in [b]} \binom{m}{a}(n-1)^{m-a}} \text{ for all } a \in [b].$$

Clearly, $\sum_{a \in [b]} p_a = 1$. Our coupling $\Gamma := \Gamma(\pi_0, \pi_1)$ is defined as follows.

1. Draw $\mathcal{I} \sim \pi_0$ and $a \sim p$ independently.

2. Let $\mathcal{I}'$ be $\mathcal{I}$ with a uniformly random subset of a indices replaced with 1. Return $(\mathcal{I}, \mathcal{I}')$.

This coupling satisfies the requirement, so it suffices to verify it has the correct marginals. This is immediate for S_0 by definition. For $\mathcal{I}' \in S_1$, suppose $\mathcal{I}'$ has a occurrences of the index 1. The total probability $\mathcal{I}'$ is drawn from Γ is then indeed

$$\frac{(n-1)^a}{(n-1)^m} \cdot \frac{p_a}{\binom{m}{a}} = \frac{1}{\sum_{a \in [b]} \binom{m}{a}(n-1)^{m-a}} = \frac{1}{|S_1|}.$$

The first equality follows as the probability we draw $\mathcal{I} \sim \pi_0$ which agrees with $\mathcal{I}'$ on all the non-1 locations is $(n-1)^{a-m}$, and the probability $\mathcal{I}'$ is drawn given that we selected $\mathcal{I}$ is $p_a \cdot \binom{m}{a}^{-1}$. □

Finally, we are ready to state our main privacy guarantee for Algorithm 13.

Lemma 2.4.10. *There is a universal constant $C_{\text{priv}} \in [1, \infty)$, such that if $\frac{T}{n} \le \frac{1}{C_{\text{priv}}}$, $\beta^2 \log^2(\frac{1}{\delta}) \le \frac{1}{C_{\text{priv}}}$, $\delta \in (0, \frac{1}{6})$, and $\frac{\rho}{r} \ge C_{\text{priv}} \log^2(\frac{\log T}{\delta})$, Algorithm 13 satisfies $(\alpha, \alpha\tau, \delta)$-RDP for*

$$\tau := C_{\text{priv}} \left(\beta \log\left(\frac{1}{\delta}\right) \cdot \frac{T}{n} \right)^2, \ \alpha \in \left(1, \frac{1}{C_{\text{priv}} \beta^2 \log^2(\frac{1}{\delta})}\right).$$

Proof. Let $\mathcal{D}$, $\mathcal{D}'$ be neighboring, and without loss of generality, suppose they differ in the first entry. Let $C_{\text{priv}} \ge 60$, and let $\mathcal{I}$ be defined as in (2.13). Let $\mathcal{E}$ be the event that $\mathcal{I}$ contains at most b copies of the index 1, where

$$b := 2 \log\left(\frac{2}{\delta}\right).$$

By a Chernoff bound, $\mathcal{E}$ occurs with probability at least $1 - \frac{\delta}{2}$ over the randomness of $\mathcal{I}$. We define P to be the distribution of the output of Algorithm 13 when run on $\mathcal{D}$, conditioned on $\mathcal{E}$ and $\mathcal{I}$ containing at least one copy of the index 1 (call this total conditioning event $\mathcal{E}_1$, i.e. there are between 1 and b copies of the index 1). Similarly, we define Q to be the distribution when run on $\mathcal{D}'$ conditioned on $\mathcal{E}_1$, and R to be the distribution conditioned on $\mathcal{E} \cap \mathcal{E}_1^c$ (when run on either $\mathcal{D}$ or $\mathcal{D}'$). We claim that for all $P_1, P_2 \in \{P, Q, R\}$, we have

$$D_{\alpha, \frac{\delta}{2}}(P_1 \| P_2) \le 1500 \alpha \beta^2 b^2, \text{ for all } \alpha > 1. \tag{2.25}$$

To see (2.25) for $P_1 = P$ and $P_2 = Q$ (or vice versa), we can view P, Q as mixtures of outcomes conditioned on the realization $\mathcal{I}$. Then, applying quasiconvexity of Rènyi divergence (over this mixture), and applying Lemma 2.4.7 (with $\delta \leftarrow \frac{\delta}{2}$), we have the desired claim. To see (2.25) for the remaining cases, we first couple the conditional distributions under $\mathcal{E}_1$ and $\mathcal{E} \cap \mathcal{E}_1^c$ by their index sets, according to the coupling in Lemma 2.4.9. Then applying quasiconvexity of Rényi divergence (over this coupling) again yields the claim, where we set $m \leftarrow \widehat{T} - 1 \le T$. Finally, let

$$s := \Pr[\mathcal{E}_1 \mid \mathcal{E}] = 1 - \frac{\left(1 - \frac{1}{n}\right)^{\widehat{T}-1}}{\Pr[\mathcal{E}]} \le 1 - \frac{1 - \frac{1.1T}{n}}{1 - \frac{\delta}{2}} \le \frac{1.2T}{n}.$$

Note that conditional on $\mathcal{E}$ and the failure event in Lemma 2.4.7 not occurring, the distributions of Algorithm 13 using $\mathcal{D}$ and $\mathcal{D}'$ respectively are $sP + (1-s)R$ and $sQ + (1-s)R$. Hence, union bounding with $\mathcal{E}^c$ (see Fact 2.1.1), the claim follows from Proposition 2.4.8 with $\tau \leftarrow 6000\beta^2 \log^2(\frac{2}{\delta})$. □

Regularized extension. We give a slight extension to Algorithm 13 which handles regularization, and enjoys similar utility and privacy guarantees as stated in Proposition 2.4.5. Let

$$x^\star_{\overline{x},\lambda} := \operatorname*{argmin}_{x \in \mathbb{B}_{\overline{x}}(r)} \left\{ \widehat{f^{\mathrm{erm}}_\rho}(x) + \frac{\lambda}{2} \|x - \overline{x}\|^2 \right\}. \tag{2.26}$$

Our extension Algorithm 13 is identical to Algorithm 13, except it requires a regularization parameter λ, allows for an arbitrary starting point with an expected distance bound (adjusting the step size accordingly), and takes composite projected steps incorporating the regularization.

Algorithm 9: Subsampled ReSQued ERM solver, regularized case, convex rate

Input: $\overline{x} \in \mathbb{R}^d$, ball radius, convolution radius, privacy parameter, and regularization parameter $r, \rho, \beta, \lambda > 0$, dataset $\mathcal{D} \in \mathcal{S}^n$, iteration count $T \in \mathbb{N}$, distance bound $r' \in [0, 2r]$, initial point $x_0 \in \mathbb{B}_{\overline{x}}(r)$ satisfying $\mathbb{E}\|x_0 - x^\star_{\overline{x},\lambda}\|^2 \le (r')^2$
$\widehat{T} \leftarrow 2^{\lfloor \log_2 T \rfloor}$, $k \leftarrow \log_2 \widehat{T}$, $\eta \leftarrow \frac{r'}{L} \min(\frac{1}{\sqrt{T}}, \frac{\beta}{\sqrt{d}})$
for $i \in [k]$ **do**
  $T_i \leftarrow 2^{-i}\widehat{T}$, $\eta_i \leftarrow 4^{-i}\eta$, $\sigma_i \leftarrow \frac{L\eta_i}{\beta}$
  $y_0 \leftarrow x_{i-1}$
  for $j \in [T_i]$ **do**
    $z_{i,j} \sim_{\text{unif.}} [n]$
    $y_j \leftarrow \operatorname{argmin}_{y \in \mathbb{B}_{\overline{x}}(r)} \{\langle \eta_i \widetilde{\nabla}_{\overline{x}} \widehat{f}^{z_{i,j}}_\rho(y_{j-1}), y\rangle + \frac{1}{2}\|y - y_{j-1}\|^2 + \frac{\eta_i\lambda}{2}\|y - \overline{x}\|^2\}$
  end
  $\overline{y}_i \leftarrow \frac{1}{T_i}\sum_{j \in [T_i]} y_j$
  $x_i \leftarrow \overline{y}_i + \zeta_i$, for $\zeta_i \sim \mathcal{N}(0, \sigma_i^2 \mathbf{I}_d)$
end
return x_k

Corollary 2.4.11. *Let $x^\star_{\overline{x},\lambda}$ be defined as in (2.26). Algorithm 13 uses at most T gradients*

and produces $x \in \mathbb{B}_{\overline{x}}(r)$ *such that, for a universal constant* C_{cvx},

$$\mathbb{E}\left[\widehat{f_\rho^{\mathrm{erm}}}(x) + \frac{\lambda}{2}\left\|x - \overline{x}\right\|^2\right] - \left(\widehat{f_\rho^{\mathrm{erm}}}(x^\star_{\overline{x},\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2\right) \le C_{\mathrm{cvx}} L r' \left(\frac{\sqrt{d}}{\beta T} + \frac{1}{\sqrt{T}}\right).$$

Moreover, there is a universal constant $C_{\mathrm{priv}} \ge 1$, *such that if* $\frac{T}{n} \le \frac{1}{C_{\mathrm{priv}}}$, $\beta^2 \log^2(\frac{1}{\delta}) \le \frac{1}{C_{\mathrm{priv}}}$, $\delta \in (0, \frac{1}{6})$, *and* $\frac{\rho}{r} \ge C_{\mathrm{priv}} \log^2(\frac{\log T}{\delta})$, *Algorithm 13 satisfies* $(\alpha, \alpha\tau, \delta)$*-RDP for*

$$\tau := C_{\mathrm{priv}}\left(\beta \log\left(\frac{1}{\delta}\right) \cdot \frac{T}{n}\right)^2, \ \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}}\beta^2 \log^2(\frac{1}{\delta})}\right).$$

Proof. The proof is almost identical to Proposition 2.4.5, so we only discuss the differences. Throughout this proof, for notational convenience, we define

$$F^\lambda(x) := \widehat{f_\rho^{\mathrm{erm}}}(x) + \frac{\lambda}{2}\left\|x - \overline{x}\right\|^2.$$

Utility. Standard results on composite stochastic mirror descent (e.g. Lemma 12 of [CJST19]) show the utility bound in (2.12) still holds with F^λ in place of F. In particular each term $\mathbb{E}[F^\lambda(\overline{y}_i) - F^\lambda(\overline{y}_{i-1})]$ as well as $\mathbb{E}[F^\lambda(x_k) - F^\lambda(\overline{y}_k)]$ enjoys the same bound as its counterpart in (2.12). The only other difference is that, defining $\zeta_0 := x_0 - x^\star_{\overline{x},\lambda}$ in the proof of Lemma 2.4.6, we have $\mathbb{E}\,\zeta_0^2 \le (r')^2$ in place of the bound r^2, and we appropriately changed η to scale as r' instead.

Privacy. The subsampling-based reduction from Lemma 2.4.10 to Lemma 2.4.7 is identical, so we only discuss how to obtain an analog of Lemma 2.4.7 for Algorithm 13. In each iteration $j \in [T_i]$, by completing the square, we can rewrite Line 8 as

$$y_j \leftarrow \operatorname*{argmin}_{y \in \mathbb{B}_{\overline{x}}(r)} \left\{\frac{1}{2}\left\|y - \left(\frac{1}{1+\eta_i\lambda} y_{j-1} + \frac{\eta_i\lambda}{1+\eta_i\lambda}\overline{x} - \frac{\eta_i}{1+\eta_i\lambda}\widetilde{\nabla}_{\overline{x}} \widehat{f}_\rho^{z_{i,j}}(y_{j-1})\right)\right\|^2\right\}.$$

Now consider our (conditional) bounds on $\mathbb{E}_{\xi_{i,j}} \Phi_j$ in (2.14) and (2.21). We claim these still hold true; before projection, the same arguments used in (2.14) and (2.21) still hold (in fact improve by $(1+\eta_i\lambda)^2$), and projection only decreases distances. Finally, note that the proof of Lemma 2.4.7 only used the choice of step size η through $\eta L\sqrt{T} \le r$ and used

the assumed bound on $\frac{r}{\rho}$ to bound the drift growth. As we now have $\eta L\sqrt{T} \le r' \le 2r$, we adjusted the assumed bound on $\frac{r}{\rho}$ by a factor of 2. The remainder of the proof of Lemma 2.4.7 is identical. □

Without loss of generality, C_{priv} is the same constant in Proposition 2.4.5 and Corollary 2.4.11, since we can set both to be the maximum of the two. The same logic applies to the following Proposition 2.4.14 and Lemma 2.4.16 (which will also be parameterized by a C_{priv}) so we will not repeat it. Finally, the following fact about initial error will also be helpful in the following Section 2.4.3.

Lemma 2.4.12. *We have*

$$\widehat{f_\rho^{\text{erm}}}(\overline{x}) - \left(\widehat{f_\rho^{\text{erm}}}(x^\star_{\overline{x},\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2\right) \le \frac{2L^2}{\lambda}.$$

Proof. By strong convexity and Lipschitzness of $\widehat{f_\rho^{\text{erm}}}$, we have

$$\begin{aligned}\frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2 &\le \widehat{f_\rho^{\text{erm}}}(\overline{x}) - \left(\widehat{f_\rho^{\text{erm}}}(x^\star_{\overline{x},\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|^2\right)\\ &\le \widehat{f_\rho^{\text{erm}}}(\overline{x}) - \widehat{f_\rho^{\text{erm}}}(x^\star_{\overline{x},\lambda}) \le L\left\|x^\star_{\overline{x},\lambda} - \overline{x}\right\|.\end{aligned}$$

Rearranging gives $\|x^\star_{\overline{x},\lambda} - \overline{x}\| \le \frac{2L}{\lambda}$, which can be plugged in above to yield the conclusion. □

We also state a slight extension to Lemma 2.4.12 which will be used in Section 2.4.5.

Lemma 2.4.13. *Define* $x^\star_{\overline{x},x',\lambda} := \operatorname{argmin}_{x\in\mathbb{B}_{\overline{x}}(r)}\{\widehat{f_\rho^{\text{erm}}}(x) + \frac{\lambda}{2}\|x - x'\|^2\}$, *where* $x' \in \mathbb{R}^d$ *is not necessarily in* $\mathbb{B}_{\overline{x}}(r)$. *Let* $x_0 := \Pi_{\mathbb{B}_{\overline{x}}(r)}(x')$. *We have*

$$\left(\widehat{f_\rho^{\text{erm}}}(x_0) + \frac{\lambda}{2}\left\|x_0 - x'\right\|^2\right) - \left(\widehat{f_\rho^{\text{erm}}}(x^\star_{\overline{x},x',\lambda}) + \frac{\lambda}{2}\left\|x^\star_{\overline{x},x',\lambda} - x'\right\|^2\right) \le \frac{2L^2}{\lambda}.$$

Proof. The proof is identical to Lemma 2.4.12, where we use $\frac{\lambda}{2}\|x_0 - x'\|^2 \le \frac{\lambda}{2}\|x^\star_{\overline{x},x',\lambda} - x'\|^2$. □

2.4.3 Subsampled smoothed ERM solver: the strongly convex case

We next give an ERM algorithm similar to Algorithm 13, but enjoys an improved optimization rate. In particular, it again attains RDP bounds improving with the subsampling parameter $\frac{T}{n}$, and we obtain error guarantees against $x^\star_{\overline{x},\lambda}$ defined in (2.26) at a rate decaying as $\frac{1}{T}$ or better.

Algorithm 10: Subsampled ReSQued ERM solver, strongly convex case

Input: $\overline{x} \in \mathbb{R}^d$, ball radius, convolution radius, privacy parameter, and regularization parameter $r, \rho, \beta, \lambda > 0$, dataset $\mathcal{D} \in \mathcal{S}^n$, iteration count $T \in \mathbb{N}$
$k \leftarrow \lceil \log \log T \rceil, x_0 \leftarrow \overline{x}$
for $i \in [k]$ **do**
  $\beta_{i-1} \leftarrow 2^{\frac{k-i+1}{2}}\beta$, $r_{i-1} \leftarrow \min(2r, \sqrt{\frac{2D_{i-1}}{\lambda}})$ (see (2.27)), $T_{i-1} \leftarrow 2^{i-1-k}T$
  $x_i \leftarrow$ output of Algorithm 13 with inputs $(\overline{x}, r, \rho, \beta_{i-1}, \lambda, \mathcal{D}, T_{i-1}, r_{i-1}, x_{i-1})$
end
return x_{k+1}

We now give our analysis of Algorithm 7 below. The proof follows a standard reduction template from the strongly convex case to the convex case (see e.g. Lemma 4.7 in [KLL21]).

Proposition 2.4.14. *Let $x^\star_{\overline{x},\lambda}$ be defined as in (2.26). Algorithm 7 uses at most T gradients and produces x such that, for a universal constant C_{sc},*

$$\mathbb{E}\left[\widehat{f^{\mathrm{erm}}_\rho}(x) + \frac{\lambda}{2}\|x - \overline{x}\|^2\right] - \widehat{f^{\mathrm{erm}}_\rho}(x^\star_{\overline{x},\lambda}) - \frac{\lambda}{2}\|x^\star_{\overline{x},\lambda} - \overline{x}\|^2 \le \frac{C_{\mathrm{sc}}L^2}{\lambda}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right).$$

Moreover, there is a universal constant $C_{\mathrm{priv}} \ge 1$, such that if $\frac{T}{n} \le \frac{1}{C_{\mathrm{priv}}}$, $\beta^2 \log^2(\frac{\log\log T}{\delta}) \le \frac{1}{C_{\mathrm{priv}}}$, $\delta \in (0, \frac{1}{6})$, and $\frac{\rho}{r} \ge C_{\mathrm{priv}} \log^2(\frac{\log T}{\delta})$, Algorithm 7 satisfies $(\alpha, \alpha\tau, \delta)$-RDP for

$$\tau := C_{\mathrm{priv}}\left(\beta \log\left(\frac{\log\log T}{\delta}\right) \cdot \frac{T}{n}\right)^2, \ \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}}\beta^2 \log^2(\frac{\log\log T}{\delta})}\right).$$

Proof. We analyze the utility and privacy separately.

Utility. Denote for simplicity $F^\lambda(x) := \widehat{f_\rho^{\mathrm{erm}}}(x) + \frac{\lambda}{2}\|x - \overline{x}\|^2$, $F_\star^\lambda := F^\lambda(x^\star_{\overline{x},\lambda})$, and $\Delta_i := \mathbb{E}[F^\lambda(x_i) - F_\star^\lambda]$. Moreover, define for all $0 \le i \le k$,

$$E_i := \frac{2C_{\mathrm{cvx}}^2 L^2}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta_i T_i} + \frac{1}{\sqrt{T_i}}\right)^2, \; D_i := 4E_i \sqrt[2^i]{\frac{2L^2}{\lambda} \cdot \frac{1}{4E_0}}, \tag{2.27}$$

where we define $T_k = T$ and $\beta_k = \beta$. By construction, for all $0 \le i \le k-1$, $E_{i+1} = \frac{1}{2}E_i$, and so

$$\frac{D_{i+1}}{4E_{i+1}} = \sqrt{\frac{D_i}{4E_i}} \implies \sqrt{D_i E_i} = D_{i+1}. \tag{2.28}$$

We claim inductively that for all $0 \le i \le k$, $\Delta_i \le D_i$. The base case of the induction follows because by Lemma 2.4.12, we have $\Delta_0 \le \frac{2L^2}{\lambda} = D_0$. Next, suppose that the inductive hypothesis is true up to iteration i. By strong convexity,

$$\mathbb{E}\left[\left\|x_i - x^\star_{\overline{x},\lambda}\right\|^2\right] \le \frac{2\Delta_i}{\lambda} \le \frac{2D_i}{\lambda},$$

where we used the inductive hypothesis. Hence, the expected radius upper bound (defined by r_i) is valid for the call to Algorithm 13. Thus, by Corollary 2.4.11,

$$\begin{aligned}\Delta_{i+1} = \mathbb{E}\left[F^\lambda(x_{i+1}) - F_\star^\lambda\right] &\le C_{\mathrm{cvx}} L r_i \left(\frac{\sqrt{d}}{\beta_i T_i} + \frac{1}{\sqrt{T_i}}\right) \\ &\le C_{\mathrm{cvx}} L \sqrt{\frac{2D_i}{\lambda}} \left(\frac{\sqrt{d}}{\beta_i T_i} + \frac{1}{\sqrt{T_i}}\right) = \sqrt{D_i E_i} = D_{i+1}.\end{aligned}$$

Here we used (2.28) in the last equation, which completes the induction. Hence, iterating (2.28) for $k = \lceil \log_2 \log_2 T \rceil$ iterations, where we use $E_0 \ge \frac{L^2}{2\lambda T}$ so that $D_k \le 8E_k$, we have

$$\Delta_k \le 8E_k \le \frac{32C_{\mathrm{cvx}}^2 L^2}{\lambda} \cdot \left(\frac{d}{\beta^2 T^2} + \frac{1}{T}\right).$$

Privacy. The privacy guarantee follows by combining the privacy guarantee in Corollary 2.4.11 and composition of approximate RDP (Lemma 2.4.4), where we adjusted the definition of δ by a factor of k. In particular, we use that the privacy guarantee in each call to Corollary 2.4.11 is a geometric sequence (i.e. $\beta_i^2 T_i^2$ is doubling), and at the end it is

$\frac{1}{2}\beta^2 T^2$. □

2.4.4 Private stochastic proximal estimator

In this section, following the development of [ACJ+21], we give an algorithm which calls Algorithm 7 with several different iteration counts and returns a (random) point $\widehat{x}$ which enjoys a substantially reduced bias for $x^\star_{\overline{x},\lambda}$ defined in (2.26) compared to the expected number of gradient queries.

Algorithm 11: Bias-reduced ReSQued stochastic proximal estimator

Input: $\overline{x} \in \mathbb{R}^d$, ball radius, convolution radius, privacy parameter, and regularization parameter $r, \rho, \beta, \lambda > 0$, dataset $\mathcal{D} \in \mathcal{S}^n$, iteration count $T \in \mathbb{N}$ with $T \le \lfloor \frac{n}{2C_{\mathrm{priv}}} \rfloor$
$T_{\max} \leftarrow \lfloor \frac{n}{C_{\mathrm{priv}}} \rfloor$, $j_{\max} \leftarrow \lfloor \log_2 \frac{T_{\max}}{T} \rfloor$
for $k \in [j_{\max}]$ **do**
  Draw $J \sim \mathrm{Geom}(\frac{1}{2})$
  $x_0 \leftarrow$ output of Algorithm 7 with inputs $(\overline{x}, r, \rho, \beta, \lambda, \mathcal{D}, T)$
  if $J \le j_{\max}$ **then**
    $x_J \leftarrow$ output of Algorithm 7 with inputs $(\overline{x}, r, \rho, 2^{-\frac{J}{2}}\beta, \lambda, \mathcal{D}, 2^J T)$
    $x_{J-1} \leftarrow$ output of Algorithm 7 with inputs $(\overline{x}, r, \rho, 2^{-\frac{J-1}{2}}\beta, \lambda, \mathcal{D}, 2^{J-1} T)$
    $\widehat{x}_k \leftarrow x_0 + 2^J (x_J - x_{J-1})$
  end
  else
    $\widehat{x}_k \leftarrow x_0$
  end
end
Return: $\widehat{x} \leftarrow \frac{1}{j_{\max}} \sum_{k \in [j_{\max}]} \widehat{x}_k$

Proposition 2.4.15. *Let $x^\star_{\overline{x},\lambda}$ be defined as in (2.26). We have, for a universal constant C_{bias}:*

$$\| \mathbb{E}\, \widehat{x} - x^\star_{\overline{x},\lambda} \| \le C_{\mathrm{bias}} \left(\frac{L}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta n} + \frac{1}{\sqrt{n}} \right) \right),$$

and, for a universal constant C_{var},

$$\mathbb{E}\,\|\widehat{x} - x^\star_{\overline{x},\lambda}\|^2 \le \frac{C_{\mathrm{var}} L^2}{\lambda^2}\left(\frac{d}{\beta^2 T^2} + \frac{1}{T}\right).$$

Proof. We begin by analyzing the output $\widehat{x}_k$ of a single loop $k \in [j_{\max}]$. For $J \sim \mathrm{Geom}(\frac{1}{2})$, we have $\Pr[J = j] = 2^{-j}$ if $j \in [j_{\max}]$, and $\Pr[J = j] = 0$ otherwise. We denote x_j to be the output of Algorithm 3 with privacy parameter $2^{-\frac{j}{2}}\beta$ and gradient bound $2^j T$. First,

$$\mathbb{E}\,\widehat{x}_k = \mathbb{E}\,x_0 + \sum_{j\in[j_{\max}]} \Pr[J = j] 2^j (\mathbb{E}\,x_j - \mathbb{E}\,x_{j-1}) = \mathbb{E}\,x_{j_{\max}}.$$

Since $T \cdot 2^{j_{\max}} \ge \frac{T_{\max}}{2} \ge \frac{n}{2C_{\mathrm{priv}}}$, applying Jensen's inequality gives

$$\|\,\mathbb{E}\,x_{j_{\max}} - x^\star_{\overline{x},\lambda}\| \le \sqrt{\mathbb{E}\,\|x_{j_{\max}} - x^\star_{\overline{x},\lambda}\|^2} \le \frac{\sqrt{2C_{\mathrm{sc}}}L}{\lambda}\left(\frac{\sqrt{d}}{\beta n} + \frac{1}{\sqrt{n}}\right),$$

where the last inequality follows from Proposition 2.4.14 and strong convexity of the regularized function to convert the function error bound to a distance bound. This implies the first conclusion, our bias bound. Furthermore, for our variance bound, we have

$$\mathbb{E}\,\|\widehat{x}_k - \mathbb{E}\,\widehat{x}_k\|^2 \le \mathbb{E}\,\|\widehat{x}_k - x^\star_{\overline{x},\lambda}\|^2 \le 2\,\mathbb{E}\,\|\widehat{x}_k - x_0\|^2 + 2\,\mathbb{E}\,\|x_0 - x^\star_{\overline{x},\lambda}\|^2.$$

By Proposition 2.4.14 and strong convexity, $\mathbb{E}\,\|x_0 - x^\star_{\overline{x},\lambda}\|^2 \le \frac{C_{\mathrm{sc}}L^2}{\lambda^2}(\frac{d}{\beta^2T^2} + \frac{1}{T})$. Next,

$$\mathbb{E}\,\|\widehat{x}_k - x_0\|^2 = \sum_{j\in[j_{\max}]} \Pr[J = j] 2^{2j}\,\mathbb{E}\,\|x_j - x_{j-1}\|^2 = \sum_{j\in[j_{\max}]} 2^j\,\mathbb{E}\,\|x_j - x_{j-1}\|^2.$$

Note that

$$\mathbb{E}\,\|x_j - x_{j-1}\|^2 \le 2\,\mathbb{E}\,\|x_j - x^\star_{\overline{x},\lambda}\|^2 + 2\,\mathbb{E}\,\|x_{j-1} - x^\star_{\overline{x},\lambda}\|^2 \le 2^{-j} \cdot \frac{6C_{\mathrm{sc}}L^2}{\lambda^2}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right),$$

and hence combining the above bounds yields

$$\mathbb{E}\,\|\widehat{x}_k - \mathbb{E}\,\widehat{x}_k\|^2 \le \frac{14C_{\mathrm{sc}}j_{\max}L^2}{\lambda^2} \cdot \left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right).$$

Now, averaging $j_{\max}$ independent copies shows that

$$\mathbb{E}\left\|\widehat{x} - x^\star_{\overline{x},\lambda}\right\|^2 = \|\widehat{x} - \mathbb{E}\,\widehat{x}\|^2 + \left\|\mathbb{E}\,\widehat{x} - x^\star_{\overline{x},\lambda}\right\|^2$$
$$\leq \frac{1}{j_{\max}} \cdot \left(\frac{14 C_{\mathrm{sc}} j_{\max} L^2}{\lambda^2} \cdot \left(\frac{d}{\beta^2 T^2} + \frac{1}{T}\right)\right) + C_{\mathrm{bias}}^2 \left(\frac{L}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta n} + \frac{1}{\sqrt{n}}\right)\right)^2,$$

where we used our earlier bias bound. The conclusion follows by letting $C_{\mathrm{var}} = C_{\mathrm{bias}}^2 + 14 C_{\mathrm{sc}}$. □

We conclude with a gradient complexity and privacy bound, depending on the sampled J.

Lemma 2.4.16. *There is a universal constant $C_{\mathrm{priv}} \geq 1$, such that if $\beta^2 \log^2(\frac{\log\log n}{\delta}) \leq \frac{1}{C_{\mathrm{priv}}}$, $\delta \in (0, \frac{1}{2})$, and $\frac{\rho}{r} \geq C_{\mathrm{priv}} \log^2(\frac{\log T}{\delta})$, the following holds. Consider one loop indexed by $k \in [j_{\max}]$, and let J be the result of the $\mathrm{Geom}(\frac{1}{2})$ draw. If $J \in [j_{\max}]$, loop k of Algorithm 11 uses at most $2^{J+1}T$ gradients. Furthermore, the loop satisfies $(\alpha, \alpha\tau, \delta)$-RDP for*

$$\tau := 2^J \cdot C_{\mathrm{priv}} \left(\beta \log\left(\frac{\log\log n}{\delta}\right) \cdot \frac{T}{n}\right)^2, \ \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}} \beta^2 \log^2\left(\frac{\log\log n}{\delta}\right)}\right).$$

If $J \notin [j_{\max}]$, Algorithm 11 uses at most T gradients, and the loop satisfies $(\alpha, \alpha\tau, \delta)$-RDP for

$$\tau := C_{\mathrm{priv}} \left(\beta \log\left(\frac{\log\log n}{\delta}\right) \cdot \frac{T}{n}\right)^2, \ \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}} \beta^2 \log^2\left(\frac{\log\log n}{\delta}\right)}\right).$$

Proof. This is immediate by Proposition 2.4.14, where we applied Lemma 2.4.4 and set $\delta \leftarrow \frac{\delta}{3}$ (taking a union bound over the at most 3 calls to Algorithm 7, adjusting C_{priv} as necessary). □

2.4.5 Private ERM solver

In this section, we give our main result on privately solving ERM in the setting of Problem 2.4.1, which will be used in a reduction framework in Section 2.4.6 to solve the SCO problem as well. Our ERM algorithm is an instantiation of Proposition 2.2.8. We first

develop a line search oracle (see Definition 2.2.5) based on the solver of Section 2.4.3 (Algorithm 7), which succeeds with high probability. To do so, we leverage the following geometric lemma for aggregating independent runs of our solver.

Lemma 2.4.17 (Claim 1, [KLL+23]). *There is an algorithm* Aggregate *which takes as input* $(S, \Delta) \in (\mathbb{R}^d)^k \times \mathbb{R}_{\geq 0}$, *and returns* $z \in \mathbb{R}^d$ *such that* $\|z - y\| \leq \Delta$, *if for some unknown point* $y \in \mathbb{R}^d$ *satisfying at least* $0.51k$ *points* $x \in S$, $\|x - y\| \leq \frac{\Delta}{3}$. *The algorithm runs in time* $O(dk^2)$.

Algorithm 12: High probability ReSQued ERM solver, strongly convex case

Input: $\overline{x} \in \mathbb{R}^d$, ball radius, convolution radius, privacy parameter, regularization parameter, and failure probability $r, \rho, \beta, \lambda, \zeta > 0$, dataset $\mathcal{D} \in \mathcal{S}^n$, iteration count $T \in \mathbb{N}$
$k \leftarrow 20 \log(\frac{1}{\zeta})$
for $i \in [k]$ **do**
  $x_i \leftarrow$ output of Algorithm 7 with inputs $(\overline{x}, r, \rho, \beta, \lambda, \mathcal{D}, T)$
end
Return: $x' \leftarrow \mathsf{Aggregate}(\{x_i\}_{i\in[k]}, \frac{9\sqrt{2C_{\mathrm{sc}}L}}{\lambda}(\frac{d}{\beta^2T^2} + \frac{1}{T})^{\frac{1}{2}})$

Proposition 2.4.18. *Let* $x^\star_{\overline{x},\lambda}$ *be defined as in* (2.26). *Algorithm 6 uses at most* $18T\log(\frac{1}{\zeta})$ *gradients and produces* x' *such that with probability at least* $1 - \zeta$, *for a universal constant* C_{ls},

$$\|x' - x^\star_{\overline{x},\lambda}\| \leq \frac{C_{\mathrm{ls}}L}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta T} + \frac{1}{\sqrt{T}}\right).$$

Moreover, there exists a universal constant $C_{\mathrm{priv}} \geq 1$ *such that* $\frac{T}{n} \leq \frac{1}{C_{\mathrm{priv}}}, \delta \in (0, \frac{1}{6})$ *and* $\frac{\rho}{r} \geq C_{\mathrm{priv}} \log^2(\frac{1}{\delta}\log(\frac{T}{\zeta}))$, *Algorithm 6 satisfies* $(\alpha, \alpha\tau, \delta)$*-RDP for*

$$\tau := C_{\mathrm{priv}} \log\left(\frac{1}{\zeta}\right)\left(\beta \log\left(\frac{1}{\delta}\log\left(\frac{T}{\zeta}\right)\right) \cdot \frac{T}{n}\right)^2, \ \alpha \in \left(1, \frac{1}{C_{\mathrm{priv}}\beta^2 \log^2\left(\frac{1}{\delta}\log\left(\frac{T}{\zeta}\right)\right)}\right).$$

Proof. For each x_i, by Proposition 2.4.14,

$$\mathbb{E}\left[\widehat{f^{\mathrm{erm}}}_r(x_i) + \frac{\lambda}{2}\|x_i - \overline{x}\|^2\right] - \widehat{f^{\mathrm{erm}}}_r(x^\star_{\overline{x},\lambda}) - \frac{\lambda}{2}\|x^\star_{\overline{x},\lambda} - \overline{x}\|^2 \le \frac{C_{\mathrm{sc}}L^2}{\lambda}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right).$$

Further, by strong convexity and Jensen's inequality we have

$$\mathbb{E}[\|x_i - x^\star_{\overline{x},\lambda}\|] \le \frac{\sqrt{2C_{\mathrm{sc}}}L}{\lambda}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right)^{\frac{1}{2}}.$$

Hence, by Markov's inequality, for each $i \in [k]$ we have

$$\Pr\left[\|x_i - x^\star_{\overline{x},\lambda}\| \ge \frac{3\sqrt{2C_{\mathrm{sc}}}L}{\lambda}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right)^{\frac{1}{2}}\right] \le \frac{1}{3}.$$

Hence by a Chernoff bound, with probability $\ge 1 - \zeta$, at least $0.51k$ points $x \in \{x_i\}_{i\in[k]}$ satisfy

$$\|x - x^\star_{\overline{x},\lambda}\| \le \frac{3\sqrt{2C_{\mathrm{sc}}}L}{\lambda}\left(\frac{d}{\beta^2T^2} + \frac{1}{T}\right)^{\frac{1}{2}}.$$

Hence the precondition of Lemma 2.4.17 holds, giving the distance guarantee with high probability. The privacy guarantee follows from Proposition 2.4.14 and the composition of approximate RDP, where we adjusted C_{priv} by a constant and the definition of δ by a factor of k. □

Now we are ready to prove our main result on private ERM.

Theorem 2.4.19 (Private ERM)**.** *In the setting of Problem 2.4.1, let $\varepsilon_{\mathrm{dp}} \in (0,1)$ and $\delta \in (0, \frac{1}{6})$. There is an $(\varepsilon_{\mathrm{dp}}, \delta)$-DP algorithm which takes as input $\mathcal{D}$ and outputs $\widehat{x} \in \mathbb{R}^d$ such that*

$$\mathbb{E}\left[f^{\mathrm{erm}}(\widehat{x}) - \min_{x\in\mathbb{B}(R)} f^{\mathrm{erm}}(x)\right] \le O\left(LR \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}\log^{1.5}(\frac{n}{\delta})\log n}{n\varepsilon_{\mathrm{dp}}}\right)\right).$$

Moreover, with probability at least $1 - \delta$, the algorithm queries at most

$$O\left(\log^6\left(\frac{n}{\delta}\right)\left(\min\left(n, \frac{n^2\varepsilon_{\mathrm{dp}}^2}{d}\right) + \min\left(\frac{(nd)^{\frac{2}{3}}}{\varepsilon_{\mathrm{dp}}}, n^{\frac{4}{3}}\varepsilon_{\mathrm{dp}}^{\frac{1}{3}}\right)\right)\right) \text{ gradients.}$$

Proof. Throughout this proof, set for a sufficiently large constant C,

$$\begin{aligned}\varepsilon_{\mathrm{opt}} &:= CLR\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log\frac{1}{\delta}}\log^{1.5}(\frac{n}{\delta})\log n}{n\varepsilon_{\mathrm{dp}}}\right),\ \kappa := \frac{LR}{\varepsilon_{\mathrm{opt}}},\\ \rho &:= \frac{\varepsilon_{\mathrm{opt}}}{L\sqrt{d}},\ r := \frac{\rho}{\sqrt{C}\log^2(\frac{n}{\delta})},\ \alpha := \frac{4\log\frac{2}{\delta}}{\varepsilon_{\mathrm{dp}}},\ \beta := \frac{\varepsilon_{\mathrm{dp}}}{C\log(\frac{n}{\delta})\sqrt{\log\frac{1}{\delta}}}.\end{aligned} \tag{2.29}$$

Note that for the given parameter settings, for sufficiently large C, we have

$$\kappa \le \frac{1}{C}\min\left(\sqrt{n}, \frac{n\varepsilon_{\mathrm{dp}}}{\sqrt{d\log\frac{1}{\delta}}\log^{1.5}(\frac{n}{\delta})\log n}\right),\ \frac{R}{r} \le n\log^2\left(\frac{\log n}{\delta}\right). \tag{2.30}$$

Our algorithm proceeds as follows:

We apply Proposition 2.2.8 with $x^\star \leftarrow \arg\min_{x\in\mathbb{B}(R)} f^{\mathrm{erm}}(x)$ and $F \leftarrow \widehat{f_\rho^{\mathrm{erm}}}$, and instantiate the necessary oracles as follows for $C_{\mathrm{ba}}K\log\kappa$ iterations.

1. We use Algorithm 6 with r, ρ, β defined in (2.29), and

$$T_1 := \sqrt{C}\left(\frac{\kappa\sqrt{d}}{\sqrt{K}\beta\log^2\kappa} + \frac{\kappa^2}{K\log^3\kappa\log\frac{n}{\delta}}\right),\ \zeta := \frac{1}{\kappa C_{\mathrm{ba}}K\log\kappa}, \tag{2.31}$$

   as a $(\frac{r}{C_{\mathrm{ba}}}, \lambda)$-line search oracle $\mathcal{O}_{\mathrm{ls}}$.

2. We use Algorithm 7 with r, ρ, β defined in (2.29), and

$$T_2 := \sqrt{C}\left(\frac{\kappa\sqrt{d}}{\sqrt{K}\beta\sqrt{\log\kappa}} + \frac{\kappa^2}{K\log\kappa}\right), \tag{2.32}$$

   as a $(\frac{\lambda r^2}{C_{\mathrm{ba}}\log^3\kappa}, \lambda)$-ball optimization oracle $\mathcal{O}_{\mathrm{bo}}$.

3. We use Algorithm 11 with r, ρ, β defined in (2.29), and

$$T_3 := \sqrt{C}\left(\frac{\kappa\sqrt{d}}{\sqrt{K}\beta} + \frac{\kappa^2}{K}\right) \tag{2.33}$$

   as a $(\frac{\varepsilon_{\mathrm{opt}}}{C_{\mathrm{ba}}R}, \frac{\varepsilon_{\mathrm{opt}}\sqrt{K}}{C_{\mathrm{ba}}R}, \lambda)$-stochastic proximal oracle $\mathcal{O}_{\mathrm{sp}}$.

We split the remainder of the proof into four parts. We first show that the oracle definitions are indeed met. We then bound the overall optimization error against f^{erm}. Finally, we discuss the privacy guarantee and the gradient complexity bound.

Oracle correctness. For the line search oracle, by Proposition 2.4.18, it suffices to show

$$\frac{C_{\mathrm{ls}}L}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta T_1} + \frac{1}{\sqrt{T_1}} \right) \leq \frac{r}{C_{\mathrm{ba}}}.$$

This is satisfied for T_1 in (2.31), since Proposition 2.2.8 guarantees $\lambda \geq \frac{\varepsilon_{\mathrm{opt}} K^2 \log^2 \kappa}{R^2 C_{\mathrm{ba}}}$. Hence,

$$\begin{aligned}
\frac{C_{\mathrm{ls}}L}{\lambda} \cdot \frac{\sqrt{d}}{\beta T_1} \cdot \frac{C_{\mathrm{ba}}}{r} &\leq C_{\mathrm{ls}} C_{\mathrm{ba}}^2 \cdot \frac{\kappa \sqrt{d}}{\beta \log^2 \kappa} \cdot \frac{1}{\sqrt{K}} \cdot \frac{1}{T_1} \leq \frac{1}{2}, \\
\frac{C_{\mathrm{ls}}L}{\lambda} \cdot \frac{1}{\sqrt{T_1}} \cdot \frac{C_{\mathrm{ba}}}{r} &\leq C_{\mathrm{ls}} C_{\mathrm{ba}}^2 \cdot \frac{\kappa}{\log^2 \kappa} \cdot \frac{1}{\sqrt{K}} \cdot \frac{1}{\sqrt{T_1}} \leq \frac{1}{2},
\end{aligned}$$

for a sufficiently large C, where we used $K^{1.5} = \frac{R}{r}$ to simplify. By a union bound, the above holds with probability at least $1 - \frac{\varepsilon_{\mathrm{opt}}}{LR}$ over all calls to Algorithm 6, since there are at most $C_{\mathrm{ba}} K \log \kappa$ iterations. For the remainder of the proof, let $\mathcal{E}_{\mathrm{ls}}$ be the event that all line search oracles succeed. For the ball optimization oracle, by Proposition 2.4.14, it suffices to show

$$\frac{C_{\mathrm{sc}}L^2}{\lambda} \left(\frac{d}{\beta^2 T_2^2} + \frac{1}{T_2} \right) \leq \frac{\lambda r^2}{C_{\mathrm{ba}} \log^3 \kappa}.$$

This is satisfied for our choice of T_2 in (2.32), again with $\lambda \geq \frac{\varepsilon_{\mathrm{opt}} K^2 \log^2 \kappa}{R^2 C_{\mathrm{ba}}}$. Hence,

$$\begin{aligned}
\frac{C_{\mathrm{sc}}L^2}{\lambda} \cdot \frac{d}{\beta^2 T_2^2} \cdot \frac{C_{\mathrm{ba}} \log^3 \kappa}{\lambda r^2} &\leq C_{\mathrm{sc}} C_{\mathrm{ba}}^3 \cdot \frac{\kappa^2 d}{\beta^2 \log \kappa} \cdot \frac{1}{K} \cdot \frac{1}{T_2^2} \leq \frac{1}{2}, \\
\frac{C_{\mathrm{sc}}L^2}{\lambda} \cdot \frac{1}{T_2} \cdot \frac{C_{\mathrm{ba}} \log^3 \kappa}{\lambda r^2} &\leq C_{\mathrm{sc}} C_{\mathrm{ba}}^3 \cdot \frac{\kappa^2}{\log \kappa} \cdot \frac{1}{K} \cdot \frac{1}{T_2} \leq \frac{1}{2},
\end{aligned}$$

again for large C. Finally, for the proximal gradient oracle, by Proposition 2.4.15, it suffices to show

$$C_{\mathrm{bias}} \left(\frac{L}{\lambda} \cdot \left(\frac{\sqrt{d}}{\beta n} + \frac{1}{\sqrt{n}} \right) \right) \leq \frac{\varepsilon_{\mathrm{opt}}}{C_{\mathrm{ba}} \lambda R},$$

$$\frac{C_{\text{var}}L^2}{\lambda^2}\left(\frac{d}{\beta^2 T_3^2}+\frac{1}{T_3}\right)\le\frac{\varepsilon_{\text{opt}}^2 K}{C_{\text{ba}}^2\lambda^2 R^2}.$$

The first inequality is clear. The second is satisfied for our choice of T_3 in (2.33), which implies

$$\frac{C_{\text{var}}L^2}{\lambda^2}\cdot\frac{d}{\beta^2 T_3^2}\cdot\frac{C_{\text{ba}}^2\lambda^2R^2}{\varepsilon_{\text{opt}}^2K}=C_{\text{var}}C_{\text{ba}}^2\cdot\frac{\kappa^2 d}{\beta^2}\cdot\frac{1}{K}\cdot\frac{1}{T_3^2}\le\frac{1}{2},$$
$$\frac{C_{\text{var}}L^2}{\lambda^2}\cdot\frac{1}{T_3}\cdot\frac{C_{\text{ba}}^2\lambda^2R^2}{\varepsilon_{\text{opt}}^2K}=C_{\text{var}}C_{\text{ba}}^2\cdot\kappa^2\cdot\frac{1}{K}\cdot\frac{1}{T_3}\le\frac{1}{2}.$$

Optimization error. By Proposition 2.2.8, the expected optimization error against $\widehat{f_\rho^{\text{erm}}}$ is bounded by ε_{opt} whenever $\mathcal{E}_{\text{ls}}$ occurs. Otherwise, the optimization error is never larger than LR as long as we return a point in $\mathbb{B}(R)$, since the function is L-Lipschitz. Further, we showed $\Pr[\mathcal{E}_{\text{ls}}]\ge 1-\frac{\varepsilon_{\text{opt}}}{LR}$, so the total expected error is bounded by $2\varepsilon_{\text{opt}}$. Finally, the additive error between $\widehat{f_\rho^{\text{erm}}}$ and f^{erm} is bounded by $\rho L\sqrt{d}=\varepsilon_{\text{opt}}$. The conclusion follows by setting the error bound to $3\varepsilon_{\text{opt}}$.

Privacy. We first claim that each call to $\mathcal{O}_{\text{ls}}$, and $\mathcal{O}_{\text{bo}}$ used by Proposition 2.2.8 satisfies

$$\left(\alpha,\frac{\varepsilon_{\text{dp}}}{6C_{\text{ba}}K\log\kappa},\frac{\delta}{18C_{\text{ba}}K\log\kappa}\right)\text{-RDP}.$$

We first analyze $\mathcal{O}_{\text{ls}}$:
The preconditions of Proposition 2.4.18 are met, where $\log(\frac{18C_{\text{ba}}K\log\kappa}{\delta}\log(\frac{T}{\zeta}))\le 2\log\frac{n}{\delta}$ for our parameter settings. Moreover, our α is in the acceptable range. Finally, by Proposition 2.4.18 it suffices to note

$$\frac{8\alpha C_{\text{priv}}\beta^2T_1^2\log^3\left(\frac{n}{\delta}\right)}{n^2}\le\frac{128CC_{\text{priv}}\beta^2\log^3(\frac{n}{\delta})\log\frac{1}{\delta}}{n^2\varepsilon_{\text{dp}}}\cdot\left(\frac{\kappa^2 d}{K\beta^2\log\kappa}+\frac{\kappa^4}{K^2\log^2\kappa}\right)\le\frac{\varepsilon_{\text{dp}}}{6C_{\text{ba}}K\log\kappa},$$

where the second inequality follows for sufficiently large C due to (2.30). Next, we analyze the privacy of $\mathcal{O}_{\text{bo}}$. The preconditions of Proposition 2.4.14 are met, where $\log(\frac{\log\log T}{\delta})\le\log\frac{n}{\delta}$ for our parameter settings, and our α is again acceptable. Finally, by Proposi-

tion 2.4.14 it suffices to note

$$\frac{\alpha C_{\mathrm{priv}}\beta^2 T_2^2 \log^2(\frac{n}{\delta})}{n^2} \le \frac{16CC_{\mathrm{priv}}\beta^2 \log^2(\frac{n}{\delta})\log\frac{1}{\delta}}{n^2\varepsilon_{\mathrm{dp}}} \cdot \left(\frac{\kappa^2 d}{K\beta^2\log\kappa} + \frac{\kappa^4}{K^2\log^2\kappa}\right) \le \frac{\varepsilon_{\mathrm{dp}}}{6C_{\mathrm{ba}}K\log\kappa},$$

again for sufficiently large C from (2.30). Hence, by applying Lemma 2.4.4, all of the at most $C_{\mathrm{ba}}K\log\kappa$ calls to $\mathcal{O}_{\mathrm{ls}}$ and $\mathcal{O}_{\mathrm{bo}}$ used by the algorithm combined satisfy

$$\left(\alpha, \frac{\varepsilon_{\mathrm{dp}}}{3}, \frac{\delta}{9}\right)\text{-RDP}.$$

Finally, we analyze the privacy of $\mathcal{O}_{\mathrm{sp}}$. Let

$$j_{\max} := \left\lfloor \log_2\left(\frac{1}{T_3}\cdot\left\lfloor\frac{n}{C_{\mathrm{priv}}}\right\rfloor\right)\right\rfloor$$

be the truncation parameter in Algorithm 11. The total number of draws from $\mathrm{Geom}(\frac{1}{2})$ in Algorithm 11 over the course of the algorithm is $C_{\mathrm{ba}}K\log\kappa\cdot j_{\max}$. It is straightforward to check that the expected number of draws where $J = j$ for all $j \in [j_{\max}]$ is

$$2^{-j_{\max}}C_{\mathrm{ba}}\kappa\log\kappa\cdot j_{\max} = \Omega\left(\frac{T_3}{n}\cdot K\log\kappa\cdot j_{\max}\right),$$

which is superconstant. By Chernoff and a union bound, with probability $\ge 1-\frac{\delta}{n}$, there is a constant C' such that for all $j \in [j_{\max}]$, the number of times we draw $J = j$ is bounded by

$$2^{-j}C'K\log\kappa\log\frac{n}{\delta}.$$

Similarly, the number of times we draw $J \notin [j_{\max}]$ is bounded by $C'K\log\kappa\log\frac{n}{\delta}$. This implies by Lemma 2.4.4 that all calls to $\mathcal{O}_{\mathrm{sp}}$ used by the algorithm combined satisfy

$$\left(\alpha, \frac{\varepsilon_{\mathrm{dp}}}{6}, \frac{\delta}{18}\right)\text{-RDP}.$$

Here, we summed the privacy loss in Lemma 2.4.16 over $0 \le J \le j_{\max}$, which gives

$$\sum_{0\le j\le j_{\max}}\left(2^j\cdot\frac{\alpha C_{\mathrm{priv}}\beta^2\log^2(\frac{n}{\delta})T_3^2}{n^2}\right)\cdot\left(2^{-j}C'K\log\kappa\log\frac{n}{\delta}\right)$$

$$\leq (j_{\max}+1)\cdot\frac{16CC'C_{\text{priv}}K\beta^2\log^3(\frac{n}{\delta})\log\frac{1}{\delta}\log\kappa}{n^2\varepsilon_{\text{dp}}}\cdot\left(\frac{\kappa^2 d}{K\beta^2}+\frac{\kappa^4}{K^2}\right)\leq\frac{\varepsilon_{\text{dp}}}{6},$$

for sufficiently large C, where we use $\log\kappa, j_{\max}\leq\log n$, and $K\geq\log\frac{1}{\delta}$ for our parameter settings. Finally, combining these bounds shows that our whole algorithm satisfies $(\alpha,\frac{\varepsilon_{\text{dp}}}{2},\frac{\delta}{6})$-RDP, and applying Corollary 2.4.3, gives the desired privacy guarantee.

Gradient complexity. We have argued that with probability at least $1-\delta$, the number of times we encounter the $J=j$ case of Lemma 2.4.16 for all $0\leq j\leq j_{\max}$ is bounded by $2^{-j}C'K\log\kappa\log\frac{n}{\delta}$. Under this event, Proposition 2.4.18, Proposition 2.4.14, and Lemma 2.4.16 imply the total gradient complexity of our algorithm is at most

$$C_{\text{ba}}K\log\kappa\cdot\left(18T_1\log\frac{1}{\zeta}+T_2+\sum_{0\leq j\leq j_{\max}}\left(2^{-j}C'\log\frac{n}{\delta}\right)\left(2^{j+1}T_3\right)\right)$$
$$\leq 36C_{\text{ba}}C'K\log n\left(T_1\log n+T_2+T_3\log n\log\frac{n}{\delta}\right),$$

where we use $\zeta\geq n^{-2}$, $j_{\max}\leq\log n$, and $\kappa\leq n$. The conclusion follows from plugging in our parameter choices from (2.31), (2.32), and (2.33). □

Finally, we note that following the strategy of Section 2.4.3, it is straightforward to extend Theorem 2.4.19 to the strongly convex setting. We state this result as follows.

Corollary 2.4.20 (Private regularized ERM)**.** *In the setting of Problem 2.4.1, let* $\varepsilon_{\text{dp}}\in(0,1)$, $\delta\in(0,\frac{1}{6})$, $\lambda\geq 0$, *and* $x'\in\mathbb{B}(R)$. *There is an* $(\varepsilon_{\text{dp}},\delta)$*-DP algorithm which outputs* $\widehat{x}\in\mathbb{B}(R)$ *such that*

$$\mathbb{E}\left[f^{\text{erm}}(\widehat{x})+\frac{\lambda}{2}\left\|x-x'\right\|^2-\min_{x\in\mathbb{B}(R)}\left\{f^{\text{erm}}(x)+\frac{\lambda}{2}\left\|x-x'\right\|^2\right\}\right]\leq O\left(\frac{L^2}{\lambda}\cdot\left(\frac{1}{n}+\frac{d\log\frac{1}{\delta}\log^3(\frac{n}{\delta})\log^2 n}{n^2\varepsilon_{\text{dp}}^2}\right)\right).$$

Moreover, with probability at least $1-\delta$, *the algorithm queries at most*

$$O\left(\log^6\left(\frac{n}{\delta}\right)\left(\min\left(n,\frac{n^2\varepsilon_{\text{dp}}^2}{d}\right)+\min\left(\frac{(nd)^{\frac{2}{3}}}{\varepsilon_{\text{dp}}},n^{\frac{4}{3}}\varepsilon_{\text{dp}}^{\frac{1}{3}}\right)\right)\right)\ \textit{gradients.}$$

Proof. We first note that similar to Corollary 2.4.11 (an extension of Proposition 2.4.5), it is straightforward to extend Theorem 2.4.19 to handle both regularization and an improved

upper bound on the distance to the optimum, with the same error rate and privacy guarantees otherwise. The handling of the improved upper bound on the distance follows because the convergence rate of the [ACJ+21] algorithm scales proportionally to the distance to the optimum, when it is smaller than R. The regularization is handled in the same way as Corollary 2.4.11, where regularization can only improve the contraction in the privacy proof. One subtle point is that for the regularized problems, we need to obtain starting points for Algorithm 7 when the constraint set is $\mathbb{B}_{\overline{x}}(r)$, but the regularization in the objective is centered around a point not in $\mathbb{B}_{\overline{x}}(r)$ (in our case, the centerpoint will be a weighted combination of $\overline{x}$ and x'). However, by initializing Algorithm 7 at the projection of the regularization centerpoint, the initial function error guarantee in Lemma 2.4.12 still holds (see Lemma 2.4.13).

The reduction from the claimed rate in this corollary statement to the regularized extension of Theorem 2.4.19 then proceeds identically to the proof of Proposition 2.4.14, which calls Corollary 2.4.11 repeatedly. □

2.4.6 Private SCO solver

Finally, we give our main result on private SCO in this section. To obtain it, we will combine Corollary 2.4.20 with a generic reduction in [FKT20, KLL21], which uses a private ERM solver as a black box. The reduction is based on the iterative localization technique proposed by [FKT20] (which is the same strategy used by Section 2.4.3), and derived in greater generality by [KLL21].

Proposition 2.4.21 (Modification of Theorem 5.1 in [KLL21])**.** *Suppose there is an* $(\varepsilon_{\text{dp}}, \delta)$*-DP algorithm* $\mathcal{A}_{\text{erm}}$ *with expected excess loss*

$$O\left(\frac{L^2}{\lambda} \cdot \left(\frac{1}{n} + \frac{d \log \frac{1}{\delta} \log^3(\frac{n}{\delta}) \log^2 n}{n^2 \varepsilon_{\text{dp}}^2}\right)\right),$$

using $N(n, \varepsilon_{\text{dp}}, \delta)$ *gradient queries, for some function* N*, when applied to an* L*-Lipschitz empirical risk (with* n *samples, constrained to* $\mathbb{B}(R) \subset \mathbb{R}^d$*) plus a* λ*-strongly convex regularizer. Then there is an* $(\varepsilon_{\text{dp}}, \delta)$*-DP algorithm* $\mathcal{A}_{\text{sco}}$ *using* $\sum_{i \in \lceil \log n \rceil} N(\frac{n}{2^i}, \frac{\varepsilon_{\text{dp}}}{2^i}, \frac{\delta}{2^i})$ *gradient*

queries, with expected excess population loss

$$O\left(LR \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d \log \frac{1}{\delta}} \log^{1.5}(\frac{n}{\delta}) \log n}{n\varepsilon_{\text{dp}}}\right)\right).$$

Theorem 5.1 in [KLL21] assumes a slightly smaller risk guarantee for $\mathcal{A}_{\text{erm}}$ (removing the extraneous $\log^3(\frac{n}{\delta}) \log^2 n$ factor), but it is straightforward to see that the proof extends to handle our larger risk assumption. Combining Proposition 2.4.21 and Corollary 2.4.20 then gives our main result.

Theorem 2.4.22 (Private SCO)**.** *In the setting of Problem 2.4.1, let $\varepsilon_{\text{dp}} \in (0, 1)$ and $\delta \in (0, \frac{1}{6})$. There is an $(\varepsilon_{\text{dp}}, \delta)$-DP algorithm which takes as input $\mathcal{D}$ and outputs $\widehat{x} \in \mathbb{R}^d$ such that*

$$\mathbb{E}\left[F_{\mathcal{P}}(\widehat{x}) - \min_{x \in \mathbb{B}(R)} F_{\mathcal{P}}(x)\right] \leq O\left(LR \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d \log \frac{1}{\delta}} \log^{1.5}(\frac{n}{\delta}) \log n}{n\varepsilon_{\text{dp}}}\right)\right).$$

Moreover, with probability at least $1 - \delta$, the algorithm queries at most

$$O\left(\log^6\left(\frac{n}{\delta}\right)\left(\min\left(n, \frac{n^2\varepsilon_{\text{dp}}^2}{d}\right) + \min\left(\frac{(nd)^{\frac{2}{3}}}{\varepsilon_{\text{dp}}}, n^{\frac{4}{3}}\varepsilon_{\text{dp}}^{\frac{1}{3}}\right)\right)\right) \text{ gradients.}$$

2.5 Helper facts

Fact 2.5.1. *Let $p \in \mathbb{N}$. For any integer r such that $0 \leq r \leq p - 1$, $\sum_{0 \leq q \leq p} (-1)^q \binom{p}{q} q^r = 0$.*

Proof. We recognize the formula as a scaling of the Stirling number of the second kind with r objects and p bins, i.e. the number of ways to put r objects into p bins such that each bin has at least one object. When $r < p$ there are clearly no such ways. □

Fact 2.5.2. *Let $p \in \mathbb{N}$ be even and $p \geq 2$. Let $\|x\|, \|y\| \leq \frac{1}{p}$. Then*

$$\sum_{0 \leq q \leq p} (-1)^q \binom{p}{q} \exp\left(\frac{1}{2}\left(((p-q)^2 - (p-q))\|x\|^2 + (q^2 - q)\|y\|^2 + 2q(p-q)\langle x, y\rangle\right)\right) \leq (12p\|x - y\|)^p.$$

Proof. Fix some x. Let $f_x(y)$ be the left-hand side displayed above, and let

$$f_x^q(y) := \exp\left(\frac{1}{2}\left(\left((p-q)^2-(p-q)\right)\|x\|^2 + (q^2-q)\|y\|^2 + 2q(p-q)\langle x, y\rangle\right)\right).$$

We will perform a p^{th} order Taylor expansion of f_x around x, where we show that partial derivatives of order at most $p-1$ are all zero at x, and we bound the largest order derivative tensor.

Derivatives of f_x^q. Fix some $0 \le q \le p$, and define

$$C_q := q^2 - q,\ F_q := f_x^q(y),\ v_q := (q^2-q)y + q(p-q)x. \tag{2.34}$$

Note that for fixed q, F_q and v_q are functions of y, and we defined them such that $\nabla_y v_q = C_q \mathbf{I}_d$, $\nabla_y F_q = v_q F_q$. Next, in the following we use $\sum_{\text{sym}}$ to mean a symmetric sum over all choices of tensor modes, e.g. $\sum_{\text{sym}} v_q^{\otimes 2} \otimes \mathbf{I}_d$ means we will choose 2 of the 4 modes where the action is $v_q^{\otimes 2}$. To gain some intuition for the derivatives of F_q, we begin by evaluating the first few via product rule:

$$\begin{aligned}
\nabla f_x^q(y) &= F_q v_q, \\
\nabla^2 f_x^q(y) &= F_q v_q^{\otimes 2} + C_q F_q \mathbf{I}_d, \\
\nabla^3 f_x^q(y) &= F_q v_q^{\otimes 3} + C_q F_q \sum_{\text{sym}} v_q \otimes \mathbf{I}_d, \\
\nabla^4 f_x^q(y) &= F_q v_q^{\otimes 4} + C_q F_q \sum_{\text{sym}} v_q^{\otimes 2} \otimes \mathbf{I}_d + 3C_q^2 F_q \mathbf{I}_d \otimes \mathbf{I}_d.
\end{aligned}$$

For any fixed $0 \le r \le p$, we claim that the r^{th} derivative tensor has the form

$$\nabla^r f_x^q(y) = F_q \left(\sum_{0 \le s \le \lfloor \frac{r}{2} \rfloor} \frac{N_{r,s}}{\binom{r}{2s}} \left((C_q)^s \sum_{\text{sym}} v_q^{\otimes (r-2s)} \otimes \mathbf{I}_d^{\otimes s} \right) \right), \tag{2.35}$$

where the $N_{r,s}$ are nonnegative coefficients which importantly do not depend on q. To see this we proceed by induction; the base cases are computed above. Every time we take the derivative of a "monomial" term of the form $F_q (C_q)^s v_q^{\otimes (r-2s)} \otimes \mathbf{I}_d^{\otimes s}$ via product rule, we

will have one term in which F_q becomes $v_q F_q$ (and hence we obtain a $F_q C_q^s v_q^{\otimes(r+1-2s)} \otimes \mathbf{I}_d^{\otimes s}$ monomial), and $r-2s$ many terms where a v_q becomes $C_q \mathbf{I}_d$ (and hence we obtain a $F_q C_q^{s+1} v_q^{\otimes(r-1-2s)} \otimes \mathbf{I}_d^{\otimes(s+1)}$ monomial). For fixed $0 \le s \le \lfloor \frac{r+1}{2} \rfloor$, we hence again see that $N_{r+1,s}$ has no dependence on q.

Next, note $\sum_{0\le s\le \lfloor \frac{r}{2} \rfloor} N_{r,s}$ has a natural interpretation as the total number of "monomial" terms of the form $F_q (C_q)^s v_q^{\otimes(r-2s)} \otimes \mathbf{I}_d^{\otimes s}$ when expanding $\nabla^r f_x^q(y)$. We claim that for all $0 \le q \le p$ and $0 \le r \le p-1$,

$$\frac{\sum_{0\le s\le \lfloor \frac{r+1}{2} \rfloor} N_{r+1,s}}{\sum_{0\le s\le \lfloor \frac{r}{2} \rfloor} N_{r,s}} \le p. \tag{2.36}$$

To see this, consider taking an additional derivative of (2.35) with respect to y. Each monomial of the form $F_q (C_q)^s v_q^{\otimes(r-2s)} \otimes \mathbf{I}_d^{\otimes s}$ contributes at most $r-2s+1 \le p$ monomials to the next derivative tensor via product rule, namely one from F_q and one from each copy of v_q. Averaging this bound over all monomials yields the claim (2.36), since each contributes at most p.

Taylor expansion at x. Next, we claim that for all $0 \le r \le p-1$,

$$\nabla^r f_x(x) = 0. \tag{2.37}$$

To see this, we have that $((p-q)^2-(p-q))+(q^2-q)+2q(p-q) = p^2-p$ is independent of q, and hence all of the F_q are equal to some value F when $y = x$. Furthermore, when $y = x$ we have that $v_q = q(p-1)x$. Now, from the characterization (2.35) and summing over all q, any monomial of the form $x^{\otimes(r-2s)} \otimes \mathbf{I}_d^{\otimes s}$ has a total coefficient of

$$F N_{r,s} \sum_{0\le q\le p} (-1)^q \binom{p}{q} (C_q)^s (q(p-1))^{r-2s} = F N_{r,s} (p-1)^{r-2s} \sum_{0\le q\le p} (-1)^q \binom{p}{q} C_q^s q^{r-2s}.$$

Since C_q is a quadratic in q, each summand $(C_q)^s q^{r-2s}$ is a polynomial of degree at most $r \le p-1$ in q, so applying Fact 2.5.1 to each monomial yields the claim (2.37).

Taylor expansion at y. Finally, we will bound the injective tensor norm of $\nabla^p f_x(y)$,

where the injective tensor norm of a degree-p symmetric tensor $\mathbf{T}$ is the maximum value of $\mathbf{T}[v^{\otimes p}]$ over unit norm v. We proceed by bounding the injective tensor norm of each monomial and then summing.

First, for any $0 \le p \le q$, under our parameter settings it is straightforward to see $\|v_q\| \le p$ and $F_q \le 2$. Also, for any $0 \le s \le \frac{p}{2}$ we have $C_q^s \le p^{2s}$, and by repeatedly applying (2.36), we have $\sum_{0 \le s \le \lfloor \frac{p}{2} \rfloor} N_{p,s} \le p^p$. In other words, each of the monomials of the form $F_q (C_q)^s v_q^{\otimes(r-2s)} \otimes \mathbf{I}_d^{\otimes s}$ has injective tensor norm at most $2p^p$ (since each C_q contributes two powers of p, and each v_q contributes one power of p), and there are at most p^p such monomials. Hence, by triangle inequality over the sum of all monomials,

$$\left|\nabla^p f_x^q(y)[(y-x)^{\otimes p}]\right| \le 2p^{2p} \|y-x\|^p .$$

By summing the above over all q (reweighting by $(-1)^q \binom{p}{q}$), and using that the unsigned coefficients sum to $\sum_{0 \le q \le p} \binom{q}{p} = 2^p$, we have

$$\left|\nabla^p f_x(y)[(y-x)^{\otimes p}]\right| \le 4^p p^{2p} \|x-y\|^p .$$

The conclusion follows by a Taylor expansion from x to y of order p, and using $p^p \le 3^p p!$. □

Proof of Lemma 2.2.4. For the first claim,

$$\begin{aligned}\int \frac{(\gamma_\rho(x-\overline{x}-\xi))^p}{(\gamma_\rho(\xi))^{p-1}} \mathrm{d}\xi &= (2\pi\rho)^{-\frac{d}{2}} \int \exp\left(-\frac{1}{2\rho^2}\left(p\|x-\overline{x}\|^2 - 2p\langle x-\overline{x}, \xi\rangle + \|\xi\|^2\right)\right) \mathrm{d}\xi \\ &= \exp\left(\frac{p^2-p}{2\rho^2}\|x-\overline{x}\|^2\right) \le 2,\end{aligned}$$

where the second equality used the calculation in (2.5), and the inequality used the assumed bound on $\|x-\overline{x}\|$. We move onto the second claim. First, we prove the statement for all even $p \in \mathbb{N}$. Denote $v := x-\overline{x}$ and $v' := x'-\overline{x}$ for simplicity. Explicitly expanding the numerator yields that

$$(2\pi\rho)^{\frac{d}{2}} \int \frac{(\gamma_\rho(v-\xi) - \gamma_\rho(v'-\xi))^p}{(\gamma_\rho(\xi))^{p-1}} \mathrm{d}\xi = \sum_{0 \le q \le p} (-1)^q \binom{p}{q} S_q$$

where we define

$$\begin{aligned} S_q &:= (2\pi\rho)^{\frac{d}{2}} \int \frac{(\gamma_\rho(v-\xi))^{p-q}(\gamma_r(v'-\xi))^q}{(\gamma_\rho(\xi))^{p-1}} \mathrm{d}\xi \\ &= \int \exp\left(-\frac{1}{2\rho^2}\left((p-q)\|v\|^2 + q\left\|v'\right\|^2 - 2(p-q)\langle v,\xi\rangle - 2q\langle v',\xi\rangle + \|\xi\|^2\right)\right)\mathrm{d}\xi \\ &= (2\pi\rho)^{\frac{d}{2}} \exp\left(\frac{1}{2\rho^2}\left(\left((p-q)^2-(p-q)\right)\|v\|^2 + (q^2-q)\left\|v'\right\|^2 + 2q(p-q)\langle v,v'\rangle\right)\right). \end{aligned}$$

In the last line, we again used (2.5) to compute the integral. When $p \geq 2$ and is even, a strengthening of the conclusion then follows from Fact 2.5.2 (where we overload $x \leftarrow \frac{v}{\rho}$, $y \leftarrow \frac{v'}{\rho}$ in its application). In particular, this shows the desired claim where the base of the exponent is $\frac{12p}{\rho}\|x-x'\|$ instead of $\frac{24p}{\rho}\|x-x'\|$. We move to general $p \geq 2$. Define the random variable

$$Z := \left|\frac{\gamma_\rho(x-\overline{x}-\xi) - \gamma_\rho(x'-\overline{x}-\xi)}{\gamma_\rho(\xi)}\right|.$$

Recall that we have shown for all even $p \geq 2$,

$$\mathbb{E}\, Z^p \leq \left(\frac{12p\|x-x'\|}{\rho}\right)^p.$$

Now, let $p \geq 2$ be sandwiched between the even integers q and $q+2$. Hölder's inequality and the above inequality (for $p \leftarrow q$ and $p \leftarrow q+2$) demonstrate

$$\mathbb{E}\, Z^p \leq \left(\mathbb{E}\, Z^q\right)^{\frac{q+2-p}{2}} \left(\mathbb{E}\, Z^{q+2}\right)^{\frac{p-q}{2}} \leq \left(\frac{12(q+2)\|x-x'\|}{\rho}\right)^p,$$

where we use $q(q+2-p)+(q+2)(p-q) = 2p$. The conclusion follows since $q+2 \leq 2p$. □

Fact 2.5.3. *Let Z be a nonnegative scalar random variable, let $C \geq 0$ be a fixed scalar, and let $p \in \mathbb{N}$ and $p \geq 2$. Then*

$$\left(\mathbb{E}\left[(Z+C)^p\right]\right)^{\frac{1}{p}} \leq \mathbb{E}\left[Z^p\right]^{\frac{1}{p}} + C.$$

Proof. Denote $A := \mathbb{E}\left[Z^p\right]^{\frac{1}{p}}$. Taking p^{th} powers of both sides, we have the conclusion if

$$(A+C)^p - \mathbb{E}\left[(Z+C)^p\right] \geq 0 \iff \sum_{q \in [p-1]} \binom{p}{q} C^{p-q}\left(A^q - \mathbb{E}\left[Z^q\right]\right) \geq 0.$$

Here we use that the $q = 0$ and $q = p$ terms cancel. We conclude since Jensen's inequality yields

$$\mathbb{E}[Z^p] \geq \mathbb{E}[Z^q]^{\frac{p}{q}} \implies A^q \geq \mathbb{E}[Z^q], \text{ for all } q \in [p-1].$$

□

2.6 *Discussion of Proposition 2.2.8*

In this section, we discuss how to obtain Proposition 2.2.8 from the analysis in [ACJ+21]. We separate the discussion into four parts, corresponding to the iteration count, the line search oracle parameters, the ball optimization oracle parameters, and the proximal gradient oracle parameters. We note that Proposition 2 in [ACJ+21] states that they obtain function error ε_{opt} with constant probability; however, examining the proof shows it actually yields an expected error bound. Additionally, Proposition 2 in [ACJ+21] is stated for $x^\star$ (the comparison point in the error guarantee) defined to be the minimizer of F, but examining the proof shows that the only property about $x^\star$ it uses is that $x^\star \in \mathbb{B}(R)$.

Iteration count. The bound $C_{\text{ba}} K \log \kappa$ on the number of iterations follows immediately from the value $K_{\max}$ stated in Proposition 2 of [ACJ+21], where we set $\lambda_{\min} \leftarrow \lambda_\star$ and $\varepsilon \leftarrow \varepsilon_{\text{opt}}$.

Line search oracle parameters. The line search oracle is called in the implementation of Line 2 of Algorithm 4 in [ACJ+21]. Our implementation follows the development of Appendix D.2.3 in [ACJ+21], which is a restatement of Proposition 2 in [CJJS21]. The bound $C_{\text{ba}} \log(\frac{R\kappa}{r})$ on the number of calls to the oracle is immediate from the statement of Proposition 2. For the oracle parameter $\Delta = \frac{r}{C_{\text{ba}}}$, we note that the proof of Proposition 2 of [CJJS21] only requires that we obtain points at distance $O(r)$ from $x^\star_{\overline{x},\lambda}$ given a choice of λ, although it is stated as requiring a function error guarantee. This is evident where the

proof applies Lemma 3 of the same paper.

Ball optimization oracle parameters. The ball optimization oracle is called in the implementation of Line 5 of Algorithm 4 in [ACJ+21]. In iteration k of the algorithm, the error requirement is derived through the potential bound in Lemma 5 of [ACJ+21]. More precisely, Lemma 5 shows that (following their notation), conditioned on all randomness through iteration k,

$$\mathbb{E}\left[A_{k+1}\left(F(x_{k+1})-F(x^\star)\right)+\|v_{k+1}-x^\star\|^2\right]-\left(A_k\left(F(x_k)-F(x^\star)\right)+\|v_k-x^\star\|^2\right)$$
$$\leq -\frac{1}{6}\lambda_{k+1}A_{k+1}\|\widehat{x}_{k+1}-y_k\|^2+A_{k+1}\varphi_{k+1}+a_{k+1}^2\sigma_{k+1}^2+2Ra_{k+1}\delta_{k+1},$$

where the terms $a_{k+1}^2\sigma_{k+1}^2+2Ra_{k+1}\delta_{k+1}$ are handled identically in [ACJ+21] and our Proposition 2.2.8 (see the following discussion). For the remaining two terms, Proposition 4 of [ACJ+21] guarantees that as long as the method does not terminate, one of the following occurs.

1. $\|\widehat{x}_{k+1}-y_k\|^2=\Omega(r^2)$.

2. $\lambda_{k+1}=O(\lambda_\star)$.

In the first case, as long as φ_{k+1} (the error tolerance to the ball optimization oracle) is set to be $\frac{\lambda_{k+1}r^2}{C_{\text{ba}}}$ for a sufficiently large C_{ba} (which it is smaller than by logarithmic factors), up to constant factors the potential proof is unaffected. The total contributions to the potential due to all $A_{k+1}\varphi_{k+1}$ losses from the iterations of the second case across the entire algorithm is bounded by

$$O\left((K\log\kappa)\cdot\frac{R^2}{\varepsilon_{\text{opt}}}\cdot\frac{\lambda_\star r^2}{\log^3\kappa}\right)=O\left(R^2\right).$$

Here, the first term is the iteration count, the second term is due to an upper bound on A_{k+1}, and the third term is bounded since $\lambda_{k+1}=O(\lambda_\star)$. The initial potential in the proof of Proposition 2 of [ACJ+21] is R^2, so the final potential is unaffected by more than constant factors. For a more formal derivation of the same improved error tolerance, we refer the reader to [CH22], Lemma 8.

Stochastic proximal oracle parameters. Our stochastic proximal oracle parameters are exactly the settings of δ_k, σ_k required by Proposition 2 of [ACJ+21], except we simplified the bound on $\sigma_k^2 = O(\frac{\varepsilon}{a_k})$ (note we use ε_{opt} in place of ε). In particular, following notation of [ACJ+21], we have

$$\frac{\varepsilon}{a_k} = \frac{\varepsilon\sqrt{\lambda_k}}{\sqrt{A_k}} = \Omega\left(\varepsilon \cdot \sqrt{\lambda_\star} \cdot \frac{\sqrt{\varepsilon}}{R}\right) = \Omega\left(\frac{\varepsilon^2 K}{R^2}\log\kappa\right).$$

The first equality used $\lambda_k a_k^2 = A_k$ for the parameter choices of Algorithm 4 in [ACJ+21]. The second equality used that all $\lambda_k = \Omega(\lambda_\star)$ and all $A_k = O(\frac{R^2}{\varepsilon})$ in Algorithm 4 in [ACJ+21], where we chose $\lambda_\star = \frac{\varepsilon K^2}{R^2}\log^2\kappa$. The final equality plugged in this bound on $\lambda_\star$ and simplified. Hence, obtaining a variance as declared in Proposition 2.2.8 suffices to meet the requirement.

2.7 Discussion of Proposition 2.2.9

In this section, we discuss how to obtain Proposition 2.2.9 (which is based on Proposition 1 in [CH22]) from the analysis in [CH22]. The iteration count discussion is the same as in Appendix 2.6. We separate the discussion into parts corresponding to the two requirements in Proposition 2.2.9. Throughout, we will show how to use the analysis in [CH22] to guarantee that with probability at least $1 - \Omega(\frac{1}{\kappa})$, the algorithm has expected function error $O(\varepsilon_{\text{opt}})$; because the maximum error over $\mathbb{B}(R)$ is $\leq LR$, this corresponds to an overall error $O(\varepsilon_{\text{opt}})$, and we may adjust C_{ba} by a constant to compensate.

Per-iteration requirements. The ball optimization error guarantees are as stated in Proposition 1 of [CH22], except we dropped the function evaluations requirement. To see that this is obtainable, note that [CH22] obtains their line search oracle (see Proposition 2.2.8) by running $O(\log(\frac{R\kappa}{r}))$ ball optimization oracles to $O(\lambda r^2)$ expected error, querying the function value, and applying Markov's inequality to argue at least one will succeed with high probability. We instead apply a Chernoff bound with $O(\log(\frac{R\kappa}{r}))$ independent runs to argue that with probability $O(\frac{1}{K\kappa\cdot\text{polylog}(K\kappa)})$, the preconditions of Aggregate in Lemma 2.4.17 are met with $\Delta = O(r)$, as required by the line search oracle (see Algo-

rithm 6). Finally, applying a union bound over all iterations implies that the overall failure probability due to these line search oracles is $O(\frac{1}{\kappa})$ as required by our earlier argument.

Additional requirements. The error requirements of the queries which occur every $\approx 2^{-j}$ iterations are as stated in [CH22]. The only difference is that we state the complexity deterministically (Proposition 1 of [CH22] implicitly states an expected gradient bound). The stochastic proximal oracle is implemented as Algorithm 2, [CH22]; it is also adapted with slightly different parameters as Algorithm 11 of this paper. The expected complexity bound is derived by summing over all $j \in [\lceil \log_2 K + C_{\text{ba}} \rceil]$, the probability j is sampled in each iteration of Algorithm 2 of [CH22]. For all j a Chernoff bound shows that the number of times in the entire algorithm j is sampled is $O(2^{-j} K \log(\frac{R\kappa}{r}))$ (within a constant of its expectation), with probability $1 - \Omega(\text{poly}(\frac{r}{R\kappa}))$. Taking a union bound over all j shows the failure probability of our complexity bound is $O(\frac{1}{\kappa})$ as required.

2.8 Discussion of Proposition 2.3.5

In this section, we discuss how to obtain Proposition 2.3.5 using results in [GL12]. We first state the following helper fact on the smoothness of a convolved function $\widehat{f}_\rho$ (see Definition 2.2.1).

Fact 2.8.1 (Lemma 8, [BJL+19]). *If $f : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz, $\widehat{f}_\rho$ (see Definition 2.2.1) is $\frac{L}{\rho}$-smooth.*

The statement of Proposition 2.3.5 then follows from recursively applying Proposition 9 of [GL12] on the objective $\Psi = \widehat{f}_\rho + \frac{\lambda}{2}\|\cdot - \overline{x}\|^2$, which is λ-strongly convex and $(\frac{L}{\rho} + \lambda)$-smooth, together with the divergence choice of $V(x_0, x^*) := \frac{1}{2}\|x_0 - x^*\|^2$, which satisfies $\nu = 1$. Our parameter choices in Algorithm 17 are the same as in [GL12], where we use that our variance bound is $3L^2$ (Lemma 2.2.3).

In particular, denote the iterate x_T^{ag} after the k^{th} outer loop by x^k. We will inductively assume that $\mathbb{E}\,\frac{1}{2}\|x^{k-1} - x^\star_{\overline{x},\lambda}\|^2 \le \frac{r^2}{2^{k-1}}$ (clearly the base case holds). This then implies

$$\mathbb{E}\left[\frac{\lambda}{2}\|x^k - x^\star_{\overline{x},\lambda}\|^2\right] \le \mathbb{E}\left[\Psi(x^k) - \Psi(x^\star_{\overline{x},\lambda})\right] \le \frac{2(\frac{L}{\rho} + \lambda)\|x^{k-1} - x^\star_{\overline{x},\lambda}\|^2}{T(T+1)} + \frac{24L^2}{\lambda N_k (T+1)} \le \frac{\lambda}{2^k} r^2$$

where the second inequality is Proposition 9 in [GL12] (cf. equation (4.21) therein), and the last is by our choice of T and N_k. Thus, when $K > \log_2(\frac{\lambda r^2}{\varphi})$ we have $\mathbb{E}\,\Psi(x_T^{\mathsf{ag}}) - \Psi(x^\star_{\overline{x},\lambda}) \leq \varphi$ as in the last outer loop $k = K$. The computational depth follows immediately from computing TK, and the total oracle queries and computational complexity follow since N_K asymptotically dominates:

$$T \cdot \left(\sum_{k \in [K]} N_k \right) = O\left(TN_K + TK\right) = O\left(\sqrt{1 + \frac{L}{\rho\lambda}} \log\left(\frac{\lambda r^2}{\varphi}\right) + \frac{L^2}{\lambda\varphi} \right).$$

Chapter 3

PRIVATE CONVEX OPTIMIZATION VIA EXPONENTIAL MECHANISM

3.1 Introduction

Differential Privacy (DP), introduced in [DMNS06, DKM+06], is increasingly becoming the universally accepted standard in privacy protection. We see an increasing array of adoptions in industry [App17, EPK14, BEM+17, DKY17] and more recently the US census bureau [Abo16, KCK+18]. Differential privacy allows us to quantify the privacy loss of an algorithm and is defined as follows.

Definition 3.1.1 ((ε, δ)-DP)**.** A randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if for any neighboring databases $\mathcal{D}, \mathcal{D}'$ and any subset S of outputs, one has

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta.$$

In this paper, we say $\mathcal{D}$ and $\mathcal{D}'$ are neighboring databases if they agree on all the user inputs except for a single user's input.

Privacy concerns are particularly acute in machine learning and optimization using private user data. Suppose we want to minimize some loss function $F(x; \mathcal{D}) : \mathcal{K} \to \mathbb{R}$ for some domain $\mathcal{K}$ where $\mathcal{D}$ is some database. We want to output a solution x^{priv} using differentially private mechanism $\mathcal{M}$ such that we minimize the *excess empirical risk*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F(x^{priv}; \mathcal{D})] - F(x^*; \mathcal{D}), \tag{3.1}$$

where $x^* \in \mathcal{K}$ is the true minimizer of $F(x; \mathcal{D})$.

Exponential Mechanism One of the first mechanisms invented in differential privacy, the *exponential mechanism*, was proposed by [MT07] precisely to solve this. It involves

sampling x^{priv} from the density

$$\pi_{\mathcal{D}}(x) \propto \exp\left(-kF(x;\mathcal{D})\right). \tag{3.2}$$

Here k controls the privacy-vs-utility tradeoff, large k ensures that we get a good solution but less privacy and small k ensures that we get good privacy but we lose utility. Suppose $\Delta_F = \sup_{\mathcal{D}\sim\mathcal{D}'} \sup_x |F(x;\mathcal{D}) - F(x;\mathcal{D}')|$ is the sensitivity of F, where the supremum is over all neighboring databases $\mathcal{D}, \mathcal{D}'$. Then choosing $k = \frac{\varepsilon}{2\Delta_F}$, the exponential mechanism satisfies $(\varepsilon, 0)$-DP.

Exponential mechanism is widely used both in theory and in practice, such as in mechanism design [HK12], convex optimization [BST14, MV21], statistics [WZ10, WM10, AKRS19], machine learning and AI [ZP19]. Even for infinite and continuous domains, exponential mechanism can be implemented efficiently for many problems [HT10, CSS13, KT13, BV19, CKS20]. There are also several variants and generalizations of the exponential mechanism which can improve its utility based on different assumptions [TS13, BNS13, RS16, LT19]. See [LT19] for a survey of these results.

DP Empirical Risk Minimization (DP-ERM) In many applications, the loss function is given by the average of the loss of each user:

$$F(x;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n} f(x;s_i). \tag{3.3}$$

where $\mathcal{D} = \{s_1, s_2, \cdots, s_n\}$ is the collection of users s_i and $f(x;s_i)$ is the loss function of user s_i.

Throughout this paper, we assume $f(x;s)$ is convex and $f(x;s) - f(x;s')$ is G-Lipschitz for all s, s', and $\mathcal{K} \subset \mathbb{R}^d$ is convex with diameter D.[1] We call the problem of minimizing the excess empirical risk in (3.3) as DP Empirical Risk Minimization (DP-ERM). This setting is well studied by the DP community with many exciting results [CM08, RBHT12, CMS11, JT14, BST14, KJ16, FTS17, ZZMW17, Wan18, INS+19, BFTGT19,

[1] Some of our results can handle the unconstrained domain, such as $\mathcal{K} = \mathbb{R}^d$.

FKT20, KLL21, BGN21, LL21, AFKT21, SSTT21, MBST21, GTU22].[2]

In particular, [BST14] shows that exponential mechanism in (3.2) achieves the optimal excess empirical risk of $O\left(\frac{GDd}{n\varepsilon}\right)$ under $(\varepsilon, 0)$-DP. On the other hand, [BST14, BFTGT19, BFGT20] show that *noisy gradient descent* on $F(x; \mathcal{D})$ achieves an excess empirical risk of

$$O\left(\frac{GD\sqrt{d\log(1/\delta)}}{n\varepsilon}\right) \tag{3.4}$$

under (ε, δ)-DP, which is also shown to be optimal [BST14]. This is a significant $\sqrt{d}$ improvement over the exponential mechanism.

Exponential mechanism is a universally powerful tool in differential privacy. However, nearly all of the previous works on DP-ERM rely on noisy gradient descent or its variants to achieve the significant $\sqrt{d}$ improvement over exponential mechanism under (ε, δ)-DP. One natural question is whether noisy gradient descent has some extra ability that exponential mechanism lacks or we didn't use exponential mechanism optimally in this setting. This brings us to the first question.

Question 3.1.2. *Can we obtain the optimal empirical risk in* (3.1) *under* (ε, δ)*-DP using exponential mechanism?*

DP Stochastic Convex Optimization (DP-SCO) Beyond the privacy guarantee and the empirical risk guarantee, another important guarantee is the generalization guarantee. Formally, we assume the users are sampled from an unknown distribution $\mathcal{P}$ over convex functions. We define the loss function as

$$\widehat{F}(x) = \mathop{\mathbb{E}}_{s\sim\mathcal{P}}[f(x; s)]. \tag{3.5}$$

[2]Most of the literature uses a stronger assumption that $f(x; s)$ is G-Lipschitz, while some of our results only need to assume the difference $f(x; s) - f(x; s')$ is G-Lipschitz.

We want to design a DP mechanism $\mathcal{M}$ which outputs x^{priv} given users $\mathcal{D} = \{s_1, s_2, \ldots, s_n\}$ independently sampled from $\mathcal{P}$ and minimize the *excess population loss*

$$\mathop{\mathbb{E}}_{\mathcal{M},\mathcal{D}\sim\mathcal{P}}[\widehat{F}(x^{priv})] - \widehat{F}(x^*) \tag{3.6}$$

where x^* is the minimizer of $\widehat{F}(x)$. We call the problem of minimizing the excess population loss in (3.6) as DP Stochastic Convex Optimization (DP-SCO). By a suitable modification of noisy stochastic gradient descent, [BFTGT19, FKT20] show that one can achieve the optimal population loss of

$$O\left(GD\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon n}\right)\right). \tag{3.7}$$

[BFTGT19] bounds the generalization error by showing that running SGD on smooth functions is stable and [FKT20] proposes an iterative localization technique. Note that only the algorithm for smooth functions in [BFTGT19] can achieve both optimal empirical risk and optimal population loss at the same time, with the price of taking more gradient queries and loss of efficiency. It is unclear to us how one can obtain both using current techniques for non-smooth functions. This brings us to the second question.

Question 3.1.3. *Can we achieve both the optimal empirical risk and the optimal population loss for non-smooth functions with the same algorithm?*

Sampling Without extra smoothness assumptions on f, currently, there is no optimally efficient algorithm for both problems. For example, with oracle access to gradients of f, the previous best algorithms for DP-SCO use:

- $\widetilde{O}(nd)$ queries to $\nabla f(x; s)$ (by combining [FKT20], Moreau-Yosida regularization and cutting plane methods),
- $\widetilde{O}(\min(n^{3/2}, n^2/\sqrt{d}))$ queries to $\nabla f(x; s)$ [AFKT21],
- $\widetilde{O}(\min(n^{5/4}d^{1/8}, n^{3/2}/d^{1/8}))$ queries to $\nabla f(x; s)$ [KLL21].

Combining these results, this gives an algorithm for DP-SCO that uses

$$\widetilde{O}(\min(nd, n^{5/4}d^{1/8}, n^{3/2}/d^{1/8}, n^2/\sqrt{d}))$$

many queries to $\nabla f(x; s)$. Although the information lower bound for non-smooth functions with the gradient queries is open, it is unlikely that the answer involves four different cases.

In this paper, we focus on the function value query (zeroth order query) on $f(x; s)$. This query is weaker than gradient query as it obtains d times less information. They are used in many practical applications such as clinical trials and ads placement when the gradient is not available and is also useful in bandit problems. This brings us to the third question.

Question 3.1.4. *Can we obtain an algorithm with optimal query complexity for DP-SCO for zeroth order query model?*

3.1.1 Our Contributions

In this paper, we give a positive answer to all these questions using the *Regularized Exponential Mechanism*. If we add an ℓ_2^2 regularizer to F and sample x^{priv} from the density

$$\exp\left(-k\left(F(x; \mathcal{D}) + \mu \|x\|_2^2/2\right)\right), \tag{3.8}$$

then, for a suitable choice of μ and k, we recover the optimal excess risk in (3.4) for DP-ERM and optimal population loss in (3.7) for DP-SCO. Finally, we give an algorithm to sample x^{priv} from the density (3.8) with nearly optimal number of queries to $f(x; s)$ (See Figure 3.1). To the best of our knowledge, our algorithm is the first whose query complexity has *polylogarithmic dependence* in both dimension and accuracy (in TV distance).

Formally, our result is follows:

Theorem 3.1.5 (DP-ERM, Informal)**.** *Let $\mathcal{K}$ be a convex set with diameter D and $\{f(\cdot; s)\}$ be a family of convex functions on $\mathcal{K}$ where $f(\cdot; s) - f(\cdot; s')$ is G-Lipschitz for all s, s'. Given a*

database $\mathcal{D} = \{s_1, s_2, \cdots, s_n\}$*, for any* $\varepsilon, \delta \in (0, \frac{1}{10})$*,* [3] *the regularized exponential mechanism*

$$x^{(priv)} \propto \exp\left(-k \cdot \left(\frac{1}{n}\sum_{i=1}^{n} f(x; s_i) + \frac{\mu}{2}\|x\|_2^2\right)\right)$$

is (ε, δ)*-DP with expected excess empirical loss*

$$\frac{2GD\sqrt{d\log(1/\delta)}}{\varepsilon n}$$

for some appropriate choices of k *and* μ*. Furthermore, if* $f(\cdot; s)$ *is* G*-Lipschitz for all* s*, we can sample* $x^{(priv)}$ *using* $O(\frac{\varepsilon^2 n^2}{\log(1/\delta)} \log^2(\frac{nd}{\delta}))$ *queries in expectation to the values of* $f(x; s)$*.*

Theorem 3.1.6 (DP-SCO, Informal)**.** *Let* $\mathcal{K}$ *be a convex set with diameter* D *and* $\{f(\cdot; s)\}$ *be a family of convex functions on* $\mathcal{K}$ *where* $f(\cdot; s) - f(\cdot; s')$ *is* G*-Lipschitz for all* s, s'*. Given a database* $\mathcal{D} = \{s_1, s_2, \cdots, s_n\}$ *of samples from some unknown distribution* $\mathcal{P}$*. For any* $\varepsilon, \delta \in (0, \frac{1}{10})$*,* [4] *the regularized exponential mechanism*

$$x^{(priv)} \propto \exp\left(-k \cdot \left(\frac{1}{n}\sum_{i=1}^{n} f(x; s_i) + \frac{\mu}{2}\|x\|_2^2\right)\right)$$

is (ε, δ)*-DP with expected excess population loss*

$$\frac{2GD}{\sqrt{n}} + \frac{2GD\sqrt{d\log(1/\delta)}}{\varepsilon n}$$

for some appropriate choice of k *and* μ*. Furthermore, if* $f(\cdot; s)$ *is* G*-Lipschitz for all* s*, we can sample* $x^{(priv)}$ *using* $O(\min\{\frac{\varepsilon^2 n^2}{\log(1/\delta)}, nd\} \log^2(\frac{nd}{\delta}))$ *queries in expectation to the values of* $f(x; s)$ *and the expected number of queries is optimal up to logarithmic terms.*

For DP-SCO, we provide a nearly matching information-theoretic lower bound on the number of value queries (Section 3.7), proving the optimality of our sampling algorithm. Moreover, when f is already strongly convex, our proof shows the exponential mechanism

[3] See Theorem 3.6.2 for general conclusions for all $\varepsilon > 0$

[4] See Theorem 3.6.9 for general conclusions for all $\varepsilon > 0$.

(without adding a regularizer) itself simultaneously achieves both the optimal excess empirical risk and optimal population loss.

In a *concurrent and independent* work, [GTU22] study the DP properties of Langevin Diffusion, and provide optimal/best known private empirical risk and population loss under both pure-DP ($\delta = 0$) and approximate-DP ($\delta > 0$) constraints. Utility/privacy trade-off of non-convex functions is also discussed.

3.2 Techniques

The main contribution of this paper is the discovery that adding regularization terms in exponential mechanism leads to optimal algorithms for DP-ERM and DP-SCO. For this, we develop some important tools that could be of independent interest. We now briefly discuss each of the main tools.

3.2.1 Gaussian Differential Privacy (GDP) of Regularized Exponential Mechanism

To analyze the privacy of the regularized exponential mechanism, we need to bound the privacy curve between a strongly log-concave distribution and its Lipschitz perturbation in the exponent. [MASN16] gave a nearly tight (up to constants) privacy guarantee of exponential mechanism if the distribution $\exp(-kF(x;\mathcal{D}))$ satisfies Logarithmic Sobolev inequality (LSI). Since strongly log-concave distributions satisfy LSI, their result immediately gives the (ε, δ)-DP guarantee of our algorithm. However, this gives a sub-optimal privacy bound because it does not fully take advantage of the strongly log-concave property.

Instead, we show directly that the privacy curve between a strongly log-concave distribution and its Lipschitz perturbation in the exponent is upper bounded by the privacy curve of an appropriate Gaussian mechanism. This new proof uses the notion of tradeoff function introduced in [DRS19] and the isoperimetric inequality for strongly log-concave distribution.

Theorem 3.2.1. *Given convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and μ-strongly convex functions $F, \tilde{F}$ over $\mathcal{K}$. Let P, Q be distributions over $\mathcal{K}$ such that $P(x) \propto e^{-F(x)}$ and $Q(x) \propto e^{-\tilde{F}(x)}$. If $\tilde{F} - F$ is*

G-Lipschitz over $\mathcal{K}$, then for all $\varepsilon > 0$,

$$\delta(P \parallel Q)(\varepsilon) \leq \delta\Big(\mathcal{N}(0,1) \Big\| \mathcal{N}\left(\frac{G}{\sqrt{\mu}}, 1\right)\Big)(\varepsilon).$$

This proves that the privacy curve for distinguishing between P, Q is upper bounded the privacy curve of a Gaussian mechanism with sensitivity $G/\sqrt{\mu}$ and noise scale 1.

Tightness: Note that Theorem 3.2.1 is completely tight because it contains the privacy of Gaussian mechanism as a special case. If $F(x) = \|x\|_2^2/2$ and $\tilde{F}(x) = \|x-a\|_2^2/2$ for some $a \in \mathbb{R}^d$, then $\tilde{F}(x) - F(x) = -\langle x, a\rangle + \|a\|_2^2/2$ is G-Lipschitz with $G = \|a\|_2$ and $F, \tilde{F}$ are 1-strongly convex. And $P = \mathcal{N}(0, I_d)$ and $Q = \mathcal{N}(a, I_d)$. Therefore:

$$\delta(P \parallel Q) = \delta(\mathcal{N}(0, I_d) \parallel \mathcal{N}(a, I_d)) = \delta(\mathcal{N}(0,1) \parallel \mathcal{N}(\|a\|_2, 1))$$

which is precisely the upper bound guaranteed by the theorem.

3.2.2 Generalization Error of Sampling

Many important and fundamental problems in machine learning, optimization and operations research are special cases of SCO, and ERM is a classic and widely-used approach to solve it, though their relationships are not well-understood. If one can solve the ERM problem optimally and get the exact optimal solution x^* to minimizing $F(\cdot; \mathcal{D})$ (see Equation 3.3), then [SSSSS09] showed x^* will also be a good solution to the SCO for strongly convex functions. But in most situations, solving ERM optimally costs too much or even impossible. Can we find a approximately good solution to ERM and hope that it is also a good solution for SCO? [Fel16] provides a negative answer and shows there is no good uniform convergence between $F(\cdot; \mathcal{D})$ and $\widehat{F}$, that is there always exists $x \in \mathcal{K}$ such that $|F(x; \mathcal{D}) - \widehat{F}(x)|$ is large. This fact forces us to find approximate solution to ERM with very high accuracy, which makes the algorithms inefficient.

Prior works proposed a few interesting ways to overcome this difficulty, such as the uniform stability in [HRS16] and the iterative localization technique in [AFKT21]. Roughly speaking, uniform stability means that if running algorithms on neighboring datasets lead to

similar output distributions, then the generalization error of the ERM algorithm is bounded. Thus a good solution to ERM obtained by a stable algorithm is also a good solution for SCO. [BFTGT19] makes use of the stability of running SGD on smooth functions to get a tight bound on the population loss for DP-SCO.

Recall $F(x;\mathcal{D})$ and $\widehat{F}(x)$ are defined in Equation (3.3) and (3.5) respectively. Our result enriches the toolbox of bounding the generalization error and provides new insights for this problem.

Theorem 3.2.2. *Suppose $\{f_i\}$ is a family of μ-strongly convex functions over $\mathcal{K}$ and $f_i - f_{i'}$ is G-Lipschitz for any two functions $f_i, f_{i'}$ in the family. For any $k > 0$ and suppose the n samples in data set $\mathcal{D}$ are drawn i.i.d from the underlying distribution, then by sampling $x^{(sol)}$ from density $\propto e^{-kF(x^{(sol)};\mathcal{D})}$, the population loss satisfies*

$$\mathbb{E}[\widehat{F}(x^{(sol)})] - \min_{x\in\mathcal{K}} \widehat{F}(x) \leq \frac{G^2}{\mu n} + \frac{d}{k}.$$

Considering two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, our result is based on bounding the Wasserstein distance between the distributions proportional to $e^{-kF(x;\mathcal{D})}$ and $e^{-kF(x;\mathcal{D}')}$, which means the sampling scheme is stable and leads to the $\frac{G^2}{\mu n}$ term in generalization error. The other term $\frac{d}{k}$ is excess empirical loss of the sampling mechanism. One advantage of our result is that it works for both smooth and non-smooth functions. Moreover, we may choose the value k carefully and get a solution with both optimal empirical loss and optimal population loss.

3.2.3 Non-smooth Sampling and DP Convex Optimization

Implementing the exponential mechanism involves sampling from a log-concave distribution. When the negative log-density function F is smooth, i.e. the gradient of F is Lipschitz, there are many efficient algorithms for this sampling tasks such as [Dal17b, LSV18, MMW+21, CV19, DMM19, SL19, CDWY20, LST20]. For example, if $F = \frac{1}{n}\sum_{i=1}^{n} f_i$ and each f_i is 1-strongly convex with κ-Lipschitz gradient,[5] we can sample $x \sim \exp(-F(x))$ in $\widetilde{O}(n +$

[5]For convenience, we used f_i to denote the function $f(\cdot; s_i)$ in this and Section 3.5.

$\kappa \max(d, \sqrt{nd}) \log(1/\delta))$ iterations with δ error in total variation distance and each iteration involves computing one $\nabla f_i(x)$ [LST21b]. Note that this is nearly linear time when $n \gg \kappa^2 d$ and the δ error in total variation distance can be translated to an extra δ error in the (ε, δ)-DP guarantee.

	Complexity	Oracle	Guarantee
[BST14]	$d^{O(1)}$	$F(x)$	$\mathrm{D}_\infty \le \varepsilon$
[CDJB20]	$G^{O(1)} d^{5/2}/\varepsilon^4$	$\nabla F(x)$	$\mathrm{W}_2 \le \delta$
[JLLV21] + [Che21a]	d^3	$F(x)$	$\mathrm{TV} \le \delta$
[GT20]	$\frac{\alpha^2 G^4 d}{\varepsilon^2}$	$\nabla F(x)$	$\mathrm{D}_\alpha \le \varepsilon$
[LC22]	$\frac{G^2}{\delta}$	$\nabla F(x)$	$\mathrm{TV} \le \delta$
This	G^2	$f_i(x)$	$\mathrm{TV} \le \delta$

Figure 3.1: The complexity of sampling from $\exp(-F(x))$ where $F = \frac{1}{n}\sum_i f_i$ is 1-strongly convex and f_i are G-Lipschitz and convex. For applications in differential privacy, ε is a constant and $\delta = n^{-\Theta(1)}$. Polylogarithmic terms are omitted. Only the last result uses the summation structure and queries only one f_i each step.

Unfortunately, when the functions f_i are only Lipschitz but not smooth, this problem is more difficult. In Table 3.1, we summarize some existing results on this topic. They use different guarantees such as Renyi divergence D_α of order α, Wasserstein distance W_2 and total variation distance TV (defined in subsection 3.3.3). For applications in differential privacy, we need either polynomially small W_2 or TV distance, or ε small D_α distance.

All previous results for non-smooth function use oracle access to F or ∇F (instead of f_i) and have iterative complexity at least d iterations for W_2 or TV distance smaller than $1/d$. Because of this, our algorithm is significantly faster than the previous algorithms and can handle the case when F is expectation of (infinitely many) f_i directly. For example, to get the optimal private empirical loss with typical settings where $\varepsilon = \Theta(1)$ and $\delta = 1/n^{\Theta(1)}$, the previous best samplers use $\widetilde{O}(n^4 d)$ many queries to $\nabla f_i(x)$ by [GT20] or $\widetilde{O}(nd^3)$ many queries to $f_i(x)$ by combining [JLLV21] and [Che21a]. In comparison, our algorithm only takes $\widetilde{O}(n^2)$ many $f_i(x)$.

Our result is based on the alternating sampler proposed in [LST21b] and a new rejection sampling scheme.

Theorem 3.2.3. *Given a μ-strongly convex function $\psi(x)$ defined on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and $+\infty$ outside. Given a family of G-Lipschitz convex functions $\{f_i(x)\}_{i\in I}$ defined on $\mathcal{K}$ and an initial point $x_0 \in \mathcal{K}$. Define the function $\widehat{F}(x) = \mathbb{E}_{i\in I}\, f_i(x) + \psi(x)$ and the distance $D = \|x_0 - x^*\|_2$ for some $x^* = \arg\min_{x\in\mathcal{K}} \widehat{F}(x)$. For any $\delta \in (0, 1/2)$, we can generate a random point x that has δ total variation distance to the distribution proportional to $\exp(-\widehat{F}(x))$ in*

$$T := \Theta\left(\frac{G^2}{\mu}\log^2\left(\frac{G^2(d/\mu + D^2)}{\delta}\right)\right) \text{ steps.}$$

Furthermore, each steps accesses only $O(1)$ many $f_i(x)$ and samples from $\exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)$ for $O(1)$ many y in expectation with $\eta = \Theta(G^{-2}/\log(T/\delta))$.

3.3 Preliminaries

3.3.1 Differential Privacy

A DP algorithm $\mathcal{M}$ usually satisfies a collection of (ε, δ)-DP guarantees for each ε, i.e., for each ε there exists some smallest δ for which $\mathcal{M}$ is (ε, δ)-DP. By collecting all of them together, we can form the privacy curve or privacy profile which fully characterizes the privacy of a DP algorithm.

Definition 3.3.1 (Privacy Curve)**.** Given two random variables X, Y supported on some set Ω, define the privacy curve $\delta(X\|Y) : \mathbb{R}_{\geq 0} \to [0, 1]$ as:

$$\delta(X\|Y)(\varepsilon) = \sup_{S\subset\Omega} \Pr[Y \in S] - e^{\varepsilon}\Pr[X \in S].$$

One can explicitly calculate the privacy curve of a Gaussian mechanism as

$$\delta(\mathcal{N}(0,1) \,\|\, \mathcal{N}(s,1))(\varepsilon) = \Phi\left(-\frac{\varepsilon}{s} + \frac{s}{2}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon}{s} - \frac{s}{2}\right) \tag{3.9}$$

where $\Phi(\cdot)$ is the Gaussian cumulative distribution function (CDF) [BW18].

We say a differentially private mechanism $\mathcal{M}$ has privacy curve $\delta : \mathbb{R}_{\geq 0} \to [0, 1]$ if for every $\varepsilon \geq 0$, $\mathcal{M}$ is $(\varepsilon, \delta(\varepsilon))$-differentially private, i.e., $\delta(\mathcal{M}(\mathcal{D})\|\,\mathcal{M}(\mathcal{D}'))(\varepsilon) \leq \delta(\varepsilon)$ for all neighbouring databases $\mathcal{D}, \mathcal{D}'$. We will also need the notion of tradeoff function introduced

in [DRS19] which is an equivalent way to describe the privacy curve $\delta(P\|Q)$.

Definition 3.3.2 (Tradeoff function)**.** Given two (continuous) distributions P, Q, we define the trade-off function[6] $T(P\|Q) : [0,1] \to [0,1]$ as

$$T(P\|Q)(z) = \inf_{S:P(S)=1-z} Q(S).$$

It is easy to compute explicitly the tradeoff function for Gaussian mechanism [DRS19],

$$T(\mathcal{N}(0,1)\|\mathcal{N}(s,1))(z) = \Phi(\Phi^{-1}(1-z) - s). \tag{3.10}$$

Note that perfect privacy is equivalent to the tradeoff function $\mathrm{Id}(z) = 1 - z$ and the closer a tradeoff function is to Id, better the privacy. The tradeoff function $T(P\|Q)$ and the privacy curve $\delta(P\|Q)$ are related via convex duality. Therefore to compare privacy curves, it is enough to compare tradeoff curves.

Proposition 3.3.3 ([DRS19])**.** $\delta(P\|Q) \leq \delta(P'\|Q')$ *iff* $T(P\|Q) \geq T(P'\|Q')$

3.3.2 Optimization

Here we collect some properties of functions which are useful for optimization and sampling.

Definition 3.3.4 (L-Lipschitz Continuity)**.** A function $f : \mathcal{K} \to \mathbb{R}$ is L-Lipschitz continuous over the domain $\mathcal{K} \subset \mathbb{R}^d$ if the following holds for all $\omega, \omega' \in \mathcal{K} : |f(\omega) - f(\omega')| \leq L\|\omega - \omega'\|_2$.

Definition 3.3.5 (μ-Strongly convex)**.** A differentiable function $f : \mathcal{K} \to \mathbb{R}$ is called strongly convex with parameter $\mu > 0$ if $\mathcal{K} \subset \mathbb{R}^d$ is convex and the following inequality holds for all points $\omega, \omega' \in \mathcal{K}$,

$$f(\omega') \geq f(\omega) + \langle \nabla f(\omega), \omega' - \omega \rangle + \frac{\mu}{2}\|\omega' - \omega\|_2^2.$$

Definition 3.3.6 (Log-concave measure and density)**.** A density function $f : \mathcal{K} \to \mathbb{R}_{\geq 0}$ is log-concave if $\int_{\mathcal{K}} f(x)dx = 1$ and $f(x) = \exp(-F(x))$ for some convex function F. We

[6] Tradeoff curves in [DRS19] are defined using type I and type II errors. The definition given here is equivalent to their definition for continuous distributions.

call f is μ-strongly log-concave if F is μ-strongly convex. Similarly, we call π a log-concave measure if its density function is log-concave, and we call π is a μ-strongly log-concave measure if its density function is μ-strongly log-concave.

3.3.3 Distribution Distance and Divergence

We present some distribution distances or divergences mentioned or used in this work.

Definition 3.3.7. [Rén61, Rényi Divergence] Suppose $1 < \alpha < \infty$ and π, ν are measures with $\pi \ll \nu$. The Rényi divergence of order α between π and ν is defined as

$$\mathrm{D}_\alpha(\pi\|\nu) = \frac{1}{\alpha}\log\int\left(\frac{\pi(x)}{\nu(x)}\right)^\alpha \nu(x)\mathrm{d}x.$$

We follow the convention that $\frac{0}{0} = 0$. Rényi Divergence of orders $\alpha = 1, \infty$ are defined by continuity. For $\alpha = 1$, the limit in Rényi Divergence equals to the Kullback-Leibler divergence of π from ν, which is defined as following:

Definition 3.3.8 (Kullback–Leibler divergence)**.** The Kullback–Leibler divergence between probability measures π and ν is defined by

$$\mathrm{D}_{KL}(\pi\|\nu) = \int \log\left(\frac{\pi}{\nu}\right)\mathrm{d}\pi.$$

Definition 3.3.9 (Wasserstein distance)**.** Let π, ν be two probability distributions on $\mathbb{R}^d$. The second Wasserstein distance W_2 between π and ν is defined by

$$\mathrm{W}_2(\pi,\nu) = \big(\inf_{\gamma\in\Gamma(\pi,\nu)}\int_{\mathbb{R}^d\times\mathbb{R}^d}\|x-y\|_2^2\mathrm{d}\gamma(x,y)\big)^{1/2},$$

where $\Gamma(\pi,\nu)$ is the set of all couplings of π and ν.

Definition 3.3.10 (Total variation distance)**.** The total variation distance between two probability measures π and ν on a sigma-algebra $\mathcal{F}$ of subsets of the sample space Ω is defined via

$$\mathrm{TV}(\pi,\nu) = \sup_{S\in\mathcal{F}}|\pi(S)-\nu(S)|.$$

3.3.4 Isoperimetric Inequality for Strongly Log-concave Distributions

The cumulative distribution function (CDF) of one-dimensional standard Gaussian distribution will be denoted by $\Phi(x) = \Pr_{y\sim\mathcal{N}(0,1)}[y \le x]$. The following Lemma relates the expanding property of log-concave measures with Φ.

Proposition 3.3.11 (Theorem 1.1. in [Led99])**.** *Let π be a μ-strongly log-concave measure supported on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$. Let $A \subset \mathcal{K}$ by any subset such that $\pi(A) = z$. For any point $x \in \mathbb{R}^d$, define $d(x, A) = \inf_{y\in A} \|x - y\|_2$. Let $A_r = \{x : d(x, A) \le r\}$. Then if $A_r \subseteq \mathcal{K}$, for every $r \ge 0$,*

$$\pi(A_r) \ge \Phi(\Phi^{-1}(z) + r\sqrt{\mu}).$$

The property above implies the concentration of Lipschitz functions over log-concave measures.

Corollary 3.3.12. *Let π be a μ-strongly log-concave measure supported on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$. Suppose $\alpha : \mathcal{K} \to \mathbb{R}$ is G-Lipschitz. For $z \in [0, 1]$, define $m(z) \in \mathbb{R}$ such that $\Pr_{x\sim\pi}[\alpha(x) \le m(z)] = z$. Then for every $r \ge 0$,*

$$\Pr_{x\sim\pi}[\alpha(x) \ge m(z) + r] \le \Phi\left(\Phi^{-1}(1 - z) - \frac{r\sqrt{\mu}}{G}\right),$$

$$\Pr_{x\sim\pi}[\alpha(x) \le m(z) - r] \le \Phi\left(\Phi^{-1}(z) - \frac{r\sqrt{\mu}}{G}\right).$$

Proof. Fix some $z \in [0, 1]$. Let $A = \{x \in \mathcal{K} : \alpha(x) \le m(z)\}$, so $\pi(A) = z$. Let $A_r = \{x : d(x, A) \le r\}$. Since α is G-Lipschitz, $\alpha(x) \ge m(z)+r$ implies that $d(x, A) \ge r/G$. Therefore $\{x : \alpha(x) \ge m(z) + r\} \subset \{x : d(x, A) \ge r/G\} = \overline{A_{r/G}}$ and so

$$\begin{aligned}
\Pr_{x\sim\pi}[\alpha(x) \ge m(z) + r] &\le \pi(\overline{A_{r/G}}) \\
&= 1 - \pi(A_{r/G}) \\
&\le 1 - \Phi\left(\Phi^{-1}(z) + \frac{r\sqrt{\mu}}{G}\right) \\
&= \Phi\left(-\Phi^{-1}(z) - \frac{r\sqrt{\mu}}{G}\right).
\end{aligned}$$

We obtain the other inequality by applying the above inequality to $-\alpha(x)$. □

3.4 GDP of Regularized Exponential Mechanism

In this section, we prove our DP result (Theorem 3.2.1). The proof uses the isoperimetric inequality for strongly log-concave measures [Led99]. Intuitively, the privacy loss random variable will be G-Lipschitz under the hypothesis and isoperimetric inequality implies that any Lipschitz function will be as concentrated as a Gaussian with appropriate standard deviation. This allows us compare the privacy curve $\delta(P \parallel Q)$ to that of a Gaussian mechanism. In our proof, it is actually more convenient to compare tradeoff curves ($T(P \parallel Q)$) which are equivalent to privacy curves via convex duality (Proposition 3.3.3 and Theorem 3.2.1).

Theorem 3.4.1. *Given convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and μ-strongly convex functions $F, \tilde{F}$ over $\mathcal{K}$. Let P, Q be distributions over $\mathcal{K}$ such that $P(x) \propto e^{-F(x)}$ and $Q(x) \propto e^{-\tilde{F}(x)}$. If $\tilde{F} - F$ is G-Lipschitz over $\mathcal{K}$, then for all $z \in [0,1]$,*

$$T(P \parallel Q)(z) \geq T\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G}{\sqrt{\mu}}, 1\right)\right)(z).$$

Proof. Let $\gamma = G/\sqrt{\mu}$. Let $\alpha(x) = \tilde{F}(x) - F(x)$ so that $Q(x) \propto e^{-\alpha(x)}P(x)$. Recall that we have $T(P\|Q)(z) = \inf_{S:P(S)=1-z} Q(S)$. Note that the infimum is achieved when we choose $S = \{x \in \mathcal{K} : \alpha(x) \geq m(z)\}$ for some $m(z)$ chosen such that $P(S) = \Pr_{x\sim P}[\alpha(x) \geq m(z)] = 1 - z$ (Neyman-Pearson lemma). Therefore:

$$\begin{aligned} T(P\|Q)(z) &= \int_{x\in S} Q(x)\mathrm{d}x \\ &= \frac{\int_{x\in S} e^{-\alpha(x)}P(x)\mathrm{d}x}{\int_{x\in\mathcal{K}} e^{-\alpha(x)}P(x)\mathrm{d}x} \\ &= \left(1 + \frac{\mathbb{E}_P[e^{-\alpha}\mathbf{1}_{\overline{S}}]}{\mathbb{E}_P[e^{-\alpha}\mathbf{1}_S]}\right)^{-1} \end{aligned}$$

We will now lower bound $\mathbb{E}_P[e^{-\alpha}\mathbf{1}_S]$. Let the random variable $Y = \alpha(x)$ where $x \sim P$. Let $f_Y(\cdot)$ be the PDF of Y.

$$\mathop{\mathbb{E}}_{P}[e^{-\alpha(x)}\mathbf{1}_S] = \int_{x:\alpha(x)\geq m(z)} e^{-\alpha(x)}P(x)\mathrm{d}x = \mathbb{E}[e^{-Y}\mathbf{1}(Y \geq m(z))] = \int_{m(z)}^{\infty} e^{-t}f_Y(t)dt$$

$$
\begin{aligned}
&= \int_{t=0}^{\infty} e^{-t-m(z)} \left(-\frac{\mathrm{d}\Pr_{x\sim P}\left[\alpha(x) \geq t + m(z)\right]}{\mathrm{d}t} \right) \mathrm{d}t \\
&= e^{-m(z)} \left(-e^{-t} \Pr_{x\sim P}\left[\alpha(x) \geq t + m(z)\right]\Big|_0^{\infty} - \int_{t=0}^{\infty} e^{-t} \Pr_{x\sim P}\left[\alpha(x) \geq t + m(z)\right] \mathrm{d}t \right) \\
&= (1-z)e^{-m(z)} - e^{-m(z)} \int_{t=0}^{\infty} e^{-t} \Pr_{x\sim P}\left[\alpha(x) \geq t + m(z)\right] \mathrm{d}t \\
&\geq (1-z)e^{-m(z)} - e^{-m(z)} \int_{t=0}^{\infty} e^{-t} \Phi(\Phi^{-1}(1-z) - t/\gamma) \mathrm{d}t \qquad \text{(Corollary 3.3.12)} \\
&= (1-z)e^{-m(z)} - e^{-m(z)} \left((1-z) - \exp\left(\frac{\gamma^2}{2} - \Phi^{-1}(1-z)\gamma \right) \Phi(\Phi^{-1}(1-z) - \gamma) \right) \\
&\qquad\qquad \text{(Claim 3.4.2)} \\
&= \exp\left(\frac{\gamma^2}{2} + \Phi^{-1}(z)\gamma - m(z) \right) \Phi(-\Phi^{-1}(z) - \gamma)
\end{aligned}
$$

We will now upper bound $\mathbb{E}_P[e^{-\alpha}\mathbf{1}_{\overline{S}}]$ in a similar way.

$$
\begin{aligned}
\mathop{\mathbb{E}}_{P}[e^{-\alpha(x)}\mathbf{1}_{\overline{S}}] &= \int_{x:\alpha(x)\leq m(z)} e^{-\alpha(x)} P(x) \mathrm{d}x \\
&= \int_{t=0}^{\infty} e^{-m(z)+t} \left(-\frac{d\Pr_{x\sim P}\left[\alpha(x) \leq m(z) - t\right]}{dt} \right) \mathrm{d}t \\
&= e^{-m(z)} \left(-e^{t} \Pr_{x\sim P}\left[\alpha(x) \leq m(z) - t\right]\Big|_0^{\infty} + \int_{t=0}^{\infty} e^{t} \Pr_{x\sim P}\left[\alpha(x) \leq m(z) - t\right] \mathrm{d}t \right) \\
&= ze^{-m(z)} + e^{-m(z)} \int_{t=0}^{\infty} e^{t} \Pr_{x\sim P}\left[\alpha(x) \leq m(z) - t\right] \mathrm{d}t \\
&\leq ze^{-m(z)} + e^{-m(z)} \int_{t=0}^{\infty} e^{t} \Phi(\Phi^{-1}(z) - t/\gamma) \mathrm{d}t \qquad \text{(Corollary 3.3.12)} \\
&= ze^{-m(z)} + e^{-m(z)} \left(-z + \exp\left(\frac{\gamma^2}{2} + \Phi^{-1}(z)\gamma \right) \Phi(\Phi^{-1}(z) + \gamma) \right) \\
&\qquad\qquad \text{(Claim 3.4.2)} \\
&= \exp\left(\frac{\gamma^2}{2} + \Phi^{-1}(z)\gamma - m(z) \right) \Phi(\Phi^{-1}(z) + \gamma)
\end{aligned}
$$

Combining the two bounds, we get:

$$
\begin{aligned}
T(P\|Q)(z) &= \left(1 + \frac{\mathbb{E}_P[e^{-\alpha}\mathbf{1}_{\overline{S}}]}{\mathbb{E}_P[e^{-\alpha}\mathbf{1}_{S}]} \right)^{-1} \\
&\geq \left(1 + \frac{\Phi(\Phi^{-1}(z) + \gamma)}{\Phi(-\Phi^{-1}(z) - \gamma)} \right)^{-1}
\end{aligned}
$$

$$= \Phi(-\Phi^{-1}(z) - \gamma) \qquad \text{(Using } \Phi(x) + \Phi(-x) = 1\text{)}$$
$$= T(N(0,1) \parallel N(\gamma, 1)). \qquad \text{(Eqn (3.10))}$$

□

We finish by calculating the integrals that showed up in the proof.

Claim 3.4.2.

$$\int_0^\infty e^{-t}\Phi\left(a - \frac{t}{\gamma}\right) \mathrm{d}t = \Phi(a) - e^{\frac{\gamma^2}{2} - a\gamma}\Phi(a - \gamma)$$
$$\int_0^\infty e^{t}\Phi\left(a - \frac{t}{\gamma}\right) \mathrm{d}t = -\Phi(a) + e^{\frac{\gamma^2}{2} + a\gamma}\Phi(a + \gamma)$$

Proof.

$$\begin{aligned}
\int_0^\infty e^{-t}\Phi(a - t/\gamma)\mathrm{d}t &= -e^{-t}\Phi(a - t/\gamma)\big|_0^\infty - \int_0^\infty e^{-t}\frac{e^{-(a-t/\gamma)^2/2}}{\gamma\sqrt{2\pi}}\mathrm{d}t \\
&= \Phi(a) - \int_0^\infty e^{\gamma^2/2 - a\gamma}\frac{e^{-(t-(\gamma a - \gamma^2))^2/2}}{\gamma\sqrt{2\pi}}\mathrm{d}t \\
&= \Phi(a) - e^{\gamma^2/2 - a\gamma}\Phi(a - \gamma).
\end{aligned}$$

$$\begin{aligned}
\int_0^\infty e^{t}\Phi(a - t/\gamma)\mathrm{d}t &= e^{t}\Phi(a - t/\gamma)\big|_0^\infty + \int_0^\infty e^{t}\frac{e^{-(a-t/\gamma)^2/2}}{\gamma\sqrt{2\pi}}\mathrm{d}t \\
&= -\Phi(a) + \int_0^\infty e^{\gamma^2/2 + a\gamma}\frac{e^{-(t-(a\gamma + \gamma^2))^2/2\gamma^2}}{\gamma\sqrt{2\pi}}\mathrm{d}t \\
&= -\Phi(a) + e^{\gamma^2/2 + a\gamma}\Phi(a + \gamma).
\end{aligned}$$

□

As a corollary to Theorem 3.4.1, we can bound any divergence measure that decreases under post-processing such as Renyi divergence or KL divergence. In particular, this also implies Renyi Differential Privacy [Mir17] of our algorithm.

Corollary 3.4.3. *Suppose $F, \tilde{F}$ are two μ-strongly convex functions over $\mathcal{K} \subseteq \mathbb{R}^d$, and $F - \tilde{F}$ is G-Lipschitz over $\mathcal{K}$. For any $k > 0$, if we let $P \propto e^{-kF}$ and $Q \propto e^{-k\tilde{F}}$ be two*

probability distributions on $\mathcal{K}$, then we have

$$\mathrm{D}(P\|Q) \le \mathrm{D}\left(\mathcal{N}(0,1)\|\mathcal{N}\left(\frac{G\sqrt{k}}{\sqrt{\mu}},1\right)\right)$$

for any divergence measure D *which decreases under post-processing. In particular,*

$$\mathrm{D}_\alpha(P\|Q) \le \frac{\alpha k G^2}{2\mu} \text{ and } \mathrm{D}_{KL}(P\|Q) \le \frac{kG^2}{2\mu}.$$

Proof. By Theorem 2.10 in [DRS19], if $T(P\|Q) \ge T(X\|Y)$, then there exists a randomized algorithm M such that $M(X) = P$ and $M(Y) = Q$. Therefore for any divergence measure which decreases under post-processing we have,

$$\mathrm{D}(P\|Q) = \mathrm{D}(M(X)\|M(Y)) \le \mathrm{D}(X\|Y).$$

The rest follows from Theorem 3.4.1. It is well-known that Renyi divergence and KL divergence decrease with post-processing (see [VEH14], for example). We can also compute $\mathrm{D}_\alpha(\mathcal{N}(0,1),\mathcal{N}(s,1)) = \alpha s^2/2$ and $\mathrm{D}_{KL}(\mathcal{N}(0,1),\mathcal{N}(s,1)) = s^2/2$ [Mir17]. □

3.5 Efficient Non-smooth Sampling

In this section, we will present an efficient sampling scheme for (non-smooth) functions to complement our main result first. Specifically, we study the following problem about sampling from a (non-smooth) log-concave distribution.

Problem 3.5.1. Given a μ-strongly convex function $\psi(x)$ defined on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and $+\infty$ outside. Given a family of G-Lipschitz convex functions $\{f_i(x)\}_{i\in I}$ defined on $\mathcal{K}$. Our goal is to sample a point $x \in \mathcal{K}$ with probability proportionally to $\exp(-\widehat{F}(x))$ where

$$\widehat{F}(x) = \mathop{\mathbb{E}}_{i\in I} f_i(x) + \psi(x).$$

Our sampler is based on the alternating sampling algorithm in [LST21b] (See algorithm 13). This algorithm reduces the problem of sampling from $\exp(-\widehat{F}(x))$ to sampling from $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x-y\|^2)$ for some fixed η and for roughly $\frac{1}{\eta\mu}$ many different y. When the

step size η is very small, the later problem is easier because the distribution is almost like a Gaussian distribution. For our problem, we will pick the largest step size η such that we can sample $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x-y\|^2)$ using only $\widetilde{O}(1)$ many steps.

Algorithm 13: Alternating Sampler

1 **Input:** μ-strongly convex function $\widehat{F}$, step size $\eta > 0$, initial point x_0
2 **for** $t \in [T]$ **do**
3 $\quad y_t \leftarrow x_{t-1} + \sqrt{\eta} \cdot \zeta$ where $\zeta \sim \mathcal{N}(0, I_d)$.
4 $\quad$ Sample $x_t \propto \exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x-y_t\|_2^2)$.
5 **end**
6 **Return** x_T

Theorem 3.5.2 ([LST21b, Theorem 1])**.** *Given a μ-strongly convex function F defined on $\mathcal{K}$ with an initial point x_0. Let the distance $D = \|x_0 - x^*\|_2$ for any $x^* = \arg\min_{x\in\mathcal{K}} \widehat{F}(x)$. Suppose the step size $\eta \le \frac{1}{\mu}$, the target accuracy $\delta > 0$ and the number of step $T \ge \Theta(\frac{1}{\eta\mu}\log(\frac{d/\mu + D^2}{\eta\delta}))$. Then, Algorithm 13 returns a random point x_T that has δ total variation distance to the distribution proportional to $\exp(-\widehat{F}(x))$.*

Now, we show that Line 4 in Algorithm 13 can be implemented by a simple rejection sampling. The idea is to pick step size η small enough such that $\widehat{F}(x)$ is essentially a constant function for a random $x \sim \mathcal{N}(y, \eta \cdot I_d)$. The precise algorithm is given in Algorithm 14.

Algorithm 14: Implementation of Line 4

1 **Input**: convex function $\widehat{F}(x) = \mathbb{E}_{i\in I} f_i(x) + \psi(x)$, step size $\eta > 0$, current point y
2 **repeat**
3 $\quad$ Sample x, z from the distribution $\propto \exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)$
4 $\quad$ Set $\rho \leftarrow 1$
5 $\quad$ **for** $\alpha = 1, 2, \cdots$ **do**
6 $\quad\quad \rho \leftarrow \rho + \Pi_{i=1}^{\alpha}(f_{j_i}(z) - f_{j_i}(x))$ where j_i are random indices in I
7 $\quad\quad$ With probability $\frac{\alpha}{1+\alpha}$, **break**
8 $\quad$ **end**
9 $\quad$ Sample u uniformly from $[0, 1]$.
10 **until** $u \le \frac{1}{2}\rho$;
11 **Return** x

Since F has the ψ term, instead of sampling x from $\mathcal{N}(y, \eta \cdot I_d)$, we sample from $\exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|^2)$ in Algorithm 14. The following lemma shows how to decompose the distribution $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x-y\|^2)$ into the distribution mentioned above and the distribution $\exp(-\mathbb{E}_{i\in I} f_i(x))$. It also calculates the distribution given by the algorithm.

Lemma 3.5.3. *Let π be the distribution proportional to $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x-y\|_2^2)$ and let $\mathcal{G}$ be the distribution proportional to $\exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|^2)$. Then, we have that*

$$\frac{d\pi}{dx} = \frac{d\mathcal{G}}{dx} \cdot \frac{\exp(-\mathbb{E}_{i\in I} f_i(x))}{\mathbb{E}_{x\sim\mathcal{G}} \exp(-\mathbb{E}_{i\in I} f_i(x))}.$$

Let $\widetilde{\pi}$ be the distribution returns by Algorithm 14. Then, we have that

$$\frac{d\widetilde{\pi}}{dx} = \frac{d\mathcal{G}}{dx} \cdot \frac{\mathbb{E}(\overline{\rho}|x)}{\mathbb{E}(\overline{\rho})}$$

where $\overline{\rho} = \min(\max(\rho, 0), 2)$ is the truncation of ρ in Algorithm 14 to $[0,2]$, $\mathbb{E}(\overline{\rho}|x)$ is the expected value of $\overline{\rho}$ conditional on x, and $\mathbb{E}(\overline{\rho}) = \mathbb{E}_{x\sim\mathcal{G}} \mathbb{E}(\overline{\rho}|x)$. Furthermore, we have that

$$\mathbb{E}(\rho|x) = \exp(-\mathop{\mathbb{E}}_{i\in I} f_i(x)) \cdot \mathop{\mathbb{E}}_{z\sim\mathcal{G}} \exp(\mathop{\mathbb{E}}_{i\in I} f_i(z)).$$

Proof. For the true distribution π, we have

$$\begin{aligned}\frac{d\pi}{dx} &= \frac{\exp(-\mathbb{E}_{i\in I} f_i(x) - \psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)}{\int \exp(-\mathbb{E}_{i\in I} f_i(x) - \psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)dx} \\ &= \frac{\exp(-\mathbb{E}_{i\in I} f_i(x))\frac{d\mathcal{G}}{dx}}{\int \exp(-\mathbb{E}_{i\in I} f_i(x))\frac{d\mathcal{G}}{dx}dx} = \frac{d\mathcal{G}}{dx} \cdot \frac{\exp(-\mathbb{E}_{i\in I} f_i(x))}{\mathbb{E}_{x\sim\mathcal{G}} \exp(-\mathbb{E}_{i\in I} f_i(x))}.\end{aligned}$$

For the distribution $\widetilde{\pi}$ by the algorithm, we sample $x \sim \mathcal{G}$, then accept the sample if $u \le \frac{1}{2}\rho$. Hence, we have

$$\frac{d\widetilde{\pi}}{dx} = \frac{d\mathcal{G}}{dx}\frac{\Pr(u \le \frac{1}{2}\rho|x)}{\Pr(u \le \frac{1}{2}\rho)}.$$

Since u is uniform between 0 and 1, we have the result.

Finally, for the expectation of ρ, we note that

$$\mathbb{E}\,\Pi_{i=1}^{\alpha}(f_{j_i}(z)-f_{j_i}(x))=(\mathop{\mathbb{E}}_{i\in I}(f_i(z)-f_i(x)))^{\alpha}$$

and that the probability that the loop pass step α is exactly $\frac{1}{\alpha!}$. Hence, we have

$$\mathbb{E}(\rho|x,z)=1+\sum_{\alpha=1}^{\infty}\frac{1}{\alpha!}(\mathop{\mathbb{E}}_{i\in I}(f_i(z)-f_i(x)))^{\alpha}=\exp(\mathop{\mathbb{E}}_{i\in I}(f_i(z)-f_i(x)).$$

Taking expectation over z gives the result. □

Note that if we always had $0\le\rho\le 2$, then $\mathbb{E}(\overline{\rho}|x)=\mathbb{E}(\rho|x)\propto\exp(-\mathbb{E}_{i\in I}\,f_i(x))$ and hence $\frac{d\pi}{dx}=\frac{d\widetilde{\pi}}{dx}$. Therefore, the only thing left is to show that $0\le\rho\le 2$ with high probability and that it does not induces too much error in total variation distance. To do this, we use Gaussian concentration to prove that $\mathbb{E}_{i\in I}\,f_i(x)$ is almost a constant over random $x\sim\mathcal{G}$.

Lemma 3.5.4 (Gaussian concentration [Led99, Eq 1.21])**.** *Let $X\sim\exp(-\widehat{F})$ for some $1/\eta$-strongly convex $\widehat{F}$ and ℓ is a G-Lipschitz function. Then, for all $t\ge 0$,*

$$\Pr[\ell(X)-\mathbb{E}[\ell(X)]\ge t]\le e^{-t^2/(2\eta G^2)}.$$

Now, we are already to prove our main result. This shows that if $\eta\ll G^{-2}$, then the algorithm indeed implements Line 4 correctly up to small error.

Lemma 3.5.5. *If the step size $\eta\le C\log^{-1}(1/\delta_{\text{inner}})G^{-2}$ for some small enough C and the inner accuracy $\delta_{\text{inner}}\in(0,1/2)$, then Algorithm 14 returns a random point x that has δ_{inner} total variation distance to the distribution proportional to $\exp(-\widehat{F}(x)-\frac{1}{2\eta}\|x-y\|_2^2)$. Furthermore, the algorithm accesses only $O(1)$ many $f_i(x)$ in expectation and samples from $\exp(-\psi(x)-\frac{1}{2\eta}\|x-y\|_2^2)$ for $O(1)$ many y.*

Proof. Let π be the distribution given by $c\cdot\exp(-\widehat{F}(x)-\frac{1}{2\eta}\|x-y\|_2^2)$ and $\widetilde{\pi}$ is the distribution outputted by the algorithm. By Lemma 3.5.3, we have

$$d_{\text{TV}}(\pi,\widetilde{\pi})=\int_{\mathbb{R}^d}\left|\frac{d\mathcal{G}}{dx}\frac{\exp(-\mathbb{E}_{i\in I}\,f_i(x))}{\mathbb{E}_{x\sim\mathcal{G}}\exp(-\mathbb{E}_{i\in I}\,f_i(x))}-\frac{d\mathcal{G}}{dx}\frac{\mathbb{E}(\overline{\rho}|x)}{\mathbb{E}(\overline{\rho})}\right|dx$$

$$= \mathop{\mathbb{E}}_{x\sim\mathcal{G}} \left| \frac{\exp(-\mathbb{E}_{i\in I} f_i(x))}{\mathbb{E}_{x\sim\mathcal{G}} \exp(-\mathbb{E}_{i\in I} f_i(x))} - \frac{\mathbb{E}(\overline{\rho}|x)}{\mathbb{E}(\overline{\rho})} \right|.$$

Let X be the random variable $\mathbb{E}(\rho|x)$ and $\widetilde{X}$ be the random variable $\mathbb{E}(\overline{\rho}|x)$. Lemma 3.5.3 shows that $X = \exp(-\mathbb{E}_{i\in I} f_i(x)) \cdot \mathbb{E}_{z\sim\mathcal{G}} \exp(\mathbb{E}_{i\in I} f_i(z))$ and hence

$$\frac{\exp(-\mathbb{E}_{i\in I} f_i(x))}{\mathbb{E}_{x\sim\mathcal{G}} \exp(-\mathbb{E}_{i\in I} f_i(x))} = \frac{X}{\mathbb{E}_{x\sim\mathcal{G}} X}.$$

Therefore, we have

$$d_{\mathrm{TV}}(\pi, \widetilde{\pi}) = \mathbb{E} \left| \frac{X}{\mathbb{E} X} - \frac{\widetilde{X}}{\mathbb{E} \widetilde{X}} \right| \le \mathbb{E} \left| \frac{X}{\mathbb{E} X} - \frac{\widetilde{X}}{\mathbb{E} X} \right| + \mathbb{E} \left| \frac{\widetilde{X}}{\mathbb{E} X} - \frac{\widetilde{X}}{\mathbb{E} \widetilde{X}} \right| \le 2 \frac{\mathbb{E} |X - \widetilde{X}|}{|\mathbb{E} X|}. \quad (3.11)$$

We simplify the right hand side by lower bounding $\mathbb{E} X$. By Lemma 3.5.4 and the fact that the negative log-density of $\mathcal{G}$ is $1/\eta$-strongly convex, we have that $\mathbb{E}_{i\in I} f_i(z) \ge \mathbb{E}_{x\sim\mathcal{G}} \mathbb{E}_{i\in I} f_i(x) - 2G\sqrt{\eta}$ with probability $\ge 1 - e^{-2}$. Hence, we have

$$\begin{aligned}
\mathbb{E} X &= \mathop{\mathbb{E}}_{x\sim\mathcal{G}} \exp(-\mathop{\mathbb{E}}_{i\in I} f_i(x)) \cdot \mathop{\mathbb{E}}_{z\sim\mathcal{G}} \exp(\mathop{\mathbb{E}}_{i\in I} f_i(z)) \\
&\ge \exp(-\mathop{\mathbb{E}}_{x\sim\mathcal{G}} \mathop{\mathbb{E}}_{i\in I} f_i(x)) \cdot \mathop{\mathbb{E}}_{z\sim\mathcal{G}} \exp(\mathop{\mathbb{E}}_{i\in I} f_i(z)) \\
&= \mathop{\mathbb{E}}_{z\sim\mathcal{G}} \exp(\mathop{\mathbb{E}}_{i\in I} f_i(z) - \mathop{\mathbb{E}}_{x\sim\mathcal{G}} \mathop{\mathbb{E}}_{i\in I} f_i(x)) \\
&\ge (1 - e^{-2}) \exp(-2G\sqrt{\eta}).
\end{aligned}$$

Using $\eta \le G^{-2}/8$, we have $\mathbb{E}[X] \ge \frac{2}{3}$. Using this, (3.11), $X = \mathbb{E}(\rho|x)$ and $\widetilde{X} = \mathbb{E}(\overline{\rho}|x)$, we have

$$d_{\mathrm{TV}}(\pi, \widetilde{\pi}) \le 3 \cdot \mathbb{E} |X - \widetilde{X}| \le 3 \cdot \mathbb{E}(|\rho| \cdot 1_{\rho\notin[0,2]}).$$

We split the ρ into two terms $\rho_{\le L}$ and $\rho_{>L}$. The first term $\rho_{\le L}$ is the sum of all terms added to ρ when $\alpha \le L$ (including the initial term 1). The second term $\rho_{>L}$ is the sum when $\alpha > L$. Hence, we have $\rho = \rho_{>L} + \rho_{\le L}$ and hence

$$d_{\mathrm{TV}}(\pi, \widetilde{\pi}) \le 3 \cdot \mathbb{E}(|\rho_{>L}| \cdot 1_{\rho\notin[0,2]}) + 3 \cdot \mathbb{E}(|\rho_{\le L}| \cdot 1_{\rho\notin[0,2]}). \quad (3.12)$$

For the term $\rho_{>L}$, by a calculation similar to Lemma 3.5.3, we have

$$\mathbb{E}(|\rho_{>L}| \cdot 1_{\rho \notin [0,2]}) \le \mathbb{E}\,|\rho_{>L}| \le \mathop{\mathbb{E}}_{x,z} \Phi(\mathop{\mathbb{E}}_{i\in I} |f_i(z) - f_i(x)|),$$

where $\Phi(t) = \sum_{\alpha=L+1}^{\infty} \frac{t^\alpha}{\alpha!}$ is a power series in t with all positive coefficients. By picking $L > C\log(1/\delta_{\text{inner}})$ for some large constant C, we have $\Phi(t) \le \frac{\delta_{\text{inner}}}{16}$ for all $|t| \le 1$. Let Δ be the random variable $\mathbb{E}_{i\in I} |f_i(z) - f_i(x)|$ whose randomness comes from x and z. Then, we have

$$\mathbb{E}(|\rho_{>L}| \cdot 1_{\rho \notin [0,2]}) \le \frac{\delta_{\text{inner}}}{16} + \mathbb{E}\, e^{\Delta} 1_{\Delta \ge 1} \le \frac{\delta_{\text{inner}}}{16} + \sum_{k=1}^{\infty} e^{k+1} \Pr_{x,z}(\Delta \ge k).$$

Denote a function $h_{x,z}(t) := \Pr_{i\in I}[|f_i(z) - f_i(x)| \ge t]$. Since each f_i is G-Lipschitz, Lemma 3.5.4 shows that

$$\Pr_{x,z}[|f_i(z) - f_i(x)| \ge t] \le 4e^{-t^2/(8\eta G^2)},$$

which implies

$$\mathop{\mathbb{E}}_{x,z}[h_{x,z}(t)] = \Pr_{x,z,i}[|f_i(z) - f_i(x)| \ge t] \le 4e^{-t^2/(8\eta G^2)}.$$

By Markov inequality, for any $k > 0$, we know

$$\Pr_{x,z}[h_{x,z}(t) \ge e^{-k}] \le 4e^{k - t^2/(8\eta G^2)}.$$

As $|f_i(z) - f_i(x)| \le G\|x - z\|_2$, if $h_{x,z}(t) = \Pr_{i\in I}[|f_i(z) - f_i(x)| \ge t] \le e^{-t^2/(16\eta G^2)}$, we know

$$\mathop{\mathbb{E}}_{i\in I} |f_i(z) - f_i(x)| \le t + e^{-t^2/(16\eta G^2)} \cdot G\|x - z\|_2.$$

Hence, one has

$$\Pr_{x,z}\left[\mathop{\mathbb{E}}_{i\in I} |f_i(z) - f_i(x)| \ge t + e^{-t^2/(16\eta G^2)} G\|x - z\|_2\right] \le \Pr_{x,z}[h_{x,z}(t) \ge e^{-t^2/(16\eta G^2)}]$$

$$\leq 4e^{-t^2/(16\eta G^2)}.$$

By Gaussian Concentration, we know

$$\begin{aligned}\Pr_{x,z}[\|x-z\|_2 \geq t] \leq & \Pr_{x,z}[\|x-\mathbb{E}\,x\|_2 \geq t/2 \text{ or } \|z-\mathbb{E}\,z\| \geq t/2] \\ & \leq 2e^{-t^2/(8\eta)}.\end{aligned}$$

Thus we know

$$\begin{aligned}&\Pr_{x,z}[\mathop{\mathbb{E}}_{i\in I}|f_i(z)-f_i(x)| \geq 2t] \\ =\ &\Pr_{x,z}[\mathop{\mathbb{E}}_{i\in I}|f_i(z)-f_i(x)| \geq 2t, \|x-z\|_2 \geq t/G] + \Pr_{x,z}[\mathop{\mathbb{E}}_{i\in I}|f_i(z)-f_i(x)| \geq 2t, \|x-z\|_2 < t/G] \\ \leq\ &2e^{-t^2/(8G^2\eta)} + \Pr_{x,z}[\mathop{\mathbb{E}}_{i\in I}|f_i(z)-f_i(x)| \geq 2t, \|x-z\|_2 < t/G] \\ \leq\ &2e^{-t^2/(8G^2\eta)} + \Pr_{x,z}[\mathop{\mathbb{E}}_{i\in I}|f_i(z)-f_i(x)| \geq t + e^{-t^2/(16\eta G^2)}G\|x-z\|_2] \\ \leq\ &6e^{-t^2/(16\eta G^2)}.\end{aligned}$$

Hence, we have $\Pr(\Delta \geq k) \leq 6\exp(-k^2/(64G^2\eta))$ and

$$\mathbb{E}(|\rho_{>L}| \cdot 1_{\rho\notin[0,2]}) \leq \frac{\delta_{\text{inner}}}{16} + 17\sum_{k=1}^{\infty} e^{k-\frac{k^2}{64G^2\eta}} \leq \frac{\delta_{\text{inner}}}{9}, \tag{3.13}$$

where we used $\eta \leq 2^{-6}G^{-2}/\log(400/\delta_{\text{inner}})$ at the end.

As for the term $\rho_{\leq L}$, we know that

$$\begin{aligned}&\mathbb{E}(|\rho_{\leq L}| \cdot 1_{\rho\notin[0,2]}) \\ =&\,\mathbb{E}(|\rho_{\leq L}| \cdot 1_{\rho\notin[0,2]} \cdot 1_{|\rho_{\leq L}|\leq 2^L}) + \mathbb{E}(|\rho_{\leq L}| \cdot 1_{\rho\notin[0,2]} \cdot 1_{|\rho_{\leq L}|\geq 2^L}) \\ \leq&\,\Pr[\rho \notin [0,2]] \cdot 2^L + \sum_{k=1}^{\infty} 2^{(k+1)L}\Pr(|\rho_{\leq L}| \geq 2^{kL}).\end{aligned} \tag{3.14}$$

Note that the term $\rho_{\leq L}$ involves only less than $\frac{L^2}{2}$ many $f_i(x)$ and $f_i(z)$. Lemma 3.5.4

shows that for any i, we have

$$\Pr_{x\sim\mathcal{G}}(|f_i(x) - \mathop{\mathbb{E}}_{x\sim\mathcal{G}} f_i(x)| \geq t) \leq 2e^{-t^2/(2\eta G^2)}.$$

By union bound, this shows

$$\Pr_{x,z\sim\mathcal{G}}(|f_i(x) - f_i(z)| \geq \frac{1}{4}2^k \text{ for any such } i) \leq L^2 \exp(-\frac{4^k}{32\eta G^2}).$$

Under the event $|f_i(x) - f_i(z)| \leq \frac{1}{3}2^k$ for all i appears in $\rho_{\leq L}$, we have

$$|\rho_{\leq L}| \leq 1 + \sum_{\alpha=1}^{L} \Pi_{i=1}^{\alpha} |f_{j_{i,\alpha}}(z) - f_{j_{i,\alpha}}(x)| \leq 1 + \sum_{\alpha=1}^{L} (\frac{2^k}{3})^\alpha \leq 2^{kL}.$$

Therefore, we have $\Pr(|\rho_{\leq L}| > 2^{kL}) \leq L^2 \exp(-\frac{4^k}{32\eta G^2})$ and

$$\sum_{k=1}^{\infty} 2^{(k+1)L} \Pr(|\rho_{\leq L}| > 2^{kL}) \leq \sum_{k=1}^{\infty} 2^{(k+1)L} L^2 \exp(-\frac{4^k}{32\eta G^2}) \leq \sum_{k=1}^{\infty} 2^{4kL} \exp(-\frac{4^k}{32\eta G^2}).$$

Picking $\eta \leq 2^{-8} G^{-2} L^{-1}$, we have that

$$\sum_{k=1}^{\infty} 2^{(k+1)L} \Pr(|\rho_{\leq L}| > 2^{kL}) \leq \sum_{k=1}^{\infty} 2^{4kL} \exp(-2 \cdot 4^k L) \leq \sum_{k=1}^{\infty} 2^{-kL} \leq \frac{\delta_{\text{inner}}}{9} \tag{3.15}$$

by picking $L > C \log(1/\delta_{\text{inner}})$ for large enough C.

It remains to bound the term $\Pr[\rho \notin [0,2]] \cdot 2^L$. We know the probability the algorithm enters the $(L+1)$-th phase is at most $\frac{1}{L!}$. Hence we know $\Pr[\rho \notin [0,2]] \leq \frac{1}{L!} + \Pr[\rho_{\leq L} \notin [0,2]]$. Similarly, by Gaussian Concentration and union bound, we have

$$\Pr_{x,z\sim\mathcal{G}}(|f_i(x) - f_i(z)| \geq 1/2 \text{ for any such } i) \leq L^2 \exp(-\frac{1}{8\eta G^2}).$$

Under the event that $|f_i(x) - f_i(z)| \leq 1/2$ for all i appears in $\rho_{\leq L}$, we have

$$1 - \sum_{\alpha=1}^{L} \Pi_{i=1}^{\alpha} |f_{j_{i,\alpha}}(z) - f_{j_{i,\alpha}}(x)| \leq \rho_{\leq L} \leq 1 + \sum_{\alpha=1}^{L} \Pi_{i=1}^{\alpha} |f_{j_{i,\alpha}}(z) - f_{j_{i,\alpha}}(x)|,$$

which implies $0 \le \rho_{\le L} \le 2$. Then we know $\Pr[\rho_{\le L} \notin [0,2]] \le L^2 \exp(-\frac{1}{8\eta G^2})$. By our setting of parameters and that $L = C\log(1/\delta_{\text{inner}})$ for some large constant C, we know

$$\Pr[\rho \notin [0,2]] \cdot 2^L \le 2^L (L^2 \exp(-\frac{1}{8\eta G^2}) + \frac{1}{L!}) \le \frac{\delta_{\text{inner}}}{9}. \tag{3.16}$$

Combining (3.12), (3.13), (3.14), (3.15) and (3.16), we have the result $d_{\text{TV}}(\pi, \widetilde{\pi}) \le \delta_{\text{inner}}$.

Finally, the accept probability is given by $\mathbb{E}\,\widetilde{X}/2$ and $\mathbb{E}\,\widetilde{X} \ge \mathbb{E}\,X - \mathbb{E}\,|X - \widetilde{X}| \ge \frac{2}{3} - \frac{\delta_{\text{inner}}}{3} \ge \frac{1}{3}$. Hence, the number of access is $O(1)$. □

Combining Theorem 3.5.2 and Lemma 3.5.5, we have the following result:

Theorem 3.5.6. *Given a μ-strongly convex function $\psi(x)$ defined on a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ and $+\infty$ outside. Given a family of G-Lipschitz convex functions $\{f_i(x)\}_{i\in I}$ defined on $\mathcal{K}$. Define the function $\widehat{F}(x) = \mathbb{E}_{i\in I}\, f_i(x) + \psi(x)$ and the distance $D = \|x_0 - x^*\|_2$ for some $x^* = \arg\min_x \widehat{F}(x)$. For any $\delta \in (0, 1/2)$, if we can get samples from $\exp(-\psi(x) - \frac{\|x-y\|_2^2}{2\eta})$ for any $y \in \mathbb{R}^d$ and $\eta > 0$, we can find a random point x that has δ total variation distance to the distribution proportional to $\exp(-\widehat{F}(x))$ in*

$$T := \Theta(\frac{G^2}{\mu}\log^2(\frac{G^2(d/\mu + D^2)}{\delta}))\ \textit{steps.}$$

Furthermore, each steps accesses only $O(1)$ many $f_i(x)$ in expectation and samples from $\exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)$ for $O(1)$ many y with $\eta = \Theta(G^{-2}/\log(T/\delta))$.

Proof. This follows from applying Lemma 3.5.5 to implement Line 4. Note that the distribution implemented has total variation distance δ_{inner} to the required one. By setting $\delta_{\text{inner}} = \delta/(2T)$, this only gives an extra $\delta/2$ error in total variation distance. Finally, setting $\eta = \Theta(G^{-2}/\log(1/\delta_{\text{inner}}))$, Theorem 3.5.2 shows that Algorithm 14 outputs the correct distribution up to $\delta/2$ error in total variation distance. This gives the result. □

In the most important case of interest when $\psi(x)$ is ℓ_2^2 regularizer, one can see $\exp(-\psi(x) - \frac{1}{2\eta}\|x-y\|_2^2)$ is a truncated Gaussian distribution, and there are many results on how to sample from truncated Gaussian, e.g. [KD99]. For more general case, there are also efficient algorithms to do the sampling, such as the Projected Langevin Monte Carlo [BEL18]. In

fact our sampling scheme matches the information-theoretical lower bound on the value query complexity up to some logarithmic terms, which can be reduced from the result in [DJWW15] with some modifications. See Section 3.7 for a detailed discussion.

3.6 DP Convex Optimization

In this section we present our results about DP-ERM and DP-SCO.

3.6.1 DP-ERM

In this subsection, we state our result for the DP-ERM problem (3.3). Briefly speaking, our main result (Theorem 3.2.1) shows that sampling from $\exp(-kF(x;\mathcal{D}))$ for some appropriately chosen k is (ε,δ)-DP and achieves the optimal empirical risk in (3.4). Our sampling scheme in Section 3.5 provides an efficient implementation. We start with the following lemma which shows the utility guarantee for the sampling mechanism.

Lemma 3.6.1 (Utility Guarantee, [DKL18, Corollary 1])**.** *Suppose $k > 0$ and F is a convex function over the convex set $\mathcal{K} \subseteq \mathbb{R}^d$. If we sample x according to distribution ν whose density is proportional to $\exp(-kF(x))$, then we have*

$$\mathop{\mathbb{E}}_{\nu}[F(x)] \le \min_{x\in\mathcal{K}} F(x) + \frac{d}{k}.$$

This is first shown by [KV06] for any linear function F, and [BST14] extends it to any convex function F with a slightly worse constant.

Theorem 3.6.2 (DP-ERM)**.** *Let $\varepsilon > 0$, $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set of diameter D and $\{f(\cdot;s)\}_{s\in\mathcal{D}}$ be a family of convex functions over $\mathcal{K}$ such that $f(x;s)-f(x;s')$ is G-Lipschitz for all s, s'. For any data-set $\mathcal{D}$ and $k > 0$, sampling $x^{(priv)}$ with probability proportional to $\exp\left(-k(F(x;\mathcal{D}) + \mu\|x\|_2^2/2)\right)$ is $(\varepsilon,\delta(\varepsilon))$-differentially private, where*

$$\delta(\varepsilon) \le \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G\sqrt{k}}{n\sqrt{\mu}}, 1\right)\right)(\varepsilon).$$

The excess empirical risk is bounded by $\frac{d}{k} + \frac{\mu D^2}{2}$. Moreover, if $\{f(\cdot,s)\}_{s\in\mathcal{D}}$ are already

μ-strongly convex, then sampling $x^{(priv)}$ with probability proportional to $\exp(-kF(x;\mathcal{D}))$ is $(\varepsilon, \delta(\varepsilon))$-differentially private where

$$\delta(\varepsilon) \leq \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G\sqrt{k}}{n\sqrt{\mu}}, 1\right)\right)(\varepsilon).$$

The excess empirical risk is bounded by $\frac{d}{k}$.

Proof. The privacy guarantee follows directly from our main result Theorem 3.2.1, and the bound on excess empirical loss can be proved by Lemma 3.6.1. □

Before we state the implementation results on DP-ERM, we need the following technical lemma:

Lemma 3.6.3. *For any constants $1/2 > \delta > 0$ and $\varepsilon > 0$, if $|s| \leq \sqrt{2\log(1/(2\delta)) + 2\varepsilon} - \sqrt{2\log(1/(2\delta))}$, one has*

$$\delta(\mathcal{N}(0,1) \,\|\, \mathcal{N}(s,1)) \leq \delta.$$

Proof. By Equation (3.9), we know that

$$\delta(\mathcal{N}(0,1) \,\|\, \mathcal{N}(s,1))(\varepsilon) \leq \Phi\left(-\frac{\varepsilon}{s} + \frac{s}{2}\right).$$

Without loss of generality, we assume $s \geq 0$ and want to find an appropriate value of s such that $\Phi\left(-\frac{\varepsilon}{s} + \frac{s}{2}\right) \leq \delta$. Denote $t := \Phi^{-1}(1-\delta)$ and since $1 - \Phi(t) \leq \frac{1}{2}\exp(-t^2/2)$ for $t > 0$, we know that $t \leq \sqrt{2\log(1/(2\delta))}$. It is equivalent to solve the equation $\frac{\varepsilon}{s} - \frac{s}{2} \geq t$, which is equivalent to $0 \leq s \leq \sqrt{t^2 + 2\varepsilon} - t$. Note that $\sqrt{t^2 + 2\varepsilon} - t$ decreases as t increases, which implies that we can set $s \leq \sqrt{2\log(1/(2\delta)) + 2\varepsilon} - \sqrt{2\log(1/(2\delta))}$. □

Combining the sampling scheme (Theorem 3.5.6) and our analysis on DP-ERM, we can get the efficient implementation results on DP-ERM directly.

Theorem 3.6.4 (DP-ERM Implementation)**.** *With same assumptions in Theorem 3.6.2, and assume $f(\cdot; s)$ is G-Lipschitz over $\mathcal{K}$ for all s. For any constants $1/10 > \delta > 0$ and $\varepsilon > 0$, there is an efficient sampler to solve DP-ERM which has the following guarantees:*

- *The scheme is (ε, δ)-differentially private;*

- *The expected excess empirical loss is bounded by $\frac{GD\sqrt{d}}{n(\sqrt{\log(1/\delta)+\varepsilon}-\sqrt{\log(1/\delta)})}$. In particular, if $\varepsilon < 1/10$, the expected excess empirical loss is bounded by $\frac{2GD\sqrt{d\log(1/\delta)}}{\varepsilon n}$. If $\varepsilon \geq \log(1/\delta)$, the expected excess empirical loss is bounded by $O(\frac{GD\sqrt{d}}{n\sqrt{\varepsilon}})$.*

- *The scheme takes*

$$\Theta\left(\frac{\varepsilon^2 n^2}{\log(1/\delta)}\log^2(\frac{nd\varepsilon}{\delta})\right)$$

queries to the values on $f(x;s)$ in expectation and takes the same number of samples from some Gaussian restricted to the convex set $\mathcal{K}$.

Proof. By Lemma 3.6.3, we can set $s = \sqrt{2\log(3/(4\delta)) + 2\varepsilon} - \sqrt{2\log(3/(4\delta))}$ to make $\delta(\mathcal{N}(0,1) \,\|\, \mathcal{N}(s,1)) \leq 2\delta/3$. For our setting, Theorem 3.6.2 shows that we have $s = \frac{G\sqrt{k}}{n\sqrt{\mu}}$ and hence we can take

$$k = \frac{2\mu n^2\left(\sqrt{\log(3/(4\delta))+\varepsilon} - \sqrt{\log(3/(4\delta))}\right)^2}{G^2}.$$

Putting it into the excess empirical loss bound of $\frac{d}{k} + \frac{\mu D^2}{2}$ and setting $\mu = \frac{G\sqrt{d}}{nD\left(\sqrt{\log(3/(4\delta))+\varepsilon}-\sqrt{\log(3/(4\delta))}\right)}$, we get the result on the empirical loss.

Particularly, consider the case when $\varepsilon < 1/10$. We know the excess empirical loss is bounded by $\frac{GD\sqrt{d}}{n(\sqrt{\log(3/(4\delta))+\varepsilon}-\sqrt{\log(3/(4\delta))})}$. Note that $1 + \frac{x}{2} - \frac{x^2}{8} \leq \sqrt{1+x} \leq 1 + \frac{x}{2}$ for $x \geq 0$. Under the assumption that $\delta, \varepsilon \in (0, \frac{1}{10})$, we know $\frac{GD\sqrt{d}}{n(\sqrt{\log(3/(4\delta))+\varepsilon}-\sqrt{\log(3/(4\delta))})} \leq \frac{2GD\sqrt{d\log(4/(5\delta))}}{n\varepsilon}$. The case when $\varepsilon \geq \log(1/\delta)$ also follows similarly.

To make it algorithmic, we apply Theorem 3.5.6 with the accuracy on the total variation distance to be $\min\{\delta/3, \frac{1}{cn^c\varepsilon}\}$ for some large enough constant c. This leads to (ε, δ)-DP and an extra empirical loss and hence we use $\log(1/\delta)$ rather than $\log(3/(4\delta))$ or $\log(4/(5\delta))$ in the final loss term.

The running time follows from Theorem 3.5.6. □

3.6.2 DP-SCO and Generalization Error

As mentioned before, one can reduce the DP-SCO (3.5) to DP-ERM (3.3) by the iterative localization technique proposed by [FKT20]. But this method forces us to design different algorithms for DP-ERM and DP-SCO, and may lead to a large constant in the final loss. In this section, we show that the exponential mechanism can achieve both the optimal empirical risk for DP-ERM and the optimal population loss for DP-SCO by simply changing the parameters. The bound on the generalization error works beyond differential privacy and can be useful for other (non-private) optimization settings.

The proof will make use of one famous inequality: *Talagrand transportation inequality*. Recall for two probability distributions ν_1, ν_2, the Wasserstein distance is equivalently defined as

$$W_2(\nu_1, \nu_2) = \inf_{\Gamma} \left(\mathop{\mathbb{E}}_{(x_1,x_2)\sim\Gamma} \|x_1 - x_2\|_2^2 \right)^{1/2},$$

where the infimum is over all couplings Γ of ν_1, ν_2.

Theorem 3.6.5 (Talagrand transportation inequality)**.** *[OV00, Theorem 1] Let* $\mathrm{d}\pi \propto e^{-F(x)}\mathrm{d}x$ *be a* μ*-strongly log-concave probability measure on* $\mathcal{K} \subseteq \mathbb{R}^d$ *with finite moments of order 2. For all probability measure* ν *absolutely continuous w.r.t.* π *and with finite moments of order 2, we have*

$$\mathrm{W}_2(\nu, \pi) \leq \sqrt{\frac{2}{\mu}\mathrm{D}_{KL}(\nu, \pi)}.$$

To prove our main result on bounding the generalization error of sampling mechanism, we need the following lemma.

Lemma 3.6.6 (Lemma 7 in [BE02])**.** *For any learning algorithm* $\mathcal{A}$ *and dataset* $\mathcal{D} = \{s_1, \cdots, s_n\}$ *drawn i.i.d from the underlying distribution* $\mathcal{P}$*, let* $\mathcal{D}'$ *be a neighboring dataset formed by replacing a random element of* $\mathcal{D}$ *with a freshly sampled* $s' \sim \mathcal{P}$*. If* $\mathcal{A}(\mathcal{D})$ *is the output of* $\mathcal{A}$ *with* $\mathcal{D}$*, then*

$$\mathop{\mathbb{E}}_{\mathcal{D}}[\widehat{F}(\mathcal{A}(\mathcal{D})) - F(\mathcal{A}(\mathcal{D}); \mathcal{D})] = \mathop{\mathbb{E}}_{\mathcal{D}, s'\sim\mathcal{P}, \mathcal{A}} \Big[f(\mathcal{A}(\mathcal{D}); s') - f(\mathcal{A}(\mathcal{D}'); s') \Big].$$

Now we begin to state and prove our main result on the generalization error.

Theorem 3.6.7. *Suppose $\{f(\cdot, s)\}$ is a family μ-strongly convex functions over $\mathcal{K}$ such that $f(x; s) - f(x; s')$ is G-Lipschitz for all s, s'. For any $k > 0$ and dataset $\mathcal{D} = \{s_1, s_2, \cdots, s_n\}$ drawn i.i.d from the underlying distribution $\mathcal{P}$, let $\mathcal{D}'$ be a neighboring dataset formed by replacing a random element of $\mathcal{D}$ with a freshly sampled $s' \sim \mathcal{P}$,*

$$\mathrm{W}_2(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'}) \leq \frac{G}{n\mu}.$$

If we sample our solution from density $\pi_{\mathcal{D}}(x) \propto e^{-kF(x;\mathcal{D})}$, we can bound the excess population loss as:

$$\mathop{\mathbb{E}}_{\mathcal{D}, x\sim\pi_{\mathcal{D}}}[\widehat{F}(x)] - \min_{x\in\mathcal{K}} \widehat{F}(x) \leq \frac{G^2}{\mu n} + \frac{d}{k}.$$

Proof. Recall that

$$F(x; \mathcal{D}) = \frac{1}{n} \sum_{s_i \in \mathcal{D}} f(x; s_i).$$

We form a neighboring data set $\mathcal{D}'$ by replacing a random element of $\mathcal{D}$ by a freshly sampled $s' \sim \mathcal{P}$. Let $\pi_{\mathcal{D}} \propto e^{-kF(x;\mathcal{D})}$ and $\pi_{\mathcal{D}'} \propto e^{-kF(x;\mathcal{D}')}$. By Corollary 3.4.3, we have

$$\mathrm{D}_{KL}(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'}) \leq \frac{G^2 k}{2n^2\mu}.$$

By the assumptions, we know both $F(x; \mathcal{D})$ and $F(x; \mathcal{D}')$ are μ-strongly convex and by Theorem 3.6.5, we have

$$\mathrm{W}_2(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'}) \leq \sqrt{\frac{2}{k\mu}\mathrm{D}_{KL}(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'})} \leq \frac{G}{n\mu}.$$

By Lemma 3.6.6 and properties of Wasserstein distance, we have

$$\mathop{\mathbb{E}}_{\mathcal{D}, x\sim\pi_{\mathcal{D}}}[\widehat{F}(x) - F(x; \mathcal{D})] = \mathop{\mathbb{E}}_{\mathcal{D}, s'\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{x\sim\pi_{\mathcal{D}}} f(x; s') - \mathop{\mathbb{E}}_{x'\sim\pi_{\mathcal{D}'}} f(x'; s')\right]$$

$$= \mathop{\mathbb{E}}_{\mathcal{D},s'\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{x\sim\pi_{\mathcal{D}}}\left[f(x;s')-f(x;s'')\right]-\mathop{\mathbb{E}}_{x'\sim\pi_{\mathcal{D}'}}\left[f(x';s')-f(x';s'')\right]\right]$$

(where s'' is chosen arbitrarily, note that $\mathbb{E}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[f(x;s'')] = \mathbb{E}_{\mathcal{D}',x'\sim\pi_{\mathcal{D}'}}[f(x';s'')]$)

$$\leq G\cdot \mathrm{W}_2(\pi_{\mathcal{D}},\pi_{\mathcal{D}'}) \qquad (f(x;s')-f(x;s'') \text{ is } G\text{-Lipschitz})$$

$$\leq \frac{G^2}{n\mu}.$$

Hence, we know that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[\widehat{F}(x)] - \min_{x\in\mathcal{K}}\widehat{F}(x) &\leq \mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[\widehat{F}(x)] - \mathop{\mathbb{E}}_{\mathcal{D}}[\min_{x\in\mathcal{K}}F(x;\mathcal{D})] \\
&\leq \mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[\widehat{F}(x)-F(x;\mathcal{D})] + \mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[F(x;\mathcal{D})-\min_{x\in\mathcal{K}}F(x;\mathcal{D})] \\
&\leq \frac{G^2}{n\mu} + \mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[F(x;\mathcal{D})-\min_{x\in\mathcal{K}}F(x;\mathcal{D})] \\
&\leq \frac{G^2}{n\mu} + \frac{d}{k},
\end{aligned}$$

where the last inequality follows from Lemma 3.6.1. □

With the bounds on generalization error, we can get our first result on DP-SCO.

Theorem 3.6.8 (DP-SCO)**.** *Let $\varepsilon > 0$, $\mathcal{K}\subseteq\mathbb{R}^d$ be a convex set of diameter D and $\{f(\cdot;s)\}_{s\in\mathcal{D}}$ be a family of convex functions over $\mathcal{K}$ such that $f(x;s)-f(x;s')$ is G-Lipschitz for all s,s'. For any data-set $\mathcal{D}$ and $k>0$, sampling $x^{(priv)}$ with probability proportional to $\exp\left(-k(F(x;\mathcal{D})+\mu\|x\|_2^2/2)\right)$ is $(\varepsilon,\delta(\varepsilon))$-differentially private, where*

$$\delta(\varepsilon) \leq \delta\left(\mathcal{N}(0,1)\,\middle\|\,\mathcal{N}\left(\frac{G\sqrt{k}}{n\sqrt{\mu}},1\right)\right)(\varepsilon).$$

If users in the data-set $\mathcal{D}$ are drawn i.i.d. from the underlying distribution $\mathcal{P}$, the excess population loss is bounded by $\frac{G}{n\mu}+\frac{d}{k}+\frac{\mu D^2}{2}$. Moreover, if $\{f(\cdot;s)\}_{s\in\mathcal{D}}$ are already μ-strongly convex, then sampling $x^{(priv)}$ with probability proportional to $\exp(-kF(x;\mathcal{D}))$ is $(\varepsilon,\delta(\varepsilon))$-differentially private where

$$\delta(\varepsilon) \leq \delta\left(\mathcal{N}(0,1)\,\middle\|\,\mathcal{N}\left(\frac{G\sqrt{k}}{n\sqrt{\mu}},1\right)\right)(\varepsilon).$$

The excess population loss is bounded by $\frac{G}{n\mu} + \frac{d}{k}$.

Proof. The first part about privacy is a restatement of our result on DP-ERM (Theorem 3.6.4). The excess population loss (See Equation (3.6)) follows from the bound on generalization error (Theorem 3.6.7) and utility guarantee (Lemma 3.6.1). □

We give an implementation result of our DP-SCO result.

Theorem 3.6.9 (DP-SCO Implementation)**.** *With same assumptions in Theorem 3.6.8, and assume* $f(\cdot; s)$ *is* G*-Lipschitz over* $\mathcal{K}$ *for all* s*. For* $0 < \delta < \frac{1}{10}$ *and* $0 < \varepsilon < \frac{1}{10}$*, there is an efficient algorithm to solve DP-SCO which has the following guarantees:*

- *The algorithm is* (ε, δ)*-differentially private;*
- *The expected population loss is bounded by*

$$GD\left(\frac{2\sqrt{\log(1/\delta)d}}{\varepsilon n} + \frac{2}{\sqrt{n}}\right),$$

  where $c > 0$ *is an arbitrary constant to be chosen.*

- *The algorithm takes*

$$O\left(\min\left\{\frac{\varepsilon^2 n^2}{\log(1/\delta)}, nd\right\}\log^2\left(\frac{\varepsilon nd}{\delta}\right)\right)$$

  queries of the values of $f(\cdot, s_i)$ *in expectation and takes the same number of samples from some Gaussian restricted to the convex set* $\mathcal{K}$*.*

Remark 3.6.10. As for the non-typical case when $\varepsilon \geq 1/10$, one can use the bound in Theorem 3.6.4 and the bound on generalization error (Theorem 3.6.7) . Particularly, one can achieve expected population loss $O\left(GD\left(\frac{\sqrt{d}/n}{\sqrt{\log(1/\delta)+\varepsilon}-\sqrt{\log(1/\delta)}} + \frac{1}{\sqrt{n}}\right)\right)$.

Proof. By Theorem 3.6.8, sampling from $\exp(-k(F(x;\mathcal{D})+\mu\|x\|_2^2/2))$ when $k \leq \frac{\varepsilon^2 n^2 \mu}{2G^2\log(3/(4\delta))}$ is $(\varepsilon, 2\delta/3)$-DP. Besides, we can set $k = \frac{\mu}{G^2}\min\{\frac{\varepsilon^2 n^2}{2\log(3/(4\delta))}, 2nd\}$ for arbitrarily large constant $c > 0$ to make the mechanism $(\varepsilon, 2\delta/3)$-differentially private, achieving tight population loss and decrease the running time. Then the population loss is upper bounded

by

$$\frac{d}{k}+\frac{\mu D^2}{2}+\frac{G^2}{\mu n}=\frac{G^2}{\mu}\max\left\{\frac{2\log(3/(4\delta))d}{\varepsilon^2n^2},\frac{1}{2n}\right\}+\frac{\mu D^2}{2}+\frac{G^2}{\mu n}.$$

By setting $\mu=\frac{G}{D}\sqrt{2(\frac{2\log(3/(4\delta))d}{\varepsilon^2n^2}+\frac{1}{2n})}$, the population loss is upper bounded by

$$GD\sqrt{\frac{4\log(3/(4\delta))d}{\varepsilon^2n^2}+\frac{1}{n}}+GD\sqrt{\frac{1}{n}}\leq GD\left(\frac{2\sqrt{\log(3/(4\delta))d}}{\varepsilon n}+\frac{2}{\sqrt{n}}\right).$$

To make it algorithmic, we also apply Theorem 3.5.6 with the accuracy on the total variation distance to be $\min\{\delta/3,\frac{1}{cn^c}\}$ for some large enough constant c. This leads to an extra empirical loss and hence we use $\log(1/\delta)$ rather than $\log(3/(4\delta))$ in the final loss term. The runtime follows from Theorem 3.5.6. □

3.7 Information-theoretic Lower Bound for DP-SCO

In this section, we prove an information-theoretic lower bound for the query complexity required for DP-SCO (with value queries), which matches (up to some logarithmic terms) the query complexity achieved by our algorithm (in Theorem 3.6.9). Our proof is similar to the previous works like [ACCD12, DJWW15] with some modifications.

Before stating the lower bound, we define some notations. Recall that we are given a set $\mathcal{D}$ of n samples (users) $\{s_1,\cdots,s_n\}$. Let $\mathbb{A}_k$ be the collection of all algorithms that observe a sequence of k data points $(Y^1,\cdots,Y^k)$ with $Y^t=f(X^t;S^t)$ where $S^t\in\mathcal{D}$ and $X^t\in\mathcal{K}$ are chosen arbitrarily and adaptively by the algorithm (and possibly using some randomness).

For the lower bound, we only consider linear functions, that is we define $f(x;s):=\langle x,s\rangle$. And let $\mathcal{P}_G$ be the collection of all distributions such that if $\mathcal{P}\in\mathcal{P}_G$, then $\mathbb{E}_{s\sim\mathcal{P}}\|s\|_2^2\leq G^2$.

And we define the optimality gap

$$\varepsilon_k(\mathcal{A},\mathcal{P},\mathcal{K}):=\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{A}}[\widehat{F}(\widehat{x}(\mathcal{D}))]-\inf_{x\in\mathcal{K}}\widehat{F}(x),$$

where $\widehat{F}(x)=\mathbb{E}_{s\sim\mathcal{P}}f(x;s)$, $\widehat{x}$ is the output the algorithm $\mathcal{A}$ given the input dataset $\mathcal{D}$ and the expectation is over the dataset $\mathcal{D}\sim\mathcal{P}^n$ and the randomness of the algorithm $\mathcal{A}$. Note

that we can rewrite the optimality gap as:

$$\begin{aligned}\varepsilon_k(\mathcal{A},\mathcal{P},\mathcal{K}) &= \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{A}}[\widehat{F}(\widehat{x}(\mathcal{D}))] - \inf_{x\in\mathcal{K}}\widehat{F}(x)\\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{A}} f(\widehat{x}(\mathcal{D});s)]\right] - \inf_{x\in\mathcal{K}}\mathop{\mathbb{E}}_{s\sim\mathcal{P}}[f(x;s)]\\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{P},\mathcal{D}\sim\mathcal{P}^n,\mathcal{A}}[\widehat{x}(\mathcal{D})^\top s] - \inf_{x\in\mathcal{K}}\mathop{\mathbb{E}}_{s\sim\mathcal{P}}[x^\top s].\end{aligned}$$

The minimax error is defined by

$$\varepsilon_k^*(\mathcal{P}_G,\mathcal{K}) := \inf_{\mathcal{A}\in\mathbb{A}_k}\sup_{\mathcal{P}\in\mathcal{P}_G}\varepsilon_k(\mathcal{A},\mathcal{P},\mathcal{K}).$$

Theorem 3.7.1. *Let $\mathcal{K}$ be the ℓ_2 ball of diameter D in $\mathbb{R}^d$, then*

$$\varepsilon_k^*(\mathcal{P}_G,\mathcal{K}) \geq \frac{GD}{16}\min\left\{1,\sqrt{\frac{d}{4k}}\right\}.$$

In particular, for any (randomized) algorithm $\mathcal{A}$ which can observe a sequence of data points $(Y^1,\cdots,Y^k)$ with $Y^t = f(X^t;S^t)$ where $S^t\in\mathcal{D}=\{s_1,s_2,\ldots,s_n\}$ and $X^t\in\mathcal{K}$ are chosen arbitrarily and adaptively by $\mathcal{A}$, there exists a distribution $\mathcal{P}$ over convex functions such that $\mathbb{E}_{s\sim\mathcal{P}}[\|\nabla f(x,s)\|_2^2]\leq G^2$ for all $x\in\mathcal{K}$, such that the output $\widehat{x}$ of the algorithm satisfies

$$\mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{A}} f(\widehat{x};s)]\right] - \min_{x\in\mathcal{K}}\mathop{\mathbb{E}}_{s\sim\mathcal{P}}[f(x;s)] \geq \frac{GD}{16}\min\left\{1,\sqrt{\frac{d}{4k}}\right\}.$$

3.7.1 Proof of Theorem 3.7.1

We reduce the optimization problem into a series of binary hypothesis tests. Recall we are considering linear functions $f(x;s) := \langle x,s\rangle$. Let $\mathcal{V}=\{-1,1\}^d$ be a Boolean hyper-cube and for each $v\in\mathcal{V}$, let $\mathcal{N}_v = \mathcal{N}(\delta v,\sigma^2 I_d)$ be a Gaussian distribution for some parameters to be chosen such that $\widehat{F}_v(x) := \mathbb{E}_{s\sim\mathcal{N}_v}[f(x;s)] = \delta\langle x,v\rangle$. Note that

$$\mathop{\mathbb{E}}_{s\sim\mathcal{N}_v}[\|\nabla f(x,s)\|_2^2] = \mathop{\mathbb{E}}_{s\sim\mathcal{N}_v}[\|s\|_2^2] = (\delta^2+\sigma^2)d.$$

Therefore $G=\sqrt{d(\delta^2+\sigma^2)}$.

Clearly the lower bound should scale linearly with D. Therefore without loss of generality, we can assume that the diameter $D = 2$ and define $\mathcal{K} = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ to be the unit ball. As in [ACCD12], we suppose that v is uniformly sampled from $\mathcal{V} = \{-1, 1\}^d$. Note that if we can find a good solution to $\widehat{F}_v(x)$, we need to determine the signs of vector v well. Particularly, we have the following claim:

Claim 3.7.2 ([DJWW15]). *For each $v \in \mathcal{V}$, let x^v minimize $\widehat{F}_v$ over $\mathcal{K}$ and obviously we know that $x^v = -v/\sqrt{d}$. For any solution $\widehat{x} \in \mathbb{R}^d$, we have*

$$\widehat{F}_v(\widehat{x}) - \widehat{F}_v(x^v) \geq \frac{\delta}{2\sqrt{d}} \sum_{j=1}^{d} \mathbb{1}\{\text{sign}(\widehat{x}_j) \neq \text{sign}(x_j^v)\},$$

where the function sign$(\cdot)$ *is defined as:*

$$\text{sign}(\widehat{x}_j) = \begin{cases} + & \text{if } \widehat{x}_j > 0 \\ 0 & \text{if } \widehat{x}_j = 0 \\ - & \text{otherwise} \end{cases}$$

Claim 3.7.2 provides a method to lower bound the minimax error. Specifically, we define the hamming distance between any two vectors $x, y \in \mathbb{R}^d$ as $d_H(x, y) = \sum_{j=1} \mathbb{1}\{\text{sign}(x_j) \neq \text{sign}(y_j)\}$, and we have

$$\varepsilon_k^*(\mathcal{P}_G, \mathcal{K}) \geq \frac{\delta}{2\sqrt{d}} \{\inf_{\widehat{v}} \mathbb{E}[d_H(\widehat{v}, v)]\}, \tag{3.17}$$

where $\widehat{v}$ denotes the output of any algorithm mapping from the observation $(Y^1, \cdots, Y^k)$ to $\{-1, 1\}^d$, and the probability is taken over the distribution of the underlying v, the observation $(Y^1, \cdots, Y^k)$ and any additional randomness in the algorithm.

By Equation (3.17), it suffices to lower bound the value of the testing error $\mathbb{E}[d_H(\widehat{v}, v)]$. As discussed in [ACCD12, DJWW15], the randomness in the algorithm can not help, and we can assume the algorithm is deterministic, i.e. (X^t, S^t) is a deterministic function of $Y^{[t-1]}$.[7] The argument is basically based on the easy direction of Yao's principle.

[7] We use $Y^{[t]}$ to denote the first t observations, i.e. $(Y^1, \cdots, Y^t)$

Now we continue our proof of the lower bound. We will make use of the property of the Bayes risk.

Lemma 3.7.3 ([ACCD12, Lemma 1])**.** *Consider the problem of testing hypothesis $H_{-1} : v \sim \mathbb{P}_{-1}$ and $H_1 : v \sim \mathbb{P}_1$, where H_{-1} and H_1 occur with prior probability π_{-1} and $\pi_1 := 1 - \pi_{-1}$ respectively prior to the experiment. For any algorithm that takes one sample v and outputs $\widehat{i} : v \to \{-1, 1\}$, we define the Bayes risk B be the minimum average probability that algorithm fails (v is not sampled from $H_{\widehat{i}(v)}$). That is $B = \inf_{\widehat{i}} \pi_{-1} \Pr[\widehat{i}(v) = 1 \mid v \sim \mathbb{P}_{-1}] + \pi_1 \Pr[\widehat{i}(v) = 0 \mid v \sim \mathbb{P}_1]$. Then, we have*

$$B \geq \min(\pi_{-1}, \pi_1)(1 - \|\mathbb{P}_1 - \mathbb{P}_{-1}\|_{\mathrm{TV}}).$$

Lemma 3.7.4. *Suppose that v is uniformly sampled from $\mathcal{V} = \{-1, 1\}^d$, then any estimate $\widehat{v}$ obeys*

$$\mathbb{E}[d_H(\widehat{v}, v)] \geq \frac{d}{2}\left(1 - \frac{\delta\sqrt{k}}{\sigma\sqrt{d}}\right).$$

Proof. Let $\pi_{-1} = \pi_1 = 1/2$. For each j, define $\mathbb{P}_{-1,j} = \mathbb{P}(Y^{[k]} \mid v_j = -1)$ and $\mathbb{P}_{1,j} = \mathbb{P}(Y^{[k]} \mid v_j = 1)$ to be distributions over the observations $(Y^1, \cdots, Y^k)$ conditional on $v_j \neq 1$ and $v_j = 1$ respectively. Let B_j be the Bayes risk of the decision problem for j-th coordinate of v between $H_{-1,j} : v_j = -1$ and $H_{1,j} : v_j = 1$. We have that

$$\begin{aligned}
\mathbb{E}[d_H(\widehat{v}, v)] \geq & \sum_{j=1}^{d} B_j \\
\geq & \pi_1 \sum_{j=1}^{d} (1 - \|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}) \\
\geq & \frac{d}{2}\left(1 - \frac{1}{\sqrt{d}}\sqrt{\sum_{j=1}^{d} \|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}^2}\right),
\end{aligned}$$

where the first inequality follows from the definition of Bayes risk B_j, the second inequality follows by Lemma 3.7.3 and the last inequality follows by the Cauchy-Schwartz inequality.

To complete the proof, it suffices to show that

$$\sum_{j=1}^{d} \|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}^2 \leq \frac{\delta^2}{\sigma^2} k. \tag{3.18}$$

Assuming Equation (3.18) first, which will be established later. Then we know that

$$\mathbb{E}[d_H(\widehat{v}, v)] \geq \frac{d}{2}(1 - \frac{\delta\sqrt{k}}{\sigma\sqrt{d}}).$$

□

We will complete the proof of Lemma 3.7.4 by showing the following bounded total variation distance.

Claim 3.7.5.

$$\sum_{j=1}^{d} \|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}^2 \leq \frac{\delta^2}{\sigma^2} k.$$

Proof. Applying Pinsker's inequality, we know $\|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}^2 \leq \frac{1}{2}\mathrm{D}_{KL}(\mathbb{P}_{-1,j}\|\mathbb{P}_{1,j})$. To bound the KL divergence between $\mathbb{P}_{-1,j}$ and $\mathbb{P}_{1,j}$ over all possible $Y^{[k]}$, consider $v' = (v_1, \cdots, v_{j-1}, v_{j+1}, \cdots, v_d)$, and define $\mathbb{P}_{-1,j,v'}(Y^{[k]}) := \mathbb{P}(Y^{[k]} \mid v_j = -1, v')$ to be the distribution conditional on $v_j = -1$ and v'. We have

$$\mathbb{P}_{-1,j}(Y^{[k]}) = \sum_{v'} \Pr[v']\mathbb{P}_{-1,j,v'}(Y^{[k]}).$$

The convexity of the KL divergence suggests that

$$\mathrm{D}_{KL}(\mathbb{P}_{-1,j}\|\mathbb{P}_{1,j}) \leq \sum_{v'} \Pr[v']\mathrm{D}_{KL}(\mathbb{P}_{-1,j,v'}\|\mathbb{P}_{1,j,v'}).$$

Fixing any possible v', we want to bound the KL divergence $\mathrm{D}_{KL}(\mathbb{P}_{-1,j,v'}\|\mathbb{P}_{1,j,v'})$.

Recall we are considering deterministic algorithms and (X^t, S^t) is a deterministic function of $Y^{[t-1]}$. Let $Q_i \in \mathbb{R}^{d\times k}$ be a (random) matrix, which records the set of points the algorithm queries for the user s_i. Specifically, for t-th step, if the algorithm queries (X^t, S^t),

then $Q_i^t = X^t$ if $S^t = s_i$, otherwise $Q_i^t = 0$, where Q_i^t is the t-th column of Q_i.

As we are considering linear functions, without loss of generality we can assume $\langle Q_i^j, Q_i^{j'} \rangle = 0$ for each i and any $j \neq j'$, and $\|Q_i^t\|_2 \in \{0, 1\}$ for any i and t. We name this assumption Orthogonal Query. Roughly speaking, for any algorithm, we can modify it to satisfy the Orthogonal Query. Whenever the algorithm wants to query some point, we can use Gram–Schmidt process to query another point and satisfy Orthogonal Query, and recover the function value at the original point queried by the algorithm.

By the chain-rule of KL-divergence, if we define $P_{-1,j,v'}(Y^t \mid Y^{[t-1]})$ to be the distribution of tth observation Y^t conditional on v', $v_j = -1$ and $Y^{[t-1]}$, then we have

$$\mathrm{D}_{KL}(\mathbb{P}_{-1,j,v'} \| \mathbb{P}_{1,j,v'}) = \sum_{t=1}^{k} \int_{\mathcal{Y}^{t-1}} \mathrm{D}_{KL}(P_{-1,j,v'}(Y^t \mid Y^{[t-1]} = y) \| P_{1,j,v'}(Y^t \mid Y^{[t-1]} = y) \mathrm{d}P_{-1,j,v'}(y).$$

Fix $Y^{[t-1]}$ such that $Y^{[t-1]} = y$. Since the algorithm is deterministic and (X^t, S^t) is fixed given $Y^{[t-1]}$. Let $S^t = s_i$ so $X^t = Q_i^t$.

Note that the n users in $\mathcal{D}$ are i.i.d. sampled. Then $\mathrm{D}_{KL}(P_{-1,j,v'}(Y^t \mid Y^{[t-1]} = y) \| P_{1,j,v'}(Y^t \mid Y^{[t-1]} = y)$ only depends on the randomness of s_i and the first t columns of Q_i, which is denoted by $Q_i^{[t]}$. We use Y_j^t to denote the observation corresponding to user s_j for the tth query (if $S^t \neq s_j$, we have $Y_j^t = 0$). Note that the observation $Y_i^{[t]} = Q_i^{[t]\top} s_i$ where $s_i \sim \mathcal{N}(\delta v, \sigma^2 I_d)$. Then we know $Y_i^{[t]}$ is normally distributed with mean $\delta Q_i^{[t]\top} v$ and co-variance $\sigma^2 Q_i^{[t]\top} Q_i^{[t]}$.

Recall that the KL divergence between two normal distributions is $\mathrm{D}_{KL}(\mathcal{N}(\mu_1, \Sigma) \| \mathcal{N}(\mu_2, \Sigma)) = \frac{1}{2}(\mu_1 - \mu_2)^\top \Sigma^{-1} (\mu_1 - \mu_2)$. Recall that we have the Orthogonal Query assumption and thus $Q_i^{[t]\top} Q_i^{[t]} \in \{0, 1\}^{t \times t}$ is a diagonal matrix. By the conditional distributions of Gaussian, we know Y_i^t only depends on the Q_i^t and it is independent of $Q_i^{[t-1]}$.

Hence we have

$$\begin{aligned}
&\mathrm{D}_{KL}(P_{-1,j,v'}(Y^t \mid Y^{[t-1]} = y) \| P_{1,j,v'}(Y^t \mid Y^{[t-1]} = y)) \\
=&\mathrm{D}_{KL}(P_{-1,j,v'}(Y_i^t \mid Y^{[t-1]} = y) \| P_{1,j,v'}(Y_i^t \mid Y^{[t-1]} = y)) \\
=&\frac{1}{2}(2\delta Q_i^t(j))^2 / \sigma^2,
\end{aligned}$$

where $Q_i^t(j)$ is the j-th coordinate of Q_i^t. Summing over the terms, one has

$$\begin{aligned}\sum_{j=1}^{d} \|\mathbb{P}_{1,j} - \mathbb{P}_{-1,j}\|_{\mathrm{TV}}^2 \leq & \frac{1}{2}\mathrm{D}_{KL}(\mathbb{P}_{-1,j}\|\mathbb{P}_{1,j}) \\ \leq & \frac{1}{2}\sum_{t=1}^{k}\sum_{j=1}^{d}\sum_{i=1}^{n}\mathbb{E}[\frac{1}{2}(2\delta Q_i^t(j))^2/\sigma^2] \\ \leq & \frac{\delta^2}{\sigma^2}k,\end{aligned}$$

where the last line follows from the fact that for each t,$\sum_{i=1}^n \|Q_i^t\|_2^2 = \sum_{i=1}^n \sum_{j=1}^d (Q_i^t(j))^2 = 1$ as we only query one user for t-th step.

This completes the proof. □

Having Lemma 3.7.4, we can complete the proof of Theorem 3.7.1.

Proof. of Theorem 3.7.1. As discussed before, we know

$$\widehat{F}_v(\widehat{x}) - \widehat{F}_v(x^v) \geq \frac{\delta}{2\sqrt{d}}\sum_{j=1}^{d}\mathbb{1}\{\mathrm{sign}(\widehat{x}_j) \neq \mathrm{sign}(x_j^v)\},$$

and hence we know that

$$\begin{aligned}\varepsilon_k^*(\mathcal{P}_G, \mathcal{K}) \geq & \frac{\delta}{2\sqrt{d}}\inf_{\widehat{v}}\mathbb{E}[d_H(\widehat{v}, v)] \\ \geq & \frac{\delta\sqrt{d}}{4}\left(1 - \frac{\delta\sqrt{k}}{\sigma\sqrt{d}}\right),\end{aligned}$$

where the last line follows from Lemma 3.7.4. We now set $\delta = \frac{\sigma\sqrt{d}}{2\sqrt{k}}$ and $\sigma = \frac{G}{\sqrt{d+d^2/4k}}$, so that $d(\sigma^2+\delta^2) = G^2$. Hence one has

$$\varepsilon_k^*(\mathcal{P}_G, \mathcal{K}) \geq \frac{\delta\sqrt{d}}{8} = \frac{D\delta\sqrt{d}}{16} = \frac{GD}{16\sqrt{1+\frac{4k}{d}}} \geq \frac{GD}{16}\min\left\{1, \sqrt{\frac{d}{4k}}\right\}.$$

Thus we complete the proof. □

Corollary 3.7.6 (Lower bound for DP-SCO)**.** *For any (non-private) algorithm which makes less than* $O\left(\min\{\frac{\varepsilon^2 n^2}{\log(1/\delta)}, nd\}\right)$ *function value queries, there exist a convex domain* $\mathcal{K} \subset \mathbb{R}^d$ *of diameter* D*, a distribution* $\mathcal{P}$ *supported on* G*-Lipschitz linear functions* $f(x;s) := \langle x, s\rangle$*, such that the output* $\widehat{x}$ *of the algorithm satisfies that*

$$\mathop{\mathbb{E}}_{s\sim\mathcal{P}}[\langle \widehat{x}, s\rangle] - \min_{x\in\mathcal{K}} \mathop{\mathbb{E}}_{s\sim\mathcal{P}}[\langle x, s\rangle] \geq \Omega\left(\frac{GD}{\sqrt{1+\log(n)/d}} \cdot \min\left\{\frac{\sqrt{\log(1/\delta)d}}{\varepsilon n} + \frac{1}{\sqrt{n}}, 1\right\}\right).$$

Proof. Note that Theorem 3.7.1 almost gives us what we want, except that the Lipschitz constant of the functions in the hard distribution is bounded only on average by G. To get distributions over G-Lipschitz functions, we just condition on the bad event not happening.

Recall that we are considering the set of distributions $\mathcal{N}_v = \mathcal{N}(\delta v, \sigma^2 I_d)$ for which $\mathbb{E}_{s\sim\mathcal{N}_v} \|s\|_2^2 \leq G^2 = d(\delta^2+\sigma^2)$. And we proved that $\inf_{\mathcal{A}\in\mathbb{A}_k} \sup_{v\in\mathcal{V}} \mathbb{E}_{s\sim\mathcal{N}_v,\mathcal{A}}[\widehat{F}_v(\widehat{x}_k) - \widehat{F}_v^*] \geq \frac{GD}{16}\min\left\{1, \sqrt{\frac{d}{4k}}\right\}$ in Theorem 3.7.1, where $\widehat{x}_k$ is the output of $\mathcal{A}$ with k observations $Y^{[k]}$. To prove Corollary 3.7.6, we need to modify the distribution of s to satisfy the Lipschitz continuity.

In particularly, for some constant c, we know

$$\begin{aligned}
&\mathbb{E}[\widehat{F}_v(\widehat{x}_k) - \widehat{F}_v^*] \\
=&\,\mathbb{E}\left[\widehat{F}_v(\widehat{x}_k) - \widehat{F}_v^* \mid \max_{s_i\in\mathcal{D}} \|s_i\|_2 \leq cG\sqrt{1+\log(nd)/d}\right] \Pr\left[\max_{s_i\in\mathcal{D}} \|s_i\|_2 \leq cG\sqrt{1+\log(nd)/d}\right] + \\
&\,\mathbb{E}\left[\widehat{F}_v(\widehat{x}_k) - \widehat{F}_v^* \mid \max_{s_i\in\mathcal{D}} \|s_i\|_2 > cG\sqrt{1+\log(nd)/d}\right] \Pr\left[\max_{s_i\in\mathcal{D}} \|s_i\|_2 > cG\sqrt{1+\log(nd)/d}\right].
\end{aligned}$$

By the concentration of spherical Gaussians, we know if $s \sim \mathcal{N}(\delta v, \sigma^2 I_d)$, then

$$\Pr\left[\|s-\delta v\|_2^2 \leq \sigma^2 d(1 + 2\sqrt{\ln(1/\eta)/d} + 2\ln(1/\eta)/d)\right] \geq 1-\eta.$$

We can choose the constant c large enough, such that $\Pr[\max_{s_i\in\mathcal{D}} \|s_i\|_2 \leq cG\sqrt{1+\log(nd)/d}] \geq 1 - 1/\operatorname{poly}(nd)$, which implies

$$\inf_{\mathcal{A}\in\mathbb{A}_k} \sup_{v\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{N}_v^n,\mathcal{A}}\left[\widehat{F}_v(\widehat{x}_k) - \widehat{F}_v^* \mid \max_{s_i\in\mathcal{D}} \|s_i\|_2 \leq cG\sqrt{1+\log(nd)/d}\right] \geq \Omega(GD\frac{\min\{\sqrt{d}, \sqrt{k}\}}{\sqrt{k}}).$$

If we use the distributions conditioned on $\max_{s_i\in\mathcal{D}}\|s_i\|_2 \le cG\sqrt{1+\log(nd)/d}$ rather than the Gaussians, and scale the constant to satisfy the assumption on Lipschitz continuity, we can prove the statement. Particularly, let $G' = cG(\sqrt{1+\log(nd)/d})$. If the algorithm can only make $k = O\left(\min\{\frac{\varepsilon^2 n^2}{\log(1/\delta)}, nd\}\right)$ observations, we know

$$\begin{aligned}
&\inf_{\mathcal{A}\in\mathbb{A}_k}\sup_{v\in\mathcal{V}}\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{N}_v^n,\mathcal{A}}\left[\widehat{F}_v(\widehat{x}_k)-\widehat{F}_v^* \mid \max_{s_i\in\mathcal{D}}\|s_i\|_2\le G'\right]\\
\ge&\Omega\left(GD\cdot\min\left\{(\frac{\sqrt{\log(1/\delta)d}}{\varepsilon n}+\frac{1}{\sqrt{n}}),1\right\}\right)\\
=&\Omega\left(\frac{G'D}{\sqrt{1+\log(nd)/d}}\cdot\min\left\{\frac{\sqrt{\log(1/\delta)d}}{\varepsilon n}+\frac{1}{\sqrt{n}},1\right\}\right),
\end{aligned}$$

which proves the lower bound claimed in the Corollary statement. □

Corollary 3.7.7 (Lower bound for sampling scheme)**.** *Given any $G>0$ and $\mu>0$. For any algorithm which takes function values queries less than $O\left(\frac{G^2}{\mu}/(1+\log(G^2/\mu)/d)\right)$ times, there is a family of G-Lipschitz linear functions $\{f_i(x)\}_{i\in I}$ defined on some ℓ_2 ball $\mathcal{K}\subset\mathbb{R}^d$, such that the total variation distance between the distribution of the output of the algorithm and the distribution proportional to $\exp(-\mathbb{E}_{i\in I} f_i(x)-\mu\|x\|^2/2)$ is at least $\min(1/2,\sqrt{d\mu/G^2})$.*

Proof. By a similar argument in the proof of Corollary 3.7.6, for any algorithm which can only make k observations, there are a family of G-Lipschitz linear functions restricted on an ℓ_2 ball $\mathcal{K}$ of diameter D centered at $\mathbf{0}$ such that

$$\mathbb{E}\left[\widehat{F}_v(\widehat{x}_k)-\widehat{F}_v^*\right]\ge\Omega\left(\frac{GD}{\sqrt{1+\log(k)/d}}\cdot\min\left\{\sqrt{\frac{d}{k}},1\right\}\right), \tag{3.19}$$

where $\widehat{F}_v^*=\min_{x\in\mathcal{K}}\widehat{F}_v(x)$ and $\widehat{x}_k\in\mathcal{K}$ is the output of $\mathcal{A}$.

Suppose we have a sampling algorithm that takes k queries. We use it to sample from $x^{(sol)}$ proportional to $p(x) := \exp(-\widehat{F}_v(x)-\frac{\mu}{2}\|x\|^2)$ on $\mathcal{K}$ with total variation distance $\eta\le\min(1/2,\sqrt{d\mu/G^2})$.

Lemma 3.6.1 shows that

$$\mathbb{E}[\widehat{F}_v(x^{(sol)}) + \frac{\mu}{2}\|x^{(sol)}\|^2] \le \min_{x\in\mathcal{K}} \left(\widehat{F}_v(x) + \frac{\mu}{2}\|x\|^2\right) + O(d) + O(\eta)\cdot(GD + \mu D^2),$$

where the last term involving η is due to the total variation distance between $x^{(sol)}$ and p. Setting $D = \sqrt{d/\mu}$ and using the diameter of $\mathcal{K}$ is D and $\eta \le \min(1/2, \sqrt{d\mu/G^2})$, we have

$$\begin{aligned}\mathbb{E}[\widehat{F}_v(x^{(sol)})] &\le \min_{x\in\mathcal{K}} \widehat{F}_v(x) + \frac{\mu}{2}D^2 + O(d + \eta\cdot(GD + \mu D^2)) \\ &\le \min_{x\in\mathcal{K}} \widehat{F}_v(x) + O(d).\end{aligned}$$

Note that we set $D = \sqrt{d/\mu}$. Comparing with (3.19), we have

$$\frac{G\sqrt{d/\mu}}{\sqrt{1+\log(k)/d}} \min\left\{\sqrt{\frac{d}{k}}, 1\right\} \le O(d).$$

If $d \le G^2/\mu \le \exp(d)$, we have

$$G\sqrt{d/\mu}\sqrt{\frac{d}{k}} \le O(d)$$

and hence $k = \Omega(G^2/\mu)$. If $G^2/\mu \ge \exp(d)$, we have

$$\frac{G\sqrt{d/\mu}}{\sqrt{\log(k)/d}}\sqrt{\frac{d}{k}} \le O(d)$$

and hence $k = \Omega(\frac{G^2 d/\mu}{\log(G^2/\mu)})$. If $G^2/\mu \le d$, we can construct our function only on the first $O(G^2/\mu)$ dimensions to get a lower bound $k = \Omega(G^2/\mu)$. Combining all cases gives the result. □

Part II

NON-EUCLIDEAN GEOMETRY

Chapter 4

PRIVATE CONVEX OPTIMIZATION IN GENERAL NORMS

4.1 Introduction

The study of convex optimization in spaces where the natural geometry is non-Euclidean, beyond being a natural question of independent interest, has resulted in many successes across algorithm design. A basic example of this is the celebrated multiplicative weights, or exponentiated gradient method [AHK12], which caters to the ℓ_1 geometry and has numerous applications in learning theory and algorithms. Moreover, optimization in real vector spaces equipped with different ℓ_p norms has found use in sparse recovery [CRT06], combinatorial optimization [KLOS14, KPSW19], multi-armed bandit problems [BC12], fair resource allocation [DFO20], and more (see e.g. [AKPS19, DG21] and references therein). Furthermore, optimization in Schatten-p norm geometries (the natural generalization of ℓ_p norms to matrix spaces) has resulted in improved algorithms for matrix completion [ANW10] and outlier-robust PCA [JLT20]. In addition to ℓ_p and Schatten-p norms, the theory of non-Euclidean geometries has been very useful in settings such as linear and semidefinite programming [Nem04] and optimization on matrix spaces [AGL+18], amongst others.

The main result of this paper is a framework for *differentially private* convex optimization in general normed spaces under a Lipschitz parameter bound. Differential privacy [DKM+06, DR14] has been adopted as the standard privacy notion for data analysis in both theory and practice, and differentially private algorithms have been deployed in many important settings in the industry as well as the U.S. census [EPK14, Abo16, Tea17, BEM+17, DKY17]. Consequently, differentially private optimization is an increasingly important and fundamental primitive in modern machine learning applications [BST14, ACG+16]. However, despite an extensive body of theoretical work providing privacy-utility tradeoffs (and more) for optimization in the Euclidean norm geometry, e.g. [CM08, CMS11, KST12, JT14, BST14, KJ16] (and many other follow-up works), more general settings have been

left relatively unexplored. This state of affairs prohibits the application of private optimization theory to problems where the natural geometry is non-Euclidean. Recent works [AFKT21, BGN21, BGM21] have investigated special cases of private convex optimization, e.g. for ℓ_p spaces or polyhedral sets, under smoothness assumptions, or under structured losses. However, the systematic study of private convex optimization in general normed spaces in the most fundamental setting of Lipschitz losses has been left open, a gap that our work addresses.

Our framework for private convex optimization is simple: we demonstrate strong privacy-utility tradeoffs for a *regularized exponential mechanism* when optimizing a loss over a set $\mathcal{X} \subset \mathbb{R}^d$ equipped with a norm $\|\cdot\|_{\mathcal{X}}$. More concretely, our algorithms sample from densities

$$\propto \exp\left(-k(F_{\mathcal{D}} + \mu r)\right)$$

where $k, \mu > 0$ are tunable parameters, $F_{\mathcal{D}}$ is a (data-dependent) empirical risk, and r is a strongly convex regularizer in $\|\cdot\|_{\mathcal{X}}$ with bounded range over $\mathcal{X}$. In the analogous non-private Lipschitz convex optimization setting, most theoretical developments (namely mirror descent frameworks) have focused on precise applications where such an r is readily available [Sha12, Bub15]. In this sense, our framework directly extends existing Lipschitz convex optimization theory to the private setting (and indeed, recovers existing non-private guarantees obtained by mirror descent [NY83]).

In the remainder of the introduction, we summarize our results (Section 4.1.1), highlight our technical contributions (Section 4.1.2), and situate our paper in the context of prior work (Section 4.1.3).

4.1.1 Our results

We study both the empirical risk minimization (ERM) problem and the stochastic convex optimization (SCO) problem in this paper; the goal in the latter case is to minimize the *population risk*. We formalize this under the following assumption, which parameterizes the space we are optimizing and the (empirical and population) risks we aim to minimize.

Assumption 4.1.1. *We make the following assumptions.*

1. *There is a compact, convex set* $\mathcal{X} \subset \mathbb{R}^d$ *equipped with a norm* $\|\cdot\|_{\mathcal{X}}$.

2. *There is a* 1*-strongly convex function* $r : \mathcal{X} \to \mathbb{R}$ *in* $\|\cdot\|_{\mathcal{X}}$*, and* $\Theta \geq \max_{x \in \mathcal{X}} r(x) - \min_{x \in \mathcal{X}} r(x)$.

3. *There is a set* Ω *such that for any* $s \in \Omega$*, there is a convex function* $f(\cdot; s) : \mathcal{X} \to \mathbb{R}$ *which is* G*-Lipschitz in* $\|\cdot\|_{\mathcal{X}}$.

For definitions used above, see Section 4.2. We remark that by strong convexity, the parameter Θ scales at least as $\Omega(D^2)$, where D is the diameter of $\mathcal{X}$ with respect to $\|\cdot\|_{\mathcal{X}}$; in many cases of interest, we may upper bound Θ by $O(D^2)$ as well up to a logarithmic factor.

Finally, throughout the paper when working under Assumption 4.1.1, $\mathcal{D} = \{s_i\}_{i \in [n]}$ denotes a dataset drawn independently from $\mathcal{P}$, a distribution supported on Ω, and we define $F_{\mathcal{D}} : \mathcal{X} \to \mathbb{R}$ and $F_{\mathcal{P}} : \mathcal{X} \to \mathbb{R}$ by

$$F_{\mathcal{D}}(x) := \frac{1}{n} \sum_{i \in [n]} f(x; s_i), \; F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s \sim \mathcal{P}}[f(x; s)]. \tag{4.1}$$

Private ERM and SCO. We first present the following general results under Assumption 4.1.1.

Theorem 4.1.2 (Informal, see Theorems 4.4.2, 4.4.6)**.** *Under Assumption 4.1.1 and following notation (4.1), drawing a sample* x *from the density* $\nu \propto \exp(-k(F_{\mathcal{D}} + \mu r))$ *for some* $k, \mu > 0$ *specified in Theorem 4.4.2 is* (ε, δ)*-differentially private, and produces* x *such that*

$$\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{D}}(x)] - \min_{x \in \mathcal{X}} F_{\mathcal{D}}(x) \leq G\sqrt{\Theta} \cdot \frac{\sqrt{8d \log \frac{1}{2\delta}}}{n\varepsilon}.$$

Moreover, drawing a sample x *from the density* $\nu \propto \exp(-k(F_{\mathcal{D}} + \mu r))$ *for some* $k, \mu > 0$ *specified in Theorem 4.4.6 is* (ε, δ)*-differentially private, and produces* x *such that*

$$\mathop{\mathbb{E}}_{\mathcal{D} \sim \mathcal{P}^n, x \sim \nu}[F_{\mathcal{P}}(x)] - \min_{x \in \mathcal{X}} F_{\mathcal{P}}(x) \leq G\sqrt{\Theta} \cdot \left(\frac{\sqrt{8d \log \frac{1}{2\delta}}}{n\varepsilon} + \sqrt{\frac{8}{n}} \right).$$

Minimizing the non-private population risk under the same setting as Assumption 4.1.1 is a very well-studied problem, with matching upper and lower bounds in many cases of interest, such as ℓ_p norms [NY83, ABRW12, DJW14]. The population utility achieved by our regularized exponential mechanism in Theorem 4.1.2 (namely, as $\varepsilon \to \infty$) matches the rate obtained by the classical mirror descent algorithm [NY83], which to our knowledge has not been previously observed. Finally, in Appendix 4.5 we provide an analog of Theorem 4.1.2 under the stronger assumption that the sample losses $f(\cdot; s)$ are strongly convex, bypassing the need for explicit regularization. Our results in Appendix 4.5 recover the optimal rate in the Euclidean case, matching known lower bounds [BST14].

We next show how to apply the results of Theorem 4.1.2 under various instantiations of Assumption 4.1.1 to derive new rates for private convex optimization under ℓ_p and Schatten-p norm geometries.

ℓ_p and Schatten-p norms. In Corollaries 4.4.10, 4.4.11, and 4.4.12, we combine known (optimal) uniform convexity estimates for ℓ_p spaces [BCL94] with the algorithms of Theorem 4.4.2 and 4.4.6 to obtain privacy-utility tradeoffs summarized in Table 4.1. Interestingly, we achieve all these bounds with a single algorithmic framework, which in all cases matches or partially matches known lower bounds.

ℓ_p norm	Optimality gap	
	ERM loss $F_{\mathcal{D}}$	SCO loss $F_{\mathcal{P}}$
$p \in (1, 2)$ $(\star)$	$GD \cdot \frac{\sqrt{d \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}}$	$GD \cdot \left(\frac{1}{\sqrt{n(p-1)}} + \frac{\sqrt{d \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}} \right)$
$p = 1$ $(\dagger)$	$GD \cdot \frac{\sqrt{d \log d \log \frac{1}{2\delta}}}{n\varepsilon}$	$GD \cdot \left(\sqrt{\frac{\log d}{n}} + \frac{\sqrt{d \log d \log \frac{1}{2\delta}}}{n\varepsilon} \right)$
$p \geq 2$ $(\dagger)$	$GD \cdot \frac{d^{1-\frac{1}{p}}\sqrt{\log \frac{1}{2\delta}}}{n\varepsilon}$	$GD \cdot \left(\frac{d^{\frac{1}{2}-\frac{1}{p}}}{\sqrt{n}} + \frac{d^{1-\frac{1}{p}}\sqrt{\log \frac{1}{2\delta}}}{n\varepsilon} \right)$

Table 4.1: Privacy-utility tradeoffs for ℓ_p norm optimization under (ε, δ)-differential privacy obtained by: Corollary 4.4.10 ($p \in (1, 2)$), Corollary 4.4.11 ($p = 1$), and Corollary 4.4.12 ($p \geq 2$). We assume $\mathcal{X}$ has ℓ_p diameter bounded by D and hide constants (stated in the formal results) for brevity. $(\star)$ indicates that our result matches the private ERM and SCO lower bound [BGN21, LL22]. $(\dagger)$ indicates that our result (as $\varepsilon \to \infty$) matches the non-private SCO lower bound [ABRW12, DJW14].

We now contextualize our results with regard to the existing literature. In the following discussion, the "privacy-dependent" loss term is the term in the SCO loss scaling with ε, δ, and the "privacy-independent" loss term is the SCO loss when $\varepsilon \to \infty$.

In the case of constant $p \in (1, 2)$, our Corollary 4.4.10 sharpens Theorem 5 of [AFKT21] by a $\sqrt{\log d}$ factor in the privacy-dependent loss term, and is the first to match lower bounds of [BGN21, LL22]. It improves bounds by [BGN21] by at least a $\log n$ factor on both parts of the SCO loss, which further loses an $n^{\frac{1}{4}}$ factor on the privacy-dependent loss and requires additional smoothness assumptions.

In the important case of $p = 1$, of fundamental interest due to its applications in sparse recovery [CRT06] as well as online learning [Sha12, AHK12], our Corollary 4.4.11 improves the privacy-dependent loss term of [AFKT21] by a $\log d$ factor, and matches the privacy-independent loss lower bound in the SCO literature [DJW14], matching the rate of entropic mirror descent. The privacy-dependent loss term incurs an additional overhead of $\sqrt{\log d}$ compared to existing lower bounds. However, just as lower bounds on the privacy-independent loss increase as $p \to 1$, it is plausible that the upper bound obtained by Corollary 4.4.11 is optimal, which we leave as an interesting open direction.

In the $p \geq 2$ case, prior work by [BGN21] obtains a rate matched by Corollary 4.4.12. The non-private population risk term in (4.15) is again known to be optimal [ABRW12]. We again find it an interesting open direction to close the gap between the upper bound (4.13) and known lower bounds for private convex optimization when $p \geq 2$, e.g. [BGN21, LL22].

We further demonstrate in Corollary 4.4.13 that all of these results have direct analogs in the case of optimization over matrix spaces equipped with Schatten norm geometries. To the best of our knowledge, this is the first such result in the private optimization literature; we believe this showcases the generality and versatility of our approach.

Finally, we mention that all of these results are algorithmic and achieved by samplers with polynomial query complexity and runtime, following developments of [LST21b, GLL22]. In all cases, by simple norm equivalence relations, the query complexity of our samplers is at most a d factor worse than the ℓ_2 case, with improvements as $p \to 2$. It is an exciting direction to develop efficient high-accuracy samplers catering to structured densities relevant to the setups considered in this paper, e.g. those whose negative log-likelihoods are

strongly convex in ℓ_p norms. The design of sampling algorithms for continuous distributions has been an area of intense research activity in the machine learning community, discussed in greater detail in Section 4.1.3. We mention here that our hope is that our results and general optimization framework serve as additional motivation for the pursuit of efficient structured sampling algorithms working directly in non-Euclidean geometries.

4.1.2 Our techniques

Our results essentially build on the recent work of [GLL22], who observed that a regularized exponential mechanism achieves optimal privacy-utility tradeoffs for empirical and population risks when losses are Lipschitz in the ℓ_2 norm. Under a Euclidean specialization of Assumption 4.1.1, [GLL22] provided variants of Theorem 4.1.2 using the regularizer $r(x) = \frac{1}{2}\|x\|_2^2$, i.e. reweighting by a Gaussian.

We demonstrate several key tools used in [GLL22] have non-Euclidean extensions by using a simple, general approach based on a convex geometric tool known as *localization*. For example, the starting point of our developments is relating the privacy curves of two nearby, strongly convex densities with the privacy curve of Gaussians (see Section 4.2 for definitions).

Theorem 4.1.3. *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, let $F, \widetilde{F} : \mathcal{X} \to \mathbb{R}$ be μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$, and let $P \propto \exp(-F)$ and $Q \propto \exp(-\widetilde{F})$. Suppose $\widetilde{F} - F$ is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$. For all $\varepsilon \in \mathbb{R}_{\geq 0}$,*

$$\delta(P \parallel Q)(\varepsilon) \leq \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G}{\sqrt{\mu}}, 1\right)\right)(\varepsilon).$$

An analog of Theorem 4.1.3 when $\|\cdot\|_{\mathcal{X}}$ is the Euclidean norm was proven as Theorem 4.1 of [GLL22]. Moreover, the analog of Theorem 4.1.2 in [GLL22] follows from combining Theorem 4.1 of that work, and their Theorem 6.10, a reduction from the SCO problem to the ERM problem (containing a generalization error bound). These proofs in [GLL22] rely on powerful inequalities from probability theory, which were initially studied in the Gaussian (Euclidean norm regularizer) setting. For example, Theorem 4.1 applied the *Gaussian isoperimetric inequality* of [ST74, Bor75a] (see also Theorem 1.1, [Led99]), which states that strongly logconcave distributions in the Euclidean norm have expansion quality at least as

good as a corresponding Gaussian. Moreover, the generalization error bound in Theorem 6.10 was proven based on a Euclidean norm *log-Sobolev inequality* and *transportation inequality*, relating Wasserstein distances, KL divergences, and Lipschitz bounds on negative log-densities. Fortunately, it turns out that all of these inequalities have non-Euclidean generalizations (possibly losing constant factors). For example, a non-Euclidean log-Sobolev inequality was shown by Proposition 3.1 of [BL00], and a non-Euclidean transport inequality sufficient for our purposes is proved as Proposition 1 of [CE17]. Finally, variants of the Gaussian isoperimetric inequality in general norms are given by [MS08, Kol11]. Plugging in these tools into the proofs of [GLL22] allows us to recover Theorems 4.1.2 and 4.1.3, as well as our applications.

In this work, we take a different (and in our opinion, simpler) strategy to proving the probability-theoretic inequalities required by Theorems 4.1.2, 4.1.3, yielding an alternative to the proofs in [GLL22] which we believe may be of independent intellectual interest to the privacy community. In particular, our technical insight is the simple observation that several of the definitions in differential privacy are naturally cast in the language of localization [KLS95, FG04], which characterizes extremal logconcave densities subject to linear constraints (see our proof of Lemma 4.3.3 for an example of this). This observation allows us to reduce the proofs of key technical tools used in Theorems 4.1.2 and 4.1.3 to proving these tools in one dimension, where *all norms are equivalent* up to constant factor rescaling.[1] After deriving several extensions of basic localization arguments in Section 4.3.1, we follow this reduction approach to give a more unified proof to Theorems 4.1.2 and 4.1.3. To our knowledge, this is the first direct application of localization techniques in differential privacy.

The interplay between the privacy and probability theory communities is an increasingly active area of exploration [DRS21, GLL22, GTU22] (discussed in more detail in Section 4.1.3). We are hence optimistic that localization-based proof strategies will have further applications in the privacy literature, especially in situations (beyond this paper) where probability theoretic tools used in the Euclidean case do not have non-Euclidean variants in

[1]The one-dimensional case can then typically be handled by more straightforward "combinatorial" arguments, see e.g. Section 2.b of [LS93] or Appendix B.3 of [CDWY20] for examples.

full generality. In such settings, it may be a valuable endeavor to see if necessary inequalities may be directly recast in the language of localization.

4.1.3 Prior work

Private optimization in Euclidean norm. Many prior works on private convex optimization have focused on variants of the ERM and SCO problems studied in this work, under ℓ_2 Lipschitz losses and ℓ_2 bounded domains, such as [CMS11, KST12, BST14, BFTGT19, BFGT20]. The optimal information-theoretic rate for these private optimization problems was given by [BST14], which was matched algorithmically up to constant factors by [BFTGT19, BFGT20].

From an algorithmic perspective, a topic of recent interest in the Euclidean case is the problem of attaining optimal privacy-utility tradeoffs in *nearly-linear time*, namely, with $\approx n$ gradient queries [FKT20, AFKT21, KLL21]. Under additional smoothness assumptions, this goal was achieved by [FKT20]; however, achieving near-optimal gradient oracle query rates in the general Lipschitz case remains open. We note that under *value oracle* access, a near-optimal bound was recently achieved by [GLL22]. This paper primarily focuses on the information-theoretic problem of achieving optimal privacy-utility tradeoffs for a given dataset size. However, we believe the corresponding problem of designing algorithms with near-optimal query complexities and runtimes (under value or gradient oracle access) is also an important open direction in the case of general norm geometries.

Private optimization in non-Euclidean norms. The study of convex optimization in non-Euclidean geometries was recently initiated by [AFKT21, BGN21], who focused primarily on developing algorithms under ℓ_p regularity assumptions and bounded domains. In follow-up work, [BGM21] gave improved guarantees for the family of generalized linear losses. We discuss the rates we achieve for ℓ_p norm geometries compared to [AFKT21, BGN21] in Section 4.1.1; in short, we improve prior results by logarithmic factors in the case $p \in [1, 2)$, and match them when $p \geq 2$. Independently from our work, [HLL+22] designed an algorithm for private optimization in ℓ_p geometries improving upon [BGN21] in gradient query complexity (matching their privacy-utility tradeoffs); both [BGN21, HLL+22] require

further smoothness assumptions on the loss functions.

One of the main motivations for this work was to develop a general theory for private convex optimization under non-Euclidean geometries, beyond ℓ_p setups. In particular, [BGN21] designed a *generalized Gaussian mechanism* for the case $p \in [1, 2)$, where gradients were perturbed by a noise distribution catering to the ℓ_p geometry. However, how to design a corresponding mechanism for more general norms may be less clear. The algorithm of [AFKT21] in the non-smooth case was based on a (Euclidean norm) Gaussian mechanism; again, this strategy is potentially more specialized to ℓ_p geometries. Beyond giving a general algorithmic framework for non-Euclidean convex optimization based on structured logconcave sampling, we hope that the information-theoretic properties we show regarding regularized exponential mechanisms (e.g. Theorem 4.1.3) may find use in designing "generalized Gaussian mechanisms" beyond ℓ_p norms.

Connections between privacy and sampling. Our work extends a line of work exploring privacy-utility tradeoffs for the exponential mechanism, a general strategy for designing private algorithms introduced by [MT07] (see additional discussion in [GLL22]). For example, the regularized exponential mechanisms we design are similar in spirit to the exponential mechanism "in the $\mathcal{X}$ norm[2]" designed by [HT10, BDKT12]. Moreover, our work continues a recent interface between the sampling and privacy literature, where (continuous and discrete-time) sampling algorithms are shown to efficiently obtain strong privacy-utility tradeoffs for optimization problems [GLL22, GTU22]. This work further develops this interface, motivating the design of efficient samplers for densities satisfying non-Euclidean regularity assumptions.

The design of sampling algorithms under general geometries (e.g. "mirrored Langevin algorithms") has been a topic of great recent interest, independently from applications in private optimization. Obtaining mixing guarantees under regularity assumptions naturally found in applications is a notoriously challenging problem in the recent algorithmic sampling literature [HKRC18, ZPFP20, AC21, Jia21, LTVW22]. For example, it has been observed both theoretically and empirically that without (potentially restrictive) relationships be-

[2]That is, the norm induced by the convex body $\mathcal{X}$, not to be confused with the $\|\cdot\|_{\mathcal{X}}$ of Assumption 4.1.1.

tween regularity parameters, natural discretizations of the mirrored Langevin dynamics may not even have vanishing bias [ZPFP20, Jia21, LTVW22]. Recently, [LST21b] gave an alternative strategy (to discretizing Langevin dynamics) for designing sampling algorithms in the Euclidean case, used in [GLL22] to obtain private algorithms for ℓ_2-structured ERM and SCO problems (see Proposition 4.4.8). Our work suggests a natural non-Euclidean generalization of these sampling problems, which is useful to study from an algorithmic perspective. We are optimistic that a non-Euclidean variant of [LST21b] may shed light on these mysteries and yield new efficient private algorithms. More generally (beyond the particular [LST21b] framework), we state the direction of designing efficient samplers for densities of the form $\exp(-F_{\mathcal{D}} - \mu r)$ satisfying Assumption 4.1.1 as an important open research endeavor with implications for both sampling and private optimization, the latter of which this paper demonstrates.

4.2 Preliminaries

General notation. Throughout, $\widetilde{O}$ hides logarithmic factors in problem parameters when clear from the context. For $n \in \mathbb{N}$, $[n]$ refers to the naturals $1 \le i \le n$. We use $\mathcal{X}$ to denote a compact convex subset of $\mathbb{R}^d$. The standard (ℓ_2) Euclidean norm is denoted $\|\cdot\|_2$. We will be concerned with optimizing functions $f : \mathcal{X} \to \mathbb{R}$, and $\|\cdot\|_{\mathcal{X}}$ will refer to a norm on $\mathcal{X}$. The diameter of such a set is denoted $\mathrm{diam}_{\|\cdot\|_{\mathcal{X}}}(\mathcal{X}) := \max_{x,y\in\mathcal{X}} \|x - y\|_{\mathcal{X}}$. We let $\mathcal{N}(\mu, \mathbf{\Sigma})$ be the Gaussian density of specified mean and covariance. We denote the convex hull of a set S (when well-defined) by $\mathrm{conv}(S)$. When $a, b \in \mathbb{R}^d$, we abuse notation and let $[a, b]$ be the line segment between a and b.

Norms. For $p \ge 1$, we let $\|\cdot\|_p$ applied to a vector-valued variable be the ℓ_p norm, namely $\|v\|_p = (\sum_{i\in[d]} |v_i|^p)^{1/p}$ for $v \in \mathbb{R}^d$; the ℓ_∞ norm is the maximum absolute value. We will use the well-known inequality

$$\|v\|_p \le \|v\|_q \le d^{\frac{1}{q}-\frac{1}{p}} \|v\|_p, \text{ for } v \in \mathbb{R}^d, \ q \le p. \tag{4.2}$$

Matrices will be denoted in boldface throughout, and $\|\cdot\|_p$ applied to a matrix-valued variable $\mathbf{M}$ is the Schatten-p norm, i.e. the ℓ_p norm of the singular values of $\mathbf{M}$.

Optimization. In the following discussion, fix some $f : \mathcal{X} \to \mathbb{R}$. We say f is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$ if for all $x, x' \in \mathcal{X}$, $|f(x) - f(x')| \leq G \|x - x'\|_{\mathcal{X}}$. We say f is μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$ if for all $x, x' \in \mathcal{X}$ and $t \in [0, 1]$,

$$f\left(tx + (1-t)y\right) \leq tf(x) + (1-t)f(y) - \frac{\mu t(1-t)}{2} \|x - y\|_{\mathcal{X}}^2 .$$

Probability. For two densities π, π', we define their total variation distance by $\|\pi - \pi'\|_{\mathrm{TV}} := \frac{1}{2}\int |\pi(x) - \pi'(x)|dx$ and (when the Radon-Nikodym derivative exists) their KL divergence by $D_{\mathrm{KL}}(\pi\|\pi') := \int \pi(x) \log \frac{\pi(x)}{\pi'(x)} dx$. We define the 2-Wasserstein distance by

$$W_2(\pi, \pi') = \inf_{\Gamma \in \Gamma(\pi, \pi')} \sqrt{\mathop{\mathbb{E}}_{(x,x') \sim \Gamma} \|x - x'\|_2^2},$$

where $\Gamma(\pi, \pi')$ is the set of couplings of π and π'. We note W_2 satisfies the following inequality.

Fact 4.2.1. *Let* $\mathrm{Lip}_2(f)$ *be the Lipschitz constant in the* ℓ_2 *norm of a function* f*. Then, for densities* π, π' *supported on* $\mathcal{X}$*,*

$$W_2(\pi, \pi') \geq \sup_{\mathrm{Lip}_2(f) \leq 1} \int_{\mathcal{X}} f(x)(\pi(x) - \pi'(x))dx.$$

Proof. This follows from the dual characterization of the 1-Wasserstein distance (which shows $\sup_{\mathrm{Lip}(f)\leq 1} \int_{\mathcal{X}} f(x)(\pi(x) - \pi'(x))dx = \inf_{\Gamma \in \Gamma(\pi,\pi')} \mathbb{E}_{(x,x')\sim\Gamma} \|x - x'\|_2$), and convexity of the square. □

We use $\propto$ to indicate proportionality, e.g. if π is a density and we write $\pi \propto \exp(-f)$, we mean $\pi(x) = \frac{\exp(-f)}{Z}$ where $Z := \int \exp(-f(x))dx$ and the integration is over the support of π.

We say that a measure π on $\mathbb{R}^d$ is logconcave if for any $\lambda \in (0,1)$ and compact $A, B \subset \mathbb{R}^d$,

$$\pi(\lambda A + (1-\lambda)B) \geq \pi(A)^\lambda \pi(B)^{1-\lambda}.$$

We have the following equivalent characterization of logconcave measures.

Proposition 4.2.2 ([Bor75b])**.** *Let π be a measure on $\mathbb{R}^d$. Let E be the least affine subspace containing the support of π, and let m_E be the Lebesgue measure in E. Then π is logconcave if and only if $d\pi = f dm_E$, f is nonnegative and locally integrable, and $-\log f : E \to \mathbb{R} \cup \{+\infty\}$ is convex.*

In particular, Proposition 4.2.2 shows that the measure of any subspace of E according to π is zero. If in the characterization of [Bor75b] the function $-\log f$ is affine, we say π is logaffine. Following [Bor75b], we analogously define strong logconcavity with respect to a norm.

Definition 4.2.3 (strong logconcavity)**.** Let π be a measure on $\mathbb{R}^d$. Let E be the least affine subspace containing the support of π, and let m_E be the Lebesgue measure in E. We say π is μ-strongly logconcave with respect to $\|\cdot\|_{\mathcal{X}}$ if $d\pi = f dm_E$, f is nonnegative and locally integrable, and $-\log f : E \to \mathbb{R} \cup \{+\infty\}$ is μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$.

Privacy. Throughout, $\mathcal{M}$ denotes a mechanism, and $\mathcal{D}$ denotes a dataset. We say $\mathcal{D}$ and $\mathcal{D}'$ are neighboring if they differ in one entry. We say a mechanism $\mathcal{M}$ satisfies (ε, δ)-differential privacy if it has output space Ω and for any neighboring $\mathcal{D}, \mathcal{D}'$,

$$\sup_{S \subseteq \Omega} \Pr[\mathcal{M}(\mathcal{D}) \in S] - \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in S] \leq \delta.$$

We define the privacy curve of two random variables X, Y supported on Ω by

$$\delta(X \| Y)(\varepsilon) := \sup_{S \subseteq \Omega} \Pr[Y \in S] - \exp(\varepsilon) \Pr[X \in S].$$

We say $\mathcal{M}$ has a privacy curve $\delta : \mathbb{R}_{\geq 0} \to [0,1]$ if for all neighboring $\mathcal{D}, \mathcal{D}'$, $\delta(\mathcal{M}(\mathcal{D}) \| \mathcal{M}(\mathcal{D}')) \leq \delta(\varepsilon)$. For any $\varepsilon \in \mathbb{R}_{\geq 0}$, it is clear that such a $\mathcal{M}$ is $(\varepsilon, \delta(\varepsilon))$-differentially private. We will

frequently compare to the privacy curve of a Gaussian, so we recall the following bound from prior work.

Fact 4.2.4 (Gaussian privacy curve, Lemma 6.3, [GLL22]). *Let $\delta \in (0, \frac{1}{2})$ and $\varepsilon > 0$. For any $|t| \le \sqrt{2\log\frac{1}{2\delta} + 2\varepsilon} - \sqrt{2\log\frac{1}{2\delta}} \le \frac{\varepsilon}{\sqrt{2\log\frac{1}{2\delta}}}$, $\delta(\mathcal{N}(0,1) \,\|\, \mathcal{N}(t,1))(\varepsilon) \le \delta$.*

We will use Fact 4.2.4 after deriving our Gaussian differential privacy guarantees [DRS21] for strongly logconcave densities in Theorem 4.1.3.

4.3 Gaussian differential privacy in general norms

In this section, we give a generalization of Theorem 4.1 of [GLL22], which demonstrates that a regularized exponential mechanism for (Euclidean norm) Lipschitz losses achieves privacy guarantees comparable to an analogous instance of the Gaussian mechanism. The proof from [GLL22] was specialized to the Euclidean setup; to show our more general result, we draw upon the localization technique from convex geometry [LS93, KLS95]. We provide the relevant localization tools we will use in Section 4.3.1, and prove our Gaussian differential privacy result in Section 4.3.2.

4.3.1 Localization

We recall the localization lemma from [FG04]. We remark that the statement in [FG04] is more refined than our statement (in that [FG04] gives a complete characterization of extreme points, whereas we give a superset), but the following form of the [FG04] result suffices for our purposes.

Proposition 4.3.1 (Theorem 1, [FG04]). *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, and let $f : \mathcal{X} \to \mathbb{R}$ be upper semi-continuous. Let $\mathcal{K}(f)$ be the set of logconcave densities ν supported in $\mathcal{X}$ satisfying $\int_{\mathcal{X}} f d\nu \ge 0$. The set of extreme points of $\mathrm{conv}(\mathcal{K}(f))$ satisfies one of the following.*

- *ν is a Dirac measure at $x \in \mathcal{X}$ such that $f(x) \ge 0$.*
- *ν is logaffine and supported on $[a,b] \subset \mathcal{X}$ such that $\int_{[a,b]} f d\nu = 0$.*

We next derive several extensions of Proposition 4.3.1.

Lemma 4.3.2 (Strongly logconcave localization)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, let $\beta : \mathcal{X} \to \mathbb{R}_{>0}$ be continuous, and let $f : \mathcal{X} \to \mathbb{R}$ be upper semi-continuous. Let $\mathcal{K}_{\mu,\beta}(f)$ be the set of probability densities π such that π is μ-strongly logconcave with respect to $\|\cdot\|_{\mathcal{X}}$ and supported in $\mathcal{X}$, such that $\pi' \propto \beta\pi$ is also μ-strongly logconcave, and $\int_{\mathcal{X}} f d\pi \geq 0$. The set of extreme points of $\mathrm{conv}(\mathcal{K}_{\mu,\beta}(f))$ satisfy one of the following.*

- *π is a Dirac measure at $x \in \mathcal{X}$ such that $f(x) \geq 0$.*
- *π is supported on $[a, b] \subset \mathcal{X}$.*

Proof. Clearly, Dirac measures at x with $f(x) \geq 0$ are extreme points, so it suffices to consider other extreme points. Given any extreme point π which is not a Dirac measure, we prove the least affine subspace containing the support of π has dimension one, i.e. denoting the least affine subspace containing the support of π by S, we prove $\dim S = 1$.

Assume for the sake of contradiction that $\dim S \geq 2$. There exists x_0 in the relative interior of the support of π and a two-dimensional subspace $E \subset \mathbb{R}^d$ such that $x_0 + E \subseteq S$. Let $S_1(E)$ be the unit circle in E, and for any $u \in S_1(E)$ denote $H_u = \{x \in S : \langle x - x_0, u\rangle = 0\}, H_u^+ = \{x \in S : \langle x - x_0, u\rangle \geq 0\}$ and $H_u^- = \{x \in S : \langle x - x_0, u\rangle \leq 0\}$. Finally, define $\varphi : S_1(E) \to \mathbb{R}$ by $\varphi(u) := \int_{H_u^+} f d\pi - \frac{1}{2}(\int f d\pi)$, such that $\varphi(u) = 0 \implies \int_{H_u^+} f d\pi = \frac{1}{2}\int f d\pi \geq 0$.

By Proposition 4.2.2, we know $\pi(H_u) = 0$. Moreover, φ is continuous since every hyperplane H_u has $\pi(H_u) = 0$. Since $\varphi(u) = -\varphi(-u)$, by the intermediate value theorem there exists $u_0 \in S_1(E)$ such that $\varphi(u_0) = 0$. We can hence rewrite π as a convex combination of its restrictions to $H_{u_0}^+$ and $H_{u_0}^-$, both of which are μ-strongly logconcave, and whose (renormalized) multiplications by β are also μ-strongly logconcave. Since $\varphi(u_0) = 0$ both of these restrictions belong to $\mathcal{K}_{\mu,\beta}(f)$, contradicting extremality of π. □

We briefly remark that the proof technique used in Lemma 4.3.2 is quite general, and the only property we used about $\mathcal{K}_{\mu,\beta}$ is that it is a subset of logconcave densities, and it is closed under restrictions to convex subsets. Similar arguments hold for other density

families with these properties. Further, we note that restrictions to compact sets are upper semi-continuous; it is straightforward to verify our applications of Lemma 4.3.2 satisfy the upper semi-continuity assumption.

We prove the following two technical lemmas using Lemma 4.3.2.

Lemma 4.3.3. *Following notation of Lemma 4.3.2, fix a continuous function $\alpha : \mathcal{X} \to \mathbb{R}$ and a subset $S \subset \mathcal{X}$. For any probability density π on $\mathcal{X}$, define the renormalized density $\tilde{\pi} \propto e^{-\alpha}\pi$. Finally, let*

$$g(\pi) := \Pr_{x\sim\tilde{\pi}}[x \in S] - e^{\varepsilon} \Pr_{x\sim\pi}[x \in S].$$

Then $\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) = \max_{\pi\in\mathcal{K}^*_{\mu,\beta}} g(\pi)$ where $\mathcal{K}^*_{\mu,\beta}$ is the subset of densities $\pi \in \mathcal{K}_{\mu,\beta}$ satisfying one of the following.

- *π is a Dirac measure at $x \in \mathcal{X}$.*
- *π is supported on $[a, b] \subset \mathcal{X}$.*

Proof. Let $\mathcal{K}_{\mu,\beta}(f) \subseteq \mathcal{K}_{\mu,\beta}$ be the set of $\pi \in \mathcal{K}_{\mu,\beta}$ such that $\int f d\pi \geq 0$. We have

$$\begin{aligned}
\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) &= \max_{\pi\in\mathcal{K}_{\mu,\beta}} \int_{x\in S} d\tilde{\pi}(x) - e^{\varepsilon} \int_{x\in S} d\pi(x) \\
&= \max_{\pi\in\mathcal{K}_{\mu,\beta}} \frac{\int_{x\in S} e^{-\alpha(x)} d\pi(x)}{\int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)} - e^{\varepsilon} \int_{x\in S} d\pi(x) \\
&= \max_{\pi\in\mathcal{K}_{\mu,\beta}} \max_{C\geq\int_{x\in\mathcal{X}} e^{-\alpha(x)}d\pi(x)} \frac{\int_{x\in S} e^{-\alpha(x)} d\pi(x)}{C} - e^{\varepsilon} \int_{x\in S} d\pi(x) \\
&= \max_{C} \max_{\pi\in\mathcal{K}_{\mu,\beta}(C-e^{-\alpha})} \int_{x\in\mathcal{X}} \left(\frac{e^{-\alpha(x)}}{C} - e^{\varepsilon} \right) \mathbf{1}_S(x) d\pi(x) \\
&= \max_{C} \max_{\pi\in\mathcal{K}_{\mu,\beta}(C-e^{-\alpha})^*} \int_{x\in\mathcal{X}} \left(\frac{e^{-\alpha(x)}}{C} - e^{\varepsilon} \right) \mathbf{1}_S(x) d\pi(x),
\end{aligned}$$

where $\mathcal{K}_{\mu,\beta}(C - e^{-\alpha})^*$ is the (super)set of extreme points of $\mathcal{K}_{\mu,\beta}(C - e^{-\alpha})$ given by the strongly logconcave localization lemma (Lemma 4.3.2). These candidate extreme points are Dirac measures at x such that $C \geq e^{-\alpha(x)}$, or are supported in $[a, b] \subset \mathcal{X}$. Hence,

$\mathcal{K}_{\mu,\beta}(C - e^{-\alpha})^* \subseteq \mathcal{K}^*_{\mu,\beta}$, and we conclude

$$\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) = \max_{C} \max_{\pi\in\mathcal{K}_{\mu,\beta}(C-e^{-\alpha})^*} \int_{x\in\mathcal{X}} \left(\frac{e^{-\alpha(x)}}{C} - e^{\varepsilon}\right) \mathbf{1}_S(x) d\pi(x) \tag{4.3}$$

$$\leq \max_{C} \max_{\pi\in\mathcal{K}_{\mu,\beta}(C-e^{-\alpha})^*} \int_{x\in\mathcal{X}} \left(\frac{e^{-\alpha(x)}}{\int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)} - e^{\varepsilon}\right) \mathbf{1}_S(x) d\pi(x) \tag{4.4}$$

$$\leq \max_{\pi\in\mathcal{K}^*_{\mu,\beta}} \int_{x\in\mathcal{X}} \left(\frac{e^{-\alpha(x)}}{\int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)} - e^{\varepsilon}\right) \mathbf{1}_S(x) d\pi(x) = \max_{\pi\in\mathcal{K}^*_{\mu,\beta}} g(\pi). \tag{4.5}$$

The first inequality used that $C \geq \int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)$ for $\pi \in \mathcal{K}_{\mu,\beta}(C - e^{-\alpha})^*$, and the second used that $\mathcal{K}_{\mu,\beta}(C - e^{-\alpha})^* \subseteq \mathcal{K}^*_{\mu,\beta}$ for any C. Since $\mathcal{K}^*_{\mu,\beta} \subseteq \mathcal{K}_{\mu,\beta}$, we have the claim. □

Lemma 4.3.4. *Following notation of Lemma 4.3.2, fix continuous function $\alpha : \mathcal{X} \to \mathbb{R}$ and upper semi-continuous function $f : \mathcal{X} \to \mathbb{R}$. For any probability density π on $\mathcal{X}$, define $\tilde{\pi} \propto e^{-\alpha}\pi$ to be a renormalized density on $\mathcal{X}$. Finally, let*

$$g(\pi) := \int_{x\in\mathcal{X}} f(x) d(\pi - \tilde{\pi})(x).$$

Then $\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) = \max_{\pi\in\mathcal{K}^*_{\mu,\beta}} g(\pi)$ where $\mathcal{K}^*_{\mu,\beta}$ is the subset of densities $\pi \in \mathcal{K}_{\mu,\beta}$ satisfying one of the following.

- *π is a Dirac measure at $x \in \mathcal{X}$.*
- *π is supported on $[a, b] \subset \mathcal{X}$.*

Proof. We follow the notation from Lemma 4.3.3. If π is a Dirac measure, $g(\pi) = 0$, so we only need to consider the case when $\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) > 0$. We have

$$\begin{aligned}
\max_{\pi\in\mathcal{K}_{\mu,\beta}} g(\pi) &= \max_{\pi\in\mathcal{K}_{\mu,\beta}} \int_{x\in\mathcal{X}} f(x) \left(1 - \frac{e^{-\alpha(x)}}{\int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)}\right) d\pi(x) \\
&= \max_{\pi\in\mathcal{K}_{\mu,\beta}} \max_{C \leq \int_{x\in\mathcal{X}} e^{-\alpha(x)} d\pi(x)} \int_{x\in\mathcal{X}} \left(f(x) - \frac{e^{-\alpha(x)} f(x)}{C}\right) d\pi(x) \\
&= \max_{C} \max_{\pi\in\mathcal{K}_{\mu,\beta}(e^{-\alpha}-C)} \int_{x\in\mathcal{X}} \left(f(x) - \frac{e^{-\alpha(x)} f(x)}{C}\right) d\pi(x)
\end{aligned}$$

$$= \max_{C} \max_{\pi \in \mathcal{K}_{\mu,\beta}(e^{-\alpha}-C)^*} \int_{x\in\mathcal{X}} \left(f(x) - \frac{e^{-\alpha(x)} f(x)}{C} \right) d\pi(x).$$

The remainder of the proof is analogous to Lemma 4.3.3. □

4.3.2 Gaussian differential privacy

Using an instantiation of the localization lemma, we prove Gaussian differential privacy in general norms by first reducing to one dimension and then using the result of [GLL22] to handle the one-dimensional case. Gaussian differential privacy was introduced by [DRS21] and is a useful tool to compare privacy curves. We first recall the (ℓ_2) Gaussian differential privacy result of [GLL22].

Proposition 4.3.5 (Theorem 4.1, [GLL22])**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, let $F, \widetilde{F} : \mathcal{X} \to \mathbb{R}$ be μ-strongly convex in $\|\cdot\|_2$, and let $P \propto \exp(-F)$ and $Q \propto \exp(-\widetilde{F})$. Suppose $\widetilde{F} - F$ is G-Lipschitz in $\|\cdot\|_2$. For all $\varepsilon \in \mathbb{R}_{\geq 0}$,*

$$\delta(P \parallel Q)(\varepsilon) \leq \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G}{\sqrt{\mu}}, 1\right)\right)(\varepsilon).$$

We next give a simple comparison result between norms.

Lemma 4.3.6. *For $f : \mathcal{X} \to \mathbb{R}$, fix $a, b \in \mathcal{X}$, and let $\tilde{f} : [a,b] \to \mathbb{R}$ be the restriction of f to $[a,b]$.*

1. *If f is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$, $\tilde{f}$ is $G \cdot \frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2}$-Lipschitz in $\|\cdot\|_2$.*

2. *If f is μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$, $\tilde{f}$ is $\mu \cdot \frac{\|b-a\|_{\mathcal{X}}^2}{\|b-a\|_2^2}$-strongly convex in $\|\cdot\|_2$.*

Proof. To see the first claim, let $c = a + r(b-a)$ and $d = a + s(b-a)$ for $s, r \in [0,1]$. We have by Lipschitzness of f that

$$\left|\tilde{f}(d) - \tilde{f}(c)\right| \leq G\,|s-r|\,\|b-a\|_{\mathcal{X}} = \left(G \cdot \frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2}\right) \cdot \|d-c\|_2 .$$

Similarly, to see the second claim, by strong convexity of f,

$$\begin{aligned}\tilde{f}\left(tc+(1-t)d\right) &\le t\tilde{f}(c)+(1-t)\tilde{f}(d)-\frac{\mu t(1-t)}{2}\left\|c-d\right\|_{\mathcal{X}}^2\\ &= t\tilde{f}(c)+(1-t)\tilde{f}(d)-\frac{\mu t(1-t)(r-s)^2}{2}\left\|a-b\right\|_{\mathcal{X}}^2\\ &= t\tilde{f}(c)+(1-t)\tilde{f}(d)-\left(\mu\cdot\frac{\left\|b-a\right\|_{\mathcal{X}}^2}{\left\|b-a\right\|_2^2}\right)\left(\frac{t(1-t)}{2}\left\|d-c\right\|_2^2\right).\end{aligned}$$

□

We now present our main result on Gaussian differential privacy with respect to arbitrary norms.

Theorem 4.1.3. *Let $\mathcal{X}\subset\mathbb{R}^d$ be compact and convex, let $F,\widetilde{F}:\mathcal{X}\to\mathbb{R}$ be μ-strongly convex in $\left\|\cdot\right\|_{\mathcal{X}}$, and let $P\propto\exp(-F)$ and $Q\propto\exp(-\widetilde{F})$. Suppose $\widetilde{F}-F$ is G-Lipschitz in $\left\|\cdot\right\|_{\mathcal{X}}$. For all $\varepsilon\in\mathbb{R}_{\ge 0}$,*

$$\delta(P\parallel Q)(\varepsilon)\le\delta\left(\mathcal{N}(0,1)\,\middle\|\,\mathcal{N}\left(\frac{G}{\sqrt{\mu}},1\right)\right)(\varepsilon).$$

Proof. Throughout this proof, fix some α which is G-Lipschitz in $\left\|\cdot\right\|_{\mathcal{X}}$ by assumption. We first claim that amongst all μ-strongly convex (in $\left\|\cdot\right\|_{\mathcal{X}}$) functions $F:\mathcal{X}\to\mathbb{R}$ such that $F+\alpha$ is also μ-strongly convex, defining $P\propto\exp(-F)$ and $Q\propto\exp(-(F+\alpha))$, some F maximizing $\delta(P\parallel Q)(\varepsilon)$ is either a Dirac measure or supported on $[a,b]\subset\mathcal{X}$. We will prove this by contradiction.

Suppose otherwise, and let F be a μ-strongly convex function that maximizes $\delta(P\parallel Q)(\varepsilon)$ defined above. Define $P\propto\exp(-F)$ and $Q\propto\exp(-(F+\alpha))$. Let $S^*\subseteq\mathcal{X}$ be the set achieving

$$\delta(P\parallel Q)(\varepsilon)=\Pr_{X\sim P}[X\in S^*]-\exp(\varepsilon)\Pr_{X\sim Q}[X\in S^*].$$

By Lemma 4.3.3, there is another μ-strongly logconcave π where the renormalized density $\propto\pi\exp(-\alpha)$ is also μ-strongly logconcave, such that (following notation of Lemma 4.3.3) $g(\pi)\ge g(P)$, where π is either a Dirac or supported on $[a,b]$. We conclude that $\delta(P\parallel Q)(\varepsilon)\le\delta(\pi\parallel\pi\exp(-\alpha))(\varepsilon)$ (since the maximizing set for π is at least as good as S^*), a contradiction.

It hence suffices to prove the theorem statement for $F, \widetilde{F}$, which are supported on some $[a, b] \in \mathcal{X}$. By Lemma 4.3.6, we have that $\widetilde{F} - F$ is $G \cdot \frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2}$-Lipschitz in $\|\cdot\|_2$ and $F, \widetilde{F}$ are $\mu \cdot \frac{\|b-a\|_{\mathcal{X}}^2}{\|b-a\|_2^2}$-strongly convex in $\|\cdot\|_2$. We conclude by Proposition 4.3.5 which shows

$$\begin{aligned}\delta(P \parallel Q)(\varepsilon) &\leq \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G}{\sqrt{\mu}} \cdot \frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2} \cdot \frac{\|b-a\|_2}{\|b-a\|_{\mathcal{X}}}, 1\right)\right)(\varepsilon) \\ &= \delta\left(\mathcal{N}(0,1) \,\middle\|\, \mathcal{N}\left(\frac{G}{\sqrt{\mu}}, 1\right)\right)(\varepsilon).\end{aligned}$$

□

Our proof strategy is a reduction to an application of Proposition 4.3.5 in one dimension. It is an interesting open question to obtain a simpler direct proof of Proposition 4.3.5 in the one-dimensional setting (without using the machinery of [GLL22]), which is tight up to constant factors.

4.4 Private ERM and SCO in general norms

In this section, we derive our results for private ERM and SCO in general norms. We will state our results for private ERM (Section 4.4.1) and SCO (Section 4.4.2) with respect to an arbitrary compact convex subset $\mathcal{X}$ of a d-dimensional normed space, satisfying Assumption 4.1.1. We then use this to derive guarantees for a variety of settings of import in Section 4.4.3.

4.4.1 Private ERM under Assumption 4.1.1

To develop our private ERM algorithms, we recall the following risk guarantee from [DKL18] of sampling from Gibbs distributions (improving upon [KV06, BST14]).

Proposition 4.4.1 ([DKL18], Corollary 1)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, let $F : \mathcal{X} \to \mathbb{R}$ be convex, and let $k > 0$. If $\nu \propto \exp(-kF)$,*

$$\mathop{\mathbb{E}}_{x \sim \nu}[F(x)] \leq \min_{x \in \mathcal{X}} F(x) + \frac{d}{k}.$$

We conclude by a simple combination of Proposition 4.4.1 (providing a risk guarantee)

and Theorem 4.1.3 (providing a privacy guarantee), which yields our main result on private ERM.

Theorem 4.4.2 (Private ERM)**.** *Under Assumption 4.1.1 and following notation (4.1), drawing a sample x from the density $\nu \propto \exp(-k(F_{\mathcal{D}} + \mu r))$ for*

$$k = \frac{\sqrt{d}n\varepsilon}{G\sqrt{2\Theta \log \frac{1}{2\delta}}}, \; \mu = \frac{G\sqrt{2d \log \frac{1}{2\delta}}}{\sqrt{\Theta}n\varepsilon},$$

is (ε, δ)-differentially private, and produces x such that

$$\mathop{\mathbb{E}}_{x\sim\nu}[F_{\mathcal{D}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{D}}(x) \le G\sqrt{\Theta} \cdot \frac{\sqrt{8d \log \frac{1}{2\delta}}}{n\varepsilon}.$$

Proof. Let $F_{\mathcal{D}'}$ be the realization of (4.1) when $\mathcal{D}$ is replaced with a neighboring dataset $\mathcal{D}'$ which agrees in all entries except some sample $s_i' \neq s_i$. By Assumption 4.1.1, we have $k(F_{\mathcal{D}} - F_{\mathcal{D}'})$ is $\frac{kG}{n}$-Lipschitz, and both $k(F_{\mathcal{D}} + \mu r)$ and $k(F_{\mathcal{D}'} + \mu r)$ are $k\mu$-strongly convex (all with respect to $\|\cdot\|_{\mathcal{X}}$). Hence, combining Theorem 4.1.3 and Fact 4.2.4 shows the mechanism is (ε, δ)-differentially private, since

$$\mu = \frac{2G^2 k \log \frac{1}{2\delta}}{n^2\varepsilon^2} \implies \frac{G\sqrt{k}}{n\sqrt{\mu}} \le \frac{\varepsilon}{\sqrt{2 \log \frac{1}{2\delta}}}. \tag{4.6}$$

Let $x_{\mathcal{D}}^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{D}}(x)$. We obtain the risk guarantee by the calculation (see Proposition 4.4.1)

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\nu}[F_{\mathcal{D}}(x)] &\le F_{\mathcal{D}}(x_{\mathcal{D}}^\star) + \left(\mu r(x_{\mathcal{D}}^\star) - \mathop{\mathbb{E}}_{x\sim\nu} \mu r(x)\right) + \frac{d}{k} \\
&\le F_{\mathcal{D}}(x_{\mathcal{D}}^\star) + \mu\Theta + \frac{d}{k}
\end{aligned}$$

and plugging in our choices of μ and k. □

4.4.2 Private SCO under Assumption 4.1.1

We first give a generic comparison result between population risk and empirical risk under Assumption 4.1.1. To do so, we use two helper results from prior work. The first was derived in [GLL22] by combining a transportation inequality and a log-Sobolev inequality (see e.g. [OV00]).

Proposition 4.4.3 ([GLL22], Theorem 6.7, Lemma 6.8)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be compact and convex, let $F, \widetilde{F} : \mathcal{X} \to \mathbb{R}$ be μ-strongly convex in $\|\cdot\|_2$, and let $P \propto \exp(-F)$ and $Q \propto \exp(-\widetilde{F})$. Suppose $\widetilde{F} - F$ is H-Lipschitz in $\|\cdot\|_2$. Then, $W_2(P, Q) \leq \frac{H}{\mu}$.*

Corollary 4.4.4. *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, and let $\alpha, f : \mathcal{X} \to \mathbb{R}$ be H-Lipschitz and G-Lipschitz respectively in $\|\cdot\|_{\mathcal{X}}$. Let $\mathcal{K}_{\mu,\exp(-\alpha)}$ be the set of densities π over $\mathcal{X}$ such that π is μ-strongly logconcave with respect to $\|\cdot\|_{\mathcal{X}}$, and $\tilde{\pi} \propto \pi \exp(-\alpha)$ is also μ-strongly logconcave. For any $\pi \in \mathcal{K}_{\mu,\exp(-\alpha)}$ define $g(\pi) := \int_{\mathcal{X}} f(x) d(\pi - \tilde{\pi})(x)$ where $\tilde{\pi} \propto \pi \exp(-\alpha)$. Then, $g(\pi) \leq \frac{GH}{\mu}$.*

Proof. By Lemma 4.3.4 (and following its notation), it suffices to show $g(\pi) \leq \frac{GH}{\mu}$ for all $\pi \in \mathcal{K}^*_{\mu,\exp(-\alpha)}$. Clearly this is true for a Dirac measure π as then $g(\pi) = 0$, so consider the other case where π is supported on $[a, b]$, such that $\pi \propto \exp(-F)$ and F is μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$. Further, define $\widetilde{F} = F + \alpha$, so that $\widetilde{F}$ is also μ-strongly convex and supported on $[a, b]$.

By Lemma 4.3.6, restricting to $[a, b]$, F and $\widetilde{F}$ are $\mu \cdot \frac{\|b-a\|_{\mathcal{X}}^2}{\|b-a\|_2^2}$-strongly convex in $\|\cdot\|_2$, $F - \widetilde{F}$ is $H \cdot \frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2}$-Lipschitz in ℓ_2 and f is $\frac{\|b-a\|_{\mathcal{X}}}{\|b-a\|_2}$-Lipschitz in $\|\cdot\|_2$. Hence, where the inequalities are by Fact 4.2.1 and Proposition 4.4.3 respectively,

$$g(\pi) = \int_{\mathcal{X}} f(x) d(\pi - \tilde{\pi})(x) \leq G W_2(\pi, \tilde{\pi}) \leq \frac{GH}{\mu}.$$

□

The second relates the population risk to the empirical risk on an independent sample.

Proposition 4.4.5 (Lemma 7, [BE02])**.** *Suppose $\mathcal{D} = \{s_i\}_{i \in [n]}$ is drawn independently from $\mathcal{P}$, let $s \sim \mathcal{P}$ be drawn independently from $\mathcal{D}$, and let $\mathcal{D}' := \{s\} \cup \{s_i\}_{i \in [n] \setminus \{1\}}$ be $\mathcal{D}$ where*

s_1 is swapped with s. Then, for any symmetric[3] *mechanism $\mathcal{M} : \mathrm{supp}(\mathcal{P})^n \to \mathbb{R}^d$,*

$$\mathbb{E}\left[F_{\mathcal{P}}(\mathcal{M}(\mathcal{D})) - F_{\mathcal{D}}(\mathcal{M}(\mathcal{D}))\right] = \mathbb{E}\left[f(\mathcal{M}(\mathcal{D}); s) - f(\mathcal{M}(\mathcal{D}'); s)\right],$$

where expectations are over $\mathcal{M}$ and the randomness used in producing $\mathcal{D}$ and s.

By applying Corollary 4.4.4 and Proposition 4.4.5 (which bound the generalization error of our mechanism), we provide the following extension of Theorem 4.4.2, our main result on private SCO.

Theorem 4.4.6 (Private SCO)**.** *Under Assumption 4.1.1 and following notation (4.1), drawing a sample x from the density $\nu \propto \exp(-k(F_{\mathcal{D}} + \mu r))$ for*

$$k = \sqrt{\frac{d + C_2}{C_1}},\ \mu = \frac{2G^2 k \log\frac{1}{2\delta}}{n^2\varepsilon^2},\ C_1 := \frac{2G^2\Theta\log\frac{1}{2\delta}}{n^2\varepsilon^2},\ C_2 := \frac{n\varepsilon^2}{2\log\frac{1}{2\delta}},$$

is (ε, δ)-differentially private, and produces x such that

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n, x\sim\nu}\left[F_{\mathcal{P}}(x)\right] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) \le G\sqrt{\Theta}\cdot\left(\frac{\sqrt{8d\log\frac{1}{2\delta}}}{n\varepsilon} + \sqrt{\frac{8}{n}}\right).$$

Proof. For the given choice of k, μ, the privacy proof follows identically to Theorem 4.4.2, so we focus on the risk proof. We follow the notation of Proposition 4.4.5 and let $s \sim \mathcal{P}$ independently from $\mathcal{D}$. By exchanging the expectation and minimum and using that $\mathbb{E}_{\mathcal{D}\sim\mathcal{P}^n} F_{\mathcal{D}} = F_{\mathcal{P}}$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n, x\sim\nu}\left[F_{\mathcal{P}}(x)\right] - \min_{x\sim\mathcal{X}} F_{\mathcal{P}}(x) &\le \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[F_{\mathcal{P}}(x)\right] - \min_{x\in\mathcal{X}}\left[F_{\mathcal{D}}(x)\right]\right] \\
&\le \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)\right]\right] \\
&\quad + \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[F_{\mathcal{D}}(x)\right] - \min_{x\in\mathcal{X}}\left[F_{\mathcal{D}}(x)\right]\right] \\
&\le \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)\right]\right] + \mu\Theta + \frac{d}{k},
\end{aligned}$$

[3]Here, a symmetric mechanism is one which only depends on the set of inputs rather than their order.

where we bounded the empirical risk in the proof of Theorem 4.4.2. Next, let ν' be the density $\propto \exp(-k(F_{\mathcal{D}'} + \mu r))$. Our mechanism is symmetric, and hence by Proposition 4.4.5,

$$\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)\right] = \mathbb{E}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[f(x;s)\right] - \mathop{\mathbb{E}}_{x\sim\nu'}\left[f(x;s)\right]\right]$$

where the outer expectations are over the randomness of drawing $\mathcal{D}, s$. Finally, for any fixed realization of $\mathcal{D}, s$, the densities ν, ν' satisfy the assumption of Corollary 4.4.4 with $H = \frac{G}{n}$, and $f(\cdot; s)$ is G-Lipschitz, so Corollary 4.4.4 shows that

$$\mathop{\mathbb{E}}_{x\sim\nu}\left[f(x;s)\right] - \mathop{\mathbb{E}}_{x\sim\nu'}\left[f(x;s)\right] \le \frac{G^2}{n\mu}.$$

Combining the above three displays bounds the population risk by

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n, x\sim\nu}\left[F_{\mathcal{P}}(x)\right] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) &\le \frac{G^2}{n\mu} + \mu\Theta + \frac{d}{k} \\ &= C_1 k + \frac{C_2 + d}{k},\end{aligned}$$

for our given value of μ. The conclusion follows by optimizing over k yielding a risk of $2\sqrt{C_1(C_2+d)}$, and using the scalar inequality $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for nonnegative a, b. □

4.4.3 Applications

To derive our private optimization algorithms for ℓ_p-norm and Schatten-p norm geometries, we recall the following results on the existence of bounded strongly convex regularizers.

Proposition 4.4.7 ([BCL94])**.** *For $1 < p \le 2$, letting $\|\cdot\|_p$ be the ℓ_p norm of a vector, $r(v) := \frac{1}{2(p-1)}\|v\|_p^2$ is 1-strongly convex in $\|\cdot\|_p$. Similarly, for $1 < p \le 2$, letting $\|\cdot\|_p$ be the Schatten-p norm of a matrix, $r(\mathbf{M}) := \frac{1}{2(p-1)}\|\mathbf{M}\|_p^2$ is 1-strongly convex in $\|\cdot\|_p$.*

We state a useful result on efficiently sampling from Lipschitz, strongly logconcave densities under value oracle access given by [GLL22] (building upon the framework of [LST21b]). We slightly specialize the result of [GLL22] by giving a rephrasing sufficient for our purposes.

Proposition 4.4.8 ([GLL22], Theorem 2.3)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex with $\mathrm{diam}_{\|\cdot\|_2}(\mathcal{X}) \le D$. Let $\mathcal{D} = \{s_i\}_{i\in[n]}$ and let $\widetilde{F}_{\mathcal{D}}(x) = \frac{1}{n}\sum_{i\in[n]} f(x; s_i) + \psi(x)$ such that all $f(\cdot; s_i) : \mathcal{X} \to \mathbb{R}$ are G-Lipschitz in $\|\cdot\|_2$ and convex, and $\psi(x) : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex in $\|\cdot\|_2$. For $\delta \in (0, \frac{1}{2})$, we can generate a sample within total variation δ of the density $\propto \exp(-\widetilde{F}_{\mathcal{D}})$ in N value queries to some $f(\cdot; s_i)$ and samples from densities $\propto \exp\left(-\psi - \frac{1}{2\eta}\|\cdot - v\|_2^2\right)$ for some $\eta > 0$, $v \in \mathbb{R}^d$, where*

$$N = O\left(\frac{G^2}{\mu}\log^2\left(\frac{G^2(D^2+\mu^{-1})d}{\delta}\right)\right).$$

ℓ_p norms. We state our results on private convex optimization under ℓ_p geometry. As a preliminary, we combine norm equivalence bounds (4.2) and Proposition 4.4.8 to give the following algorithmic result on sampling from a logconcave distribution under value oracle access under ℓ_p geometry.

Proposition 4.4.9. *Let $p \ge 1$ and let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex with $\mathrm{diam}_{\|\cdot\|_p}(\mathcal{X}) \le D$. Let $\mathcal{D} = \{s_i\}_{i\in[n]}$ and let $\widetilde{F}_{\mathcal{D}}(x) = \frac{1}{n}\sum_{i\in[n]} f(x; s_i) + \psi(x)$ such that all $f(\cdot; s_i) : \mathcal{X} \to \mathbb{R}$ are G-Lipschitz in $\|\cdot\|_p$ and convex, and $\psi(x) : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex in $\|\cdot\|_p$. For $\delta \in (0, \frac{1}{2})$, we can generate a sample within total variation δ of the density $\propto \exp(-\widetilde{F}_{\mathcal{D}})$ in N value queries to some $f(\cdot; s_i)$ and samples from densities $\propto \exp\left(-\psi - \frac{1}{2\eta}\|\cdot - v\|_2^2\right)$ for some $\eta > 0$, $v \in \mathbb{R}^d$, where*

$$N = O\left(\frac{G^2 d^{\frac{2}{p}-1}}{\mu}\log^2\left(\frac{G^2(D^2+\mu^{-1})d}{\delta}\right)\right) \quad \text{if } p \in [1,2],$$
$$N = O\left(\frac{G^2 d^{1-\frac{2}{p}}}{\mu}\log^2\left(\frac{G^2(D^2+\mu^{-1})d}{\delta}\right)\right) \quad \text{if } p \in [2,\infty).$$

Proof. For $p \in [1,2]$, note that each $f(\cdot; s_i)$ is $d^{\frac{1}{p}-\frac{1}{2}}G$-Lipschitz in the ℓ_2 norm by combining (4.2) and the definition of Lipschitzness. Moreover, because the ℓ_p norm is larger than the ℓ_2 norm, ψ remains μ-strongly convex in the ℓ_2 norm. The diameter D is only affected by poly(d) factors when converting norms, which is accounted for by the logarithmic term. Hence, the complexity bound follows by applying Proposition 4.4.8 under this change of parameters. For the other case of $p \in [2,\infty)$, the Lipschitz bound is G, and the strong

convexity bound is $d^{\frac{2}{p}-1}\mu$ by a similar argument. □

In the following discussion, we primarily focus on the value oracle query complexity of our samplers. Generic results on logconcave sampling (see e.g. [LV07], or more recent developments by [JLLV21, Che21a, KL22]) imply the samples from the densities $\propto \exp(-\psi - \frac{1}{2\eta}\|\cdot - v\|_2^2)$ can be performed in polynomial time, for all the ψ that are relevant in our applications (which are all squared ℓ_p distances). We expect samplers which run in nearly-linear time (in d) may be designed for applications where $\mathcal{X}$ is structured, such as an ℓ_p ball, but for brevity we omit this discussion.

Corollary 4.4.10. *Let $1 < p \le 2$ be a constant, and let $\varepsilon > 0$, $\delta \in (0,1)$. Let $\mathcal{X} \subset \mathbb{R}^d$ have $\operatorname{diam}_{\|\cdot\|_p}(\mathcal{X}) \le D$, and let $F_{\mathcal{P}} = \mathbb{E}_{s\sim\mathcal{P}}[f(\cdot;s)]$ where all $f(\cdot;s) : \mathbb{R}^d \to \mathbb{R}$ are convex and G-Lipschitz in $\|\cdot\|_p$. Finally, let $\mathcal{D} = \{s_i\}_{i\in[n]} \sim \mathcal{P}^n$ independently and $F_{\mathcal{D}} := \frac{1}{n}\sum_{i\in[n]} f(\cdot;s_i)$.*

1. *There is an (ε,δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{D}}(x) \le 2GD \cdot \frac{\sqrt{d\log\frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}} \tag{4.7}$$

using

$$O\left(\frac{n^2\varepsilon^2 d^{\frac{2}{p}-1}}{\log\frac{1}{\delta}}\log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot;s_i). \tag{4.8}$$

2. *There is an (ε,δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{M}}[F_{\mathcal{P}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) \le 2GD \cdot \left(\sqrt{\frac{1}{n(p-1)}} + \frac{\sqrt{d\log\frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}}\right). \tag{4.9}$$

using

$$O\left(\frac{n^2\varepsilon^2 d^{\frac{2}{p}-1}}{\log\frac{1}{\delta}}\log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot;s_i).$$

Proof. We will parameterize Assumption 4.1.1 with the function $r(x) := \frac{1}{2(p-1)}\|x - x_0\|_p^2$, where $x_0 \in \mathcal{X}$ is an arbitrary point, and strong convexity follows from Proposition 4.4.7. By assumption, we may set $\Theta = \frac{1}{2(p-1)}D^2$. The conclusions follow by combining Theorem 4.4.2,

Theorem 4.4.6. To obtain (ε, δ)-differential privacy, it suffices to run the mechanism with privacy level $\delta \leftarrow \frac{\delta}{2}$, run to total variation $\frac{\delta}{2}$ using Proposition 4.4.9, and take a union bound. For both ERM and SCO, note that our choices of k and μ satisfy the relation (4.6), namely $\frac{kG^2}{\mu} = O(n^2\varepsilon^2/\log\frac{1}{\delta})$. Since both the Lipschitz and strong convexity parameters are scaled up by k in our application of Proposition 4.4.9, we have the leading-order term is $\frac{kG^2}{\mu}$ which yields the conclusion. □

For any p such that $p - 1$ is bounded away from 0, Corollary 4.4.10 matches the information-theoretic lower bound of [BGN21] (and its subsequent sharpening by [LL22]). When this is not the case, we use norm equivalence (4.2) to obtain a weaker bound.

Corollary 4.4.11. *Let $\varepsilon > 0$, $\delta \in (0, 1)$. Let $\mathcal{X} \subset \mathbb{R}^d$ have $\operatorname{diam}_{\|\cdot\|_1}(\mathcal{X}) \le D$, and let $F_{\mathcal{P}} = \mathbb{E}_{s\sim\mathcal{P}}[f(\cdot; s)]$ where all $f(\cdot; s) : \mathbb{R}^d \to \mathbb{R}$ are convex and G-Lipschitz in $\|\cdot\|_1$. Finally, let $\mathcal{D} = \{s_i\}_{i\in[n]} \sim \mathcal{P}^n$ independently and $F_{\mathcal{D}} := \frac{1}{n}\sum_{i\in[n]} f(\cdot; s_i)$.*

1. *There is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{D}}(x) \le 6GD\sqrt{\log d} \cdot \frac{\sqrt{d \log \frac{1}{2\delta}}}{n\varepsilon} \tag{4.10}$$

   using

$$O\left(\frac{n^2\varepsilon^2 d}{\log\frac{1}{\delta}} \log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot; s_i). \tag{4.11}$$

2. *There is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{M}}[F_{\mathcal{P}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) \le 6GD\sqrt{\log d} \cdot \left(\sqrt{\frac{1}{n}} + \frac{\sqrt{d \log \frac{1}{2\delta}}}{n\varepsilon}\right) \tag{4.12}$$

   using

$$O\left(\frac{n^2\varepsilon^2 d}{\log\frac{1}{\delta}} \log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot; s_i).$$

Proof. We will parameterize Assumption 4.1.1 with the function $r(x) := \frac{e^2}{2(q-1)}\|x - x_0\|_q^2$, where $q = 1 + \frac{1}{\log d}$. By combining Proposition 4.4.7 (which shows r is e^2-strongly convex in

ℓ_q) and (4.2), we have that r is 1-strongly convex in ℓ_1. The remainder of the proof follows identically to Corollary 4.4.10. □

The term scaling as $\sqrt{\log d/n}$ in (4.12), namely the non-private population risk, is known to be optimal from existing lower bounds on SCO [DJW14]. Up to a $\sqrt{\log d}$ factor, the non-private empirical risk is optimal with respect to current private optimization lower bounds [BGN21, LL22].

Corollary 4.4.12. *Let $p \geq 2$, and let $\varepsilon > 0$, $\delta \in (0, 1)$. Let $\mathcal{X} \subset \mathbb{R}^d$ have $\operatorname{diam}_{\|\cdot\|_p}(\mathcal{X}) \leq D$, and let $F_{\mathcal{P}} = \mathbb{E}_{s\sim\mathcal{P}}[f(\cdot; s)]$ where all $f(\cdot; s) : \mathbb{R}^d \to \mathbb{R}$ are convex and G-Lipschitz in $\|\cdot\|_p$. Finally, let $\mathcal{D} = \{s_i\}_{i\in[n]} \sim \mathcal{P}^n$ independently and $F_{\mathcal{D}} := \frac{1}{n}\sum_{i\in[n]} f(\cdot; s_i)$.*

1. *There is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{D}}(x) \leq 2GD \cdot \frac{d^{1-\frac{1}{p}}\sqrt{\log\frac{1}{2\delta}}}{n\varepsilon} \tag{4.13}$$

using

$$O\left(\frac{n^2\varepsilon^2 d^{1-\frac{2}{p}}}{\log\frac{1}{\delta}} \log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot; s_i). \tag{4.14}$$

2. *There is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces x such that*

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{M}}[F_{\mathcal{P}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) \leq 2GD \cdot \left(\frac{d^{\frac{1}{2}-\frac{1}{p}}}{\sqrt{n}} + \frac{d^{1-\frac{1}{p}}\sqrt{\log\frac{1}{2\delta}}}{n\varepsilon}\right). \tag{4.15}$$

using

$$O\left(\frac{n^2\varepsilon^2 d^{1-\frac{2}{p}}}{\log\frac{1}{\delta}} \log^2\left(\frac{GDdn\varepsilon}{\delta}\right)\right) \text{ value queries to some } f(\cdot; s_i).$$

Proof. We will parameterize Assumption 4.1.1 with the function $r(x) := \frac{1}{2}\|x - x_0\|_2^2$. By combining Proposition 4.4.7 (which shows r is 1-strongly convex in ℓ_2, and hence also ℓ_p) and (4.2), we may set $\Theta = \frac{1}{2}d^{1-2/p}D^2$. The remainder of the proof follows identically to Corollary 4.4.10. □

Schatten-p norms. Our results extend immediately to matrix spaces equipped with Schatten-p norm geometries. We record our relevant results in the following.

Corollary 4.4.13. *Let $p \in [1, \infty)$, $\varepsilon > 0$, $\delta \in (0, 1)$, and let $d_1, d_2 \in \mathbb{N}$ have $d_1 > d_2$. Let $\mathcal{X} \subset \mathbb{R}^{d_1 \times d_2}$ have $\operatorname{diam}_{\|\cdot\|_p}(\mathcal{X}) \le D$, and let $F_{\mathcal{P}} = \mathbb{E}_{s \sim \mathcal{P}}[f(\cdot; s)]$ where all $f(\cdot; s) : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ are convex and G-Lipschitz in $\|\cdot\|_p$. Finally, let $\mathcal{D} = \{s_i\}_{i \in [n]} \sim \mathcal{P}^n$ independently and $F_{\mathcal{D}} := \frac{1}{n} \sum_{i \in [n]} f(\cdot; s_i)$.*

1. *For constant $1 < p \le 2$, there is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces $\mathbf{M}$ such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(\mathbf{M})] - \min_{\mathbf{M} \in \mathcal{X}} F_{\mathcal{D}}(\mathbf{M}) \le 2GD \cdot \frac{\sqrt{d_1 d_2 \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}},$$
$$\mathop{\mathbb{E}}_{\mathcal{D} \sim \mathcal{P}^n, \mathcal{M}}[F_{\mathcal{P}}(\mathbf{M})] - \min_{\mathbf{M} \in \mathcal{X}} F_{\mathcal{P}}(\mathbf{M}) \le 2GD \cdot \left(\sqrt{\frac{1}{n(p-1)}} + \frac{\sqrt{d_1 d_2 \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}} \right).$$

   The value oracle complexity of the algorithm is bounded as in (4.8) for $d \leftarrow d_2$ in the non-logarithmic term, and $d \leftarrow d_1$ in the logarithmic term.

2. *For $p = 1$, there is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces $\mathbf{M}$ such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(\mathbf{M})] - \min_{\mathbf{M} \in \mathcal{X}} F_{\mathcal{D}}(\mathbf{M}) \le 6GD\sqrt{\log d_2} \cdot \frac{\sqrt{d_1 d_2 \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}},$$
$$\mathop{\mathbb{E}}_{\mathcal{D} \sim \mathcal{P}^n, \mathcal{M}}[F_{\mathcal{P}}(\mathbf{M})] - \min_{\mathbf{M} \in \mathcal{X}} F_{\mathcal{P}}(\mathbf{M}) \le 6GD\sqrt{\log d_2} \cdot \left(\sqrt{\frac{1}{n}} + \frac{\sqrt{d_1 d_2 \log \frac{1}{2\delta}}}{n\varepsilon\sqrt{p-1}} \right).$$

   The value oracle complexity of the algorithm is bounded as in (4.11) for $d \leftarrow d_2$ in the non-logarithmic term, and $d \leftarrow d_1$ in the logarithmic term.

3. *For $p \ge 2$, there is an (ε, δ)-differentially private algorithm $\mathcal{M}$ which produces $\mathbf{M}$ such*

that

$$\mathop{\mathbb{E}}_{\mathcal{M}}[F_{\mathcal{D}}(\mathbf{M})] - \min_{\mathbf{M}\in\mathcal{X}} F_{\mathcal{D}}(\mathbf{M}) \le 2GD \cdot \frac{d_2^{\frac{1}{2}-\frac{1}{p}}\sqrt{d_1 d_2 \log\frac{1}{2\delta}}}{n\varepsilon},$$
$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n,\mathcal{M}}[F_{\mathcal{P}}(\mathbf{M})] - \min_{\mathbf{M}\in\mathcal{X}} F_{\mathcal{P}}(\mathbf{M}) \le 2GD \cdot \left(\frac{d_2^{\frac{1}{2}-\frac{1}{p}}}{\sqrt{n}} + \frac{d_2^{\frac{1}{2}-\frac{1}{p}}\sqrt{d_1 d_2 \log\frac{1}{2\delta}}}{n\varepsilon}\right).$$

The value oracle complexity of the algorithm is bounded as in (4.14) *for* $d \leftarrow d_2$ *in the non-logarithmic term, and* $d \leftarrow d_1$ *in the logarithmic term.*

Proof. The privacy and utility proofs follow identically to Corollaries 4.4.10, 4.4.11, and 4.4.12, where we use the second portion of Proposition 4.4.7 instead of the first. We note that the "dimension-dependent" term in the risk inherited from Proposition 4.4.1 scales as $d_1 d_2$ (the dimensionality of the matrix space). However, the terms in the risk due to the size of regularizers (inherited from the tradeoffs in (4.2), for $p = 1$ and $p > 2$) scales as a power of d_2, the maximum dimension of singular values. To obtain the value oracle complexity, we note that by definition of the Schatten norm, it satisfies the relationship (4.2) as well. Moreover, the Schatten-2 norm agrees with the vector ℓ_2 norm (when the matrix is flattened into a vector), since they are both the Frobenius norm. Hence, we may directly apply Proposition 4.4.8 after paying a norm conversion, in the same way as was done in Proposition 4.4.9. □

Remark on high-probability bounds. One advantage of using a sampling-based algorithm is an immediate high-probability bound which follows due to the good concentration of Lipschitz functions over samples from strongly logconcave distributions, stated below.

Lemma 4.4.14 (Concentration of Lipschitz functions, [Led99], Section 2.3 and [BL00], Proposition 3.1)**.** *Let* ℓ *be a* G*-Lipschitz function and* $X \sim \exp(-F)$ *for a* μ*-strongly convex function* F*, all with respect to the same norm* $\|\cdot\|_{\mathcal{X}}$*. For all* $t \ge 0$*,*

$$\Pr\left[\ell(X) - \mathbb{E}[\ell(X)] \ge t\right] \le \exp\left(-\frac{t^2\mu}{2G^2}\right).$$

In particular, we have demonstrated that the population and empirical risks (which are

Lipschitz) have good expectations. Naively combining Lemma 4.4.14 and our main results on the expectation utility bound then yields tight concentration around the mean in some parameter regimes, but we suspect the resulting bound is loose in general. We leave it as an interesting open problem to obtain tight high-probability bounds in all parameter regimes.

4.5 Private ERM and SCO under strong convexity

In this section, we derive our results for private ERM and SCO in general norms under the assumption that the sample losses are strongly convex. We will state our results for private ERM (Theorem 4.5.2) and SCO (Theorem 4.5.3) with respect to an arbitrary compact convex subset $\mathcal{X}$ of a d-dimensional normed space, satisfying the following Assumption 4.5.1.

Assumption 4.5.1. *We make the following assumptions.*

1. *There is a compact, convex subspace $\mathcal{X} \subset \mathbb{R}^d$ equipped with a norm $\|\cdot\|_{\mathcal{X}}$.*

2. *There is a set Ω such that for any $s \in \Omega$, there is a function $f(\cdot; s) : \mathcal{X} \to \mathbb{R}$ which is G-Lipschitz and μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$.*

Theorem 4.5.2 (Private ERM)**.** *Under Assumption 4.5.1 and following notation (4.1), drawing a sample x from the density $\nu \propto \exp(-kF_{\mathcal{D}})$ for*

$$k = \frac{n^2\varepsilon^2\mu}{2G^2 \log \frac{1}{2\delta}},$$

is (ε, δ)-differentially private, and produces x such that

$$\mathop{\mathbb{E}}_{x\sim\nu}[F_{\mathcal{D}}(x)] - \min_{x\in\mathcal{X}} F_{\mathcal{D}}(x) \le \frac{2dG^2 \log \frac{1}{2\delta}}{n^2\varepsilon^2\mu}.$$

Proof. Let $F_{\mathcal{D}'}$ be the realization of (4.1) when $\mathcal{D}$ is replaced with a neighboring dataset $\mathcal{D}'$ which agrees in all entries except some sample $s_i' \neq s_i$. By Assumption 4.5.1, we have $k(F_{\mathcal{D}} - F_{\mathcal{D}'})$ is $\frac{kG}{n}$-Lipschitz, and both $kF_{\mathcal{D}}$ and $kF_{\mathcal{D}'}$ are $k\mu$-strongly convex (all with respect to $\|\cdot\|_{\mathcal{X}}$). Hence, combining Theorem 4.1.3 and Fact 4.2.4 shows the mechanism is

(ε, δ)-differentially private, since

$$k = \frac{n^2\varepsilon^2\mu}{2G^2 \log \frac{1}{2\delta}} \implies \frac{G\sqrt{k}}{n\sqrt{\mu}} \le \frac{\varepsilon}{\sqrt{2 \log \frac{1}{2\delta}}}.$$

Let $x^\star_{\mathcal{D}} := \operatorname{argmin}_{x \in \mathcal{X}} F_{\mathcal{D}}(x)$. We obtain the risk guarantee by the calculation (see Proposition 4.4.1)

$$\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{D}}(x)] \le F_{\mathcal{D}}(x^\star_{\mathcal{D}}) + \frac{d}{k} \le F_{\mathcal{D}}(x^\star_{\mathcal{D}}) + \frac{2dG^2 \log \frac{1}{2\delta}}{n^2\varepsilon^2\mu}.$$

□

Theorem 4.5.3 (Private SCO)**.** *Under Assumption 4.5.1 and following notation (4.1), drawing a sample x from the density $\nu \propto \exp(-kF_{\mathcal{D}})$ for*

$$k = \frac{n^2\varepsilon^2\mu}{2G^2 \log \frac{1}{2\delta}}$$

is (ε, δ)-differentially private, and produces x such that

$$\mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n, x \sim \nu}[F_{\mathcal{P}}(x)] - \min_{x \in \mathcal{X}} F_{\mathcal{P}}(x) \le \frac{G^2}{n\mu}\left(1 + \frac{2d \log \frac{1}{2\delta}}{n\varepsilon^2}\right).$$

Proof. For the given choice k, μ, the privacy proof follows identically to Theorem 4.4.2, so we focus on the risk proof. We follow the notation of Proposition 4.4.5 and let $s \sim \pi$ independently from π. By exchanging the expectation and minimum and using that $\mathbb{E}_{\mathcal{D} \sim \pi^n} F_{\mathcal{D}} = F_{\mathcal{P}}$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n, x \sim \nu}[F_{\mathcal{P}}(x)] - \min_{x \sim \mathcal{X}} F_{\mathcal{P}}(x) &\le \mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n}\left[\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{P}}(x)] - \min_{x \in \mathcal{X}}[F_{\mathcal{D}}(x)]\right] \\
&\le \mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n}\left[\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)]\right] \\
&\quad + \mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n}\left[\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{D}}(x)] - \min_{x \in \mathcal{X}}[F_{\mathcal{D}}(x)]\right] \\
&\le \mathop{\mathbb{E}}_{\mathcal{D} \sim \pi^n}\left[\mathop{\mathbb{E}}_{x \sim \nu}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)]\right] + \frac{d}{k},
\end{aligned}$$

where we bounded the empirical risk in the proof of Theorem 4.5.2. Next, let ν' be the density $\propto \exp(-kF_{\mathcal{D}'})$. Our mechanism is symmetric, and hence by Proposition 4.4.5,

$$\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)\right] = \mathbb{E}\left[\mathop{\mathbb{E}}_{x\sim\nu}\left[f(x;s)\right] - \mathop{\mathbb{E}}_{x\sim\nu'}\left[f(x;s)\right]\right]$$

where the outer expectations are over the randomness of drawing $\mathcal{D}, s$. Finally, for any fixed realization of $\mathcal{D}, s$, the densities ν, ν' satisfy the assumption of Corollary 4.4.4 with $H = \frac{G}{n}$, and $f(\cdot; s)$ is G-Lipschitz, so Corollary 4.4.4 shows that

$$\mathop{\mathbb{E}}_{x\sim\nu}\left[f(x;s)\right] - \mathop{\mathbb{E}}_{x\sim\nu'}\left[f(x;s)\right] \le \frac{G^2}{n\mu}.$$

Combining the above three displays bounds the population risk by

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\pi^n, x\sim\nu}\left[F_{\mathcal{P}}(x)\right] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x) \le \frac{G^2}{n\mu} + \frac{d}{k} = \frac{G^2}{n\mu}\left(1 + \frac{2d\log\frac{1}{2\delta}}{n\varepsilon^2}\right).$$

for our given value of k. □

Chapter 5

ALGORITHMIC ASPECTS OF THE LOG-LAPLACE TRANSFORM AND A NON-EUCLIDEAN PROXIMAL SAMPLER

5.1 Introduction

The development of samplers for continuous distributions, under weak oracle access to the corresponding densities, has seen a flurry of recent research activity. For applications in settings inspired by machine learning or computational statistics, this development has in large part built upon connections between sampling and continuous optimization. Inspired by perspectives on sampling as optimization in the space of measures [JKO98] and starting with pioneering work of [Dal17b], a long sequence of results, e.g. [Dal17a, CCBJ18, DCWY19, DM19, CV19, DMM19, SL19, CDWY20, LST20, CLA+21], has used analysis techniques from convex optimization to bound the convergence rates of sampling algorithms for densities. We refer the reader to the survey [Che23] for a more complete account, but note in almost all cases, the focus has been on sampling from densities satisfying regularity assumptions stated in the Euclidean (ℓ_2) norm, e.g. ℓ_2-bounded derivatives.

The theory of continuous optimization under regularity assumptions stated for non-Euclidean geometries has played an important role in algorithm design. These geometries naturally arise when the optimization problem is over a structured constraint set, such as an ℓ_p ball or a polytope. In diverse applications such as learning from experts [AHK12], sparse recovery [CRT06], multi-armed bandits [BC12], matrix completion [ANW10], fair resource allocation [DFO20], and robust PCA [JLT20], first-order mirror descent techniques for ℓ_p or Schatten-p geometries have been a remarkable success story. Beyond these applications, the theory of self-concordant barriers (and the Riemannian geometries induced by their Hessians) has been greatly influential to the theory of convex programming and interior

point methods [NT02, Nem04].[1]

Non-Euclidean samplers. A natural direction for building the theory of logconcave sampling (the analog of convex optimization) is thus to develop samplers which can handle non-Euclidean regularity assumptions and constraint sets. Unfortunately, progress in this direction has relatively lagged behind optimization counterparts, as discretization tools which work well in the Euclidean case do not readily generalize. Briefly (with an extended discussion deferred to Section 5.1.3), most prior attempts at giving non-Euclidean samplers have focused on analyzing variants of the *mirrored Langevin dynamics*, building upon the ubiquitous mirror descent algorithm in optimization [NY83]. The key idea of mirror descent is to choose a regularizer $\varphi : \mathcal{X} \to \mathbb{R}$ over a constraint set $\mathcal{X}$, such that φ is strongly convex in an appropriate (possibly non-Euclidean) norm $\|\cdot\|_{\mathcal{X}}$. The regularizer φ is then used to define iterative methods for optimizing functions f with regularity in $\|\cdot\|_{\mathcal{X}}$.

The sampling analog of this non-Euclidean generalization is to extend the *Langevin dynamics*, a stochastic process inherently catered to the ℓ_2 geometry, to use Brownian motion reweighted by the Hessian of a regularizer φ. This process, which we call the mirrored Langevin dynamics (MLD), was introduced recently by [ZPFP20] (see also [HKRC18] for an earlier incarnation). Several follow-up works attempted to bound convergence rates for discretizations of the MLD process, e.g. [AC21, Jia21, LTVW22]. Unfortunately, many of these analyses have imposed rather strong conditions on φ beyond strong convexity, e.g. a "modified self-concordance" assumption used in [ZPFP20, Jia21, LTVW22] which (to our knowledge) is not known to be satisfied by standard regularizers. Even more problematically, these analyses (as well as an empirical evaluation by [Jia21]) suggest that without strong relative regularity assumptions between the target density and φ, naive discretizations of MLD inherently do not converge to the target even in the limit. A notable exception is the work of [AC21], which circumvented both issues (the modified self-concordance assumption and a biased limit) using a different MLD discretization; however, it is not always clear that this discretization is feasible for standard choices of φ and $\mathcal{X}$.

[1] Self-concordance requires that the second derivative of a function is stable to perturbations which are measured in the induced norm. For notation and definitions used throughout the paper, see Section 5.2.

An alternative to directly discretizing MLD is to use a filter to control bias, akin to the MALA or Metropolized HMC algorithms which are well-studied in the Euclidean case [Bes94, RT96, BRH12, DCWY19, CDWY20, LST20]. However, here too generalizing existing analyses runs into obstacles: for example, typical analyses of MALA and Metropolized HMC rely on bounding the conductance of random walks via isoperimetric inequalities on the target distribution. Prior isoperimetry bounds appear to be tailored to the ℓ_2 geometry and properties of Gaussians (the basic strongly logconcave distribution in Euclidean settings). Potentially due to this difficulty, to our knowledge no general-purpose extension of MALA or its variants to non-Euclidean norms exists in the literature.[2]

Proximal samplers. In this paper, we overcome these difficulties by following a third strategy for the design of efficient samplers: a proximal approach recently proposed by [LST21b]. To sample from a density π on $\mathbb{R}^d$ proportional to $\exp(-f)$, the algorithm of [LST21b] first extends the space to $\mathbb{R}^d \times \mathbb{R}^d$, and defines a joint density $\widehat{\pi}$ such that, for some parameter $\eta > 0$,

$$\mathrm{d}\widehat{\pi}(z) \propto \exp\left(-f(x) - \frac{1}{2\eta}\|x - y\|_2^2\right) \mathrm{d}z \text{ where } z = (x, y) \in \mathbb{R}^d \times \mathbb{R}^d. \tag{5.1}$$

It is straightforward to see that for any η, the x-marginal of $\widehat{\pi}$ is the original distribution π, and further [LST21b] shows that alternating sampling from the conditional distributions of $\widehat{\pi}$, i.e. $\widehat{\pi}(x \mid y)$ or $\widehat{\pi}(y \mid x)$, mixes rapidly. We give an extended discussion on recent activity on designing and harnessing proximal samplers building upon [LST21b] in Section 5.1.3, but mention that instantiations of the framework have resulted in state-of-the-art runtimes for many structured density families [CCSW22, LC22, GLL22]. Motivated by the success of proximal methods in the Euclidean setting, one goal of our work is to extend this technique to non-Euclidean geometries.

[2] We mention that in certain geometries induced by structured manifolds (discussed in part in Section 5.1.3), generalizations of MALA or Metropolized HMC have been previously proposed, e.g. [GC11, Bar20]. These works are motivated by related, but different, settings to the ones considered in this work (we mainly study norm regularity, akin to first-order convex optimization), and their focus is not on establishing non-asymptotic mixing time bounds.

Our approach. Our main insight is that a generalization of the strategy in [LST21b] induces a well-studied object in probability theory called the *log-Laplace transform* (LLT). Letting $\varphi : \mathbb{R}^d \to \mathbb{R}$ be a convex function in the dual space $y \in \mathbb{R}^d$, our generalization of (5.1) defines the joint density

$$\begin{aligned} \mathrm{d}\widehat{\pi}(z) &\propto \exp\left(-f(x) + (\langle x, y\rangle - \varphi(y) - \psi(x))\right) \mathrm{d}z, \\ \text{where } \psi(x) &:= \log\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y\right). \end{aligned} \tag{5.2}$$

The function ψ is called the LLT of φ, and it has an interpretation as a normalizing constant for induced densities $\mathcal{D}_x^\varphi$ on the dual space proportional to $\exp(\langle x, \cdot\rangle - \varphi)$. Indeed, $\mathcal{D}_x^\varphi$ is defined exactly so the x-marginal of $\widehat{\pi}$ is $\pi \propto \exp(-f)$. When $\eta = 1$ and φ, ψ are quadratics, this is exactly (5.1); we discuss the case of general η in Section 5.1.2. Moreover, the LLT is a well-studied mathematical object: it arises in probability theory as a *cumulant-generating function*, i.e. derivatives of the LLT yield cumulants of the induced distributions $\mathcal{D}_x^\varphi$, just as derivatives of the MGF yield moments.

The LLT famously appeared in Cramér's theorem on large deviations [Cra38], and its cumulant-generating properties have yielded fundamental concentration results in convex geometry [Kla06, EK11, KM12]. More recently, algorithmically-motivated properties of the LLT have been studied in settings such as optimization [BE19], where it was used to define an optimal self-concordant barrier, as well as connections to localization schemes for sampling from discrete distributions [CE22].

We continue this investigation by demonstrating new mathematical properties of the LLT with an algorithmic flavor, and showcasing uses of the LLT as a tool for continuous logconcave sampling. In particular, armed with a deeper understanding of the LLT, we overcome several of the aforementioned barriers to non-Euclidean sampler design and develop a generalized proximal sampler. We further give applications of our sampler to obtain new complexity results for non-Euclidean differentially private convex optimization, building upon a connection discovered by [GLL22, GLL+23]. We are optimistic that the LLT will find additional uses in sampler design (potentially beyond the proximal sampling framework, building upon the new properties we prove), and suggest a number of avenues

of future exploration to the community in Section 5.6.

5.1.1 Our results

In this section, we overview our results, which separate cleanly into three categories.

Algorithmic aspects of the LLT. It is well-known that the derivatives of the LLT at a point $x \in \mathbb{R}^d$ are *cumulants* of the induced density on $y \in \mathbb{R}^d$:

$$\mathrm{d}\mathcal{D}_x^\varphi(y) \propto \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y.$$

For example, $\nabla\psi(x) = \mathbb{E}_{y\sim\mathcal{D}_x^\varphi}[y]$, and $\nabla^2\psi(x)$ is the covariance of $\mathcal{D}_x^\varphi$. Further, it was shown in [BE19] that if ψ is the LLT of a convex function φ, then ψ is convex and self-concordant. Building upon these facts, in Section 5.3, we prove the following new properties of the LLT.

- *Strong convexity-smoothness duality.* Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$. We prove that if $\varphi : \mathbb{R}^d \to \mathbb{R}$ is L-smooth in the dual norm $\|\cdot\|_*$, its LLT $\psi : \mathbb{R}^d \to \mathbb{R}$ is $\frac{1}{L}$-strongly convex in $\|\cdot\|$.[3] This fact parallels a similar, well-known form of strong convexity-smoothness duality for Fenchel conjugates [Sha07, KST09]. Our proof does not require φ to be convex. We further show that the converse holds as well: a $\frac{1}{L}$-strongly convex φ has a L-smooth LLT.

- *Isoperimetry in the Hessian norm.* We prove a one-dimensional isoperimetric inequality for densities of the form $\exp(-\varphi)$, where $\varphi : \mathbb{R} \to \mathbb{R}$ is self-concordant and convex. By appealing to (a strong variant of) the localization lemma of [LS93], this proves that measures which are strongly logconcave with respect to convex and self-concordant $\varphi : \mathbb{R}^d \to \mathbb{R}$ satisfy a similar isoperimetric inequality in the Riemannian geometry induced by $\nabla^2\varphi$. Importantly, due to self-concordance of the LLT, this applies to strongly logconcave measures in an LLT.

[3] The constant factor 1 here is optimal, as demonstrated by quadratics.

- *Overlap of induced distributions $\mathcal{D}_x^\varphi$.* We provide a KL divergence bound on the distributions $\mathcal{D}_x^\varphi$ and $\mathcal{D}_{x'}^\varphi$ for x and x' which are close in the Riemannian distance induced by ψ. Combined with our isoperimetric inequality and a classical argument of [DFK91], this proves a lower bound on the conductance of an alternating sampler for densities of the form (5.2).

These new properties of the LLT suggest that it may find uses in designing samplers under non-Euclidean geometries beyond those explored in Sections 5.4 and 5.5 of our paper. For example, the LLT of a smooth function is strongly convex and self-concordant, which are exactly the properties required by the mirror Langevin discretization scheme of [AC21]. In optimization, regularizers φ for mirror descent typically only require strong convexity (and not self-concordance). However, controlling the evolution of the geometry induced by $\nabla^2\varphi$ is critical for discretizing MLD schemes, so imposing self-concordance (as opposed to more non-standard regularity such as the modified self-concordance of [ZPFP20, Jia21, LTVW22]) may be viewed as a minimal assumption. Problematically, standard strongly convex regularizers for mirror descent such as entropy or ℓ_p^2 *are not* self-concordant, so LLTs are a way of bridging this gap for sampling. Moreover, our new isoperimetric inequality and conductance bounds suggest that LLTs may find use in Metropolized sampling schemes, paving the way for non-Euclidean generalizations of MALA and its variants.

In some sense, our new duality result is a generic way of taking a strongly convex regularizer and transform it, via the *Fenchel transform* and the *log-Laplace transform*, to another regularizer which is strongly convex in the same norm, but also self-concordant. The first transform makes the function smooth in the dual [KST09], and the second effectively undoes this change. We will later discuss an application of this framework in improving the oracle complexity of the problem of private stochastic convex optimization in the ℓ_p geometry, using the LLT of the ℓ_q^2 regularizer.

Non-Euclidean proximal sampling. In Section 5.4, we build upon these aforementioned tools to analyze the mixing time of an alternating scheme for sampling densities π on convex, compact $\mathcal{X} \subset \mathbb{R}^d$ equipped with a norm $\|\cdot\|_{\mathcal{X}}$, where π is proportional to

$\exp\left(-F(x) - \eta\mu\psi(x)\right) \mathbb{1}_{\mathcal{X}}(x)$. Here, $F : \mathcal{X} \to \mathbb{R}$ is convex, $\eta, \mu > 0$ are tunable parameters, and ψ is the LLT of η-smooth $\varphi : \mathbb{R}^d \to \mathbb{R}$ in the dual norm $\|\cdot\|_{\mathcal{X}^*}$. We prove in Theorem 5.4.12 that alternately sampling from conditional distributions of the extended density on $z = (x, y) \in \mathcal{X} \times \mathbb{R}^d$ proportional to

$$\exp\left(-F(x) - \eta\mu\psi(x) + \left(\langle x, y\rangle - \varphi(y) - \psi(x)\right)\right) \mathbb{1}_{\mathcal{X}}(x) \tag{5.3}$$

has stationary distribution π, and converges in $\approx \frac{1}{\eta\mu}$ iterations for a warm start. More specifically, the convergence rate of our sampler depends polylogarithmically on both the warmness β of the point it is initialized with, and the inverse of the total variation error δ. The form of (5.3) is the same as (5.2), but we impose that f is $\eta\mu$-relatively strongly convex in ψ.

We first compare this result to the Euclidean proximal sampler of [LST21b], who proved a similar result for alternating sampling densities of the form (5.1). The main result of [LST21b] shows that if f is μ-strongly convex in the ℓ_2 norm, then alternating sampling from the marginals of (5.1) converges in $\approx \frac{1}{\eta\mu}$ iterations, also with polylogarithmic dependence on the target total variation error. Our result can be viewed as an extension of this result; instead of requiring μ-strong convexity in the ℓ_2 norm (which is equivalent to relative strong convexity with respect to the function $x \to \frac{1}{2}\|x\|_2^2$), we require μ-relative strong convexity in the function $\eta\psi$. In light of our duality result, $\eta\psi$ is 1-strongly convex in $\|\cdot\|_{\mathcal{X}}$, so it is the natural "unit" for measuring strong convexity.

We remark that the parameters η and μ play different roles: μ governs the strong logconcavity of the stationary distribution, and η controls the strong logconcavity of the x-conditional distribution of (5.3), which is tuned to govern the convergence rate of sampling from the conditional distribution. In particular, we further show that when F is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$, then as long as $\eta \lesssim G^{-2}$, the conditional sampling required by (5.3) can be performed in constant calls to a value oracle to F in expectation. This result holds even when F is a distribution over G-Lipschitz functions, and we only have sample access to this distribution. This extends a similar implementation of the marginal sampler required by [LST21b] for log-Lipschitz densities in the ℓ_2 norm, given by [GLL22]. The remaining

complexity of the marginal sampling depends on the structure of the chosen φ and $\mathcal{X}$, but is independent of F; we give a discussion of this aspect of our sampler in Sections 5.5.3 and 5.6.

One shortcoming of Theorem 5.5.8's rate is that it depends polylogarithmically on the warmness parameter. In contrast, the rate of [LST21b] depends *doubly logarithmically* on the warmness, which is important because in many sampling applications, standard starting distributions have warmness bounds growing exponentially in problem parameters such as the dimension d. We refer the reader to a discussion in Section 1.1 of [LST21a] on warmness assumptions under ℓ_2 geometry, which have created a $\approx \sqrt{d}$-sized gap on mixing time bounds for MALA, with and without a polynomially-bounded warm start [CLA+21, LST20]. We believe it is an interesting future direction to close this gap in warmness assumptions for our sampler in Section 5.4, analogously to the result of [LST21b]. Notably, there has been an ongoing exploration of new proof techniques for the convergence of proximal samplers by the community [CCSW22, CE22], and we are optimistic similar advancements can be made in non-Euclidean settings, discussed further in Section 5.1.3.

Zeroth-order private convex optimization. As the main application of our techniques, in Section 5.5 we design LLTs based on the smoothness of the function $\varphi_q(x) = \frac{p-1}{2}\|x\|_q^2$ in the norm ℓ_q, where $\frac{1}{p} + \frac{1}{q} = 1$ and $p \in [1, 2]$, $q \geq 2$. We show that the additive range of $\psi_{\eta,p}$,[4] the LLT of $\eta\varphi_q$ for $\eta \lesssim \frac{1}{d}$,[5] is bounded by $O(\frac{1}{(p-1)\eta})$ over the unit ℓ_p ball. This makes $\eta\psi_{\eta,p}$ competitive with the canonical choice of regularizer in ℓ_p norms for optimization, namely $r_p(x) := \frac{1}{2(p-1)}\|x\|_p^2$, which has the same additive range and strong convexity parameters as $\eta\psi_{\eta,p}$ (up to constants). We further build efficient value oracles and samplers for induced densities for $\psi_{\eta,p}$ in Section 5.5.3.

A critical difference between $\eta\psi$ and r_p, however, is that regularizing by a multiple of $\eta\psi$ admits efficient samplers via the machinery in Section 5.4; to our knowledge no similar technique is known for r_p. This difference is particularly important in the setting of *differentially private convex optimization*: see Problem 5.5.5 for a formal statement of the

[4] We use slightly different notation than in Section 5.5 for convenience of exposition here.

[5] This restriction is discussed further in Section 5.1.2, but does not bottleneck our privacy applications.

problem we study. Recently, [GLL+23] showed that to privately minimize either population or empirical risk for a distribution over convex functions which are Lipschitz in a (possibly non-Euclidean) norm $\|\cdot\|_{\mathcal{X}}$, it suffices to sample from a regularized density $\propto \exp(-k(f^{\text{erm}} + \mu r))$. Here, $f^{\text{erm}} = \frac{1}{n}\sum_{i\in[n]} f_i$ is the empirical risk over n samples $\{f_i\}_{i\in[n]}$, k, μ are tunable parameters, and r is a 1-strongly convex regularizer in $\|\cdot\|_{\mathcal{X}}$.

Our new sampling results show a demonstrable algorithmic advantage of using $\eta\psi_{\eta,p}$ as a regularizer for ℓ_p geometries, as opposed to r_p. In Theorem 5.5.8, we give algorithms for private convex optimization matching the state-of-the-art excess risk bounds for private convex optimization recently attained by [GLL+23] (who used r_p as their regularizer). Under a warm start, our new algorithms further improve the *value (zeroth-order) oracle* complexities of private convex optimization under ℓ_p regularity in dimension d compared to [GLL+23] by $\text{poly}(d)$ factors, i.e. the number of queries to $\{f_i\}_{i\in[n]}$ used. We also show these new value oracle complexities extend straightforwardly to improve private convex optimization over matrix spaces satisfying Schatten-p norm regularity.

We note that our results match (up to logarithmic factors) the value oracle complexities in the ℓ_2 setting obtained by [GLL22], for all ℓ_p norms where $p \in [1, 2]$. In Appendix 5.7, we extend lower bounds for stochastic optimization from [DJWW15, GLL22] to the ℓ_p setting to show the value oracle complexities of Theorems 5.4.12 and 5.5.8 are near-optimal, assuming a polynomially warm start.

5.1.2 Our techniques

Analogously to Section 5.1.1, in this section we split our discussion of our techniques into three parts.

Algorithmic aspects of the LLT. We first discuss our strong convexity-smoothness duality result. From a convex geometry perspective, smoothness of φ (with LLT ψ) ensures that the induced distributions $\propto \exp(\langle x, \cdot\rangle - \varphi)$ are heavy-tailed (because their log-densities cannot grow quickly), which means their variances are "large." We also know that $\nabla^2\psi$ is the covariance matrix of the induced distribution which means that $\nabla^2\psi$ should be lower-bounded. We formalize this using a version of the Cramér-Rao bound from [CP22]. An

older arXiv version of this paper contains a more elementary proof of this result inspired by differential privacy, achieving a worse constant of $\approx \frac{1}{12}$; the (optimal) improvement was suggested by Sam Power. Our converse proof is similar, and follows by applying the Brascamp-Lieb inequality [BL76].

To prove our isoperimetric inequality, we draw inspiration from a similar bound shown in Lemma 35 of [LV18], but for a family of convex functions φ satisfying a strange condition that φ'' was convex (which fortunately includes the log barrier function). Noticing that $-\log$ is self-concordant, we extend the [LV18] result to hold for all self-concordant functions. Further we show by a direct calculation that the KL divergence between the induced distributions of two nearby points x and x' is essentially the LLT ψ at one of the points, up to a linear term. This lets us use stability of the Hessian of self-concordance functions to demonstrate stability of nearby induced distributions, a key ingredient in proving conductance bounds by the machinery of [DFK91].

Non-Euclidean proximal sampling. Given the results of Section 5.3, establishing our main proximal sampling result Theorem 5.4.12 is fairly routine. Our algorithm consists of an "outer loop" and an "inner loop" for sampling from the x marginal of (5.3) which is stated and analyzed in Section 5.4.1. Our outer loop analysis is directly based on the mixing time-to-conductance reduction of [LS93] and the technique of [DFK91] to lower bound conductance, using facts from Section 5.3. Our inner loop handling functions F in (5.3) which are Lipschitz (or distributions over Lipschitz functions) is a small modification of a similar result in [GLL22]. The only property we need of the LLT is strong convexity: this implies a rejection sampler terminates quickly via the concentration of Lipschitz functions under strongly logconcave distributions (in any norm) [Led99, BL00].

We do note there is a design decision to be made on how to define "scaling up the LLT by $\frac{1}{\eta}$," unlike in the case of (5.1) where using the induced density $\mathcal{N}(x, \eta^{-1}\mathbf{I}_d)$ is natural. Given r, a 1-strongly convex function in $\|\cdot\|_{\mathcal{X}}$, and letting r^* be its (smooth) Fenchel conjugate, two natural ways of defining a scaled up induced distribution at x are to choose densities

$$\propto \exp\left(\langle x, y\rangle - \eta r^*(y) - \psi(x)\right), \tag{5.4}$$

or

$$\propto \exp\left(\frac{1}{\eta}\left(\langle x, y\rangle - r^*(y) - \psi(x)\right)\right). \tag{5.5}$$

The choice (5.4) clearly results in ψ which is $\Omega(\eta^{-1})$-strongly convex, rendering it suitable for our proximal sampling applications. It is not difficult to see that the second results in $\eta^{-1}\psi$ which is also $\Omega(\eta^{-1})$-strongly convex. More interestingly, plugging in $r = r^* = \frac{1}{2}\|\cdot\|_2^2$ makes (5.1) agree with (5.5) rather than (5.4). Unfortunately, the ψ which results from (5.5) is not self-concordant, as its Hessian scales with η^{-1} and its third derivative with η^{-2}. Our choice to use (5.4) has further implications, elaborated on next, but a deeper understanding of this discrepancy seems interesting.

Zeroth-order private convex optimization. As outlined in Section 5.1.1, the frameworks of [GLL22, GLL+23] show that to use our proximal sampler for ℓ_p norm private convex optimization, it suffices to design an LLT which has small additive range. Perhaps surprisingly, we exploit the *non-scale invariance* of LLT for this task: the LLT of $\eta\varphi$ does not behave like η^{-1} times the LLT of φ.[6] To see why this is helpful, consider the case when $\varphi = \frac{1}{2}\|\cdot\|_\infty^2$: then,

$$\psi(x) = \log\left(\int \exp\left(\langle x, y\rangle - \frac{1}{2}\|y\|_\infty^2\right) \mathrm{d}y\right).$$

Although one would hope $\psi(x)$ has additive range comparable to $\frac{1}{2}\|x\|_1^2$, the Fenchel conjugate of $\frac{1}{2}\|x\|_\infty^2$, it is not hard to show that $\psi(e_1) - \psi(0) = \Omega(\sqrt{d})$; we give a proof in Appendix 5.8. Intuitively, the ℓ_∞ radius of a typical point $\sim \exp(-\frac{1}{2}\|\cdot\|_\infty^2)$ is about $\sqrt{d}$, and a constant fraction of points on the surface of this ℓ_∞ ball have inner product with e_1 of $\Omega(\sqrt{d})$. This shows the additive range of ψ on the ℓ_1 ball is larger than $\frac{1}{2}\|\cdot\|_1^2$ by dimension-dependent factors.

We show that the non-scale invariance of (5.4) is actually helpful in controlling additive ranges. Specifically, letting ψ_η denote the LLT of $\eta\|x\|_q^2$, we show the additive range of $\eta\psi_\eta$ (a $\approx$ 1-strongly convex function) is $\approx \max(\eta, 1, \sqrt{d\eta})$. For sufficiently small η, this implies $\eta\psi_\eta$ is actually a much smaller regularizer than ψ; graciously, our differential privacy applications require $\eta \lesssim \frac{1}{d^2}$. We find it potentially useful to explore how generic this non-

[6] On the other hand, the Fenchel conjugate of $\eta\varphi$ is η^{-1} times the Fenchel conjugate of φ.

scale invariance of the LLT is.

5.1.3 Prior work

Non-Euclidean sampling. A recurring issue that arises in bounding the convergence rate of non-Euclidean samplers is that naive discretizations can result in significant error. As a result, most prior works either require strong assumptions or oracles for accurate discretization or adopt more sophisticated discretization methods that are difficult to analyze. For example, earlier in the introduction this was discussed for discretizations of MLD [ZPFP20, Jia21, AC21, LTVW22]. Part of the intrinsic difficulty of bounding discretized MLD lies in third-order error terms emerging from non-Euclidean geometries, which are hard to control under standard assumptions.

Under structured settings different than, but related to, those in this paper, an interesting alternative sampling strategy is discretizing Riemannian Langevin or Hamiltonian dynamics. For example, [GV22] studied the Riemmanian Langevin dynamics assuming access to an oracle to sample from Brownian motion on a manifold, whose complexity heavily depends on the manifold. Further, the convergence rate of Riemannian Hamiltonian Monte Carlo (RHMC) in polytopes was studied in [LV18], and a discretized version was analyzed in [KLSV22]; the results apply to a limited family of distributions, and the convergence rate is fairly large. For RHMC to converge to the correct target distribution, sophisticated discretization methods such as Implicit Midpoint Method are necessary. Though efficient in practice, these methods are challenging to analyze theoretically.

Proximal sampling. A long line of works has studied the use of proximal methods in sampling (inspired by optimization). Several considered proximal Langevin algorithms [Per16, BDMP17, Ber18, Wib19], which combine proximal methods and discretizations of Langevin dynamics. Further, [MFWB22] proposed a sampler based on a proximal sampling oracle. However, these algorithms required either stringent assumptions or a large mixing time. Recently, [LST21b] proposed a new proximal sampler overcoming many of the assumptions and efficiency issues in prior methods. Several works have focused on generalizing [LST21b] and applying it in different settings: [CCSW22] proved convergence results

using weaker assumptions than strong logconcavity. The framework has been used to obtain state-of-the-art samplers for various structured families, including smooth, composite, and finite-sum densities [LST21b] as well as non-smooth densities [GLL22, LC22].

Log-Laplace transform. The LLT is a powerful tool that emerges frequently in probability theory and convex geometry. Notably, [BE19, Che21b] showed that the Legendre-Fenchel dual of LLT of the uniform measure on a convex body in $\mathbb{R}^n$ is an n-self-concordant barrier, giving the first universal barrier for convex bodies with optimal self-concordance parameter. In [CE22], the LLT serves as one of the key ingredients of entropy conservation in localization schemes for sampling. In addition, the LLT shows up in the solution to the entropic optimal transport problem, where a KL divergence is added to regularize the optimal transport objective [CP22].

Private convex optimization. Differentially private convex optimization is one of the most extensively studied problems in the privacy literature and captures an increasing number of critical applications in various domains, including machine learning, statistics, and data analysis. There is a rich body of works on this topic [CM08, CMS11, KST12, BST14, WYX17, BFTGT19, FKT20], which have mainly focused on the Euclidean geometry, e.g. assuming the ℓ_2 diameter of the domain and ℓ_2 norms of gradients are bounded. Motivated by applications not captured by these assumptions, there has been growing interest in studying differentially private convex optimization in non-Euclidean geometries, as seen in [TTZ15, AFKT21, BGN21, HLL+22, GLL+23]. Of particular relevance, [GLL+23] develops an exponential mechanism based method attaining state-of-the-art excess risk bounds for ℓ_p and Schatten-p norms, which are matched by our algorithms in Section 5.5.

5.2 Preliminaries

General notation. In Section 2.1 only, $\widetilde{O}$, $\approx$, and $\lesssim$ hide logarithmic factors in problem parameters for expositional convenience. For $n \in \mathbb{N}$, $[n]$ refers to the naturals $1 \le i \le n$. We use $\mathcal{X}$ to denote a compact convex subset of $\mathbb{R}^d$. For all $p \ge 1$ including $p = \infty$, we let $\|\cdot\|_p$ applied to a vector argument denote the ℓ_p norm. We denote matrices in boldface and

when $\|\cdot\|_p$ is applied to a matrix argument it denotes the corresponding Schatten-p norm (ℓ_p norm of the singular values).

For any $\mathcal{X} \subset \mathbb{R}^d$ we let its indicator function (i.e. the function which is 1 on $\mathcal{X}$ and 0 otherwise) be denoted $\mathbb{1}_{\mathcal{X}}$. We will be concerned with optimizing functions $f : \mathcal{X} \to \mathbb{R}$, and $\|\cdot\|_{\mathcal{X}}$ refers to a norm on $\mathcal{X}$. We let $\mathcal{X}^*$ be the dual space to $\mathcal{X}$, and equip it with the dual norm $\|y\|_{\mathcal{X}^*} := \sup_{\|x\|_{\mathcal{X}}=1} x^\top y$. We let $\mathcal{N}(\mu, \mathbf{\Sigma})$ be the Gaussian density of given mean and covariance. For a positive definite matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$, we denote the induced norm by $\|v\|_{\mathbf{M}} := \sqrt{v^\top \mathbf{M} v}$. When making asymptotic statements we will typically assume the dimension d is at least a sufficently large constant, else we can pad and affect statements by at most constant factors.

Optimization. In the following, fix $f : \mathcal{X} \to \mathbb{R}$. We say f is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$ if for all $x, x' \in \mathcal{X}$, $|f(x) - f(x')| \le G \|x - x'\|_{\mathcal{X}}$. If f is differentiable, we say it is L-smooth in $\|\cdot\|_{\mathcal{X}}$ if for all $x, x' \in \mathcal{X}$, $\|\nabla f(x) - \nabla f(x')\|_{\mathcal{X}^*} \le L \|x - x'\|_{\mathcal{X}}$. Taylor expanding then shows $f(x') \le f(x) + \langle \nabla f(x), x' - x\rangle + \frac{L}{2}\|x - x'\|_{\mathcal{X}}^2$. We say f is m-relatively strongly convex in φ if $f - m\varphi$ is convex. For k-times differentiable f, $\nabla^k f(x)[v_1, v_2, \ldots, v_k]$ denotes the corresponding k^{th} order directional derivative at f. We say twice-differentiable f is m-strongly convex in $\|\cdot\|_{\mathcal{X}}$ if for all $x \in \mathcal{X}$, $v \in \mathbb{R}^d$, $\nabla^2 f(x)[v, v] \ge m\|v\|_{\mathcal{X}}^2$. We say convex $\varphi : \mathbb{R}^d \to \mathbb{R}$ is self-concordant if

$$\left|\nabla^3 \varphi(x)[h, h, h]\right| \le 2\left(\nabla^2 \varphi(x)[h, h]\right)^{\frac{3}{2}}, \text{ for all } x, h \in \mathbb{R}^d.$$

A key fact we use about self-concordant functions is that their Hessians are stable under small distances, where the distance is measured in the Hessian norm: see Lemma 5.2.2 for a formal statement.

Probability. For a density π supported on $\mathcal{X}$, we let $\pi(S) := \Pr_{x\sim\pi}[x \in S]$. For two densities μ, π, we define their total variation distance by $\|\mu - \pi\|_{\text{TV}} := \frac{1}{2}\int |\mu(x) - \pi(x)| \mathrm{d}x$ and (when the Radon-Nikodym derivative exists) their KL divergence by $D_{\text{KL}}(\mu\|\pi) := \int \mu(x) \log \frac{\mu(x)}{\pi(x)} \mathrm{d}x$. For $1 < \alpha < \infty$, we also define the α-Rényi divergence between densities

μ, π by

$$D_\alpha(\mu\|\pi) := \frac{1}{\alpha - 1} \log \left(\int \left(\frac{\mu(x)}{\pi(x)} \right)^\alpha \pi(x) \mathrm{d}x \right).$$

We say density π is logconcave (respectively, m-strongly logconcave in $\|\cdot\|_{\mathcal{X}}$) if $-\log \pi$ is convex (respectively, m-strongly convex in $\|\cdot\|_{\mathcal{X}}$). We similarly say π is m-relatively strongly logconcave in φ if $-\log \pi$ is m-relatively strongly convex in φ. If $\log \pi$ is affine, we say π is logaffine. We say a density π_0 is β-warm with respect to a density π if for all x in the support of π, $\frac{\mathrm{d}\pi_0(x)}{\mathrm{d}\pi(x)} \leq \beta$.

Log-Laplace transform. We define the log-Laplace transform (LLT) of $\varphi : \mathbb{R}^d \to \mathbb{R}$ by

$$\psi(x) := \log \left(\int \exp\left(\langle x, y \rangle - \varphi(y) \right) \mathrm{d}y \right).$$

When φ, ψ are clear from context, we define the density

$$\mathcal{D}_x^\varphi(y) = \exp\left(\langle x, y \rangle - \varphi(y) - \psi(x) \right). \tag{5.6}$$

Note that the normalization constant is exactly given by $\psi(x)$ and hence $\mathcal{D}_x^\varphi$ is indeed a valid density. We use $\propto$ to indicate proportionality, e.g. if μ is a density and we write $\mu \propto \exp(-f)$, we mean $\mu(x) = \frac{\exp(-f)}{Z}$ where $Z := \int \exp(-f(x)) \mathrm{d}x$ and the integration is over the support of μ.

Riemannian geometry. In Sections 5.3 and 5.4 we will use geometry induced by the Hessian of a self-concordant, convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$. We summarize the important points here, and defer a more extended treatment to [NT02]. When φ is clear from context, we denote the norm $\|h\|_x := \|h\|_{\nabla^2 \varphi(x)}$. Throughout this discussion let $M \subseteq \mathbb{R}^d$ be a Riemannian manifold equipped with the local metric $\|\cdot\|_x$. The induced Riemannian distance of a curve $c : [0, 1] \to M$ is defined as

$$L_\varphi(c) := \int_0^1 \left\| \frac{\mathrm{d}}{\mathrm{d}t} c(t) \right\|_{c(t)} \mathrm{d}t,$$

where $\frac{\mathrm{d}}{\mathrm{d}t}c(t)$ is the velocity element of the curve in the tangent space at $c(t)$. For $x, y \in M$, we then define $d_\varphi(x, y)$ to be the infimum of the length $L_\varphi(c)$ over all curves c such that $c(0) = x$ and $c(1) = y$. We will use the following two important properties of the Riemannian geometry over $M = \mathbb{R}^d$ induced by self-concordant, convex functions.

Lemma 5.2.1 ([NT02], Lemma 3.1)**.** *Suppose $\varphi : \mathbb{R}^d \to \mathbb{R}$ is convex and self-concordant. For $x, y \in \mathbb{R}^d$, if $d_\varphi(x, y) \le \delta - \delta^2 < 1$ for some $\delta \in (0, 1)$, then $\|y - x\|_x \le \delta$.*

Lemma 5.2.2 ([Nem04], Section 2.2.1)**.** *Suppose $\varphi : \mathbb{R}^d \to \mathbb{R}$ is convex and self-concordant. For any $h, x \in \mathbb{R}^d$ such that $\|h\|_x < 1$, $(1-\|h\|_x)^2\nabla^2\varphi(x) \preceq \nabla^2\varphi(x+h) \preceq (1-\|h\|_x)^{-2}\nabla^2\varphi(x)$.*

5.3 Properties of the LLT

In this section, we collect a variety of facts about the log-Laplace transform which we will use to develop our sampling scheme in Section 5.4. We begin by proving basic facts about the LLT in Section 5.3.1. We then use them to derive isoperimetric properties of induced distributions in Section 5.3.2 and total variation bounds in Section 5.3.3. Throughout this section we will fix a convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$, and let $\psi : \mathbb{R}^d \to \mathbb{R}$ be its LLT. We will also follow the notation (5.6).

5.3.1 Basic properties and duality

The log-Laplace transform ψ at x is the cumulant-generating function of the distribution $\mathcal{D}_x^\varphi$, which means that ψ is infinitely-differentiable and that $\nabla^k\psi$ is the k^{th} cumulant tensor of $\mathcal{D}_x^\varphi$. We will only use the first three derivatives of ψ which we compute below for completeness.

Lemma 5.3.1 (LLT derivatives)**.** *For any $x, h \in \mathbb{R}^d$, we have*

$$\begin{aligned}
\nabla\psi(x) &= \mu(\mathcal{D}_x^\varphi) := \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}[y], \\
\nabla^2\psi(x) &= \mathrm{Cov}(\mathcal{D}_x^\varphi) := \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[(y - \mu(\mathcal{D}_x^\varphi))(y - \mu(\mathcal{D}_x^\varphi))^\top\right], \\
\nabla^3\psi(x)[h, h, h] &= \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\langle y - \mu(\mathcal{D}_x^\varphi), h\rangle^3\right].
\end{aligned}$$

Proof. For any $x \in \mathbb{R}^d$, a straightforward calculation shows that

$$\nabla\psi(x) = \nabla\left(\log \int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y\right) = \frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y} = \mu(\mathcal{D}_x^\varphi).$$

Further,

$$\begin{aligned}\nabla^2\psi(x) &= \nabla\left(\frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y}\right) \\ &= \frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y y^\top \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y} - \frac{\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y \mathrm{d}y\right)\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y \mathrm{d}y\right)^\top}{\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y\right)^2}.\end{aligned}$$

Finally,

$$\begin{aligned}\nabla^3\psi(x)[h,h,h] &= h^\top \nabla\left(\frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right)\left(y^\top h\right)^2 \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y} - \frac{\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y^\top h \mathrm{d}y\right)^2}{\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y\right)^2}\right) \\ &= \frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right)\left(y^\top h\right)^3 \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y} + 2\left(\frac{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) y^\top h \mathrm{d}y}{\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y}\right)^3 \\ &\quad - \frac{3\int \exp\left(\langle x, y\rangle - \varphi(y)\right)\left(y^\top h\right)^2 \mathrm{d}y \int \exp\left(\langle x, y\rangle - \varphi(y)\right) y^\top h \mathrm{d}y}{\left(\int \exp\left(\langle x, y\rangle - \varphi(y)\right) \mathrm{d}y\right)^2}.\end{aligned}$$

□

By using a fact on one-dimensional logconcave distributions in [BE19], this implies the following.

Lemma 5.3.2 (Self-concordance)**.** *If ψ is the LLT of a convex function, it is self-concordant.*

Proof. By the definition of self-concordance and Lemma 5.3.1, it suffices to show for any $h \in \mathbb{R}^d$,

$$\mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\langle y - \mu(\mathcal{D}_x^\varphi), h\rangle\right]^3 \le 2\left(\mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\langle y - \mu(\mathcal{D}_x^\varphi), h\rangle^2\right]\right)^{\frac{3}{2}}. \tag{5.7}$$

We then note that the random variable $\langle y - \mu(\mathcal{D}_x^\varphi), h\rangle$ for $y \sim \mathcal{D}_x^\varphi$ follows a logconcave distribution because affine transformations preserve logconcavity. Finally Lemma 2 of [BE19] implies (5.7) holds. □

Next, we prove that a form of strong convexity-smoothness duality (and its converse)

holds with respect to φ and ψ, analogous to the type of duality satisfied by Fenchel conjugates [KST09].

Lemma 5.3.3 (Strong convexity-smoothness duality)**.** *If $\varphi : \mathbb{R}^d \to \mathbb{R}$ is L-smooth with respect to $\|\cdot\|_*$, then $\psi : \mathbb{R}^d \to \mathbb{R}$ is $\frac{1}{L}$-strongly convex with respect to $\|\cdot\|$.*

Proof. By definition of strong convexity it suffices to prove for any $x, v \in \mathbb{R}^d$, $v^\top \nabla^2 \psi(x) v \geq \frac{1}{L}\|v\|^2$. Without loss of generality, by scale invariance we can assume $\|v\| = 1$. Let $Y = \langle y, v\rangle$, where $y \sim \mathcal{D}_x^\varphi$. By Lemma 5.3.1, $\nabla^2\psi(x) = \mathbf{Cov}(\mathcal{D}_x^\varphi)$, so it suffices to prove that

$$\mathbf{Var}(Y) = \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\langle y - \mu(\mathcal{D}_x^\varphi), v\rangle^2\right] \geq \frac{1}{L}.$$

Letting $\mathbf{M} := \mathbb{E}_{y\sim\mathcal{D}_x^\varphi} \nabla^2\varphi(y)$, we first observe

$$\frac{L}{2} v^\top \mathbf{M}^{-1} v = \max_{u\in\mathbb{R}^d} \langle u, v\rangle - \frac{1}{2L} u^\top \mathbf{M} u \geq \max_{u\in\mathbb{R}^d} \langle u, v\rangle - \frac{1}{2}\|u\|_*^2 = \frac{1}{2}\|v\|^2.$$

In the only inequality, we used that $u^\top \mathbf{M} u = \mathbb{E}_{y\sim\mathcal{D}_x^\varphi} u^\top \nabla^2\varphi(y) u \leq L\|u\|_*^2$ by smoothness of φ, and the last equality follows by optimizing over $\|u\|_*$. This shows $v^\top \mathbf{M}^{-1} v \geq \frac{1}{L}$. The Cramér-Rao inequality (see Lemma 2, [CP22]) then implies

$$\mathbf{Var}(Y) \geq v^\top \mathbf{M}^{-1} v \geq \frac{1}{L},$$

since the Hessian of $-\log \mathcal{D}_x^\varphi$ at any $x \in \mathbb{R}^d$ is $\nabla^2\varphi$. □

Lemma 5.3.4 (Smoothness-strong convexity duality)**.** *If $\varphi : \mathbb{R}^d \to \mathbb{R}$ is $\frac{1}{L}$-strongly convex with respect to $\|\cdot\|_*$, then $\psi : \mathbb{R}^d \to \mathbb{R}$ is L-smooth with respect to $\|\cdot\|$.*

Proof. Let $v, x \in \mathbb{R}^d$ and assume $\|v\| = 1$. As in Lemma 5.3.3, defining $Y = \langle y, v\rangle$ for $y \sim \mathcal{D}_x^\varphi$, we have $v^\top \nabla^2\psi(x) v = \mathbf{Var}(Y)$, and want to show $\mathbf{Var}(Y) \leq L$. First note that for any $y \in \mathbb{R}^d$,

$$\frac{1}{2L} v^\top \left(\nabla^2\varphi(y)\right)^{-1} v = \max_{u\in\mathbb{R}^d} \langle u, v\rangle - \frac{L}{2} u^\top \nabla^2\varphi(y) u \leq \max_{u\in\mathbb{R}^d} \langle u, v\rangle - \frac{1}{2}\|u\|_*^2 = \frac{1}{2}\|v\|^2.$$

The first inequality used strong convexity of φ and the last equality follows by optimizing over $\|u\|_*$. This shows $v^\top(\nabla^2\varphi(y))^{-1}v \le L$ for all y. The Brascamp-Lieb inequality [BL76] then implies

$$\mathbf{Var}(Y) \le \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[v^\top\left(\nabla^2\varphi(y)\right)^{-1}v\right] \le L,$$

since the Hessian of $-\log\mathcal{D}_x^\varphi$ at any $x\in\mathbb{R}^d$ is $\nabla^2\varphi$. □

5.3.2 Isoperimetry

In this section we prove an isoperimetric inequality for densities which are relatively strongly logconcave with respect to an appropriate LLT. The only LLT property we use is Lemma 5.3.2, i.e. self-concordance, via the following generic fact which generalizes Lemma 35 of [LV18].

Lemma 5.3.5. *Suppose $\varphi:\mathbb{R}\to\mathbb{R}$ is convex and self-concordant. For any $x\in\mathbb{R}$,*

$$\frac{\exp(-\varphi(x))}{\sqrt{\varphi''(x)}} \ge \frac{1}{12}\min\left\{\int_{-\infty}^{x}\exp(-\varphi(t))dt, \int_{x}^{\infty}\exp(-\varphi(t))dt\right\}.$$

Proof. Assume $\varphi'(x)\ge 0$ (the other case will follow analogously by bounding the integral on $(-\infty,x]$). Define $r := x + \frac{1}{4\sqrt{\varphi''(x)}}$. By self-concordance (Lemma 5.2.2), for all $t\in[x,r]$,

$$\frac{1}{2}\varphi''(x) \le \varphi''(t) \le 2\varphi''(x).$$

Hence, we have for all $t\in[x,r]$, since $\varphi'(x)\ge 0$,

$$\varphi(t) = \varphi(x) + \varphi'(x)(t-x) + \int_x^t (t-s)\varphi''(s)\mathrm{d}s \ge \varphi(x) + \frac{1}{4}(t-x)^2\varphi''(x). \tag{5.8}$$

We use (5.8) to bound the integral on $[x,r]$:

$$\begin{aligned}\int_x^r \exp(-\varphi(t))\mathrm{d}t &\le \exp(-\varphi(x))\int_x^r \exp\left(-\frac{1}{4}(t-x)^2\varphi''(x)\right)\mathrm{d}t \\ &\le \exp(-\varphi(x))\int_{-\infty}^{\infty}\exp\left(-\frac{1}{4}(t-x)^2\varphi''(x)\right)\mathrm{d}t = 2\sqrt{\pi}\cdot\frac{\exp(-\varphi(x))}{\sqrt{\varphi''(x)}}.\end{aligned} \tag{5.9}$$

Next, to bound the integral on $[r, \infty)$, we first observe

$$\varphi'(r) \geq \varphi'(x) + \int_x^r \varphi''(r)\mathrm{d}t \geq \frac{1}{2}\int_x^r \varphi''(x)\mathrm{d}t \geq \frac{1}{8}\sqrt{\varphi''(x)}.$$

Hence, by convexity from r,

$$\begin{aligned}\int_r^\infty \exp(-\varphi(t))\mathrm{d}t &\leq \int_r^\infty \exp\left(-\varphi(r) - \varphi'(r)(t-r)\right)\mathrm{d}t \\ &\leq \exp(-\varphi(x))\int_r^\infty \exp\left(-\frac{1}{8}\sqrt{\varphi''(x)}(t-r)\right)\mathrm{d}t = 8\cdot\frac{\exp(-\varphi(x))}{\sqrt{\varphi''(x)}}.\end{aligned} \tag{5.10}$$

We used $\varphi(r) \geq \varphi(x)$ by convexity and $\varphi'(x) \geq 0$. Combining (5.9) and (5.10) yields the claim. □

Next, we reduce the problem of proving isoperimetry for relatively strongly logconcave densities to the same problem in one dimension (captured via Lemma 5.3.5), via the localization lemma.

Lemma 5.3.6 (Modification of the localization lemma, [KLS95], Theorem 2.7)**.** *Let f_1, f_2, f_3, f_4 be four nonnegative functions on $\mathbb{R}^d$ such that f_1 and f_2 are upper semicontinuous and f_3 and f_4 are lower semicontinuous, let $c_1, c_2 > 0$, and let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be convex. Then, the following are equivalent:*

- *For every density $\pi : \mathbb{R}^d \to \mathbb{R}$ which is 1-relatively strongly logconcave in φ,*

$$\left(\int f_1(x)\pi(x)\mathrm{d}x\right)^{c_1}\left(\int f_2(x)\pi(x)\mathrm{d}x\right)^{c_2} \leq \left(\int f_3(x)\pi(x)\mathrm{d}x\right)^{c_1}\left(\int f_4(x)\pi(x)\mathrm{d}x\right)^{c_2}.$$

- *For every $a, b \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}$,*

$$\begin{aligned}&\left(\int_0^1 f_1((1-t)a+tb)e^{\gamma t - \varphi((1-t)a+tb)}\mathrm{d}t\right)^{c_1}\left(\int_0^1 f_2((1-t)a+tb)e^{\gamma t - \varphi((1-t)a+tb)}\mathrm{d}t\right)^{c_2} \\ &\leq \left(\int_0^1 f_3((1-t)a+tb)e^{\gamma t - \varphi((1-t)a+tb)}\mathrm{d}t\right)^{c_1}\left(\int_0^1 f_4((1-t)a+tb)e^{\gamma t - \varphi((1-t)a+tb)}\mathrm{d}t\right)^{c_2}.\end{aligned}$$

Proof. The proof follows identically to the case where $\varphi = 0$, which was proven in [LS93,

KLS95] via a bisection argument (see Lemma 2.5, [LS93]). The only fact the bisection argument relies on is that restricting logconcave densities to subsets of $\mathbb{R}^d$ preserves logconcavity, which remains true for densities which are relatively strongly logconcave with respect to a given convex function. For a more formal treatment of this generalized bisection argument, see Lemma 1 of [GLL+23]. Finally the change on the continuity assumptions on the $\{f_i\}_{i\in[4]}$ follows by Remark 2.3 of [KLS95]. □

Finally, we combine these tools to prove the main result of this section.

Lemma 5.3.7 (Self-concordant isoperimetry)**.** *Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be convex and self-concordant, and let $f : \mathbb{R}^d \to \mathbb{R}$ be m-relatively strongly convex in φ. Given any partition S_1, S_2, S_3 of $\mathbb{R}^d$,*

$$\frac{\int_{S_3} \exp\left(-f(x)\right) \mathrm{d}x}{\min\left\{\int_{S_1} \exp(-f(x))\mathrm{d}x, \int_{S_2} \exp(-f(x))\mathrm{d}x\right\}} = \Omega\left(\sqrt{m} d_\varphi(S_1, S_2)\right),$$

where $d_\varphi(S_1, S_2) = \min_{x\in S_1, y\in S_2} d_\varphi(x, y)$.

Proof. We assume $m = 1$ by rescaling $\varphi \leftarrow m\varphi$ which results in $d_\varphi(S_1, S_2) \leftarrow \sqrt{m} d_\varphi(S_1, S_2)$. We first show that without loss of generality, we can assume

$$\max_{i\in\{1,2\}} \frac{\int_{S_i} \exp(-f(x))\mathrm{d}x}{\int \exp(-f(x))\mathrm{d}x} = \Omega(1). \tag{5.11}$$

To see this, let $S_1^\star, S_2^\star$ and $S_3^\star$ be the partition that achieves the minimum of

$$\beta(S_1, S_2, S_3) = \frac{\int_{S_3} \exp\left(-f(x)\right) \mathrm{d}x}{d_\varphi(S_1, S_2) \min\left\{\int_{S_1} \exp(-f(x))\mathrm{d}x, \int_{S_2} \exp(-f(x))\mathrm{d}x\right\}}.$$

Let $\delta = d_\varphi(S_1^\star, S_2^\star)$. For any $z \in S_3^\star$, let $x \in S_1^\star$ minimize $d_\varphi(x, z)$ and let $y \in S_2^\star$ minimize $d_\varphi(y, z)$. By the triangle inequality we have

$$d_\varphi(x, z) + d_\varphi(y, z) \geq \delta$$

and hence $\max(d_\varphi(x, z), d_\varphi(y, z)) \geq \frac{\delta}{2}$. Consequently we can partition $S_3^\star$ into S_3' and S_3'' such that $d_\varphi(S_1^\star, S_3') \geq \frac{\delta}{2}$ and $d_\varphi(S_2^\star, S_3'') \geq \frac{\delta}{2}$ by placing each z into an appropriate set.

Moreover, we can assume without loss of generality that

$$\frac{\int_{S_3'} \exp\left(-f(x)\right) \mathrm{d}x}{\frac{\delta}{2} \min\left\{\int_{S_1^\star} \exp(-f(x))\mathrm{d}x, \int_{S_2^\star} \exp(-f(x))\mathrm{d}x\right\}} \leq \beta.$$

as otherwise the above is true for S_3''. Thus, $\beta(S_1^\star \cup S_3'', S_2^\star, S_3') \leq \beta(S_1^\star, S_2^\star, S_3^\star)$, proving (5.11) (else we may halve the measure of S_3). Given (5.11), it suffices to show that there is a constant C with

$$\begin{aligned} C d_\varphi(S_1, S_2) \int \exp(-f(x)) \mathbb{1}_{S_1}(x) \mathrm{d}x \int \exp(-f(x)) \mathbb{1}_{S_2}(x) \mathrm{d}x \\ \leq \int \exp(-f(x)) \mathrm{d}x \int \exp(-f(x)) \mathbb{1}_{S_3}(x) \mathrm{d}x. \end{aligned}$$

Using the localization lemma (Lemma 5.3.6), letting $f_i = \mathbb{1}_{S_i}$ for $i \in [3]$ and $f_4 = (C d_\varphi(S_1, S_2))^{-1}$,[7] it suffices to prove for every $a, b \in \mathbb{R}^d$ and $\gamma \in \mathbb{R}$,

$$\begin{aligned} & C d_\varphi(S_1, S_2) \int_0^1 \exp\left(\gamma t - \varphi((1-t)a + tb)\right) \mathbb{1}_{S_1}((1-t)a + tb) \mathrm{d}t \\ & \cdot \int_0^1 \exp\left(\gamma t - \varphi((1-t)a + tb)\right) \mathbb{1}_{S_2}((1-t)a + tb) \mathrm{d}t \\ & \leq \int_0^1 \exp\left(\gamma t - \varphi((1-t)a + tb)\right) \mathrm{d}t \int_0^1 \exp\left(\gamma t - \varphi((1-t)a + tb)\right) \mathbb{1}_{S_3}((1-t)a + tb) \mathrm{d}t. \end{aligned}$$

Redefine $\varphi(t) \leftarrow \varphi((1-t)a + tb) - \gamma t$ for $t \in \mathbb{R}$, which is a one-dimensional self-concordant function, and redefine $S_i \leftarrow \{t \mid (1-t)a + tb \in S_i\}$ for $i \in [3]$, such that each S_i is a union of intervals. It is straightforward to check that the distance $d_\varphi(S_1, S_2)$ only increases under this transformation, because it can only take fewer paths, and each path has the same length (the change in $\sqrt{\varphi''}$ is negated by the change in distance traveled by the path).

So, it suffices to consider the special one-dimensional case with $\gamma = 0$, where $d_\varphi(x, y) = \int_x^y \sqrt{\varphi''(t)} \mathrm{d}t$. We next note that it suffices to consider the case when S_3 is a single interval, i.e. for any $a \leq a' \leq b' \leq b$, we have $S_1 = [a, a']$, $S_2 = [b', b]$, $S_3 = [a', b']$, and wish to show

[7] Without loss of generality we can assume S_1 and S_2 are closed (implying S_3 is open) by taking their closures. This implies f_1, f_2 are upper semicontinuous and f_3, f_4 are lower semicontinuous.

for some constant C

$$\frac{\int_{a'}^{b'} \exp(-\varphi(t))\mathrm{d}t}{\int_{a'}^{b'} \sqrt{\varphi''(t)}dt} \geq C \frac{\int_{a}^{a'} \exp(-\varphi(t))\mathrm{d}t \int_{b'}^{b} \exp(-\varphi(t))\mathrm{d}t}{\int_{a}^{b} \exp(-\varphi(t))\mathrm{d}t}. \tag{5.12}$$

When S_3 has multiple intervals, by Theorem 2.6 in [LS93], we show (5.12) for each interval in S_3 and its adjacent segments in S_1 and S_2, and sum over all such inequalities. By Lemma 5.3.5, when φ is convex and self-concordant, we have for any $x \in [a, b]$,

$$\frac{\exp(-\varphi(x))}{\sqrt{\varphi''(x)}} \geq \frac{1}{12} \min\left(\int_a^x \exp(-\varphi(t))\mathrm{d}t, \int_x^b \exp(-\varphi(t))\mathrm{d}t\right)$$

which combined with $\frac{\int_{a'}^{b'} \exp(-\varphi(t))\mathrm{d}t}{\int_{a'}^{b'} \sqrt{\varphi''(t)}\mathrm{d}t} \geq \min_{x\in[a',b']} \frac{\exp(-\varphi(x))}{\sqrt{\varphi''(x)}}$ shows (5.12). □

5.3.3 Total variation bounds

In this section, we provide a bound on the total variation distance of induced distributions $\mathcal{D}_x^\varphi$ and $\mathcal{D}_{x'}^\varphi$, when x and x' are close in the Riemannian distance given by ψ.

Lemma 5.3.8 (TV distance between $\mathcal{D}_x^\varphi$ and $\mathcal{D}_{x'}^\varphi$)**.** *For any $x, x' \in \mathbb{R}^d$ such that $d_\psi(x, x') \leq \frac{1}{4}$,*

$$\left\|\mathcal{D}_x^\varphi - \mathcal{D}_{x'}^\varphi\right\|_{\mathrm{TV}} \leq \frac{1}{2}.$$

Proof. Let $h = x' - x$ and note that the KL divergence between $\mathcal{D}_x^\varphi$ and $\mathcal{D}_{x'}^\varphi$ may be rewritten as

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathcal{D}_x^\varphi \| \mathcal{D}_{x'}^\varphi\right) &= \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\log \frac{\mathrm{d}\mathcal{D}_x^\varphi}{\mathrm{d}\mathcal{D}_{x'}^\varphi}(y)\right] = \mathop{\mathbb{E}}_{y\sim\mathcal{D}_x^\varphi}\left[\psi(x') - \psi(x) - \langle h, y\rangle\right] \\ &= \psi(x') - \psi(x) - \langle h, \nabla\psi(x)\rangle\,. \end{aligned}$$

In the last equation, we used Lemma 5.3.1. We recognize that the KL divergence is the Bregman divergence (first-order Taylor approximation) in ψ, and hence letting $x_t = x + th$ for $t \in [0, 1]$ such that $x_0 = x$ and $x_1 = x'$, we continue bounding

$$D_{\mathrm{KL}}\left(\mathcal{D}_x^\varphi \| \mathcal{D}_{x'}^\varphi\right) = \int_0^1 (1-t)\nabla^2\psi(x_t)[h, h]\mathrm{d}t$$

$$\leq \int_0^1 4(1-t)\nabla^2\psi(x)[h,h]\mathrm{d}t \leq \frac{1}{2}.$$

The first inequality used that when $d_\psi(x,x') \leq \frac{1}{4}$, Lemma 5.2.1 shows $\|x_t - x\|_x \leq \|x' - x\|_x \leq \frac{1}{2}$, so Lemma 5.2.2 gives $\nabla^2\psi(x_t) \preceq 4\nabla^2\psi(x)$; the second used $\|h\|_x \leq \frac{1}{2}$. Finally by Pinsker's inequality,

$$\left\|\mathcal{D}_x^\varphi - \mathcal{D}_{x'}^\varphi\right\|_{\mathrm{TV}} \leq \sqrt{\frac{1}{2}D_{\mathrm{KL}}(\mathcal{D}_x^\varphi \| \mathcal{D}_{x'}^\varphi)} \leq \frac{1}{2}.$$

□

5.4 Proximal LLT sampler

In this section, we study a sampling problem in the following setting, assumed throughout.

Problem 5.4.1. For $D, G, \eta > 0$, let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, with diameter in a norm $\|\cdot\|_{\mathcal{X}}$ at most D. Let $F : \mathcal{X} \to \mathbb{R}$ have the stochastic form $F(x) := \mathbb{E}_{i\sim\mathcal{I}}[f_i(x)]$, for a distribution $\mathcal{I}$ over (a possibly infinite) family of indices i, such that each $f_i : \mathcal{X} \to \mathbb{R}$ is convex and G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$. Finally, let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be convex and η-smooth in the dual norm $\|\cdot\|_{\mathcal{X}^*}$. Given $\mu > 0$, and letting $\psi : \mathbb{R}^d \to \mathbb{R}$ be the LLT of φ, the goal is to sample from the density π satisfying

$$\mathrm{d}\pi(x) \propto \exp\left(-F(x) - \eta\mu\psi(x)\right) \mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x. \tag{5.13}$$

Note that by Lemma 5.3.3, $\eta\mu\psi$ is μ-strongly convex in $\|\cdot\|_{\mathcal{X}}$. Letting $z = (x, y)$ denote a variable on $\mathcal{X} \times \mathbb{R}^d$, it is convenient for us to define the extended density on the joint space of z:

$$\mathrm{d}\widehat{\pi}(z) \propto \exp\left(-F(x) - \eta\mu\psi(x) + (\langle x, y\rangle - \psi(x) - \varphi(y))\right) \mathbb{1}_{\mathcal{X}}(x)\mathrm{d}z. \tag{5.14}$$

Our sampling framework for (5.13) generalizes an approach pioneered by [LST21b], and is stated in the following Algorithm 6. The algorithm simply alternately samples from each marginal of (5.14). Before stating it, we define the following notation for conditional

densities throughout the section:

$$\begin{aligned} \mathrm{d}\pi_x(y) &= \exp\left(\langle x, y\rangle - \psi(x) - \varphi(y)\right) \mathrm{d}y \text{ for all } x \in \mathcal{X}, \\ \mathrm{d}\pi_y(x) &\propto \exp\left(-F(x) - (1+\eta\mu)\psi(x) + \langle x, y\rangle\right) \mathbb{1}_{\mathcal{X}}(x) \mathrm{d}x \text{ for all } y \in \mathbb{R}^d. \end{aligned} \tag{5.15}$$

In particular, we observe that $\mathrm{d}\pi_x(y) = \mathrm{d}\widehat{\pi}(\cdot \mid x)$ and $\mathrm{d}\pi_y(x) = \mathrm{d}\widehat{\pi}(\cdot \mid y)$.

Algorithm 15: `AlternateSample`$(\mathcal{X}, F, \varphi, T, \mu, x_0)$

1 **Input:** $\mathcal{X}, F, \varphi$ in the setting of Problem 5.4.1, $T \in \mathbb{N}$, $\mu > 0$, $x_0 \in \mathcal{X}$;
2 **for** $k \in [T]$ **do**
3 | Sample $y_k \sim \pi_{x_{k-1}}$.;
4 | Sample $x_k \sim \pi_{y_k}$.;
5 **end**
6 **Return** x_T

Correctness of Algorithm 6 for sampling from (5.14) builds upon the following basic facts.

Lemma 5.4.2. *The total x-marginal of $\widehat{\pi}$ in (5.14) is π in (5.13). Furthermore, the stationary distribution of Algorithm 6 is $\widehat{\pi}$, and the induced Markov chains in Algorithm 6 restricted to either $\{x_k\}_{0 \le k \le T}$ (a Markov chain on $\mathcal{X}$) or $\{y_k\}_{k \in [T]}$ (a Markov chain on $\mathbb{R}^d$) are both reversible.*

Proof. The first conclusion is a direct calculation, and the remainder is Lemma 1 in [LST21b]. □

In Section 5.4.1 we develop a subroutine based on rejection sampling for implementing Line 4 of Algorithm 6, extending [GLL22]. We then give our complete analysis of Algorithm 6 in Section 5.4.2.

5.4.1 Sampling from the x-conditional distribution

Throughout this section, we assume the setting in Problem 5.4.1, and fix some $y \in \mathbb{R}^d$. We provide a sampler for the marginal density π_y (following notation (5.15)), and denote the

component of the density independent of F by γ_y, i.e.

$$\mathrm{d}\gamma_y(x) \propto \exp\left(-\eta\mu\psi(x) - (\psi(x) - \langle x, y\rangle)\right) \mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x. \tag{5.16}$$

By Lemma 5.3.3, γ_y (and hence π_y) is $\frac{1}{\eta}$-strongly logconcave in $\|\cdot\|_{\mathcal{X}}$. Our rejection sampler leverages this fact and the stochastic nature of F to build a rejection sampling scheme similarly to [GLL22]. For completeness, we state our Algorithm 13 below, and provide the details of its analysis here.

Algorithm 16: `InnerLoop`$(y, \delta, \mathcal{X}, F, \varphi, \mu)$

textbfInput: $\delta \in (0, \frac{1}{2})$, $y \in \mathbb{R}^d$, $\mathcal{X}, F, \varphi$ in the setting of Problem 5.4.1 for $\frac{1}{\eta} \geq 10^4 G^2 \log \frac{1}{\delta}$

Output: Sample within total variation distance δ of

$$\mathrm{d}\pi_y(x) \propto \exp\left(-F(x) - \eta\mu\psi(x) - (\psi(x) - \langle x, y\rangle)\right) \mathbb{1}_{x\in\mathcal{X}}\mathrm{d}x.$$

$u \leftarrow 1, \rho \leftarrow 1$;
while $u > \frac{1}{2}\rho$ **do**
  Sample $x_1, x_2 \sim \gamma_y$ defined in (5.16) independently;
  $\rho \leftarrow 1$, $u \sim_{\text{unif.}} [0, 1]$;
  Draw $a \in \mathbb{N}$ such that for all $b \in \mathbb{N}$, $\Pr[a \geq b] = \frac{1}{b!}$;
  for $b \in [a]$ **do**
    Draw $j_{i,b} \sim \mathcal{I}$ for $i \in [b]$;
    $\rho \leftarrow \rho + \prod_{i\in[b]}(f_{j_{i,b}}(x_2) - f_{j_{i,b}}(x_1))$;
  end
end
Return: x_1

In order to analyze Algorithm 13, we first state a general result about concentration of Lipschitz functions with respect to a strongly logconcave measure, in general norms. The following is a direct adaptation of standard results on log-Sobolev inequalities contained in [Led99, BL00].

Lemma 5.4.3 ([Led99], Section 2.3 and [BL00], Proposition 3.1)**.** *Let $X \sim \pi$ for density $\pi : \mathcal{X} \to \mathbb{R}$ which is μ-strongly logconcave in $\|\cdot\|_{\mathcal{X}}$, and let $\ell : \mathcal{X} \to \mathbb{R}$ be G-Lipschitz in*

$\|\cdot\|_{\mathcal{X}}$. *For all* $t \geq 0$,

$$\Pr_{x\sim\pi}\left[\ell(x) \geq \mathop{\mathbb{E}}_{\pi}[\ell] + t\right] \leq \exp\left(-\frac{\mu t^2}{2G^2}\right).$$

In the remainder of the section, let $\tilde{\pi}_y$ be the distribution of the output of Algorithm 13 and recall the target stationary distribution is π_y. When ρ is clear from context, we define $\overline{\rho} := \mathrm{med}(0, \rho, 2)$ to be the truncation of ρ to $[0, 2]$. We also denote the index set drawn on Line 9 by

$$\mathcal{J} := \{j_{i,b}\}_{b\in[a],i\in[b]},$$

when a is clear from context. We first provide the following characterization of $\|\pi_y - \tilde{\pi}_y\|_{\mathrm{TV}}$.

Lemma 5.4.4. *Define* r_x *to be the random variable* $\mathbb{E}[\rho \mid x_1 = x]$ *(where the expectation is over* x_2*,* a*, and the random indices* $\mathcal{J}$*, and similarly let* $\overline{r}_x := \mathbb{E}[\overline{\rho} \mid x_1 = x]$*. Then,*

$$\|\pi_y - \tilde{\pi}_y\|_{\mathrm{TV}} \leq \mathop{\mathbb{E}}_{x\sim\gamma_y} |r_x - \overline{r}_x|.$$

Proof. First, by definition of π_y, we have

$$\pi_y(x) = \frac{\exp(-F(x))\gamma_y(x)}{\int \exp(-F(w))\gamma_y(w)\mathrm{d}w} = \gamma_y(x) \cdot \frac{\exp(-F(x))}{\mathbb{E}_{w\sim\gamma_y}\exp(-F(w))}. \tag{5.17}$$

Moreover, by definition of the algorithm,

$$\tilde{\pi}_y(x) = \frac{\gamma_y(x)\Pr[u \leq \frac{1}{2}\rho \mid x_1 = x]}{\Pr[u \leq \frac{1}{2}\rho]} = \frac{\gamma_y(x)\,\mathbb{E}[\overline{\rho} \mid x_1 = x]}{\mathbb{E}[\overline{\rho}]} \tag{5.18}$$

where all probabilities and expectations are x_2, a, and $\mathcal{J}$. Furthermore, note that for fixed $b \in [a]$,

$$\mathop{\mathbb{E}}_{\mathcal{J}}\left[\prod_{i\in[b]}(f_{j_{i,b}}(x_2) - f_{j_{i,b}}(x_1))\right] = \left(\mathop{\mathbb{E}}_{j\sim\mathcal{I}}[f_j(x_2) - f_j(x_1)]\right)^b = (F(x_2) - F(x_1))^b.$$

Hence, taking expectations over a, we have for any fixed x_1, x_2,

$$\begin{aligned}\mathbb{E}\left[\rho \mid x_1, x_2\right] &= \sum_{b\geq 0} \Pr[a \geq b](F(x_2) - F(x_1))^b \\ &= \sum_{b\geq 0} \frac{1}{b!}(F(x_2) - F(x_1))^b = \exp\left(F(x_2) - F(x_1)\right).\end{aligned} \tag{5.19}$$

Next, by combining (5.17) and (5.18), we have

$$\begin{aligned}\|\pi - \tilde{\pi}\|_{\mathrm{TV}} &= \frac{1}{2}\int \left|\frac{\exp(-F(x))}{\mathbb{E}_{w\sim\gamma_y}\exp(-F(w))} - \frac{\mathbb{E}[\overline{\rho} \mid x_1 = x]}{\mathbb{E}[\overline{\rho}]}\right| \gamma_y(x)\mathrm{d}x \\ &= \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[\left|\frac{\exp(-F(x))}{\mathbb{E}_{w\sim\gamma_y}\exp(-F(w))} - \frac{\mathbb{E}[\overline{\rho} \mid x_1 = x]}{\mathbb{E}[\overline{\rho}]}\right|\right].\end{aligned}$$

By taking expectations over x_2 in (5.19), and recalling the definitions of $r_x, \overline{r}_x$, we obtain $r_x = \mathbb{E}[\rho \mid x_1 = x] = \exp(-F(x))\,\mathbb{E}_{x_2\sim\gamma_y}\exp(F(x_2))$. We thus have

$$\|\pi - \tilde{\pi}\|_{\mathrm{TV}} = \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[\left|\frac{r_x}{\mathbb{E}_{w\sim\gamma_y} r_w} - \frac{\overline{r}_x}{\mathbb{E}_{w\sim\gamma_y}\overline{r}_w}\right|\right].$$

Next, we lower bound $\mathbb{E}_{w\sim\gamma_y} r_w$ as follows. By taking expectations over (5.19) and using independence of x_1 and x_2, we have that for the random variable $Z = \exp(-F(x))$ where $x \sim \gamma_y$, we have

$$\mathop{\mathbb{E}}_{w\sim\gamma_y} r_w = (\mathbb{E}\, Z)\cdot\left(\mathbb{E}\, Z^{-1}\right) \geq 1, \tag{5.20}$$

where we used Jensen's inequality which implies the last inequality for any nonnegative random variable Z. Finally, combining the above two displays, we derive the desired bound as follows:

$$\begin{aligned}\frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[\left|\frac{r_x}{\mathbb{E}_{w\sim\gamma_y} r_w} - \frac{\overline{r}_x}{\mathbb{E}_{w\sim\gamma_y}\overline{r}_w}\right|\right] &\leq \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[\left|\frac{r_x}{\mathbb{E}_{w\sim\gamma_y} r_w} - \frac{\overline{r}_x}{\mathbb{E}_{w\sim\gamma_y} r_w}\right|\right] \\ &\quad + \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[\left|\frac{\overline{r}_x}{\mathbb{E}_{w\sim\gamma_y} r_w} - \frac{\overline{r}_x}{\mathbb{E}_{w\sim\gamma_y}\overline{r}_w}\right|\right] \\ &\leq \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[|r_x - \overline{r}_x|\right] + \frac{\mathbb{E}_{x\sim\gamma_y}[|\overline{r}_x|]}{2}\cdot\left|\frac{1}{\mathbb{E}_{w\sim\gamma_y}\overline{r}_w} - \frac{1}{\mathbb{E}_{w\sim\gamma_y} r_w}\right| \\ &= \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\gamma_y}\left[|r_x - \overline{r}_x|\right] + \frac{1}{2}\left|1 - \frac{\mathbb{E}_{x\sim\gamma_y}\overline{r}_x}{\mathbb{E}_{x\sim\gamma_y} r_x}\right|\end{aligned}$$

$$\leq \frac{1}{2} \mathop{\mathbb{E}}_{x\sim\gamma_y} [|r_x - \overline{r}_x|] + \frac{1}{2|\mathop{\mathbb{E}}_{x\sim\gamma_y} r_x|} \cdot \mathop{\mathbb{E}}_{x\sim\gamma_y} [|r_x - \overline{r}_x|]$$
$$\leq \mathop{\mathbb{E}}_{x\sim\gamma_y} [|r_x - \overline{r}_x|] .$$

In the second and last inequalities, we use the bound (5.20). The third line follows since $\overline{r}_x$ is always nonnegative by definition, and the third inequality used convexity of $|\cdot|$. □

Lemma 5.4.4 shows it remains to bound $\mathbb{E}_{x\sim\gamma_y} |r_x - \overline{r}_x|$. Fixing x_1 and x_2, we know ρ and $\overline{\rho}$ as random variables of a and $\mathcal{J}$ are equal, except for the effect of truncating ρ to $[0, 2]$. Hence,

$$\mathop{\mathbb{E}}_{x\sim\gamma_y} |r_x - \overline{r}_x| \leq \mathbb{E}[|\rho| \mathbb{1}_{\rho\notin[0,2]}]. \tag{5.21}$$

In the remainder of the section, define

$$H := \left\lceil 10 \log \frac{1}{\delta} \right\rceil . \tag{5.22}$$

We then let

$$\begin{aligned} \lambda &:= \sum_{b>H} \mathbb{1}_{a\geq b} \prod_{i\in[b]} (f_{j_{i,b}}(x_2) - f_{j_{i,b}}(x_1)), \\ \sigma &:= \sum_{b=0}^{H} \mathbb{1}_{a\geq b} \prod_{i\in[b]} (f_{j_{i,b}}(x_2) - f_{j_{i,b}}(x_1)), \end{aligned} \tag{5.23}$$

be random variables depending on the choices of $x_1, x_2, a, \mathcal{J}$, where λ captures the effect of the "large" b, and σ captures the effect of the "small" b (where the $b = 0$ term is 1 by convention). Since $\rho = \sigma + \lambda$, in light of (5.21) it suffices to bound $\mathbb{E}[|\sigma| \mathbb{1}_{\rho\notin[0,2]}] + \mathbb{E}[|\lambda| \mathbb{1}_{\rho\notin[0,2]}]$, as

$$\mathop{\mathbb{E}}_{x\sim\gamma_y} |r_x - \overline{r}_x| \leq \mathbb{E}[|\rho| \mathbb{1}_{\rho\notin[0,2]}] \leq \mathbb{E}[|\sigma| \mathbb{1}_{\rho\notin[0,2]}] + \mathbb{E}[|\lambda| \mathbb{1}_{\rho\notin[0,2]}]. \tag{5.24}$$

We defer proofs of the following to Appendix 5.9, using small modifications to [GLL22].

Lemma 5.4.5. *For λ defined in (5.23),*

$$\mathbb{E}\left[|\lambda| \mathbb{1}_{\rho\notin[0,2]}\right] \leq \frac{\delta}{4}.$$

Lemma 5.4.6. *For σ defined in (5.23),*

$$\mathbb{E}\left[|\sigma|\mathbb{1}_{\rho\notin[0,2]}\right] \le \frac{\delta}{4}.$$

Putting together these pieces, we finally obtain the following guarantee on Algorithm 13.

Proposition 5.4.7. *The output of Algorithm 13 has total variation distance to π_y bounded by δ. In expectation, Algorithm 13 queries $O(1)$ random f_i and draws $O(1)$ samples from γ_y.*

Proof. The total variation distance bound comes from combining Lemma 5.4.4, (5.24), Lemma 5.4.5, and Lemma 5.4.6. Further, the end probability of each "while" loop is $\Pr[u \le \frac{1}{2}\rho] = \mathbb{E}[\overline{\rho}] = \mathbb{E}_{x\sim\gamma}\,\overline{r}_x \ge \mathbb{E}_{x\sim\gamma_y}\,r_x - \mathbb{E}_{x\sim\gamma_y}\,|\overline{r}_x - r_x|$. We proved in (5.20) that $\mathbb{E}_{x\sim\gamma_y}\,r_x \ge 1$, and combining (5.24), Lemma 5.4.5 and Lemma 5.4.6, shows $\mathbb{E}_{x\sim\gamma_y}\,|\overline{r}_x - r_x| \le \delta \le \frac{1}{2}$. Hence the expected number of loops is ≤ 2, and each loop draws two samples from γ_y, and $O(1)$ many f_i in expectation since $\mathbb{E}\,a^2 = O(1)$. □

5.4.2 Analysis of Algorithm 6

We now prove a mixing time on Algorithm 6 using a standard conductance argument, by using tools developed in Section 5.3. We first define our notion of conductance.

Definition 5.4.8. For a reversible Markov chain with stationary distribution π supported on $\mathcal{X}$ and transition distributions $\{\mathcal{T}_x\}_{x\in\mathcal{X}}$, we define the conductance of the Markov chain by

$$\Phi := \inf_{S\subset\mathcal{X}} \frac{\int_S \mathcal{T}_x(\mathcal{X}\backslash S)\mathrm{d}\pi(x)}{\min\{\pi(S), \pi(\mathcal{X}\backslash S))\}}.$$

We further recall a standard way of lower bounding conductance via isoperimetry.

Lemma 5.4.9 ([LV18], Lemma 13)**.** *In the setting of Definition 5.4.8, let $d : \mathcal{X}\times\mathcal{X}$ be a metric on $\mathcal{X}$. Suppose for any $x, x' \in \mathcal{X}$ with $d(x, x') \le \Delta$,*

$$\|\mathcal{T}_x - \mathcal{T}_{x'}\|_{\mathrm{TV}} \le \frac{1}{2}.$$

Also, suppose that for any partition S_1, S_2, S_3 of $\mathbb{R}^d$, π satisfies the isoperimetric inequality

$$\pi(S_3) \geq C_{\text{iso}} \left(\min_{x \in S_1, y \in S_2} d(x, y) \right) \min \{\pi(S_1), \pi(S_2)\}.$$

Then $\Phi = \Omega(\Delta C_{\text{iso}})$.

Finally, a classical result of [LS93] shows how to upper bound mixing time via conductance.

Lemma 5.4.10 ([LS93], Corollary 1.5)**.** *In the setting of Definition 5.4.8, let π_t be the distribution after t steps of the Markov chain. If the starting distribution π_0 is β-warm with respect to π*

$$\|\pi_t - \pi\|_{\text{TV}} \leq \sqrt{\beta} \left(1 - \frac{\Phi^2}{2}\right)^t.$$

Leveraging Lemmas 5.4.9 and 5.4.10, we prove the following mixing time bound.

Proposition 5.4.11. *Assume the input x_0 to Algorithm 6 is drawn from a β-warm distribution with respect to π, $\eta\mu \leq 1$, and $T = \Omega(\frac{1}{\eta\mu} \log \frac{\beta}{\delta})$ for a sufficiently large constant. Then the output of Algorithm 6 has total variation distance to π bounded by δ.*

Proof. Following the optimal coupling characterization of total variation, whenever the optimal coupling of $y \sim \mathcal{D}_x^{\varphi}$ and $y' \sim \mathcal{D}_{x'}^{\varphi}$ sets $y = y'$ in Line 3 of Algorithm 6, we can couple the resulting distributions in Line 4 as well. This shows that $\|\mathcal{T}_x - \mathcal{T}_{x'}\|_{\text{TV}} \leq \|\mathcal{D}_x^{\varphi} - \mathcal{D}_{x'}^{\varphi}\|_{\text{TV}}$. By Lemma 5.3.2, since φ is convex, ψ is a self-concordant function. Then, combined with Lemma 5.3.8, for any $d_\psi(x, x') \leq \frac{1}{4}$,

$$\|\mathcal{T}_x - \mathcal{T}_{x'}\|_{\text{TV}} \leq \left\|\mathcal{D}_x^{\varphi} - \mathcal{D}_{x'}^{\varphi}\right\|_{\text{TV}} \leq \frac{1}{2}.$$

By Lemma 5.3.7, since $F + \eta\mu\psi$ is $\eta\mu$-relatively strongly convex in ψ, π satisfies the isoperimetric inequality such that for any partition S_1, S_2, S_3 of $\mathbb{R}^d$,

$$\pi(S_3) = \Omega(\sqrt{\eta\mu}) \left(\min_{x \in S_1, y \in S_2} d_\psi(x, y) \right) \min \{\pi(S_1), \pi(S_2)\}.$$

By Lemma 5.4.9, we can then lower bound the conductance by $\Phi = \Omega(\sqrt{\eta\mu})$. Choosing a

sufficiently large constant in T, we conclude by Lemma 5.4.10 the desired $\|\pi_T - \pi\|_{\mathrm{TV}} \leq \sqrt{\beta}\exp(-\frac{T\Phi^2}{2}) \leq \delta$. □

By combining Proposition 5.4.7 with Proposition 5.4.11, we can now complete our analysis.

Theorem 5.4.12. *In the setting of Problem 5.4.1, let $\eta\mu \leq 1$ and assume x_0 has a β-warm distribution with respect to π defined in (5.13). Further for sufficiently large constants suppose $\frac{1}{\eta} = \Omega(G^2 \log \frac{\log \beta}{\delta\eta\mu})$ and*

$$T = \Theta\left(\frac{1}{\eta\mu}\log\frac{\beta}{\delta}\right).$$

Algorithm 6 using Algorithm 13 with error parameter $\frac{\delta}{2T}$ to implement Line 4 returns a point with δ total variation distance to π, querying $O(T)$ random f_i in expectation.

Proof. Proposition 5.4.11 guarantees that if each call to Line 4 of Algorithm 6 is implemented exactly, we obtain $\frac{\delta}{2}$ total variation to π. Further, the total variation error accumulated over T calls to Algorithm 13 is less than $\frac{\delta}{2}$ by a union bound on Proposition 5.4.7. Combining these bounds results in the desired total variation guarantee, and the complexity bound follows from Proposition 5.4.7. □

We note that given sample access to $\exp(-\eta\mu\psi(x))\mathbb{1}_{x\in\mathcal{X}}$, a distribution which only depends on the choice of φ and $\mathcal{X}$ (and not the function F), we obtain $\beta \leq \exp(GD)$ in Theorem 5.4.12.

Lemma 5.4.13. *In the setting of Problem 5.4.1, the density ν satisfying*

$$\mathrm{d}\nu(x) \propto \exp(-\eta\mu\psi(x))\mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x$$

is $\exp(GD)$-warm for π defined in (5.13).

Proof. Note that for all $x, w \in \mathcal{X}$, $|F(x) - F(w)| \leq GD$. Further recall $\pi \propto \exp(-F)\nu$. We conclude by observing that for all $x \in \mathcal{X}$,

$$\frac{\exp\left(-F(x)\right)\nu(x)}{\int_{\mathcal{X}}\exp(-F(w))\nu(w)\mathrm{d}w} \cdot \frac{\int_{\mathcal{X}}\nu(w)\mathrm{d}w}{\nu(x)} = \frac{\int_{\mathcal{X}}\nu(w)\mathrm{d}w}{\int_{\mathcal{X}}\exp(F(x)-F(w))\nu(w)\mathrm{d}w} \leq \exp\left(GD\right).$$

□

5.5 Applications

In this section, we discuss applications of the sampling scheme we develop in Section 5.4. We begin by specializing our machinery to ℓ_p and Schatten-p norms in Section 5.5.1. We then give new algorithms with improved zeroth-order query complexity for private convex optimization in Section 5.5.2. Finally, in Section 5.5.3 we discuss computational issues regarding the specific LLT we introduce.

5.5.1 LLT for ℓ_p and Schatten-p norms

Throughout this section we fix some $p \in [1, 2]$, and define the dual value $q \geq 2$ such that $\frac{1}{q} + \frac{1}{p} = 1$. It is well-known that the ℓ_q norm and ℓ_p norm are dual, as are the corresponding Schatten norms. In light of Lemma 5.3.3, to obtain a sampler catering to the ℓ_p geometry for example, it suffices to take the LLT of a smooth function in ℓ_q. We provide the latter by recalling the following fact.

Fact 5.5.1. *Let $p \in [1, 2]$, $q \geq 2$ satisfy $\frac{1}{p} + \frac{1}{q}$. If $\|\cdot\|_q$ is a vector ℓ_q norm, $\frac{1}{2}\|\cdot\|_q^2$ is $\frac{1}{p-1}$-smooth in the ℓ_q norm, and if $\|\cdot\|_q$ is a matrix Schatten-q norm, $\frac{1}{2}\|\cdot\|_q^2$ is $\frac{1}{p-1}$-smooth in the Schatten-q norm.*

Proof. This follows (for example) from three well-known facts: 1) that $\frac{1}{2}\|\cdot\|_q^2$ and $\frac{1}{2}\|\cdot\|_p^2$ are conjugate functions in both the vector and matrix cases, 2) that the conjugate of a m-strongly convex function in a norm is $\frac{1}{m}$-smooth in the dual norm [KST09], and 3) that $\frac{1}{2}\|\cdot\|_p^2$ is $(p-1)$-strongly convex in $\|\cdot\|_p$ in both the vector and matrix cases [BCL94]. □

ℓ_p norms. Next, for any $a > 0$, when the context is clearly about vector spaces, we define

$$\psi_{p,a}(x) := \log\left(\int \exp\left(\langle x, y\rangle - a\|y\|_q^2\right) \mathrm{d}y\right). \tag{5.25}$$

Note that as the LLT of a $\frac{2a}{p-1}$-smooth function in ℓ_q, $\psi_{p,a}$ is $\Omega(\frac{p-1}{a})$-strongly convex in ℓ_p by Lemma 5.3.3. In applications we fix a value of $\eta > 0$, set $a = \Theta((p-1)\eta)$, and use $\eta\psi_{p,a}$ as our strongly convex regularizer in ℓ_p. We next provide a bound on the range of $\psi_{p,a}$.

Lemma 5.5.2. *Let $a > 0$ and let $d \in \mathbb{N}$ be at least a sufficiently large constant. The additive range of $\psi_{p,a}$ over $\{x \in \mathbb{R}^d \mid \|x\|_p \le 1\}$ is*

$$O\left(1 + \frac{1}{a} + \sqrt{\frac{d}{a}\log\left(a + \frac{d}{a}\right)}\right).$$

In particular, for $a \le \frac{1}{d \log d}$, the additive range is $O(\frac{1}{a})$.

Proof. Throughout the proof denote for simplicity $\psi := \psi_{p,a}$ and let

$$\mathcal{D}_x^{\varphi}(y) \propto \exp\left(\langle x, y\rangle - a\|y\|_q^2\right)$$

be the associated density. By the characterization of $\nabla\psi$ in Lemma 5.3.1 and the fact that the associated density $\mathcal{D}_x^{\varphi}$ is symmetric in y for $x = 0$, we have $\nabla\psi(0) = 0$ and hence it suffices to bound $\psi(x) - \psi(0)$ for $\|x\|_q \le 1$. We simplify this expression as

$$\begin{aligned}\psi(x) - \psi(0) &= \log\left(\int \exp\left(\langle x, y\rangle - a\|y\|_q^2\right)\mathrm{d}y\right) - \log\left(\int \exp\left(-a\|y\|_q^2\right)\mathrm{d}y\right) \\ &= \log\left(\int \exp\left(\langle x, y\rangle\right)\frac{\exp\left(-a\|y\|_q^2\right)}{\int \exp\left(-a\|y\|_q^2\right)\mathrm{d}y}\mathrm{d}y\right) = \log\left(\mathop{\mathbb{E}}_{y\sim\mathcal{D}_0^{\varphi}}\left[\exp\left(\langle x, y\rangle\right)\right]\right).\end{aligned} \tag{5.26}$$

Next, let π be the probability density on $\mathbb{R}_{\ge 0}$ such that

$$\mathrm{d}\pi(r) \propto r^{d-1}\exp\left(-ar^2\right)\mathrm{d}r.$$

We note $\mathrm{d}\pi(r)$ is the density of the scalar quantity $r = \|y\|_q$ for $y \sim \mathcal{D}_0^{\varphi}$. This can be seen by taking a derivative of the volume of the ℓ_p ball of radius r, which scales as r^d, so the surface area of the ball scales as r^{d-1}. By Hölder's inequality, $\langle x, y\rangle \le \|y\|_q$ for all y, since $\|x\|_p \le 1$. We then continue (5.26) and bound $\psi(x) - \psi(0) \le \log(\mathbb{E}_{r\sim\pi}\exp(r))$, and the conclusion follows from Lemma 5.5.3. □

Lemma 5.5.3. *For any $a > 0$ and $d \in \mathbb{N}$ at least a sufficiently large constant,*

$$\log\left(\frac{\int_0^\infty \exp\left((d-1)\log r + r - ar^2\right)\mathrm{d}r}{\int_0^\infty \exp\left((d-1)\log r - ar^2\right)\mathrm{d}r}\right) \le 8 + \frac{8}{a} + \sqrt{\frac{8d}{a}\log\left(a + \frac{d}{a}\right)}.$$

Proof. Throughout this proof let

$$Z := \int_0^\infty \exp\left((d-1)\log r - \alpha r^2\right)\mathrm{d}r = \frac{\Gamma(\frac{d}{2})}{2a^{\frac{d}{2}}},\ \tau := 7 + \frac{8}{a} + \sqrt{\frac{8d}{a}\log\left(a + \frac{d}{a}\right)}.$$

Next we split the numerator of the left-hand side into two integrals:

$$I_1 := \int_0^\tau \exp\left((d-1)\log r + r - ar^2\right)\mathrm{d}r,$$
$$I_2 := \int_\tau^\infty \exp\left((d-1)\log r + r - ar^2\right)\mathrm{d}r.$$

It is immediate that $I_1 \le \exp(\tau)Z$. Further, we recognize that for $r \ge \tau$,

$$\max\left(r, (d-1)\log r\right) \le \frac{ar^2}{4}.$$

The first piece in the maximum is clear from $\tau \ge \frac{4}{a}$. The second follows since $\frac{r^2}{\log r}$ is an increasing function for $r \ge 7$, and either $\frac{4d}{a} \le 10$ in which case we use $\frac{7^2}{\log 7} \ge 10$, or we let $C := \frac{4d}{a}$ and use

$$\frac{r^2}{\log r} \ge C \text{ for } r \ge \sqrt{2C\log\frac{C}{4}},\ C \ge 10.$$

Hence we may bound

$$I_2 \le \int_\tau^\infty \exp\left(-\frac{ar^2}{2}\right) = \sqrt{\frac{2\pi}{a}} \Pr_{t\sim\mathcal{N}(0,a^{-1})}[t \ge \tau] \le \frac{2}{a\tau}\exp\left(-\frac{a\tau^2}{2}\right).$$

Above, we used Mill's inequality

$$\Pr_{t\sim\mathcal{N}(0,\sigma^2)}[t \ge \tau] \le \sqrt{\frac{2}{\pi}}\frac{\sigma}{\tau}\exp\left(-\frac{\tau^2}{2\sigma^2}\right).$$

Further for our τ, our upper bound on I_1 is larger than our upper bound on I_2. To see this,

$$\tau\left(1+\frac{a\tau}{2}\right)+\frac{d}{3}\log d \geq \frac{d}{2}\log a \implies \exp\left(\tau\left(1+\frac{a\tau}{2}\right)\right)\Gamma\left(\frac{d}{2}\right) \geq a^{\frac{d}{2}}$$
$$\implies \frac{\exp(\tau)\,\Gamma(\frac{d}{2})}{2a^{\frac{d}{2}}} \geq \frac{4}{a\tau}\exp\left(-\frac{a\tau^2}{2}\right).$$

The first inequality is because $a\tau^2 \geq d\log a$. The first implication then follows by exponentiating and using $\log\Gamma(\frac{d}{2}) \geq \frac{d}{3}\log d$ for sufficiently large d, and the second implication follows by rearranging and using $a\tau \geq 4$. Finally the conclusion follows from

$$\log\left(\frac{\int_0^\infty \exp\left((d-1)\log r + r - ar^2\right)\mathrm{d}r}{\int_0^\infty \exp\left((d-1)\log r - ar^2\right)\mathrm{d}r}\right) \leq \log\left(\frac{2\exp(\tau)Z}{Z}\right) \leq \tau+1.$$

□

Schatten-p norms. When the context is clearly about matrix spaces, we analogously define

$$\psi_{p,a}(\mathbf{X}) := \log\left(\int \exp\left(\langle \mathbf{X},\mathbf{Y}\rangle - a\left\|\mathbf{Y}\right\|_q^2\right)\mathrm{d}y\right).$$

The proof of Lemma 5.5.2 implies the following analogous range bound in this setting.

Corollary 5.5.4. *Let $a>0$ and let $d_1,d_2\in\mathbb{N}$ be at least sufficiently large constants. The additive range of $\psi_{p,a}$ over $\{\mathbf{X}\in\mathbb{R}^{d_1\times d_2} \mid \|\mathbf{X}\|_p \leq 1\}$ is*

$$O\left(1+\frac{1}{a}+\sqrt{\frac{d_1d_2}{a}\log\left(a+\frac{d_1d_2}{a}\right)}\right).$$

In particular, for $a \leq \frac{1}{d_1d_2\log(d_1d_2)}$, the additive range is $O(\frac{1}{a})$.

5.5.2 Zeroth-order private convex optimization

In this section, we consider a pair of closely-related problems in private convex optimization. Let $\mathcal{S}$ be a domain, and let $n\in\mathbb{N}$. We say that a mechanism (randomized algorithm) $\mathcal{M}:\mathcal{S}^n\to\Omega$ satisfies (ε,δ)-differential privacy (DP) if for any event $S\subseteq\Omega$ where Ω is the

output space, and any two datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$ which differ in exactly one element,

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \le \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta.$$

We next define the private optimization problems we study.

Problem 5.5.5 (DP-ERM and DP-SCO)**.** Let $n \in \mathbb{N}$, $\varepsilon, \delta \in (0,1)$, $D, G \ge 0$, and let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex with diameter in a norm $\|\cdot\|_{\mathcal{X}}$ at most D. Let $\mathcal{P}$ be a distribution over a set $\mathcal{S}$ such that for any $s \in \mathcal{S}$, there is a $f(\cdot; s) : \mathcal{X} \to \mathbb{R}$ which is convex and G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$. Let $\mathcal{D} := \{s_i\}_{i \in [n]}$ consist of n independent draws from $\mathcal{P}$, and let $f_i := f(\cdot; s_i)$ for all $i \in [n]$.

In the *differentially private empirical risk minimization (DP-ERM)* problem, we receive $\mathcal{D}$ and wish to design a mechanism $\mathcal{M}$ which satisfies (ε, δ)-DP and approximately minimizes

$$f^{\text{erm}}(x) := \frac{1}{n} \sum_{i \in [n]} f_i(x).$$

In the *differentially private stochastic convex optimization (DP-SCO)* problem, we receive $\mathcal{D}$ and wish to design a mechanism $\mathcal{M}$ which satisfies (ε, δ)-DP and approximately minimizes

$$F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s \sim \mathcal{P}} \left[f(x; s) \right].$$

The following powerful general-purpose result was proven in [GLL+23] reducing the DP-ERM and DP-SCO problems to logconcave sampling problems catered to the $\|\cdot\|_{\mathcal{X}}$ geometry. We slightly improve the parameter settings used by Theorem 4 of [GLL+23] for DP-SCO by noting that a smaller value of k also suffices (due to the larger error bound), as observed by [GLL22].

Proposition 5.5.6 (Theorem 3, Theorem 4, [GLL+23], Theorem 6.9, [GLL22])**.** *In the setting of Problem 5.5.5, let $k \ge 0$, and let $r : \mathcal{X} \to \mathbb{R}$ be 1-strongly convex with respect to $\|\cdot\|_{\mathcal{X}}$, with additive range at most Θ. Let ν be the density on $\mathcal{X}$ satisfying $\mathrm{d}\nu(x) \propto$*

$\exp(-k(f^{\mathrm{erm}}(x)+\mu r(x)))\mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x$. *Then the algorithm which returns a sample from* ν *for*

$$k=\frac{\sqrt{d}n\varepsilon}{G\sqrt{2\Theta\log\frac{1}{2\delta}}},\ \mu=\frac{2G^2k\log\frac{1}{2\delta}}{n^2\varepsilon^2},$$

satisfies (ε,δ)*-DP, and guarantees*

$$\mathop{\mathbb{E}}_{x\sim\nu}\left[f^{\mathrm{erm}}(x)\right]-\min_{x\in\mathcal{X}}f^{\mathrm{erm}}(x)\leq O\left(G\sqrt{\Theta}\cdot\frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}\right).$$

Further, the algorithm which returns a sample from ν *for*

$$k=\frac{1}{G\sqrt{\Theta}}\cdot\sqrt{\left(\frac{d\log\frac{1}{2\delta}}{\varepsilon^2n^2}+\frac{1}{n}\right)}\cdot\min\left(\frac{\varepsilon^2n^2}{\log\frac{1}{2\delta}},nd\right),\ \mu=G^2k\cdot\max\left(\frac{\log\frac{1}{2\delta}}{n^2\varepsilon^2},\frac{1}{nd}\right)$$

satisfies (ε,δ)*-DP, and guarantees*

$$\mathop{\mathbb{E}}_{x\sim\nu}\left[F_{\mathcal{P}}(x)\right]-\min_{x\in\mathcal{X}}F_{\mathcal{P}}(x)\leq O\left(G\sqrt{\Theta}\cdot\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{n\varepsilon}+\frac{1}{\sqrt{n}}\right)\right).$$

Armed with Proposition 5.5.6 and the sampler in Theorem 5.4.12, we give our main results on Problem 5.5.5.

Assumption 5.5.7. *Fix* $p\in[1,2]$ *and* $k,a,\eta,\mu>0$. *In the setting of Problem 5.5.5, assume there is an algorithm* $\mathcal{A}$ *which returns a point drawn from a* β*-warm start to the density* ν *satisfying*

$$\mathrm{d}\nu(x)\propto\exp\left(-k\left(f^{\mathrm{erm}}(x)+\eta\mu\psi_{p,a}(x)\right)\right)\mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x.$$

Theorem 5.5.8. *Let* $p\in[1,2]$, $\varepsilon,\delta\in(0,1)$. *In the setting of Problem 5.5.5 where* $\|\cdot\|_{\mathcal{X}}$ *is the* ℓ_p *norm on* $\mathbb{R}^d$, *there is an* (ε,δ)*-differentially private algorithm* $\mathcal{M}_{\mathrm{erm}}$ *which produces*

$x \in \mathcal{X}$ such that

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{erm}}}[f^{\mathrm{erm}}(x)] - \min_{x \in \mathcal{X}} f^{\mathrm{erm}}(x) = O\left(\frac{GD}{\sqrt{p-1}} \cdot \frac{\sqrt{d \log \frac{1}{\delta}}}{n\varepsilon}\right) \text{ for } p \in (1, 2],$$

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{erm}}}[f^{\mathrm{erm}}(x)] - \min_{x \in \mathcal{X}} f^{\mathrm{erm}}(x) = O\left(GD\sqrt{\log d} \cdot \frac{\sqrt{d \log \frac{1}{\delta}}}{n\varepsilon}\right) \text{ for } p = 1.$$

Further, there is an (ε, δ)-differentially private algorithm $\mathcal{M}_{\mathrm{sco}}$ which produces $x \in \mathcal{X}$ such that

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{sco}}}[F_{\mathcal{P}}(x)] - \min_{x \in \mathcal{X}} F_{\mathcal{P}}(x) = O\left(\frac{GD}{\sqrt{p-1}} \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d \log \frac{1}{\delta}}}{n\varepsilon}\right)\right) \text{ for } p \in (1, 2],$$

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{sco}}}[F_{\mathcal{P}}(x)] - \min_{x \in \mathcal{X}} F_{\mathcal{P}}(x) = O\left(GD\sqrt{\log d} \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d \log \frac{1}{\delta}}}{n\varepsilon}\right)\right) \text{ for } p = 1.$$

Both $\mathcal{M}_{\mathrm{erm}}$ and $\mathcal{M}_{\mathrm{sco}}$ call $\mathcal{A}$ in Assumption 5.5.7, appropriately parameterized, once. $\mathcal{M}_{\mathrm{erm}}$ uses

$$O\left(\left(1 + \frac{n^2\varepsilon^2}{\log \frac{1}{\delta}}\right) \log\left(\frac{(1 + n\varepsilon)\log \beta}{\delta}\right) \log \frac{\beta}{\delta}\right).$$

additional value queries to some $f(\cdot; s_i)$, and $\mathcal{M}_{\mathrm{sco}}$ uses

$$O\left(\min\left(nd,\ 1 + \frac{n^2\varepsilon^2}{\log \frac{1}{\delta}}\right) \log\left(\frac{(1 + n\varepsilon)\log \beta}{\delta}\right) \log \frac{\beta}{\delta}\right)$$

additional value queries to some $f(\cdot; s_i)$.

Proof. First, we slightly simplify the setting of Problem 5.5.5. We may first assume that $D = 1$, i.e. $\mathcal{X}$ has diameter at most 1 in $\|\cdot\|_{\mathcal{X}}$. If the diameter is bounded by some $D \neq 1$, we can rescale the domain $\mathcal{X} \leftarrow \frac{1}{D}\mathcal{X}$, and remap to the modified functions $f(x; s) \leftarrow f(Dx; s)$ over this modified domain for all $s \in \mathcal{S}$. It is clear the Lipschitz constant rescales as $G \leftarrow GD$ as a result. Next, we assume $(n\varepsilon)^2 \geq d\Theta \log \frac{1}{\delta}$ where $\Theta = \min(\frac{1}{p-1}, \log d)$. In the other case, in light of the diameter bound on $\mathcal{X}$ and the Lipschitz assumption, returning a random point in $\mathcal{X}$ attains the error bound claimed. Finally, assume $p \in (1, 2]$, as otherwise

we set $p \leftarrow 1 + \frac{1}{\log d}$, which only affects bounds by constant factors, since $\|\cdot\|_p$ is affected by $O(1)$ multplicatively everywhere under this change.

Under these simplifications, we choose the parameters k and μ according to Proposition 5.5.5 for each problem. Assume for now that Θ for the regularizer r we choose is bounded by a universal constant times $\frac{1}{p-1}$. Then the Lipschitz constant of kf^{erm} in either case of Proposition 5.5.5 is

$$kG = \Omega\left(\min\left(\frac{\sqrt{(p-1)d}n\varepsilon}{\sqrt{\log\frac{1}{\delta}}}, d\sqrt{n}\right)\right) = \Omega(d),$$

as implied by our earlier simplification. We hence may choose $\mathcal{I}$ to be uniform over $[n]$, and

$$\eta = O\left(\frac{1}{k^2G^2\log\frac{(1+n\varepsilon)\log\beta}{\delta}}\right)$$

for a sufficiently small constant to use Theorem 5.4.12. Under this setting we certainly have $\eta = O(\frac{1}{d^2})$, so letting $r := \eta\psi_{p,a}$ for $a := \frac{\eta(p-1)}{2}$ shows that r is η times the LLT of an η-smooth function in ℓ_q. By Lemma 5.3.3, r is indeed 1-strongly convex in ℓ_p, and Lemma 5.5.2 bounds its range by $\Theta = O(\frac{1}{p-1})$ satisfying our earlier assumption, where we use $a = O(\frac{1}{d^2})$. The runtime finally follows by applying our choices of k, μ in Proposition 5.5.6, with our choice of η, in Theorem 5.4.12, where we ensure that $\eta \cdot k\mu \leq 1$ by choosing a smaller η if this is not the case (so Theorem 5.4.12 applies). Finally, to account for total variation error in our sampler, it suffices to adjust the failure probability δ by a constant and take a union bound over the privacy definition and the failure of Theorem 5.4.12. □

By combining the proof strategy of Theorem 5.5.8 with Corollary 5.5.4 instead of Lemma 5.5.2, we immediately obtain the following corollary in the case of Schatten norms.

Corollary 5.5.9. *Let $p \in [1, 2]$, $\varepsilon, \delta \in (0, 1)$. In the setting of Problem 5.5.5 where $\|\cdot\|_{\mathcal{X}}$ is the Schatten-p norm on $\mathbb{R}^{d_1 \times d_2}$, there is an (ε, δ)-differentially private algorithm $\mathcal{M}_{\text{erm}}$*

which produces $\mathbf{X} \in \mathcal{X}$ *such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{erm}}}[f^{\mathrm{erm}}(\mathbf{X})] - \min_{\mathbf{X}\in\mathcal{X}} f^{\mathrm{erm}}(\mathbf{X}) = O\left(\frac{GD}{\sqrt{p-1}} \cdot \frac{\sqrt{d_1 d_2 \log\frac{1}{\delta}}}{n\varepsilon}\right) \text{ for } p \in (1,2],$$

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{erm}}}[f^{\mathrm{erm}}(\mathbf{X})] - \min_{\mathbf{X}\in\mathcal{X}} f^{\mathrm{erm}}(\mathbf{X}) = O\left(GD\sqrt{\log(d_1 d_2)} \cdot \frac{\sqrt{d_1 d_2 \log\frac{1}{\delta}}}{n\varepsilon}\right) \text{ for } p = 1.$$

Further, there is an (ε, δ)*-differentially private algorithm* $\mathcal{M}_{\mathrm{sco}}$ *which produces* $\mathbf{X} \in \mathcal{X}$ *such that*

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{sco}}}[F_{\mathcal{P}}(\mathbf{X})] - \min_{\mathbf{X}\in\mathcal{X}} F_{\mathcal{P}}(\mathbf{X}) = O\left(\frac{GD}{\sqrt{p-1}} \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d_1 d_2 \log\frac{1}{\delta}}}{n\varepsilon}\right)\right) \text{ for } p \in (1,2],$$

$$\mathop{\mathbb{E}}_{\mathcal{M}_{\mathrm{sco}}}[F_{\mathcal{P}}(\mathbf{X})] - \min_{\mathbf{X}\in\mathcal{X}} F_{\mathcal{P}}(\mathbf{X}) = O\left(GD\sqrt{\log(d_1 d_2)} \cdot \left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d_1 d_2 \log\frac{1}{\delta}}}{n\varepsilon}\right)\right) \text{ for } p = 1.$$

Both $\mathcal{M}_{\mathrm{erm}}$ *and* $\mathcal{M}_{\mathrm{sco}}$ *call* $\mathcal{A}$ *in Assumption 5.5.7, appropriately parameterized, once.* $\mathcal{M}_{\mathrm{erm}}$ *uses*

$$O\left(\left(1 + \frac{n^2\varepsilon^2}{\log\frac{1}{\delta}}\right)\log\left(\frac{(1+n\varepsilon)\log\beta}{\delta}\right)\log\frac{\beta}{\delta}\right).$$

additional value queries to some $f(\cdot; s_i)$*, and* $\mathcal{M}_{\mathrm{sco}}$ *uses*

$$O\left(\min\left(nd_1d_2,\ 1 + \frac{n^2\varepsilon^2}{\log\frac{1}{\delta}}\right)\log\left(\frac{(1+n\varepsilon)\log\beta}{\delta}\right)\log\frac{\beta}{\delta}\right)$$

additional value queries to some $f(\cdot; s_i)$*.*

5.5.3 Oracle access for $\psi_{p,a}$

In Theorem 5.5.8 and Corollary 5.5.9, we only bounded the value oracle complexity of our sampling algorithms. The remainder of the steps in Algorithm 6 and its subroutine Algorithm 13 require samples from densities of the form $\mathrm{d}\pi_x$ (for some $x \in \mathcal{X}$) or $\mathrm{d}\gamma_y$ (for

some $y \in \mathbb{R}^d$), defined in (5.15) and (5.16) respectively and reproduced here for convenience:

$$\begin{aligned} \mathrm{d}\pi_x(y) &= \exp\left(\langle x, y\rangle - \psi(x) - \varphi(y)\right) \mathrm{d}y, \\ \mathrm{d}\gamma_y(x) &\propto \exp\left(-\eta\mu\psi(x) - (\psi(x) - \langle x, y\rangle)\right) \mathbb{1}_{\mathcal{X}}(x)\mathrm{d}x. \end{aligned} \tag{5.27}$$

These densities are independent of the function F in Problem 5.4.1 and hence do not require additional value oracle queries in the setting of Problem 5.4.1. In general, the complexity of these steps depends on the complexity of the functions φ and ψ, and the set $\mathcal{X}$. We now discuss strategies for sampling from π_x and γ_y in specific settings described by Section 5.5.1, which we first briefly summarize.

1. We describe a method based on the inverse Laplace transform for sampling from π_x and evaluating $\psi_{p,a}$ with complexity linear in the dimension d in the vector setting.

2. Under efficient value oracle access to $\psi_{p,a}$ and membership oracle access to $\mathcal{X}$, general-purpose results [LV07, JLLV21, JLV22] imply polynomial-time samplers for γ_y.

3. We discuss generalizations of these methods to the matrix setting, and naive sampling methods. We draw a loose connection to the HCIZ integral from harmonic analysis, and suggest how it may potentially help in the structured sampling task for LLTs in Schatten norms.

ℓ_p setting. We first discuss the case when $\mathcal{X} \subset \mathbb{R}^d$ is a set on vectors equipped with the ℓ_p norm for some $p \in [1, 2]$, and we let $q \geq 2$ satisfy $\frac{1}{p} + \frac{1}{q} = 1$. We follow the notation (5.25).

In order to sample from the density π_x, we use an *inverse Laplace transform* decomposition. For a parameter $c \in [0, 1)$, we define the density μ_c supported on $\mathbb{R}_{\geq 0}$, such that for all $t \geq 0$,

$$\exp(-t^c) = \int_0^\infty \exp\left(-\lambda t\right) \mu_c(\lambda)\mathrm{d}\lambda. \tag{5.28}$$

Intuitively, the density $\mu_c(\lambda)$ and the corresponding decomposition (inverse Laplace transform) (5.28) aims to express the more heavy-tailed function $\exp(-t^c)$ as a distribution over

the lighter-tailed functions $\exp(-\lambda t)$. The inverse Laplace transform densities μ_c are well-studied in the probability theory literature, and correspond to *stable count distributions* parameterized by c. For example, it is well-known that $\mu_{\frac{1}{2}}$ is the Lévy distribution

$$\mathrm{d}\mu_{\frac{1}{2}}(\lambda) = \frac{1}{2\sqrt{\pi}\lambda^{\frac{3}{2}}} \exp\left(-\frac{1}{4\lambda}\right) \mathrm{d}\lambda.$$

We refer the reader to references e.g. [Mai07] on properties of the densities μ_c, and for now assume we can access and sample from these one-dimensional distributions in closed form for simplicity. Given this decomposition, we can then write

$$\begin{aligned} \exp(\psi_{p,a}(x)) &= \int \exp\left(\langle x, y\rangle - a\left\|y\right\|_q^2\right) \mathrm{d}y \\ &= \int_0^\infty \left(\int \exp\left(\langle x, y\rangle - \lambda a^{\frac{q}{2}} \left\|y\right\|_q^q\right) \mathrm{d}y\right) \mu_{\frac{2}{q}}(\lambda)\mathrm{d}\lambda \\ &= \int_0^\infty \prod_{i\in[d]} \left(\int_{-\infty}^\infty \exp\left(x_i y_i - \lambda a^{\frac{q}{2}} y_i^q\right) \mathrm{d}y_i\right) \mu_{\frac{2}{q}}(\lambda)\mathrm{d}\lambda. \end{aligned} \tag{5.29}$$

The decomposition (5.29) reduces the problem of sampling from π_x to d one-dimensional problems. To sample $\propto \exp(\langle x, y\rangle - a\left\|y\right\|_q^2)$, we can first sample λ from the density μ_c for $c = \frac{2}{q}$, and then sample each coordinate y_i proportionally to $\exp(x_i y_i - \lambda a^{\frac{q}{2}} y_i^q)$ conditioned on the sampled λ.

This decomposition also gives us an efficient value oracle for $\psi_{p,a}$, by evaluating (5.29) as a one-dimensional integral over λ, where the integrand may be evaluated as a product of d one-dimensional integrals. Under membership oracle access to $\mathcal{X}$, the problem of sampling from γ_y then falls under a generic logconcave sampling setup studied in a long line of work building upon [DFK91]. The state-of-the-art general-purpose logconcave sampler, which combines the algorithms of [LV07, JLLV21] with the isoperimetric bound in [JLV22] (improving recent breakthroughs by [Che21a, KL22]), requires roughly d^3 value oracle calls to $\psi_{p,a}$ and membership oracle calls to $\mathcal{X}$.

In principle, for structured sets $\mathcal{X}$ (such as ℓ_p balls), the particular explicit structure of $\psi_{p,a}$ and $\mathcal{X}$ may be exploited to design more efficient samplers for the densities γ_y, analogously to our custom linear-time sampler for π_x. However, it should be noted that

the sampling problem for γ_y appears to be quite a bit more challenging than the problem for π_x. We leave the investigation of explicit sampler design for γ_y as an interesting open problem for future work.

Schatten-p setting. The situation is somewhat less straightforward in the matrix case. Here, the key computational problem in replicating the strategy suggested by (5.29) is evaluating the integral

$$\int \exp\left(\langle \mathbf{X}, \mathbf{Y}\rangle - C\,\|\mathbf{Y}\|_q^q\right) \mathrm{d}\mathbf{Y}, \tag{5.30}$$

where the integral is over $\mathbf{Y} \in \mathbb{R}^{d_1 \times d_2}$, and $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2}$, $C > 0$ are fixed. The difficulty is $\langle \mathbf{X}, \mathbf{Y}\rangle$ decomposes coordinatewise, whereas $\|\mathbf{Y}\|_q^q$ decomposes spectrally.[8] At least superficially, this is similar to the challenge faced when evaluating the Harish-Chandra-Itzykson-Zuber (HCIZ) formula

$$\int \exp\left(\mathrm{Tr}\left(\mathbf{AUBU}^\dagger\right)\right) \mathrm{d}\mathbf{U}, \tag{5.31}$$

where the integral is over the Haar measure on (complex) unitary matrices $\mathbf{U}$, and $\mathbf{A}$, $\mathbf{B}$ are Hermitian. By dropping the $-C\,\|\mathbf{Y}\|_q^q$ term in (5.30) and only integrating over unitary conjugations of a fixed matrix $\mathbf{Y}$, we arrive at a generalization of (5.31). The difficulty in evaluating (5.31) is also a sort of tension between the eigenspaces of $\mathbf{A}$ and $\mathbf{B}$. Nonetheless, (5.31) has a (polynomial-time computable) exact formula, which was famously discovered independently by [HC57, IZ80]. Furthermore, [LMV21] recently obtained a polynomial-time sampler for the density induced by (5.31); while a sampler for (5.30) would follow from logconcavity and general-purpose results, it would be far from cheap, so ways of exploiting structure are fruitful to explore.

As a proof-of-concept, evaluating the integral (5.30) in (polynomial-time computable) closed form is a minimal requirement for implementing the $\mathbf{X}$-oracles in (5.27) used by our algorithm. Even this problem appears challenging, but (as summarized cleanly by [Tao13, McS21]) a plethora of techniques exist for proving the HCIZ formula, some based

[8] Note that because $\|\cdot\|_q$ is unitarially invariant, we may assume $\mathbf{X}$ is diagonal.

on tools from stochastic processes. We pose the efficient computability of the integral (5.30) as another explicit open question.

5.6 Conclusion

We believe our work is a significant step towards developing the theory of LLTs and paving the way for their use in designing sampling algorithms. There are a number of important questions left open by our work, which we find interesting and potentially fruitful for the community to explore.

Stronger mixing time bounds. Perhaps the most immediate open question regarding our alternating sampling framework in Section 5.4 is to obtain a better understanding of its mixing time. As discussed in Section 4.1.1, Theorem 5.4.12's mixing time scales linearly in $\log \beta$, which as demonstrated by Lemma 5.4.13 (and related other settings, e.g. MALA [CLA+21, LST21a]) can result in additional polynomial overhead in problem parameters: for what φ, ψ is this avoidable? Notably, it is avoided for the Euclidean proximal sampler [LST21b] by working directly with KL divergence (as opposed to the larger χ^2 distance typically used by proofs using conductance bounds). Different proofs of this $\log \log \beta$ dependency for the Euclidean proximal sampler were then subsequently obtained by [CCSW22, CE22]. We also mention that $\log \log \beta$ dependences may sometimes follow via average conductance techniques (e.g. [LK99]), which may apply to our Markov chain.

Samplers for explicit distributions. Our results Theorem 5.4.12 and 5.5.8 mainly focused on bounding the query complexity to the function F, or samples f_i from the distribution defining it. The total computational complexity of a practical implementation of Algorithm 6 also includes the cost of sampling from the distributions (5.27), which are "data-independent" for this problem (only depending on explicit functions and sets instead of F). In Section 5.5.3, we give a linear-time sampler for π_x and a polynomial-time sampler for γ_y under the ℓ_p geometry, but it is interesting to obtain faster samplers for particular structured choices of $(\varphi, \mathcal{X})$ of importance in applications.

LLT beyond proximal sampling. More generally, we believe it is worthwhile to obtain a better understanding of specific choices of (φ, ψ), e.g. the examples in Section 5.5.1, from an algorithmic perspective. LLTs satisfy appealing properties such as self-concordance, strong convexity, and isoperimetry making them well-suited for frameworks beyond Algorithm 6, such as discretized MLD [AC21] and Metropolized sampling methods discussed in Section 2.1. Bounding the complexity of their use in these applications necessitates an improved understanding of specific LLTs.

LLT as a dual object. Finally, a tantalizing open question in the theory of well-conditioned sampling (even in the ℓ_2 setting) is whether acceleration is achievable, i.e. mixing times scaling with the square root of the condition number (which is famously possible in optimization [Nes83]). The duality of Fenchel conjugates appears to play a key role in acceleration, as made explicit by [WA18, CST21], so a better understanding of duality may be helpful in the corresponding endeavor for sampling. The LLT is a natural candidate for a dual object in sampling, as it arises via joint densities on an extended space (5.2), and satisfies properties such as strong convexity-smoothness duality. Can we demystify this relationship, and use it to obtain faster samplers?

5.7 Information-theoretic lower bound

In this section, we show that prior information-theoretic lower bounds from [DJWW15] and [GLL22] can be straightforwardly extended to the settings studied by this paper to show that the value oracle complexities used by our algorithms in Sections 5.3 and 5.5 are near-optimal. We first recall some notation from prior work and summarize previous results we will leverage.

Setup. We consider the setting of stochastic optimization where there is a distribution over distributions $\{\mathcal{P}_v\}_v$ indexed by v. An index v is randomly selected, and we consider algorithms interacting with $\mathcal{P}_v$ in one of two different ways. Letting $k \in \mathbb{N}$ and $\mathcal{X} \subset \mathbb{R}^d$, [DJWW15] defined a family of algorithms $\mathbb{A}_k$ such that $\mathcal{A} \in \mathbb{A}_k$ can (adaptively) query a sequence of k values $f(x; s)$ where $x \in \mathcal{X}$ and s is a fresh random sample from $\mathcal{P}_v$.

The follow-up work [GLL22] defined another family of algorithms $\mathbb{B}_k$ which takes as input a dataset $\mathcal{D} = \{s_i\}_{i\in[n]}$ and can (adaptively) query a sequence of k values $f(x;s)$ where $x \in \mathcal{X}$ and $s \in \mathcal{D}$. These algorithm families model the SCO and ERM problems stated in Problem 5.5.5, without the privacy requirement. In a slight abuse of notation, we denote the output of an algorithm $\mathcal{A} \in \mathbb{A}_k \cup \mathbb{B}_k$ in a SCO or ERM problem corresponding to a distribution $\mathcal{P}$ by $\mathcal{A}(\mathcal{P})$, where $\mathcal{A} \in \mathbb{B}_k$ also depends on the dataset received.

Both [DJWW15, GLL22] let v be drawn uniformly at random from $\mathcal{V} := \{-1, 1\}^d$ and let

$$\mathcal{P}_v := \mathcal{N}\left(\kappa v, \sigma^2 \mathbf{I}_d\right), \; f(x;s) := \langle s, x\rangle$$

for parameters κ, σ to be chosen. We fix this notation throughout this section. For any algorithm $\mathcal{A} \in \mathbb{A}_k \cup \mathbb{B}_k$ corresponding to a set $\mathcal{X}$ and a distribution $\mathcal{P}$, we define the optimality gap

$$\varepsilon_k(\mathcal{A}, \mathcal{X}, \mathcal{P}) := \mathbb{E}\left[\mathop{\mathbb{E}}_{s\sim\mathcal{P}} f(\mathcal{A}(\mathcal{P}); s)\right] - \min_{x\in\mathcal{X}} \mathop{\mathbb{E}}_{s\sim\mathcal{P}} f(x;s),$$

where the first outer expectation is over any randomness in $\mathcal{A}$, as well as in the samples used. We also define the minimax risk over a family of distributions P,

$$\varepsilon_k^\star(\mathbb{A}_k \cup \mathbb{B}_k, P, \mathcal{X}) := \inf_{\mathcal{A}\in\mathbb{A}_k\cup\mathbb{B}_k} \sup_{\mathcal{P}\in P} \varepsilon_k\left(\mathcal{A}, \mathcal{P}, \mathcal{X}\right).$$

For $p \in [1, 2]$, we let $P_{G,p}$ denote the family of distributions $\mathcal{P}$ over vectors s such that

$$\mathop{\mathbb{E}}_{s\sim\mathcal{P}} \|s\|_q^2 \le G^2, \text{ where } \frac{1}{p} + \frac{1}{q} = 1.$$

Our lower bounds in this section will be on $\varepsilon_k^\star(\mathbb{A}_k \cup \mathbb{B}_k, P_{G,p}, \mathcal{X})$, where $\mathcal{X}$ is a scaled ℓ_p ball. The family $P_{G,p}$ induces random linear functions $\langle s, \cdot\rangle$ with gradient s, and hence $\mathcal{P} \in P_{G,p}$ implies that the induced function $\mathbb{E}_{s\sim\mathcal{P}}\langle s, \cdot\rangle$ has a bounded-variance gradient oracle in the ℓ_p norm via queries to $\mathcal{P}$. We use the following facts from prior work in our proofs.

Lemma 5.7.1 (Section 5.1, [DJWW15])**.** *Let $\mathcal{X}$ be the ℓ_p ball of diameter D for $p \in [1, 2]$. For any $v \in \mathcal{V}$ and $x \in \mathcal{X}$, letting $x_v^\star := \min_{x\in\mathcal{X}} \mathbb{E}_{s\sim\mathcal{P}_v} f(x;s)$, and letting $\mathbb{1}(\text{sign}(a) =$*

$\operatorname{sign}(b)$) *be the* 0-1 *function which is* 1 *if and only if the signs of* a *and* b *agree,*

$$\mathop{\mathbb{E}}_{s\sim\mathcal{P}_v}[f(x;s)] - \mathop{\mathbb{E}}_{s\sim\mathcal{P}_v}[f(x_v^\star;s)] \geq \frac{(1-\frac{1}{p})\kappa D}{2d^{\frac{1}{p}}} \sum_{j\in[d]} \mathbb{1}\left(\operatorname{sign}(x_j) = \operatorname{sign}(v_j)\right).$$

Lemma 5.7.1 shows that it suffices to lower bound the expected Hamming distance between the signs of an estimate x and a randomly sampled $-v$. Such a lower bound was given in [DJWW15, GLL22] for estimates returned by $\mathcal{A} \in \mathbb{A}_k \cup \mathbb{B}_k$ via information-theoretic arguments.

Lemma 5.7.2 (Section 5.1, [DJWW15], Lemma 7.4, [GLL22])**.** *Let* $\mathcal{X}$ *be the* ℓ_p *ball of diameter* D*, and let* $\mathcal{A} \in \mathbb{A}_k \cup \mathbb{B}_k$ *be parameterized by* $\mathcal{X}$ *and* $\mathcal{P}_v$*. Then*

$$\mathop{\mathbb{E}}_{v\sim_{\text{unif.}}\mathcal{V}}\left[\sum_{j\in[d]} \mathbb{1}(\operatorname{sign}(\mathcal{A}(\mathcal{P}_v)_j) = \operatorname{sign}(v_j))\right] \geq \frac{d}{2}\left(1 - \frac{\kappa\sqrt{k}}{\sigma\sqrt{d}}\right).$$

To lower bound the oracle query complexity of our sampler we use the following standard result.

Lemma 5.7.3 ([DKL18], Corollary 1)**.** *Let* $\mathcal{X} \subset \mathbb{R}^d$ *be compact and convex,* $f : \mathcal{X} \to \mathbb{R}$ *be convex,* $k > 0$*, and* π *be the density over* $\mathcal{X}$ *proportional to* $\exp(-kf)$*. Then,*

$$\mathop{\mathbb{E}}_{x\sim\pi}[f(x)] - \min_{x\in\mathcal{X}} f(x) \leq \frac{d}{k}.$$

Lower bounds. We now state three lower bounds generalizing results from [DJWW15, GLL22]. Our results follow straightforwardly from Lemmas 5.7.1, 5.7.2, and 5.7.3 with appropriate parameters.

Proposition 5.7.4 (Minimax risk lower bound, $P_{G,p}$)**.** *Let* $G, D > 0$*, and let* $p \in [1, 2]$*,* $q \geq 2$ *satisfy* $\frac{1}{p} + \frac{1}{q} = 1$*. Let* $\mathcal{X}$ *be the* ℓ_p *ball of diameter* D*. Then,*

$$\varepsilon_k^\star\left(\mathbb{A}_k \cup \mathbb{B}_k, P_{G,p}, \mathcal{X}\right) = \Omega\left(GD\max\left(1 - \frac{1}{p}, \frac{1}{\log d}\right)\min\left(1, \sqrt{\frac{d}{k\log d}}\right)\right).$$

Proof. Throughout the proof, let $\kappa = \frac{\sigma\sqrt{d}}{2\sqrt{k}}$, and let

$$\sigma = \frac{Gd^{-\frac{1}{q}}}{\sqrt{\frac{d}{k} + 4\log d}}. \tag{5.32}$$

By well-known bounds on the expected maximum of d standard Gaussians, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{s\sim\mathcal{P}_v}\left[\|s\|_q^2\right] &\le 2\kappa^2\|v\|_q^2 + 2\mathop{\mathbb{E}}_{u\sim\mathcal{N}(0,\sigma^2\mathbf{I}_d)}\left[\|u\|_q^2\right] \\
&\le 2\kappa^2 d^{\frac{2}{q}} + 2d^{\frac{2}{q}}\mathop{\mathbb{E}}_{u\sim\mathcal{N}(0,\sigma^2\mathbf{I}_d)}\left[\|u\|_\infty^2\right] \\
&\le \sigma^2 d^{\frac{2}{q}}\left(\frac{d}{k} + 4\log d\right) \le G^2.
\end{aligned}$$

Hence, $\mathcal{P}_v \in P_{G,p}$ for all $v \in \mathcal{V}$, so it suffices to lower bound $\varepsilon_k(\mathcal{A}, \mathcal{P}_v, \mathcal{X})$. Combining Lemmas 5.7.1 and 5.7.2 with our choices of parameters,

$$\varepsilon_k(\mathcal{A}, \mathcal{P}_v, \mathcal{X}) \ge \frac{(1-\frac{1}{p})\kappa D d^{1-\frac{1}{p}}}{8} = \Omega\left(GD\left(1-\frac{1}{p}\right)\min\left(1, \sqrt{\frac{d}{k\log d}}\right)\right).$$

The conclusion then follows because for $p \le 1 + \frac{1}{\log d}$, choosing a larger value of p only affects problem parameters by constant factors by norm conversions. □

We give a slight extension of Proposition 5.7.4 for the family $\overline{P}_{G,p}$ of distributions over linear functions $\langle s, \cdot\rangle$, where s is required to satisfy $\|s\|_q \le G$ with probability 1, by simply truncating a draw from $\mathcal{P}_v$. This family is compatible with the setting in Problem 5.5.5.

Corollary 5.7.5 (Minimax risk lower bound, $\overline{P}_{G,p}$)**.** *In the setting of Proposition 5.7.4,*

$$\varepsilon_k^\star\left(\mathbb{A}_k \cup \mathbb{B}_k, \overline{P}_{G,p}, \mathcal{X}\right) = \Omega\left(GD\max\left(1-\frac{1}{p}, \frac{1}{\log d}\right)\min\left(1, \sqrt{\frac{d}{k\log(dk)}}\right)\right).$$

Proof. We define a distribution $\overline{\mathcal{P}}_v$ as follows: first $s \sim \mathcal{P}_v$, and then if $\|s\|_q \ge G$, we set $s \leftarrow 0$. By adjusting the logarithmic term in (5.32) to be $O(\log(dk))$, with probability at most $\mathrm{poly}((dk)^{-1})$, all k draws from $\mathcal{P}_v$ and $\overline{\mathcal{P}}_v$ used are identical by a union bound. Further, due to problem constraints the function error is always at most GD. So, the risk

is affected by at most $GD \cdot \mathrm{poly}((dk)^{-1})$. □

Corollary 5.7.5 shows that when β in Assumption 5.5.7 is polynomially bounded, the value oracle complexities used by Theorem 5.5.8 for both DP-SCO and DP-ERM are optimal up to logarithmic factors for the expected excess risk bounds they produce, even without the requirement of privacy. Finally, we show that the value oracle complexity of our sampler in Theorem 5.4.12 is also near-optimal.

Corollary 5.7.6. *In the setting of Proposition 5.7.4, let $r : \mathcal{X} \to \mathbb{R}$ be 1-strongly convex in $\|\cdot\|_p$ with additive range $O(D^2 \min(\log d, \frac{1}{p-1}))$. Let $\mathcal{I}$ be a distribution over i such that all $f_i : \mathcal{X} \to \mathbb{R}$ are G-Lipschitz in $\|\cdot\|_p$, and let $F := \mathbb{E}_{i\sim\mathcal{I}} f_i$. No algorithm using $o(\frac{G^2}{\mu} \log^{-4} d)$ value oracle queries to some f_i samples within total variation*

$$o\left(\min\left(\frac{1}{\log d}, \sqrt{\frac{d}{k \log^3(dk)}}\right)\right)$$

of the density proportional to $\exp(-F - \mu r(x))\mathbb{1}_{\mathcal{X}}(x)$.

Proof. Assume for contradiction that $\mathcal{A}$ is an algorithm satisfying the stated criterion using $k = o(\frac{G^2}{\mu} \log^{-4} d)$ value oracle queries, and let F be minimized by $x^\star \in \mathcal{X}$. We choose

$$\mu = \frac{d}{D^2 \min(\log d, \frac{1}{p-1})}.$$

Lemma 5.7.3 then shows that the sampled x satisfies

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\mathcal{A}}[F(x)] - F(x^\star) &\le \mu\left(r(x^\star) - r(x)\right) + d + GD \cdot o\left(\min\left(\frac{1}{\log d}, \sqrt{\frac{d}{k \log^3(dk)}}\right)\right) \\ &= O(d) + o\left(\frac{GD}{\log d} \min\left(1, \sqrt{\frac{d}{k \log(dk)}}\right)\right).\end{aligned}$$

For the given values of k and μ, this contradicts Corollary 5.7.5. □

Corollary 5.7.6 implies that for samplers with value query complexity depending polylogarithmically on the total variation distance, $\frac{G^2}{\mu}$ queries are required (up to polylogarithmic factors). This applies to the setting of our sampler in Theorem 5.4.12; we also note that

the LLT-based regularizers we use in our ℓ_p applications (Section 5.5.2) satisfy the additive range bound in Corollary 5.7.6.

5.8 *Lower bound on the range of* $\psi_{1,1}$

In this section, we provide a lower bound on the range of $\psi_{1,1}$ (5.25) which grows with the dimension d, demonstrating non-scale invariance of our family of LLTs. Recall that $\psi_{1,1}(x)$ is defined by

$$\psi_{1,1}(x) := \log\left(\int \exp\left(\langle x, y\rangle - \|y\|_\infty^2\right) \mathrm{d}y\right).$$

Lemma 5.8.1. *The additive range of* $\psi_{1,1}$ *over* $\{x \in \mathbb{R}^d \mid \|x\|_1 \le 1\}$ *is* $\Omega(\sqrt{d})$.

Proof. Throughout the proof denote for simplicity $\psi := \psi_{1,1}$ and let

$$\mathcal{D}_x^\varphi(y) \propto \exp\left(\langle x, y\rangle - \|y\|_\infty^2\right).$$

Then, following (5.26), we can write $\psi(x) - \psi(0)$ as

$$\psi(x) - \psi(0) = \log\left[\mathop{\mathbb{E}}_{y\sim\mathcal{D}_0^\varphi} \exp(\langle x, y\rangle)\right],$$

where $\mathcal{D}_0^\varphi \propto \exp(-\|y\|_\infty^2)$. Let π be the probability density on $\mathbb{R}_{\ge 0}$ such that

$$\mathrm{d}\pi(r) \propto r^{d-1}\exp(-r^2)\mathrm{d}r.$$

Here, $\mathrm{d}\pi(r)$ is the density of the scalar quantity $r = \|y\|_\infty$ for $y \sim \mathcal{D}_0^\varphi$. Note that the distribution of y conditioned on $\|y\|_\infty = r$ is uniform over the surface of the ℓ_∞ ball, where one random coordinate is set to $\pm r$, and the remaining coordinates are uniform on a $d-1$ dimensional hypercube with side length r. We denote this distribution as $\mathcal{P}_r$, and write

$$\begin{aligned}\mathop{\mathbb{E}}_{y\sim\mathcal{D}_0^\varphi} \exp(\langle x, y\rangle) &= \mathop{\mathbb{E}}_{r\sim\pi}\left[\mathop{\mathbb{E}}_{y\sim\mathcal{P}_r} \exp(\langle x, y\rangle)\right] \\ &= \mathop{\mathbb{E}}_{r\sim\pi}\left[\frac{1}{d}\sum_{i^\star\in[d]}\frac{1}{2}\sum_{y_{i^\star}\in\{-r,r\}}\exp(x_{i^\star}y_{i^\star})\prod_{i\neq i^\star}\int_{-r}^{r}\frac{1}{2r}\exp(x_i y_i)\mathrm{d}y_i\right].\end{aligned}$$

Let $x = e_1$ and $g_{i^\star}^{(r)} = \exp(x_{i^\star} r) \prod_{i \neq i^\star} \int_{-r}^{r} \frac{1}{2r} \exp(x_i y_i) \mathrm{d}y_i$. Then,

$$\mathop{\mathbb{E}}_{y \sim \mathcal{D}_0^\varphi} \exp(\langle x, y \rangle) \geq \frac{1}{2d} \sum_{i^\star \in [d]} \mathop{\mathbb{E}}_{r \sim \pi(r)} g_{i^\star}^{(r)}$$

since this drops terms where $y_{i^\star} = -r$. When $i^\star = 1$, we have $g_{i^\star}^{(r)} = \exp(r)$. When $i^\star \neq 1$, we have

$$g_{i^\star}^{(r)} = \int_{-r}^{r} \frac{1}{2r} \exp(y_1) \mathrm{d}y_1 = \frac{1}{2r} \left(\exp(r) - \exp(-r) \right).$$

Now, consider $r_1 = \sqrt{\frac{d-1}{2}}$. For any $r \leq r_1$, $\frac{\mathrm{d}}{\mathrm{d}r}[(d-1)\log r - r^2] = \frac{d-1}{r} - 2r \geq 0$. Thus, we have

$$I := \int_0^{\frac{1}{2} r_1} \exp((d-1)\log r - r^2) \mathrm{d}r \leq \int_{\frac{1}{2} r_1}^{r_1} \exp((d-1)\log r - r^2) \mathrm{d}r. \tag{5.33}$$

Letting $Z := \int_0^\infty \exp((d-1)\log r - r^2) \mathrm{d}r$, (5.33) shows that

$$\int_{\frac{1}{2} r_1}^{\infty} \exp((d-1)\log r - r^2) \mathrm{d}r = Z - I \geq Z - \frac{1}{2} Z = \frac{1}{2} Z.$$

Then, for all $i^\star \in [d]$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{r \sim \pi} g_{i^\star} &= \frac{\int_0^\infty \exp((d-1)\log r - r^2) g_{i^\star}^{(r)} \mathrm{d}r}{Z} \\
&\geq \frac{\int_{\frac{1}{2} r_1}^\infty \exp((d-1)\log r - r^2) g_{i^\star}^{(r)} \mathrm{d}r}{Z} \\
&\geq \frac{2 \int_{\frac{1}{2} r_1}^\infty \exp((d-1)\log r - r^2) g_{i^\star}^{(r)} \mathrm{d}r}{\int_{\frac{1}{2} r_1}^\infty \exp((d-1)\log r - r^2) \mathrm{d}r} \\
&\geq 2 \min_{r \geq r_1} \exp(r - \log(4r)) = 2 \exp(r_1 - \log(4 r_1)).
\end{aligned}$$

The fourth step follows from $g_{i^\star}^{(r)} \geq \frac{1}{4r} \exp(r)$ for $r \geq r_1$. The last step follows from $r - \log 4r$

increases on $r \geq r_1$. Combining with $\mathbb{E}_{y\sim\mathcal{P}_0} \exp(\langle x, y\rangle) \geq \frac{1}{2d}\sum_{i^\star\in[d]} \mathbb{E}_{r\sim\pi(r)}\, g_{i^\star}$,

$$\psi(x) - \psi(0) = \log \mathop{\mathbb{E}}_{y\sim\mathcal{P}_0} \exp(\langle x, y\rangle) \geq \log\left(\frac{d-1}{d}\exp(r_1 - \log(4r_1))\right) = \Omega(\sqrt{d}).$$

□

5.9 *Deferred proofs from Section 5.4*

Lemma 5.4.5. *For λ defined in (5.23),*

$$\mathbb{E}\left[|\lambda|\mathbb{1}_{\rho\notin[0,2]}\right] \leq \frac{\delta}{4}.$$

Proof. Clearly, it suffices to show $\mathbb{E}\,|\lambda| \leq \frac{\delta}{4}$. Define random variables,

$$\Delta_i := |f_i(x_2) - f_i(x_1)|,\ \Delta := \mathop{\mathbb{E}}_{i\sim\mathcal{I}} \Delta_i,$$

whose randomness comes from $x_1, x_2 \sim \gamma_y$. By definition,

$$\mathbb{E}\,|\lambda| = \sum_{b>H} \frac{1}{b!} \mathop{\mathbb{E}}_{x_1,x_2\sim\gamma} [\Delta]^B.$$

Define $\Phi(t) := \sum_{b>H} \frac{t^b}{b!}$. For $H = \lceil 10 \log \frac{1}{\delta}\rceil$, it is straightforward to check $\Phi(t) \leq \frac{\delta}{16}$ for any $|t| \leq 1$, and for all nonnegative t, $\Phi(t) \leq \exp(t)$. Hence, letting p_Δ be the density of Δ,

$$\begin{aligned}\mathbb{E}\,|\lambda| &\leq \frac{\delta}{16} + \mathbb{E}[\mathbb{1}_{\Delta>1} e^{\Delta}] \leq \frac{\delta}{16} + \int_1^\infty \exp\left(\lceil\Delta\rceil\right) p_\Delta(\Delta)\mathrm{d}\Delta \\ &\leq \frac{\delta}{16} + \sum_{k\geq 1} \exp(k+1) \Pr_{x_1,x_2\sim\gamma}[\Delta \geq k].\end{aligned} \tag{5.34}$$

It now suffices to bound on $\Pr[\Delta \geq k]$. Define a function $h_{x_1,x_2}(k) := \Pr_{i\sim\mathcal{I}}[|f_i(x_1) - f_i(x_2)| \geq k]$. Since each f_i is G-Lipschitz, and γ_y is $\frac{1}{12\eta}$-strongly logconcave in by Lemma 5.3.3, by Lemma 5.4.3:

$$\mathop{\mathbb{E}}_{x_1,x_2}[h_{x_1,x_2}(k)] = \Pr_{x_1,x_2,i\sim\mathcal{I}}[|f_i(x_1) - f_i(x_2)| \geq k] \leq 4\exp\left(-\frac{k^2}{96\eta G^2}\right),$$

and so by Markov's inequality we have

$$\Pr_{x_1,x_2}[h_{x_1,x_2}(k) \geq e^{-t}] \leq 4\exp\left(t - \frac{k^2}{96\eta G^2}\right). \tag{5.35}$$

For fixed x_1, x_2, as each f_i is G-Lipschitz in $\|\cdot\|_{\mathcal{X}}$, $|f_i(x_1) - f_i(x_2)| \leq G\|x_1 - x_2\|_{\mathcal{X}}$, and hence

$$\mathop{\mathbb{E}}_{i\sim\mathcal{I}}[|f_i(x_1) - f_i(x_2)|] \leq \min_{k\geq 0} k + h_{x_1,x_2}(k)\cdot G\|x_1 - x_2\|_{\mathcal{X}}.$$

This then shows that if for some k, $h_{x_1,x_2}(k) \leq \exp(-\frac{k^2}{192\eta G^2})$,

$$\mathop{\mathbb{E}}_{i\sim\mathcal{I}}[|f_i(x_1) - f_i(x_2)|] \leq k + \exp\left(-\frac{k^2}{192\eta G^2}\right)\cdot G\|x_1 - x_2\|_{\mathcal{X}},$$

which implies via (5.35) that

$$\begin{aligned}&\Pr_{x_1,x_2}\left[\Delta \geq k + \exp\left(-\frac{k^2}{192\eta G^2}\right)\cdot G\|x_1 - x_2\|_{\mathcal{X}}\right]\\ \leq\ &\Pr_{x_1,x_2}\left[h_{x_1,x_2}(k) \geq \exp\left(-\frac{k^2}{192\eta G^2}\right)\right] \leq 4\exp\left(-\frac{k^2}{192\eta G^2}\right).\end{aligned} \tag{5.36}$$

Further, since $\|x_1 - \mathbb{E}\,x_1\|_{\mathcal{X}}$ is a 1-Lipschitz function in x_1 with a nonnegative mean, by Lemma 5.4.3,

$$\Pr[\|x_1 - x_2\|_{\mathcal{X}} \geq k] \leq 2\Pr[\|x_1 - \mathbb{E}\,x_1\|_{\mathcal{X}} \geq k] \leq 2\exp\left(-\frac{k^2}{96\eta G^2}\right). \tag{5.37}$$

Combining (5.36) and (5.37),

$$\begin{aligned}\Pr_{x_1,x_2}[\Delta \geq 2k] &= \Pr_{x_1,x_2}\left[\Delta \geq 2k \wedge \|x_1 - x_2\|_{\mathcal{X}} \geq \frac{k}{G}\right] + \Pr_{x_1,x_2}\left[\Delta \geq 2k \wedge \|x_1 - x_2\|_{\mathcal{X}} \leq \frac{k}{G}\right]\\ &\leq 2\exp\left(-\frac{k^2}{96\eta G^2}\right) + \Pr_{x_1,x_2}\left[\Delta \geq k + \exp\left(-\frac{k^2}{192\eta G^2}\right)G\|x_1 - x_2\|_{\mathcal{X}}\right]\\ &\leq 6\exp\left(-\frac{k^2}{192\eta G^2}\right).\end{aligned} \tag{5.38}$$

Plugging (5.38) into (5.34), and using $\eta^{-1} \geq 10^4 G^2 \log \frac{1}{\delta}$, we have the desired

$$\mathbb{E}(|\lambda| \mathbb{1}_{\rho \notin [0,2]}) \leq \frac{\delta}{16} + \sum_{k=1}^{\infty} 6 \exp\left(k - \frac{k^2}{768 \eta G^2}\right) \leq \frac{\delta}{4}.$$

□

Lemma 5.4.6. *For σ defined in (5.23),*

$$\mathbb{E}\left[|\sigma| \mathbb{1}_{\rho \notin [0,2]}\right] \leq \frac{\delta}{4}.$$

Proof. We begin by bounding, analogously to (5.34),

$$\mathbb{E}[|\sigma| \mathbb{1}_{\rho \notin [0,2]}] \leq 2^H \Pr[\rho \notin [0,2]] + \sum_{k \geq 1} \Pr\left[|\sigma| > 2^{kH}\right] 2^{(k+1)H}. \tag{5.39}$$

Recall when $a \leq H$, $|\mathcal{J}| \leq \frac{1}{2} H^2$. By a union bound over Lemma 5.4.3,

$$\Pr_{x_1, x_2}\left[|f_i(x_1) - f_i(x_2)| \geq \frac{2^k}{3} \; \forall i \in \mathcal{J}\right] \leq H^2 \exp\left(-\frac{4^k}{864 \eta G^2}\right).$$

If for each $i \in \mathcal{J}$, $|f_i(x_1) - f_i(x_2)| \leq \frac{2^k}{3}$, we have for $k \geq 1$

$$|\sigma| = \sum_{b=0}^{H} \mathbb{1}_{a \geq b} \prod_{i \in [b]} (f_{j_{i,b}}(x_2) - f_{j_{i,b}}(x_1)) \leq 1 + \sum_{b=1}^{H} \left(\frac{2^k}{3}\right)^b \leq 2^{kH},$$

which implies that $\Pr[|\sigma| \geq 2^{kH}] \leq H^2 \exp(-\frac{4^k}{864 \eta G^2})$ and hence using our choice of $\eta \leq \frac{1}{500 G^2 H}$,

$$\begin{aligned} \sum_{k=1}^{\infty} 2^{(k+1)H} \Pr\left[|\sigma| > 2^{kH}\right] &\leq \sum_{k=1}^{\infty} 2^{(k+1)H} H^2 \exp\left(-\frac{4^k}{864 \eta G^2}\right) \\ &\leq \sum_{k=1}^{\infty} 2^{4kH} \exp(-2 \cdot 4^k H) \leq \sum_{k=1}^{\infty} 2^{-kH} \leq \frac{\delta}{8}. \end{aligned} \tag{5.40}$$

It remains to bound $\Pr[\rho \notin [0,2]]$. Recall $\Pr[a > H] \leq \frac{1}{H!}$ so since $a \leq H \implies \sigma = \rho$, $\Pr[\rho \notin [0,2]] \leq \frac{1}{H!} + \Pr[\sigma \notin [0,2]]$. Next, by a union bound over Lemma 5.4.3 and $\frac{1}{2} H^2$

indices in $\mathcal{J}$,

$$\Pr_{x_1,x_2}\left[|f_i(x_1) - f_i(x_2)| \geq \frac{1}{2}\ \forall i \in \mathcal{I}\right] \leq 2H^2 \exp\left(-\frac{1}{384\eta G^2}\right).$$

Under the event that $|f_i(x_1) - f_i(x_2)| \leq \frac{1}{2}$ for all $i \in \mathcal{I}$, $0 \leq \sigma \leq 2$ by definition. Hence we know $\Pr[\sigma \notin [0,2]] \leq 2H^2 \exp(-\frac{1}{384\eta G^2})$ and by our setting that $H > 10 \log \frac{1}{\delta}$, we have

$$\Pr[\rho \notin [0,2]] \cdot 2^H \leq 2^H \left(2H^2 \exp\left(-\frac{1}{384\eta G^2}\right) + \frac{1}{H!}\right) \leq \frac{\delta}{8}. \tag{5.41}$$

Combining (5.39), (5.40) and (5.41) completes the proof. □

Part III

NON CONVEX OPTIMIZATION

Chapter 6

PRIVATE (STOCHASTIC) NON-CONVEX OPTIMIZATION REVISITED: SECOND-ORDER STATIONARY POINTS AND EXCESS RISKS

6.1 Introduction

Differential privacy [DMNS06] is a standard privacy guarantee for training machine learning models. Given a randomized algorithm $\mathcal{A} : P^* \to R$, where P is a data domain and R is a range of outputs, we say $\mathcal{A}$ is (ε, δ)-differentially private (DP) for some $\varepsilon \geq 0$ and $\delta \in [0, 1]$ if for any neighboring datasets $\mathcal{D}, \mathcal{D}' \in P^*$ that differ in at most one element and any $\mathcal{R} \subseteq R$, the distribution of the outcome of the algorithm, e.g., pair of models trained on the respective datasets, are similar:

$$\Pr_{x \sim \mathcal{A}(\mathcal{D})}[x \in \mathcal{R}] \leq e^{\varepsilon} \Pr_{x \sim \mathcal{A}(\mathcal{D}')}[x \in \mathcal{R}] + \delta.$$

Smaller ε and δ imply the distributions are closer; hence, an adversary accessing the trained model cannot tell with high confidence whether an example x was in the training dateset. Given this measure of privacy, we consider the problem of optimizing a non-convex loss while ensuring a desired level of privacy. In particular, suppose we are given a dataset $\mathcal{D} = \{z_1, \ldots, z_n\}$ drawn i.i.d. from underlying distribution $\mathcal{P}$. Each loss function $f(\cdot; z) : \mathcal{K} \to \mathbb{R}$ is G-Lipschitz over the convex set $\mathcal{K} \subset \mathbb{R}^d$ of diameter D. Let the population risk function be $F_{\mathcal{P}}(x) := \mathbb{E}_{z \sim \mathcal{P}}[f(x; z)]$ and the empirical risk function be $F_{\mathcal{D}}(x) := \frac{1}{n} \sum_{z \in \mathcal{D}} f(x; z)$. We also denote $F_S(x) := \frac{1}{|S|} \sum_{z \in S} f(x; z)$ for $S \subseteq \mathcal{D}$.

Our focus is in minimizing non-convex risk functions, both empirical and population, which may have multiple local minima. Since finding the global optimum of a non-convex function can be challenging, an alternative goal in the field is to find stationary points: A first-order stationary point is a point with a small gradient of the function, and a second-order stationary point is a first-order stationary point where additionally the function has

	α-SOSP		Excess population risk	
	empirical	population	poly-time	exp-time
SOTA	$\min(\frac{d^{\frac{1}{4}}}{n^{\frac{1}{2}}}, \frac{d^{\frac{4}{7}}}{n^{\frac{4}{7}}})$	N/A	$\frac{d}{\varepsilon^2 \log n}$ ♠	N/A
Ours	$\frac{d^{\frac{1}{3}}}{n^{\frac{2}{3}}}$	$\frac{1}{n^{\frac{1}{3}}} + \left(\frac{\sqrt{d}}{n}\right)^{\frac{3}{7}}$	$\frac{d \log\log n}{\varepsilon \log(n)}$	$\frac{d}{n\varepsilon} + \sqrt{\frac{d}{n}}$
LB	$\frac{\sqrt{d}}{n}$	$\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n}$	$\frac{d}{n\varepsilon} + \sqrt{\frac{d}{n}}$	$\frac{d}{n\varepsilon} + \sqrt{\frac{d}{n}}$

Table 6.1: SOTA refers to the best previously known bounds on α for α-SOSP by [WCX19] and on the excess population risk by [WCX19]. We introduce algorithm 18 that finds an α-SOSP (columns 2–3) with an improved rate. We show exponential mechanism can minimize the excess risk in polynomial time and exponential time, respectively (columns 4 and 5). ♠ requires extra assumption on bounded smoothness. The lower bounds for SOSP are from [ABG+23], and the lower bound on excess population risk is from Theorem 6.4.11. We omit logarithmic factors in n and d.

a positive or nearly positive semi-definite Hessian. As first order stationary points can be saddle points or even a local maximum, we focus on the problem of finding a second order stationary point, i.e., a local minimum, privately. Existing works in finding approximate SOSP privately only give guarantees for the empirical function $F_{\mathcal{D}}$. We improve upon the state-of-the-art result for empirical risk minimization and give the first guarantee for the population function $F_{\mathcal{P}}$. This requires standard assumptions on bounded Lipschitzness, smoothness, and Hessian Lipschitzness, which we make precise in Section 6.2 and in Assumption 6.3.1.

Compared to finding a local minimum, finding a global minimum can be extremely challenging. Progress towards finding the global minima is measured in the excess empirical risk, $\mathbb{E}[F_{\mathcal{D}}(x^{priv})] - \min_{x \in \mathcal{K}} F_{\mathcal{D}}(x)$, and the excess population risk, $\mathbb{E}[F_{\mathcal{P}}(x^{priv})] - \min_{x \in \mathcal{K}} F_{\mathcal{P}}(x)$ for a private solution x^{priv}. We provide two approaches, in polynomial time and exponential time, that improve upon the state-of-the-art guarantees as measured in the excess risks for the respective families of computational complexity.

6.1.1 Main results

Our main contribution is a private non-convex optimization algorithm based on the variance-reduced SpiderBoost [WJZ+19]; Algorithm 18 achieves improved rates on the approximation error for finding SOSP of the empirical and population risks privately. Table 6.1 summarizes our main results.

Finding second-order stationary points. Advances in private non-convex optimization have focused on finding a first-order stationary point (FOSP), whose performance is measured in (i) the norm of the empirical gradient at the solution x, i.e., $\|\nabla F_{\mathcal{D}}(x)\|$, and (ii) the norm of the population gradient, i.e., $\|\nabla F_{\mathcal{P}}(x)\|$.

Definition 6.1.1 (First-order stationary point)**.** We say $x \in \mathbb{R}^d$ is a First-Order Stationary Point (FOSP) of $g : \mathbb{R}^d \to \mathbb{R}$ iff $\nabla g(x) = 0$. x is an α-FOSP of g, if $\|\nabla g(x)\|_2 \leq \alpha$.

Since FOSP can be a saddle point or a local maxima, finding a second-order stationary point is desired. Exact second-order stationary points can be extremely challenging to find [GHJY15]. Instead, progress is commonly measured in terms of how well the solution approximates an SOSP.

Definition 6.1.2 (Second-order stationary point, [AAZB+17])**.** We say a point $x \in \mathbb{R}^d$ is a Second-Order Stationary Point (SOSP) of a twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ iff $\|\nabla g(x)\|_2 = 0$ and $\nabla^2 g(x) \succeq 0$. We say $x \in \mathbb{R}^d$ is an α-SOSP for ρ-Hessian Lipschitz function g, if $\|\nabla g(x)\|_2 \leq \alpha \bigwedge \nabla^2 g(x) \succeq -\sqrt{\rho\alpha} I$.

On the empirical risk $F_{\mathcal{D}}$, the SOTA on privately finding α-SOSP is by [WCX19, WX20], which achieves $\alpha = \tilde{O}(\min\{(\sqrt{d}/n)^{1/2}, (d/n)^{4/7}\})$. In Theorem 6.3.9, we show that the proposed Algorithm 18 achieves a rate bounded by $\alpha = \tilde{O}((\sqrt{d}/n)^{2/3})$, which improves over the SOTA in all regime.[1] There remains a factor $(\sqrt{d}/n)^{-1/6}$ gap to a known lower bound of $\alpha = \Omega(\sqrt{d}/n)$ that holds even if privacy is not required and even if finding only an α-FOSP [ABG+23]. On the population risk $F_{\mathcal{P}}$, Algorithm 18 is the first private algorithm to guarantee finding an α-SOSP with $\alpha = \tilde{O}(n^{-1/3} + (\sqrt{d}/n)^{3/7})$ in Theorem 6.3.12. There

[1] We want $\alpha = o(1)$ and hence can assume $d \leq n^2$.

is a gap to a known lower bound of $\alpha = \Omega(1/\sqrt{n} + \sqrt{d}/n)$ that holds even if privacy is not required and even if finding only an α-FOSP [ABG+23].

Minimizing excess risk. We also provide sampling-based algorithms that aims to tackle the ultimate objective of finding a private solution $x^{priv} \in \mathbb{R}^d$ that minimizes the excess EMPIRICAL RISK: $\mathbb{E}[F_{\mathcal{D}}(x^{priv})] - \min_{x\in\mathcal{K}} F_{\mathcal{D}}(x)$, and the excess POPULATION RISK, $\mathbb{E}[F_{\mathcal{P}}(x^{priv})] - \min_{x\in\mathcal{K}} F_{\mathcal{P}}(x)$, where the expectation is over the randomness on the solution x^{priv}. With a mild smoothness assumption, [WCX19] achieves in polynomial time a bound of $O(d\sqrt{\log(1/\delta)}/(\varepsilon^2 \log n))$ for both excess empirical and population risks. In Table 6.1 we omit excess empirical risk, as the bounds are the same. We introduce a sampling-based algorithm from the exponential mechanism, which runs in polynomial time and achieves excess empirical and population risks bounded by $O(d\sqrt{\log(1/\delta)}/(\varepsilon \log(nd)))$ with improved dependence on ε (Theorem 6.4.6). Moreover, we do not need the smoothness assumption required by [WCX19].

If we allow an exponential running time, [GTU22] demonstrated $\tilde{O}(d/(\varepsilon n))$ upper bound for non-convex excess empirical risks along with a nearly matching lower bound. It remained an open question to obtain a tight bound for the excess population risk. We close this gap by providing a nearly matching upper and lower bounds of $\tilde{\Theta}(d/(\varepsilon n) + \sqrt{d/n})$ for the excess population risk (Theorem 6.4.8).

6.1.2 Our techniques

Stationary points. We propose a simple framework based on SpiderBoost [WJZ+19] and its private version [ABG+23] that achieves the current best rate for finding the first order stationary point privately. In SGD and its variants, we usually get an estimation Δ_t of the gradient $\nabla f(x_t)$. In the stochastic variance-reduced algorithm SpiderBoost, it only queries the gradient $\mathcal{O}_1(x_t) \approx \nabla f(x_t)$ directly every q steps with some oracle $\mathcal{O}_1$, and for the other $q-1$ steps in each period, it queries the difference between two steps, that is $\mathcal{O}_2(x_t, x_{t-1}) \approx \nabla f(x_t) - \nabla f(x_{t-1})$, and maintain $\Delta_t = \Delta_{t-1} + \mathcal{O}_2(x_t, x_{t-1})$. One interpretation of the difference between these two kinds of oracles is that, in many situations, one can treat $\mathcal{O}_1$ as more accurate and more costly (e.g., in computation or privacy budget),

though our framework does not necessarily assume this.

As SpiderBoost queries $\mathcal{O}_1$ every q steps, the error on the estimation may accumulate and $\|\Delta_t - \nabla f(x_t)\|$ can be large. Though on average, as shown in [ABG+23], these estimations can be good enough to find a private first-order stationary point, such a large deviation makes it challenging to analyze the behavior near a saddle point and to provide a tight analysis of the population risk.

In our framework, rather than using $\mathcal{O}_1$ once every q steps, we introduce a new technique of keeping track of the total drift we make, i.e., $\text{drift}_t = \sum_{i=\tau_t}^{t} \|x_i - x_{i-1}\|_2^2$, where τ_t is the last time stamp when we used $\mathcal{O}_1$. As we are considering smooth functions, the worst error to estimate $\nabla f(x_t) - \nabla f(x_{t-1})$ is proportional to $\|x_t - x_{t-1}\|_2$. When the drift_t is small, we know the current estimation should still be good enough, and we do not need to get an expensive fresh estimation from $\mathcal{O}_1$. When drift_t is large, the gradient estimation error may be large and we query $\mathcal{O}_1$ and get $\Delta_t = \mathcal{O}_1(x_t)$. To control the total cost, we need an appropriate threshold to determine when the drift is large. The smaller the threshold is, we can guarantee more accurate estimations but may need to pay more cost for querying $\mathcal{O}_1$ more frequently.

We want to bound the total occurrences of the event that drift_t is large, which leads to querying $\mathcal{O}_1$. A crucial observation is that, if drift_t increases quickly, then the gradient norms are large and hence function values decrease quickly, which we know does not happen frequently under the standard assumption that the function is bounded.

In our framework, we assume $\mathcal{O}_1(x)$ is an unbiased estimation of $\nabla f(x)$, and $\mathcal{O}_1(x) - \nabla f(x)$ is Norm-SubGaussian (Definition 6.2.3), and similarly $\mathcal{O}_2(x, y)$ is an unbiased estimation of $\nabla f(x) - \nabla f(y)$ whose error is also Norm-SubGaussian. In the empirical case, we can simply add Gaussian noises with appropriately chosen variances to the gradients of the empirical function $\nabla F_{\mathcal{D}}$ for simplicity, and one can choose a smaller batch size to reduce the computational complexity. In the population case, we draw samples from the dataset without replacement to avoid dependence issues, and add the Gaussian noises to the sampled gradients. Hence we only need the gradient oracle complexity to be linear in the number of samples for the population case.

Minimizing excess risk. Our polynomial time approach relies on the Log-Sobolev Inequality (LSI) and the classic Stroock perturbation lemma. The previous work of [MASN16] shows that if the density $\exp\big(-\beta F_{\mathcal{D}}(x) - r(x)\big)$ satisfies the LSI for some regularizer r, then sampling a model x from this density satisfies differential privacy with an appropriate (ε, δ). If r is a μ strongly convex function, then the density proportional to $\exp(-r)$ satisfies LSI with constant $1/\mu$, and $\exp(-\beta F_{\mathcal{D}}(x) - r(x))$ satisfies LSI with constant $\exp(\max_{x,y} |F_{\mathcal{D}}(x) - F_{\mathcal{D}}(y)|)/\mu$ by the Stroock perturbation lemma. Our bound on the empirical risk follows from choosing the appropriate inverse temperature β and regularizer r to satisfy (ε, δ)-DP. The final bound on the population risk also follows from LSI, which bounds the stability of the sample drawn from the respective distribution.

When running time is not concerned, we apply an exponential mechanism over a discretization of $\mathcal{K}$ to get the upper bound. The empirical risk bound follows from [BST14], and we use concentration of sums of bounded random variables to bound the maximum difference over the discretizations between the empirical and population risk. We show this is nearly tight by reductions from selection to non-convex Lipschitz optimization of [GTU22].

6.1.3 Further related work

In the convex setting, it is feasible to achieve efficient algorithms with good risk guarantees. In turn, differentially private empirical risk minimization (DP-ERM) [CM08, CMS11, CYS21, INS+19, KST12, BST14, TTZ15, SCS13, SSTT21] and differentially private stochastic optimization [ALD21, BFTGT19, BFGT20, FKT20, KLL21, AFKT21, KLZ22, GLL22, GTU22, CJJ+23, GLL+23] have been two of the most extensively studied problems in the DP literature. Most common approaches are variants of DP-SGD [CMS11] or the exponential mechanism [MT07].

As for the non-convex optimization, due to the intrinsic challenges in minimizing general non-convex functions, most of the previous works [WYX17, WJEG19, WX19, WCX19, ZCH+20, SSTT21, TC22, YZCL22, ABG+23, WB23, GW23] adopted the gradient norm as the accuracy metric rather than risk. Instead of minimizing the gradient norm discussed before, [BGM21] tried to minimize the stationarity gap of the population function privately,

which is defined as $\mathrm{Gap}_{F_{\mathcal{P}}}(x) := \max_{y\in\mathcal{K}}\langle \nabla F_{\mathcal{P}}(x), x-y\rangle$, which requires $\mathcal{K}$ to be a bounded domain. There are also some different definitions of the second order stationary point. We refer the readers to [LRY+20] for more details.

The risk bound achieved by algorithms with polynomial running time is weak and requires $n \gg d$ to be meaningful. Many previous works consider minimizing risks of non-convex functions under stronger assumptions, such as, Polyak-Lojasiewicz condition [WYX17, ZMLX21], Generalized linear model (GLM) [WCX19] and weakly convex functions [BGM21].

In the (non-private) classic stochastic optimization, there is a long line of influential works on finding the first and second-order stationary points for non-convex functions, [AAZB+17, JGN+17, FLLZ18, XJY18, CO19].

6.2 Preliminary

Throughout the paper, if not stated explicitly, the norm $\|\cdot\|$ means the ℓ_2 norm.

Definition 6.2.1 (Lipschitz and Smoothness)**.** Given a function $f : \mathcal{K} \to \mathbb{R}$, we say f is G-Lipschitz, if for all $x_1, x_2 \in \mathcal{K}$, $|f(x_1) - f(x_2)| \le G\|x_1 - x_2\|$, and we say a function $f : \mathcal{K} \to \mathbb{R}$ is M-smooth, if for all $x_1, x_2 \in \mathcal{K}$, $\|\nabla f(x_1) - \nabla f(x_2)\| \le M\|x_1 - x_2\|$.

Definition 6.2.2. We say a twice-differentiable function $f : \mathcal{K} \to \mathbb{R}$ is ρ-Hessian Lipschitz if for all $x_1, x_2 \in \mathcal{K}$, $\|\nabla^2 f(x_1) - \nabla^2 f(x_2)\|_2 \le \rho\|x_1 - x_2\|_2$.

Definition 6.2.3 (SubGaussian, and Norm-SubGaussian)**.** A random vector $x \in \mathbb{R}^d$ is SubGaussian (SG(ζ)) if there exists a positive constant ζ such that $\mathbb{E}\, e^{\langle v, x-\mathbb{E}\,x\rangle} \le e^{\|v\|^2\zeta^2/2}$, $\forall v \in \mathbb{R}^d$. $x \in \mathbb{R}^d$ is norm-SubGaussian (nSG(ζ)) if there exists ζ such that $\Pr[\|x - \mathbb{E}\,x\| \ge t] \le 2e^{-\frac{t^2}{2\zeta^2}}, \forall t \in \mathbb{R}$.

Fact 6.2.4. *For a Gaussian $\theta \sim \mathcal{N}(0, \sigma^2 I_d)$, θ is* SG(σ) *and* nSG($\sigma\sqrt{d}$).

Lemma 6.2.5 (Hoeffding type inequality for norm-subGaussian, [JNG+19])**.** *Let $x_1, \cdots, x_k \in \mathbb{R}^d$ be random vectors, and for each $i \in [k]$, $x_i \mid \mathcal{F}_{i-1}$ is zero-mean* nSG(ζ_i) *where $\mathcal{F}_i$ is the corresponding filtration. Then there exists an absolute constant c such that for any $\delta > 0$,*

with probability at least $1-\omega$, $\|\sum_{i=1}^k x_i\| \le c\cdot\sqrt{\sum_{i=1}^k \zeta_i^2 \log(2d/\omega)}$, *which means* $\sum_{i=1}^k x_i$ *is* $\mathrm{nSG}(\sqrt{c\log(d)\sum_{i=1}^k \zeta_i^2})$.

Definition 6.2.6 (Laplace distribution)**.** We say $X \sim \mathrm{Lap}(b)$ if X has density $f(X=x) = \frac{1}{2b}\exp(\frac{-|x|}{b})$.

Theorem 6.2.7 (Matrix Bernstein inequality, [Tro15])**.** *Consider a sequence* $\{X_i\}_{i\in m}$ *of independent, mean-zero, symmetric* $d\times d$ *random matrices. If for each matrix* X_i*, we know* $\|X_i\|_{op} \le M$*, then for all* $t \ge 0$*, we have* $\Pr\left[\|\sum_{i\in[m]} X_i\|_{op} \ge t\right] \le d\exp\left(\frac{-t^2}{2(\sigma^2+Mt/3)}\right)$*, where* $\sigma^2 = \|\sum_{i\in[m]} \mathbb{E}\, X_i^2\|_{op}$.

Theorem 6.2.8 (Basic composition, [DR14])**.** *If* $\mathcal{A}_1$ *is* (ε_1,δ_1)*-DP and* $\mathcal{A}_2$ *is* (ε_2,δ_2)*-DP, then their combination is* $(\varepsilon_1+\varepsilon_2, \delta_1+\delta_2)$*-DP.*

Theorem 6.2.9 (Advanced composition, [KOV15])**.** *For* $\varepsilon \le 0.9$*, an end-to-end guarantee of* (ε,δ)*-differential privacy is satisfied if a database is accessed at most* k *times, where each time with a* $(\varepsilon/(2\sqrt{2k\log(2/\delta)}), \delta/(2k))$*-differentially private mechanism.*

Due to space limit, some proofs are left in the Appendix.

6.3 Convergence to Stationary points

We follow the assumptions of [WCX19], which also studies privately finding an α-SOSP.

Assumption 6.3.1. *Any function drawn from* $\mathcal{P}$ *is* G*-Lipschitz,* ρ*-Hessian Lipschitz, and* M*-smooth, almost surely, and the risk is upper bounded by* B.

As discussed before, we define two different kinds of gradient oracles, one for estimating the gradient at one point and the other for estimating the gradient difference at two points.

Definition 6.3.2 (SubGaussian gradient oracles)**.** For a G-Lipschitz and M-smooth function F:

(1) We say $\mathcal{O}_1$ is a first kind of ζ_1 norm-subGaussian Gradient oracle if given $x\in\mathbb{R}^d$, $\mathcal{O}(x)$ satisfies $\mathbb{E}\,\mathcal{O}_1(x) = \nabla F(x)$ and $\mathcal{O}_1(x) - \nabla F(x)$ is $\mathrm{nSG}(\zeta_1)$.

(2) We say $\mathcal{O}_2$ is a second kind of ζ_2 norm-subGaussian stochastic Gradient oracle if given $x, y \in \mathbb{R}^d$, $\mathcal{O}_2(x, y)$ satisfies that $\mathbb{E}\,\mathcal{O}_2(x, y) = \nabla F(x) - \nabla F(y)$ and $\mathcal{O}_2(x, y) - (\nabla F(x) - \nabla F(y))$ is $\mathrm{nSG}(\zeta_2\|x - y\|)$.

Note that we should assume $M \geq \sqrt{\rho\alpha}$ to make finding a second-order stationary point strictly more challenging than finding a first-order stationary point. We use $\mathrm{smin}(\cdot)$ to denote the smallest eigenvalue of a matrix.

6.3.1 Meta framework

Algorithm 17: Stochastic Spider

Input: Objective function F, Gradient Oracle $\mathcal{O}_1, \mathcal{O}_2$ with SubGaussian parameters ζ_1 and ζ_2, parameters of objective function B, M, G, ρ, parameter κ, failure probability ω;
Set $\gamma = \sqrt{4C(\zeta_2^2\kappa + 4\zeta_1^2) \cdot \log(BMd/\rho\omega)}, \Gamma = \frac{M \log(\frac{dMB}{\rho\gamma\omega})}{\sqrt{\rho\gamma}}$;
Set $\eta = 1/M, t = 0, T = BM \log^4(\frac{dMB}{\rho\gamma\omega})/\gamma^2$;
Set $\mathrm{drift}_0 = \kappa, \mathrm{frozen} = 1, \nabla_{-1} = 0$;
while $t \leq T$ **do**
  if $\|\nabla_{t-1}\| \leq \gamma \log^3(BMd/\rho\omega) \bigwedge \mathrm{frozen}_{t-1} \leq 0$ **then**
    $\mathrm{frozen}_t = \Gamma, \mathrm{drift}_t = 0$;
    $\nabla_t = \mathcal{O}_1(x_t) + g_t$, where $g_t \sim \mathcal{N}(0, \frac{\zeta_1^2}{d} I_d)$;
  end
  else if $\mathrm{drift}_{t-1} \geq \kappa$ **then**
    $\nabla_t = \mathcal{O}_1(x_t)$, $\mathrm{drift}_t = 0$, $\mathrm{frozen}_t = \mathrm{frozen}_{t-1} - 1$;
  end
  else
    $\Delta_t = \mathcal{O}_2(x_t, x_{t-1})$, $\nabla_t = \nabla_{t-1} + \Delta_t$, $\mathrm{frozen}_t = \mathrm{frozen}_{t-1} - 1$;
  end
  $x_{t+1} = x_t - \eta\nabla_t, \mathrm{drift}_t = \mathrm{drift}_{t-1} + \eta^2\|\nabla_t\|_2^2$, $t = t + 1$;
end
Return: $\{x_1, \cdots, x_T\}$;

We demonstrate a framework based on the SpiderBoost in Algorithm 18. Our analysis of Algorithm 18 builds upon three key properties we prove in this section: (i) ∇_t is consistently close to the true gradient $\nabla F(x_t)$ with high probability; (ii) the algorithm can escape the saddle point with high probability, and (iii) a large drift implies significant decrease in the

function value, allowing us to limit the number of queries to the more accurate but more expensive first kind of gradient oracle $\mathcal{O}_1$.

Lemma 6.3.3. *For any $0 \le t \le T$ and letting $\tau_t \le t$ be the largest integer such that drift_{τ_t} is set to be 0, with probability at least $1-\omega/T$, for some universal constant $C > 0$, we have*

$$\|\nabla_t - \nabla F(x_t)\|^2 \le \big(\zeta_2^2 \cdot \sum_{i=\tau_t+1}^{t} \|x_i - x_{i-1}\|^2 + 4\zeta_1^2\big) \cdot C \cdot \log(Td/\omega). \tag{6.1}$$

Hence with probability at least $1-\omega$, we know for each $t \le T$, $\|\nabla_t - \nabla F(x_t)\|^2 \le \gamma^2/16$, where $\gamma^2 := 16C(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(Td/\omega)$ and κ is a parameter we can choose in the algorithm.

As shown in Lemma 6.3.3, the error on the gradient estimation for each step is bounded with high probability. Then we can show the algorithm can escape the saddle point efficiently based on previous results.

Lemma 6.3.4 (Essentially from [WCX19])**.** *Under Assumption 6.3.1, run SGD iterations $x_{t+1} = x_t - \eta\nabla_t$, with step size $\eta = 1/M$. Suppose x_0 is a stationary point satisfying $\|\nabla F(x_0)\| \le \alpha$ and $\mathrm{smin}(\nabla^2 F(x_0)) \le -\sqrt{\rho\alpha}$, $\alpha = \gamma\log^3(dBM/\rho\omega)$. If $\nabla_0 = \nabla F(x_0) + \zeta_1 + \zeta_2$ where $\|\zeta_1\| \le \gamma$, $\zeta_2 \sim \mathcal{N}(0, \frac{\gamma^2}{d\log(d/\omega)} I_d)$, and $\|\nabla_t - \nabla F(x_t)\| \le \gamma$ for all $t \in [\Gamma]$, with probability at least $1 - \omega\cdot\log(1/\omega)$, one has*

$$F(x_\Gamma) - F(x_0) \le -\Omega\Big(\frac{\gamma^{3/2}}{\sqrt{\rho}\log^3(\frac{dMB}{\rho\gamma\omega})}\Big),$$

where $\Gamma = \frac{M\log(\frac{dMB}{\rho\gamma\omega})}{\sqrt{\rho\gamma}}$.

We discuss this lemma in the Appendix 6.5.2 in more details. The next lemma is standard, showing how large the function values can decrease in each step.

Lemma 6.3.5. *By setting $\eta = 1/M$, we have*

$$F(x_{t+1}) \le F(x_t) + \eta\|\nabla_t\| \cdot \|\nabla F(x_t) - \nabla_t\| - \frac{\eta}{2}\|\nabla_t\|^2.$$

Moreover, with probability at least $1-\omega$*, for each* $t \le T$ *such that* $\|\nabla F(x_t)\| \ge \gamma$*, we have*

$$F(x_{t+1}) - F(x_t) \le -\eta\|\nabla_t\|^2/6 \le -\eta\gamma^2/6.$$

With the algorithm designed to control the drift term, the guarantee for Stochastic Spider to find the second order stationary point is stated below:

Lemma 6.3.6. *Suppose* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *are* ζ_1 *and* ζ_2 *norm-subGaussian respectively. If one sets* $\gamma = O(1)\sqrt{(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(Td/\omega)}$*, with probability at least* $1-\omega$*, at least one point in the output set* $\{x_1, \cdots, x_T\}$ *of Algorithm 18 is* α*-SOSP, where*

$$\alpha = \gamma\log^3(BMd/\rho\omega\gamma) = \sqrt{(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(\frac{d/\omega}{\zeta_2^2\kappa+\zeta_1^2})}\cdot\log^3(\frac{BMd}{\rho\omega(\zeta_2^2\kappa+\zeta_1^2)}).$$

As mentioned before, we can bound the number of occurrences where the drift gets large and hence bound the total time we query the oracle of the first kind.

Lemma 6.3.7. *Under the event that* $\|\nabla_t - \nabla F(x_t)\| \le \gamma/4$ *for all* $t \in [T]$ *and our parameter settings, letting* $K = \{t \in [T] : \text{drift}_t \ge \kappa\}$ *be the set of iterations where the drift is large, we know* $|K| \le O\big(\frac{B\eta}{\kappa} + T\gamma^2\eta^2/\kappa\big) = O\big(B\eta\log^4(\frac{dMB}{\rho\gamma\omega})/\kappa\big)$*.*

6.3.2 Convergence to the SOSP of the empirical risk

We use Stochastic Spider to improve the convergence to α-SOSP of the empirical risk, and aim at getting $\alpha = \tilde{O}(d^{1/3}/n^{2/3})$. We let $F_{\mathcal{D}}$ be the objective function F and use the gradient oracles

$$\mathcal{O}_1(x) := \nabla F_{\mathcal{D}}(x) + g_1, \text{ and } \quad \mathcal{O}_2(x,y) := \nabla F_{\mathcal{D}}(x) - \nabla F_{\mathcal{D}}(y) + g_2, \tag{6.2}$$

where $g_1 \sim \mathcal{N}(0, \sigma_1^2 I_d)$ and $g_2 \sim \mathcal{N}(0, \sigma_2^2 I_d)$ ensures privacy.

Before stating the formal results, note that by Lemma 6.3.6, the framework can only guarantee the existence of an α-SOSP in the outputted set. In order to find the SOSP privately from the set, we adopt the well-known AboveThreshold algorithm, whose pseudo-code can be found in Algorithm 18.

Algorithm 18: AboveThreshold

Input: A set of points $\{x_i\}_{i=1}^T$, dataset S, parameters of objective function B, M, G, ρ, objective error α;
Set $\widehat{T}_1 = \alpha + \mathrm{Lap}(\frac{4G}{n\varepsilon}) + \frac{16\log(2T/\omega)G}{n\varepsilon}, \widehat{T}_2 = -\sqrt{\rho\alpha} + \mathrm{Lap}(\frac{4M}{n\varepsilon}) - \frac{16\log(2T/\omega)M}{n\varepsilon}$;
for $i = 1, \cdots, T$ **do**
  if $\|\nabla F_S(x_i)\| + \mathrm{Lap}(\frac{8G}{n\varepsilon}) \le \widehat{T}_1 \bigwedge \mathrm{smin}(\nabla^2 F_S(x_i)) + \mathrm{Lap}(\frac{8M}{n\varepsilon}) \ge \widehat{T}_2$ **then**
    Output: x_i;
    Halt;
  end
end
Output: 0;

Algorithm 18 is a slight modification of the AboveThreshold algorithm [DR14], and we get the following guarantee immediately.

Lemma 6.3.8. *Algorithm 18 is $(\varepsilon, 0)$-DP. Given the point set $\{x_1, \cdots, x_T\}$ and S of size n as the input,*

- *if it outputs any point x_i, then with probability at least $1 - \omega$, we know*

$$\|\nabla F_S(x_i)\| \le \alpha + \frac{32\log(2T/\omega)G}{n\varepsilon}, \text{ and } \mathrm{smin}(\nabla^2 F_S(x_i)) \ge -\sqrt{\rho\alpha} - \frac{32\log(2T/\omega)M}{n\varepsilon}$$

- *if there exists a α-SOSP point $x \in \{x_i\}_{i\in[T]}$, then with probability at least $1 - \omega$, Algorithm 18 will output one point.*

Combining Algorithm 18 and Algorithm 18, we can find the SOSP we want, which is stated formally below:

Theorem 6.3.9 (Empirical)**.** *Using full batch in Algorithm 18, and setting $\kappa = \frac{G^{4/3}B^{1/3}}{M^{5/3}}(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{2/3}$, $\sigma_1 = \frac{G\sqrt{B\eta\log^2(1/\delta)/\kappa}\log^2(ndMB/\omega)}{n\varepsilon}, \sigma_2 = \frac{M\sqrt{\log^2(1/\delta)BM/\alpha_1^2}\log^5(ndMB/\omega)}{n\varepsilon}$, Algorithm 18 is (ε, δ)-DP, and with probability at least $1 - \omega$, at least one point in the output set $\{x_i\}_{i\in[T]}$ is α_1-SOSP of $F_{\mathcal{D}}$ with*

$$\alpha_1 = O\left((\frac{\sqrt{dBGM\log^2(1/\delta)}}{n\varepsilon})^{2/3} \cdot \log^6\frac{nBMd}{\rho\omega}\right).$$

Moreover, if we run Algorithm 18 with inputs $\{x_i\}_{i\in[T]}, \mathcal{D}, B, M, G, \rho, \alpha_1$*, with probability at least* $1-\omega$*, we can get an* α_2*-SOSP of* $F_{\mathcal{D}}$ *with*

$$\alpha_2 = O\left(\alpha_1 + \frac{G\log(n/G\omega)}{n\varepsilon} + \frac{M\log(ndBGM/\rho\omega)}{n\varepsilon\sqrt{\rho}}\sqrt{\alpha_1}\right).$$

6.3.3 Convergence to the SOSP of the population risk

This subsection aims at getting an α-SOSP for $F_{\mathcal{P}}$ (the population function). Differing from the stochastic oracles used for empirical function $F_{\mathcal{D}}$, we do not use full batch in the oracle. As an alternative, we draw fresh samples from $\mathcal{D}$ without replacement with a smaller batch size:

$$\mathcal{O}_1(x) := \frac{1}{b_1}\sum_{z\in S_1}\nabla f(x;z) + g_1, \text{ and } \mathcal{O}_2(x,y) := \frac{1}{b_2}\sum_{z\in S_2}(\nabla f(x;z) - \nabla f(y;z)) + g_2, \quad (6.3)$$

where S_1 and S_2 are sets of size of b_1 and b_2 respectively drawn from $\mathcal{D}$ without replacement, $g_1 \sim \mathcal{N}(0, \sigma_1^2 I_d)$ and $g_2 \sim \mathcal{N}(0, \sigma_2^2\|x-y\|_2^2 \cdot I_d)$. These gradient oracles satisfy the following.

Claim 6.3.10. *The gradient oracles* $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *constructed in Equation (6.3) are a first kind of* $O(\frac{L\sqrt{\log d}}{\sqrt{b_1}} + \sqrt{d}\sigma_1)$ *norm-subGaussian gradient oracle and second kind of* $O(\frac{M\sqrt{\log d}}{\sqrt{b_2}} + \sqrt{d}\sigma_2)$ *norm-subGaussian gradient oracle respectively.*

Proof. For the oracle $\mathcal{O}_1$, we know for each $z \in S_1$, $\mathbb{E}_{z\sim\mathcal{P}}[\nabla f(x,z)] = \nabla F_{\mathcal{P}}(x)$ and $\nabla f(x,z) - \nabla F_{\mathcal{P}}(x)$ is $\mathrm{nSG}(L)$ due to the Lipschitzness assumption. The statement follows from Fact 6.2.4 and Lemma 6.2.5. As for the $\mathcal{O}_2$, the statement follows similarly with the smoothness assumption. □

Recall that in the empirical case, we use Algorithm 18 to choose the SOSP for $F_{\mathcal{D}}$. But in the population case, we need to find SOSP for $F_{\mathcal{P}}$, and what we have are samples from $\mathcal{P}$. We need the following technical results to help us find the SOSP from the set, which follows from Hoeffding inequality for norm-subGaussians (Lemma 6.2.5) and Matrix Bernstein inequality (Theorem 6.2.7).

Lemma 6.3.11. *Fix a point $x \in \mathbb{R}^d$. Given a set S of m samples drawn i.i.d. from the distribution $\mathcal{P}$, then we know with probability at least $1-\omega$, we have*

$$\|\nabla F_S(x) - \nabla F_{\mathcal{P}}(x)\|_2 \le O\big(\frac{G\log(d/\omega)}{\sqrt{m}}\big) \bigwedge \|\nabla^2 F_S(x) - \nabla^2 F_{\mathcal{P}}(x)\|_{op} \le O\big(\frac{M\log(d/\omega)}{\sqrt{m}}\big).$$

We can bound the population bound similar to the empirical bound with these tools.

Theorem 6.3.12 (Population)**.** *Divide the dataset $\mathcal{D}$ into two disjoint datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ of size $\lceil n/2\rceil$ and $\lfloor n/2\rfloor$ respectively. Setting $b_1 = \frac{n\kappa}{B\eta}, b_2 = \frac{n\alpha_1^2}{BM}, \sigma_1 = \frac{G\sqrt{\log(1/\delta)}}{b_1\varepsilon}, \sigma_2 = \frac{M\sqrt{\log(1/\delta)}}{b_2\varepsilon}$ and $\kappa = \max(\frac{G^{4/3}B^{1/3}\log^{1/3}d}{M^{5/3}}n^{-1/3}, (\frac{GB^{2/3}}{M^{5/3}})^{6/7}(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{4/7})$ in Equation (6.3) and using them as gradient oracles, Algorithm 18 with $\mathcal{D}_1$ is (ε,δ)-DP, and with probability at least $1-\omega$, at least one point in the output is α_1-SOSP of $F_{\mathcal{P}}$ with*

$$\alpha_1 = O\Big(\big((BGM\cdot\log d)^{1/3}\frac{1}{n^{1/3}} + (G^{1/7}B^{3/7}M^{3/7})(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{3/7}\big)\log^3(nBMd/\rho\omega)\Big).$$

Moreover, if we run Algorithm 18 with inputs $\{x_i\}_{i\in[T]}, \mathcal{D}_2, B, M, G, \rho, \alpha_1$, with probability at least $1-\omega$, Algorithm 18 can output an α_2-SOSP of $F_{\mathcal{P}}$ with

$$\alpha_2 = O\left(\alpha_1 + \frac{M\log(ndBGM/\rho\omega)}{\sqrt{\rho}\min(n\varepsilon, n^{1/2})}\sqrt{\alpha_1} + G(\frac{\log(n/G\omega)}{n\varepsilon} + \frac{\log(d/\omega)}{\sqrt{n}})\right).$$

6.4 Bounding the excess risk

In this section, we consider the risk bounds.

6.4.1 Polynomial time approach

If we want the algorithm to be efficient and implementable in polynomial time, to our knowledge the only known bound is $O(\frac{d\log(1/\delta)}{\varepsilon^2\log n})$ in [WCX19] for smooth functions. [WCX19] used Gradient Langevin Dynamics, a popular variant of SGD to solve this problem, and prove the privacy by advanced composition. We generalize the exponential mechanism to the non-convex case and implement it without smoothness assumption.

First recall the Log-Sobolev inequality: We say a probability distribution π satisfies LSI with constant C_{LSI} if for all $f:\mathbb{R}^d\to\mathbb{R}$, $\mathbb{E}_\pi[f^2\log f^2] - \mathbb{E}_\pi[f^2]\log\mathbb{E}_\pi[f^2] \le 2C_{\mathrm{LSI}}\,\mathbb{E}_\pi\|\nabla f\|_2^2$.

A well-known result ([OV00]) says if f is μ-strongly convex, then the distribution proptional to $\exp(-f)$ satisfies LSI with constant $1/\mu$. Recall the results from previous results [MASN16] about LSI and DP:

Theorem 6.4.1 ([MASN16]). *Sampling from* $\exp(-\beta F(x;\mathcal{D}) - r(x))$ *for some public regularizer* r *is* (ε,δ)*-DP, where* $\varepsilon \le 2\frac{G\beta}{n}\sqrt{C_{\mathrm{LSI}}}\sqrt{1+2\log(1/\delta)}$*, and* C_{LSI} *is the worst LSI constant.*

We can apply the classic perturbation lemma to get the new LSI constant in the non-convex case. Suppose we add a regularizer $\frac{\mu}{2}\|x\|^2$, and try to sample from $\exp(-\beta(F(x;\mathcal{D})+\frac{\mu}{2}\|x\|^2))$.

Lemma 6.4.2 (Stroock perturbation). *Suppose* π *satisfies LSI with constant* $C_{\mathrm{LSI}}(\pi)$*. If* $0 < c \le \frac{\mathrm{d}\pi'}{\mathrm{d}\pi} \le C$*, then* $C_{\mathrm{LSI}}(\pi') \le \frac{C}{c}C_{\mathrm{LSI}}(\pi)$.

Lemma 6.4.3 is a more general version of Theorem 3.4 in [GTU22] and can be used to bound the empirical risk.

Lemma 6.4.3. *Let* $\pi(x) \propto \exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$*. Then for* $\beta GD > d$*, we know*

$$\mathop{\mathbb{E}}_{x\sim\pi}(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2) - \min_{x^*\in\mathcal{K}}(F_{\mathcal{D}}(x^*) + \frac{\mu}{2}\|x^*\|_2^2) \le \frac{d}{\beta}\log(\beta GD/d)$$

We now turn to bound the generalization error, and use the notion of uniform stability:

Lemma 6.4.4 (Stability and Generalization [BE02]). *Given a dataset* $\mathcal{D} = \{s_i\}_{i\in[n]}$ *drawn i.i.d. from some underlying distribution* $\mathcal{P}$*, and given any algorithm* $\mathcal{A}$*, suppose we randomly replace a sample* s *in* $\mathcal{D}$ *by an independent fresh one* s' *from* $\mathcal{P}$ *and get the neighoring dataset* $\mathcal{D}'$*, then* $\mathbb{E}_{\mathcal{D},\mathcal{A}}[F_{\mathcal{P}}(\mathcal{A}(\mathcal{D})) - F_{\mathcal{D}}(\mathcal{A}(\mathcal{D}))] = \mathbb{E}_{\mathcal{D},s',\mathcal{A}}[f(\mathcal{A}(\mathcal{D});s')) - f(\mathcal{A}(\mathcal{D}');s'))]$*, where* $\mathcal{A}(\mathcal{D})$ *is the output of* $\mathcal{A}$ *with input* $\mathcal{D}$.

As each function $f(;s')$ is G-Lipschitz, it suffices to bound the W_2 distance of $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$. If $\mathcal{A}$ is sampling from the exponential mechanism, letting $\pi_{\mathcal{D}} \propto \exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|^2))$ and $\pi_{\mathcal{D}'} \propto \exp(-\beta(F_{\mathcal{D}'}(x) + \frac{\mu}{2}\|x\|^2))$, it suffices to bound the W_2 distance between

$\pi_{\mathcal{D}}$ and $\pi_{\mathcal{D}'}$. The following lemma can bound the generalization risk of the exponential mechanism under LSI:

Lemma 6.4.5 (Generalization error bound)**.** *Let* $\pi_{\mathcal{D}} \propto \exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$. *Then we have*

$$\mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)] \leq O(\frac{G^2\exp(\beta GD)}{n\mu}).$$

We get the following results:

Theorem 6.4.6 (Risk bound)**.** *We are given* $\varepsilon, \delta \in (0, 1/2)$. *Sampling from* $\exp(-\beta(F_{\mathcal{D}}(x)+\frac{\mu}{2}\|x\|_2^2))$ *with* $\beta = O(\frac{\varepsilon\log(nd)}{GD\sqrt{\log(1/\delta)}}), \mu = \frac{d}{D^2\beta}$ *is* (ε,δ)*-DP. The empirical risk and population risk are bounded by* $O(GD\frac{d\cdot\log\log(n)\sqrt{\log(1/\delta)}}{\varepsilon\log(nd)})$.

Implementation

There are multiple existing algorithms that can sample efficiently from density with LSI, under mild assumptions. For example, when the functions are smooth or weakly smooth, one can turn to the Langevin Monte Carlo [CEL+22], and [LC22]. The algorithm in [WCX19] also requires mild smoothness assumptions. We discuss the implementation of non-smooth functions in bit more details, which is more challenging.

We can adopt the rejection sampler in [GLL22], which is based on the alternating sampling algorithm in [LST21b]. Both [LST21b] and [GLL22] are written in the language of log-concave and strongly log-concave densities, but their results hold as long as LSI holds. By combining them together, we can get the following risk bounds. The details of the implementation can be found in Appendix 6.6.3.

Theorem 6.4.7 (Implementation, risk bound)**.** *For* $\varepsilon, \delta \in (0, 1/2)$, *there is an* $(\varepsilon, 2\delta)$*-DP efficient sampler that can achieve the empirical and population risks* $O(GD\frac{d\cdot\log\log(n)\sqrt{\log(1/\delta)}}{\varepsilon\log(nd)})$. *Moreover, in expectation, the sampler takes* $\tilde{O}\left(n\varepsilon^3\log^3(d)\sqrt{\log(1/\delta)}/(GD)\right)$ *function values query and some Gaussian random variables restricted to the convex set* $\mathcal{K}$ *in total.*

6.4.2 Exponential time approach

In [GTU22], it is shown that sampling from $\exp(-\frac{\varepsilon n}{GD}F_{\mathcal{D}}(x))$ is ε-DP, and a nearly tight empirical risk bound of $\tilde{O}(\frac{DGd}{n\varepsilon})$ is achieved for convex functions. It is open what is the bound we can get for non-convex DP-SO.

Upper Bound

Given exponential time we can use a discrete exponential mechanism as considered in [BST14]. We recap the argument and extend it to DP-SO. The proof is based on a simple packing argument, and can be found in Appendix 6.6.4.

Theorem 6.4.8. *There exists an ε-DP differentially private algorithm that achieves a population risk of $O\left(GD\left(d\log(\varepsilon n/d)/(\varepsilon n) + \sqrt{d\log(\varepsilon n/d)}/(\sqrt{n})\right)\right)$.*

Lower Bound

Results in [GTU22] imply that the first term of $\tilde{O}(GDd/\varepsilon n)$ is tight, even if we relax to approximate DP with $\delta > 0$. A reduction from private selection problem shows the $\tilde{O}(\sqrt{d/n})$ generalization term is also nearly-tight (Theorem 6.4.11). In the selection problem, we have k coins, each with an unknown probability p_i. Each coin is flipped n times such that $\{x_{i,j}\}_{j\in[n]}$, each $x_{i,j}$ i.i.d. sampled from $\text{Bern}(p_i)$, and we want to choose a coin i with the smallest p_i. The risk of choosing i is $p_i - \min_{i^*} p_{i^*}$.

Theorem 6.4.9. *Any algorithm for the selection problem has excess population risk $\tilde{\Omega}(\sqrt{\frac{\log k}{n}})$.*

This follows from a folklore result on the selection problem (see e.g. [BU17]). We can combine this with the following reduction from selection to non-convex optimization:

Theorem 6.4.10 (Restatement of results in [GTU22])**.** *If any (ε,δ)-DP algorithm for selection has risk $R(k)$, then any (ε,δ)-DP algorithm for minimizing 1-Lipschitz losses over $B_d(0,1)$ (the d-dimensional unit ball) has risk $R(2^{\Theta(d)})$.*

From this and the aforementioned lower bounds in empirical non-convex optimization we get the following:

Theorem 6.4.11. *For $\varepsilon \le 1, \delta \in [2^{-\Omega(n)}, 1/n^{1+\Omega(1)}]$, any (ε,δ)-DP algorithm for minimizing 1-Lipschitz losses over $B_d(0,1)$ has excess population risk $\max\{\Omega(d\log(1/\delta)/(\varepsilon n)), \tilde{\Omega}(\sqrt{d/n})\}$.*

6.5 Omitted Proof of Section 6.3

6.5.1 Proof of Lemma 6.3.3

Lemma 6.3.3. *For any $0 \le t \le T$ and letting $\tau_t \le t$ be the largest integer such that drift_{τ_t} is set to be 0, with probability at least $1-\omega/T$, for some universal constant $C > 0$, we have*

$$\|\nabla_t - \nabla F(x_t)\|^2 \le \left(\zeta_2^2 \cdot \sum_{i=\tau_t+1}^{t} \|x_i - x_{i-1}\|^2 + 4\zeta_1^2\right) \cdot C \cdot \log(Td/\omega). \tag{6.1}$$

Hence with probability at least $1-\omega$, we know for each $t \le T$, $\|\nabla_t - \nabla F(x_t)\|^2 \le \gamma^2/16$, where $\gamma^2 := 16C(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(Td/\omega)$ and κ is a parameter we can choose in the algorithm.

Proof. If $\mathrm{drift}_{\tau_t} = 0$ happens, we use the first kind oracle to query the gradient, and hence $\nabla_{\tau_t} - \nabla F(x_{\tau_t})$ is zero-mean and $\mathrm{nSG}(2\zeta_1)$. If $t = \tau_t$, Equation (6.1) holds by the property of norm-subGaussian.

For each $\tau_t + 1 \le i \le t$, conditional on ∇_{i-1}, we know $\Delta_i - (\nabla F(x_i) - F(x_{i-1}))$ is zero-mean and $\mathrm{nSG}(\zeta_2\|x_i - x_{i-1}\|)$. Note that

$$\nabla_t - \nabla F(x_t) = \nabla_{\tau_t} - \nabla F(x_{\tau_t}) + \sum_{i=\tau_t+1}^{t} [\Delta_i - (\nabla F(x_i) - \nabla F(x_{i-1}))].$$

Equation (6.1) follows from Lemma 6.2.5.

We know $\mathrm{drift}_{t-1} = \sum_{i=\tau_t+1}^{t} \|x_i - x_{i-1}\|^2 \le \kappa$ almost surely by the design of the algorithm. By union bound, we know with probability at least $1-\omega$, for each $t \in [T]$,

$$\|\nabla_t - \nabla F(x_t)\|^2 \le C(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(Td/\omega) = \gamma^2/16.$$

□

6.5.2 Discussion of Lemma 6.3.4

Lemma 6.3.4 (Essentially from [WCX19]). *Under Assumption 6.3.1, run SGD iterations* $x_{t+1} = x_t - \eta\nabla_t$, *with step size* $\eta = 1/M$. *Suppose* x_0 *is a stationary point satisfying* $\|\nabla F(x_0)\| \le \alpha$ *and* $\mathrm{smin}(\nabla^2 F(x_0)) \le -\sqrt{\rho\alpha}$, $\alpha = \gamma\log^3(dBM/\rho\omega)$. *If* $\nabla_0 = \nabla F(x_0) + \zeta_1 + \zeta_2$ *where* $\|\zeta_1\| \le \gamma$, $\zeta_2 \sim \mathcal{N}(0, \frac{\gamma^2}{d\log(d/\omega)} I_d)$, *and* $\|\nabla_t - \nabla F(x_t)\| \le \gamma$ *for all* $t \in [\Gamma]$, *with probability at least* $1 - \omega\cdot\log(1/\omega)$, *one has*

$$F(x_\Gamma) - F(x_0) \le -\Omega\Big(\frac{\gamma^{3/2}}{\sqrt{\rho}\log^3(\frac{dMB}{\rho\gamma\omega})}\Big),$$

where $\Gamma = \frac{M\log(\frac{dMB}{\rho\gamma\omega})}{\sqrt{\rho\gamma}}$.

We briefly recap the proof of Lemma 6.3.4 in [WCX19]. One observation between the decreased function value, and the distance solutions moved is stated below:

Lemma 6.5.1 (Lemma 11, [WCX19]). *For each* $t \in [\Gamma]$, *we know*

$$\|x_{t+1} - x_0\|_2^2 \le 8\eta(\Gamma(F(x_0) - F(x_\Gamma)) + 50\eta^2\Gamma\sum_{i\in[\Gamma]}\|\nabla_i - \nabla F(x_t)\|_2^2.$$

The difference between our algorithm and the DP-GD in [WCX19] is the noise on the gradient. Note that with high probability, $\sum_{i\in[\Gamma]}\|\nabla_i - \nabla F(x_t)\|_2^2$ in our algorithm is controlled and small, and hence does not change the other proofs in [WCX19]. Hence if $F(x_0) - F(x_\Gamma)$ is small, i.e., the function value does not decrease significantly, we know x_t is close to x_0.

Let $B_x(r)$ be the unit ball of radius r around point x. Denote the $(x)_\Gamma$ the point x_Γ after running SGD mentioned in Lemma 6.3.4 for Γ steps beginning at x. With this observation, denote $B^\gamma(x_0) := \{x \mid x \in B_{x_0}(\eta\alpha), \Pr[F((x)_\Gamma) - F(x) \ge -\Phi] \ge \omega\}$. [WCX19] demonstrates the following lemma:

Lemma 6.5.2. *If* $\|\nabla F(x_0)\| \le \alpha$ *and* $\mathrm{smin}(\nabla^2 F(x_0)) \le -\sqrt{\rho\gamma}$, *then the width of* $B^\gamma(x_0)$ *along the along the minimum eigenvector of* $\nabla^2 F(x_0)$ *is at most* $\frac{\omega\eta\gamma}{\log(1/\omega)}\sqrt{\frac{2\pi}{d}}$.

The intuition is that if two different points $x^1, x^2 \in B_{x_0}(\eta\alpha)$, and $x^1 - x^2$ is large along the minimum eigenvector, then with high probability, the distance between $\|(x^1)_\Gamma - (x^2)_\Gamma\|$ will

be large, and either $\|(x^1)_\Gamma - x^1\|$ or $\|(x^2)_\Gamma - x^2\|$ is large, and hence either $F(x^1) - F((x^1)_\Gamma)$ or $F(x^2) - F((x^2)_\Gamma)$ is large. The Lemma 6.3.4 follows from Lemma 6.5.2 by using the Gaussian ζ_2 to kick off the point.

6.5.3 Proof of Lemma 6.3.5

Lemma 6.3.5. *By setting $\eta = 1/M$, we have*

$$F(x_{t+1}) \le F(x_t) + \eta\|\nabla_t\| \cdot \|\nabla F(x_t) - \nabla_t\| - \frac{\eta}{2}\|\nabla_t\|^2.$$

Moreover, with probability at least $1-\omega$, for each $t \le T$ such that $\|\nabla F(x_t)\| \ge \gamma$, we have

$$F(x_{t+1}) - F(x_t) \le -\eta\|\nabla_t\|^2/6 \le -\eta\gamma^2/6.$$

Proof. By the assumption on smoothness, we know

$$\begin{aligned} F(x_{t+1}) \le & F(x_t) + \langle \nabla F(x_t), x_{t+1} - x_t\rangle + \frac{M}{2}\|x_{t+1} - x_t\|^2 \\ = & F(x_t) - \eta/2\|\nabla_t\|^2 - \langle \nabla F(x_t) - \nabla_t, \eta\nabla_t\rangle \\ \le & F(x_t) + \eta\|\nabla F(x_t) - \nabla_t\| \cdot \|\nabla_t\| - \frac{\eta}{2}\|\nabla_t\|^2. \end{aligned}$$

By Lemma 6.3.3, with probability at least $1-\omega$, for each $t \in [T]$ we have $\|\nabla F(x_t) - \nabla_t\|_2 \le \gamma/4$. Hence we know if $\nabla F(x_t) \ge \gamma$, we have

$$F(x_{t+1}) - F(x_t) \le -\eta\|\nabla_t\|^2/6 \le -\eta\gamma^2/6.$$

□

6.5.4 Proof of Lemma 6.3.6

Lemma 6.3.6. *Suppose $\mathcal{O}_1$ and $\mathcal{O}_2$ are ζ_1 and ζ_2 norm-subGaussian respectively. If one sets $\gamma = O(1)\sqrt{(\zeta_2^2\kappa + 4\zeta_1^2)\cdot\log(Td/\omega)}$, with probability at least $1-\omega$, at least one point in*

the output set $\{x_1, \cdots, x_T\}$ *of Algorithm 18 is* α*-SOSP, where*

$$\alpha = \gamma \log^3(BMd/\rho\omega\gamma) = \sqrt{(\zeta_2^2\kappa + 4\zeta_1^2) \cdot \log(\frac{d/\omega}{\zeta_2^2\kappa + \zeta_1^2})} \cdot \log^3(\frac{BMd}{\rho\omega(\zeta_2^2\kappa + \zeta_1^2)}).$$

Proof. By Lemma 6.3.5, we know if the gradient $\|\nabla F(x_t)\| \geq \gamma$, then with high probability that $F(x_{t+1}) - F(x_t) \leq -\eta\gamma^2/6$. By Lemma 6.3.4, if x_t is a saddle point (with small gradient norm but the Hessian has a small eigenvalue), then with high probability that $F(x_{\Gamma+t}) - F(x_t) \leq -\Omega(\frac{\gamma^{3/2}}{\sqrt{\rho}\log^3(\frac{dMB}{\rho\gamma\omega})})$, and the function values decrease $\Omega\big(\frac{\gamma^2}{M\log^4(\frac{dMB}{\rho\gamma\omega})}\big)$ on average for each step.

Recall the assumption that the risk is upper bounded by B, by our setting $T = \Omega\big(\frac{BM}{\gamma^2}\log^4(\frac{dMB}{\rho\gamma\omega})\big)$, the statement is proved. □

6.5.5 *Proof of Lemma 6.3.7*

Lemma 6.3.7. *Under the event that* $\|\nabla_t - \nabla F(x_t)\| \leq \gamma/4$ *for all* $t \in [T]$ *and our parameter settings, letting* $K = \{t \in [T] : \text{drift}_t \geq \kappa\}$ *be the set of iterations where the drift is large, we know* $|K| \leq O\big(\frac{B\eta}{\kappa} + T\gamma^2\eta^2/\kappa\big) = O\big(B\eta\log^4(\frac{dMB}{\rho\gamma\omega})/\kappa\big)$.

Proof. By Lemma 6.3.5, if $\|F(x_t)\| \geq \gamma$, we know $F(x_{t+1}) - F(x_t) \leq -\eta\|\nabla_t\|^2/6$, and $F(x_{t+1}) - F(x_t) \leq \eta\gamma^2$ otherwise. Index the items in $K = \{t_1, t_2, \cdots, t_{|K|}\}$ such that $t_i < t_{i+1}$. We know

$$F(x_{t_{i+1}}) - F(x_{t_i}) \leq -\frac{1}{6\eta}\text{drift}_{t_{i+1}} + (t_{i+1} - t_i)\gamma^2\eta \leq -\frac{1}{6\eta}\kappa + (t_{i+1} - t_i)\gamma^2\eta.$$

Recall by the assumption that $\max_y F(y) - \min_x F(x) \leq B$. And hence $-B \leq F(x_{t_{|L|}}) - F(x_{t_1}) \leq -\frac{|K|}{6\eta}\kappa + T\gamma^2\eta$, and we know

$$|K| \leq O\big(\frac{B\eta}{\kappa} + T\gamma^2\eta^2/\kappa\big) = O\big(B\eta\log^4(\frac{dMB}{\rho\gamma\omega})/\kappa\big).$$

□

6.5.6 *Proof of Theorem 6.3.9*

Theorem 6.3.9 (Empirical)**.** *Using full batch in Algorithm 18, and setting* $\kappa = \frac{G^{4/3}B^{1/3}}{M^{5/3}}(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{2/3}$, $\sigma_1 = \frac{G\sqrt{B\eta\log^2(1/\delta)/\kappa}\log^2(ndMB/\omega)}{n\varepsilon}, \sigma_2 = \frac{M\sqrt{\log^2(1/\delta)BM/\alpha_1^2}\log^5(ndMB/\omega)}{n\varepsilon}$*, Algorithm 18 is* (ε,δ)*-DP, and with probability at least* $1-\omega$*, at least one point in the output set* $\{x_i\}_{i\in[T]}$ *is* α_1*-SOSP of* $F_{\mathcal{D}}$ *with*

$$\alpha_1 = O\left((\frac{\sqrt{dBGM\log^2(1/\delta)}}{n\varepsilon})^{2/3}\cdot\log^6\frac{nBMd}{\rho\omega}\right).$$

Moreover, if we run Algorithm 18 with inputs $\{x_i\}_{i\in[T]}, \mathcal{D}, B, M, G, \rho, \alpha_1$*, with probability at least* $1-\omega$*, we can get an* α_2*-SOSP of* $F_{\mathcal{D}}$ *with*

$$\alpha_2 = O\left(\alpha_1 + \frac{G\log(n/G\omega)}{n\varepsilon} + \frac{M\log(ndBGM/\rho\omega)}{n\varepsilon\sqrt{\rho}}\sqrt{\alpha_1}\right).$$

Proof. The privacy guarantee can be proved by composition theorems (Theorem 6.2.8 and Theorem 6.2.9) and Lemma 6.3.7.

As for the utility, we know the $\mathcal{O}_1$ and $\mathcal{O}_2$ constructed in Equation (6.2) are first kind of $\sigma_1\sqrt{d}$ and second kind of $\sigma_2\sqrt{d}$ norm-subGaussian gradient oracle by Fact 6.2.4. Hence by Lemma 6.3.6, the utility α_1 satisfies that

$$\begin{aligned}\alpha_1 =& O(\sigma_1\sqrt{d}+\sigma_2\sqrt{d\kappa})\cdot\log^3(BMd/\rho\omega)\\ =& O\Big(\frac{L\sqrt{dB\eta\log^2(1/\delta)/\kappa}}{n\varepsilon} + \frac{M\log^3(ndMB/\omega)\sqrt{\log^2(1/\delta)BM}}{n\varepsilon\alpha_1}\sqrt{d\kappa}\Big)\cdot\log^5(nBMd/\rho\omega).\end{aligned}$$

Choosing the best κ demonstrates the bound on α_1. The bound for α_2 follows from the value of α_1 and Lemma 6.3.8. Combining the two items in Lemma 6.3.8, we know with probability at least $1-\omega$, the output point x of Algorithm 18 satisfies that

$$\|\nabla F_{\mathcal{D}}(x)\| \le \alpha_1 + \frac{32\log(2T/\omega)G}{n\varepsilon}, \text{ and } \operatorname{smin}(\nabla^2 F_{\mathcal{D}}(x)) \ge -\sqrt{\rho\alpha_1} - \frac{32\log(2T/\omega)M}{n\varepsilon}.$$

Hence we know x is an α_2-SOSP for α_2 stated in the statement. □

6.5.7 Proof of Lemma 6.3.11

Lemma 6.3.11. *Fix a point $x \in \mathbb{R}^d$. Given a set S of m samples drawn i.i.d. from the distribution $\mathcal{P}$, then we know with probability at least $1-\omega$, we have*

$$\|\nabla F_S(x) - \nabla F_{\mathcal{P}}(x)\|_2 \le O\big(\frac{G\log(d/\omega)}{\sqrt{m}}\big) \bigwedge \|\nabla^2 F_S(x) - \nabla^2 F_{\mathcal{P}}(x)\|_{op} \le O\big(\frac{M\log(d/\omega)}{\sqrt{m}}\big).$$

Proof. As for any $s \in S$, $\nabla f(x;s) - \nabla F_{\mathcal{P}}(x)$ is zero-mean nSG(G). Then the Hoeffding inequality for norm-subGuassians (Lemma 6.2.5) demonstrates with probability at least $1-\omega/2$, we have $\|\nabla F_S(x) - \nabla F_{\mathcal{P}}(x)\|_2 \le O\big(\frac{G\log(d/\omega)}{\sqrt{m}}\big)$.

As for the other term, we know for any $s \in S, \mathbb{E}[\nabla^2 f(x;s) - \nabla^2 F_{\mathcal{P}}(x)] = 0$, and $\|\nabla^2 f(x;s) - \nabla^2 F_{\mathcal{P}}(x)\|_{op} \le 2M$ almost surely. Hence applying Matrix Bernstein inequality (Theorem 6.2.7) with $\sigma^2 = 4M^2 m, t = O\big(\sqrt{m}M\log(d/\omega)\big)$, we know with probability at least $1-\omega/2$, $\|\nabla^2 F_S(x) - \nabla^2 F_{\mathcal{P}}(x)\|_{op} \le t/m$.

Applying the Union bound completes the proof. □

6.5.8 Proof of Theorem 6.3.12

Theorem 6.3.12 (Population)**.** *Divide the dataset $\mathcal{D}$ into two disjoint datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ of size $\lceil n/2\rceil$ and $\lfloor n/2\rfloor$ respectively. Setting $b_1 = \frac{n\kappa}{B\eta}, b_2 = \frac{n\alpha_1^2}{BM}, \sigma_1 = \frac{G\sqrt{\log(1/\delta)}}{b_1\varepsilon}, \sigma_2 = \frac{M\sqrt{\log(1/\delta)}}{b_2\varepsilon}$ and $\kappa = \max(\frac{G^{4/3}B^{1/3}\log^{1/3}d}{M^{5/3}}n^{-1/3}, (\frac{GB^{2/3}}{M^{5/3}})^{6/7}(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{4/7})$ in Equation (6.3) and using them as gradient oracles, Algorithm 18 with $\mathcal{D}_1$ is (ε,δ)-DP, and with probability at least $1-\omega$, at least one point in the output is α_1-SOSP of $F_{\mathcal{P}}$ with*

$$\alpha_1 = O\Big(\big((BGM\cdot\log d)^{1/3}\frac{1}{n^{1/3}} + (G^{1/7}B^{3/7}M^{3/7})(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{3/7}\big)\log^3(nBMd/\rho\omega)\Big).$$

Moreover, if we run Algorithm 18 with inputs $\{x_i\}_{i\in[T]}, \mathcal{D}_2, B, M, G, \rho, \alpha_1$, with probability at least $1-\omega$, Algorithm 18 can output an α_2-SOSP of $F_{\mathcal{P}}$ with

$$\alpha_2 = O\left(\alpha_1 + \frac{M\log(ndBGM/\rho\omega)}{\sqrt{\rho}\min(n\varepsilon, n^{1/2})}\sqrt{\alpha_1} + G(\frac{\log(n/G\omega)}{n\varepsilon} + \frac{\log(d/\omega)}{\sqrt{n}})\right).$$

Proof. We should have all samples to be fresh to avoid dependency, and hence we need

$$b_1 \cdot |K| + b_2 \cdot T \leq n/2,$$

which is satisfied by the parameter settings and Lemma 6.3.7. As we never reuse a sample, the privacy guarantee follows directly from the Gaussian Mechanism [DR14]. By lemma 6.3.6, we have

$$\begin{aligned}
&\frac{\alpha_1}{\log^3(nBMd/\rho\omega)} \\
=&O(\sigma_1\sqrt{d} + \frac{G\sqrt{\log d}}{\sqrt{b_1}} + \sigma_2\sqrt{d\kappa} + \frac{M\sqrt{\kappa\log d}}{\sqrt{b_2}}) \cdot \\
=&O(\frac{GB\eta\sqrt{d\log(1/\delta)}}{n\varepsilon\kappa} + \frac{BM^2\sqrt{\log(1/\delta)}}{n\varepsilon\alpha_1^2}\sqrt{d\kappa} + \frac{G\sqrt{B\eta\log d}}{\sqrt{n\kappa}} + M\sqrt{\kappa}\frac{\sqrt{BM\log d}}{\sqrt{n}\alpha_1}).
\end{aligned}$$

Setting $\kappa = \max(\frac{G^{4/3}B^{1/3}\log^{1/3}d}{M^{5/3}}(n)^{-1/3}, (\frac{GB^{2/3}}{M^{5/3}})^{6/7}(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{4/7})$, we get

$$\alpha_1 = O\Big(((BGM\log d)^{1/3}\frac{1}{n^{1/3}} + (G^{1/7}B^{3/7}M^{3/7})(\frac{\sqrt{d\log(1/\delta)}}{n\varepsilon})^{3/7})\log^3(nBMd/\rho\omega)\Big).$$

Then we use the other half fresh samples $\mathcal{D}_2$ to find the point in the set by Algorithm 18. By Lemma 6.3.8 and Lemma 6.3.11, we know with probability at least $1-\omega$, for some large enough constant $C > 0$, the output point x of Algorithm 18 satisfies that

$$\begin{aligned}
\|\nabla F_{\mathcal{P}}(x)\|_2 \leq& \alpha_1 + G(\frac{32\log(2T/\omega)}{n\varepsilon} + \frac{C\log(dn/\omega)}{\sqrt{n}}), \\
\mathrm{smin}(\nabla^2 F_{\mathcal{P}}(x)) \geq& -\sqrt{\rho\alpha_1} - M(\frac{32\log(2T/\omega)}{n\varepsilon} + \frac{C\log(dn/\omega)}{\sqrt{n}})
\end{aligned}$$

Hence we know x is an α_2-SOSP for α_2 stated in the statement. The privacy guarantee follows from Basic composition and Lemma 6.3.8. □

6.6 Omitted proof of Section 6.4

6.6.1 Proof of Lemma 6.4.5

Lemma 6.4.5 (Generalization error bound)**.** *Let* $\pi_{\mathcal{D}} \propto \exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$*. Then we have*

$$\mathop{\mathbb{E}}_{\mathcal{D},x\sim\pi_{\mathcal{D}}}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)] \leq O(\frac{G^2\exp(\beta GD)}{n\mu}).$$

Proof. We know how to bound the KL divergence by LSI:

$$\begin{aligned} KL(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'}) :=& \int \log \frac{\mathrm{d}\pi_{\mathcal{D}}}{\mathrm{d}\pi_{\mathcal{D}'}} \mathrm{d}\pi_{\mathcal{D}} \\ \leq& \frac{C_{\mathrm{LSI}}}{2} \mathop{\mathbb{E}}_{\pi_{\mathcal{D}}} \left\| \nabla \log \frac{\mathrm{d}\pi_{\mathcal{D}}}{\mathrm{d}\pi_{\mathcal{D}'}} \right\|_2^2 \\ \leq& 2C_{\mathrm{LSI}} G^2\beta^2/n^2. \end{aligned}$$

LSI can lead to Talagrand transportation inequality [Theorem 1 in [OV00]], i.e.,

$$W_2(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'}) \lesssim \sqrt{C_{\mathrm{LSI}} \cdot KL(\pi_{\mathcal{D}}, \pi_{\mathcal{D}'})} = C_{\mathrm{LSI}} G\beta/n.$$

The generalization error is bounded by $O(C_{\mathrm{LSI}}G^2\beta/n)$. Using Holley-Stroock perturbation, we know $C_{\mathrm{LSI}}(\pi_{\mathcal{D}}) \leq \frac{\exp(\beta GD)}{\beta\mu}$ and hence the W_2 distance between $\pi_{\mathcal{D}}$ and $\pi_{\mathcal{D}'}$ can be bounded by $O(\frac{G\exp(\beta GD)}{n\mu})$. The statement follows the Lipschitzness constant and Lemma 6.4.4. □

6.6.2 Proof of Theorem 6.4.6

Theorem 6.4.6 (Risk bound)**.** *We are given* $\varepsilon, \delta \in (0, 1/2)$*. Sampling from* $\exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$ *with* $\beta = O(\frac{\varepsilon\log(nd)}{GD\sqrt{\log(1/\delta)}}), \mu = \frac{d}{D^2\beta}$ *is* (ε, δ)*-DP. The empirical risk and population risk are bounded by* $O(GD\frac{d\cdot\log\log(n)\sqrt{\log(1/\delta)}}{\varepsilon\log(nd)})$.

Proof. Denote $\pi(x) \propto \exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$. By Lemma 6.4.2, we know $C_{\mathrm{LSI}}(\pi) \leq$

$\frac{1}{\beta\mu} \cdot \exp(\beta GD)$. Plugging in the parameters and applying Theorem 6.4.1, we get

$$\frac{2G\beta}{n} \cdot \sqrt{\frac{\exp(\beta GD)}{\beta\mu}} \sqrt{3\log(1/\delta)} = O(1)\frac{GD\beta}{n\sqrt{d}} \sqrt{\exp(\beta GD)\log(1/\delta)} \leq 1$$

and hence prove the privacy guarantee.

As for the empirical risk bound, by Lemma 6.4.3, we know

$$\mathop{\mathbb{E}}_{x\sim\pi}(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2) - \min_{x^*\in\mathcal{K}}(F_{\mathcal{D}}(x^*) + \frac{\mu}{2}\|x^*\|_2^2) \lesssim \frac{d\log(\beta GD/d)}{\beta},$$

and we know

$$\mathop{\mathbb{E}}_{x\sim\pi} F_{\mathcal{D}}(x) - \min_{x^*\in\mathcal{K}} F_{\mathcal{D}}(x^*) \lesssim \frac{d\log(\beta GD/d)}{\beta} + \mu D^2.$$

Replacing the value of β achieves the empirical risk bound.

As for the population risk, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim\pi} F_{\mathcal{P}}(x) - \min_{y^*\in\mathcal{K}} F_{\mathcal{P}}(y^*) \\
=& \mathop{\mathbb{E}}_{x\sim\pi}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)] + \mathbb{E}[F_{\mathcal{D}}(x) - \min_{x^*\in\mathcal{K}} F_{\mathcal{D}}(x^*)] + \mathbb{E}[\min_{x^*\in\mathcal{D}} F_{\mathcal{D}}(x^*) - \min_{y^*\in\mathcal{K}} F_{\mathcal{P}}(y^*)] \\
\leq& \mathop{\mathbb{E}}_{x\sim\pi}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)] + \mathbb{E}[F_{\mathcal{D}}(x) - \min_{x^*\in\mathcal{K}} F_{\mathcal{D}}(x^*)].
\end{aligned}$$

We can bound $\mathbb{E}_{x\sim\pi}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)] \leq O(\frac{G^2\exp(\beta GD)}{n\mu}) \leq O(\frac{GD\varepsilon\log(n)}{n^{1-c}d\sqrt{\log(1/\delta)}})$ by Lemma 6.4.5 for an arbitrarily small constant $c > 0$. Hence the empirical risk is dominated term compared to $\mathbb{E}_{x\sim\pi}[F_{\mathcal{P}}(x) - F_{\mathcal{D}}(x)]$, and we complete the proof. □

6.6.3 Implementation

We rewrite them below: Let $\widehat{F}(x) := F(x) + r(x)$ where $r(x)$ is some regularizer, and $F = \mathbb{E}_{i\in I} f_i$ is the expectation of a family of G-Lipschitz functions.

Theorem 6.6.1 (Guarantee of Algorithm 19, [CCSW22])**.** *Let $\mathcal{K} \subset \mathbb{R}^d$ be a convex set of diameter D, and $\widehat{F} : \mathcal{K} \to \mathbb{R}$, and $\pi \propto \exp(-\widehat{F})$ satisfies LSI with constant C_{LSI}. Then set*

Algorithm 19: AlternateSample, [LST21b]

Input: Function $\widehat{F}$, initial point $x_0 \sim \pi_0$, step size η;
for $t \in [T]$ **do**
  $y_t \leftarrow x_{t-1} + \sqrt{\eta}\zeta$ where $\zeta \sim \mathcal{N}(0, I_d)$;
  Sample $x_t \leftarrow \exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x - y_t\|_2^2)$;
end
Output: x_T;

$\eta \geq 0$, we have

$$R_q(\pi_t, \pi) \leq \frac{R_q(\pi_0, \pi)}{(1 + \eta/C_{\mathrm{LSI}})^{2t/q}},$$

where $R_q(\pi', \pi)$ is the q-th order of Renyi divergence between π' and π.

To get a sample from $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x - y_t\|_2^2)$, we use the rejection sampler from [GLL22], whose guarantee is stated below:

Lemma 6.6.2 (Rejection Sampler, [GLL22])**.** *If the step size $\eta \lesssim G^{-2}\log^{-1}(1/\delta_{inner})$ and the inner accuracy $\delta_{inner} \in (0, 1/2)$, there is an algorithm that can return a random point x that has δ_{inner} total variation distance to the distribution proportional to $\exp(-\widehat{F}(x) - \frac{1}{2\eta}\|x - y\|_2^2)$. Moreover, the algorithm accesses $O(1)$ different f_i function values and $O(1)$ samples from the density proportional to $\exp(-r(x) - \frac{1}{2\eta}\|x - y\|_2^2)$.*

Combining Theorem 6.4.6, Theorem 6.6.1 and Lemma 6.6.2, we can get the following implementation of the exponential mechanism for non-smooth functions.

Theorem 6.4.7 (Implementation, risk bound)**.** *For $\varepsilon, \delta \in (0, 1/2)$, there is an $(\varepsilon, 2\delta)$-DP efficient sampler that can achieve the empirical and population risks $O(GD\frac{d\cdot\log\log(n)\sqrt{\log(1/\delta)}}{\varepsilon\log(nd)})$. Moreover, in expectation, the sampler takes $\tilde{O}\left(n\varepsilon^3\log^3(d)\sqrt{\log(1/\delta)}/(GD)\right)$ function values query and some Gaussian random variables restricted to the convex set $\mathcal{K}$ in total.*

Proof. By Theorem 6.4.6, it suffices to get a good sample from π with density proportional to $\exp(-\beta(F_{\mathcal{D}}(x) + \frac{\mu}{2}\|x\|_2^2))$ where $\beta = O(\frac{\varepsilon\log(nd)}{GD\sqrt{\log(1/\delta)}}), \mu = \frac{d}{D^2\beta}$. Set $q = 1$, which gives that $R_q(\cdot,\cdot)$ is the KL-divergence. Suppose we let x_0 is drawn from density proportional to $\exp(-\frac{\beta}{2}\mu\|x\|_2^2)$, then the KL divergence between π_0 and π is bounded by $\exp(q\beta GD)$.

Now let $\pi_T^{(i)}$ be the distribution we get over x_T from Algorithm 19 if we use an exact sampler for i iterations, then the sampler of Lemma 6.6.2 for the remaining $T-i$ iterations. The output of Algorithm 19 that we actually get is $\pi_T^{(0)}$. Note that $C_{\text{LSI}} \leq D^2 n$, and $\eta \lesssim \beta^{-2} G^{-2} \log^{-1}(2T/\delta)$. Setting

$$T = O\left(\frac{C_{\text{LSI}}}{\eta} \log(\exp(q\beta GD)/\delta^2)\right) = \tilde{O}\left(\frac{n\varepsilon^3 \log^3(d)\sqrt{\log(1/\delta)}}{GD}\right)$$

we get $\delta_{inner} = \delta/2T$ in Lemma 6.6.2 and that $R_1(\pi_T^{(T)}, \pi) \leq \delta^2/8$. This implies the total variation distance between $\pi_T^{(T)}$ and π is at most $\delta/2$ by Pinsker's inequality. Furthermore, by the post-processing inequality, the total variation distance between $\pi_T^{(i)}$ and $\pi_T^{(i+1)}$ is at most $\delta/2T$ for all i. Then by triangle inequality the total variation distance between $\pi_T^{(0)}$ and π is at most δ. □

6.6.4 Proof of Theorem 6.4.8

Theorem 6.4.8. *There exists an ε-DP differentially private algorithm that achieves a population risk of $O\left(GD\left(d\log(\varepsilon n/d)/(\varepsilon n) + \sqrt{d\log(\varepsilon n/d)}/(\sqrt{n})\right)\right)$.*

Proof. We pick a maximal packing P of $O((D/r)^d)$ points, such that every point in $\mathcal{K}$ is distance at most r from some point in P. By G-Lipschitzness, the risk of any point in P for the DP-ERM/SCO problems over $\mathcal{K}$ are at most Gr plus the risk of the same point for DP-ERM/SCO over P. The exponential mechanism over P gives a DP-ERM risk bound of $O\left(\frac{GD}{\varepsilon n}\log|P|\right)$. Next, note that the empirical loss of each point in P is the average of n random variables in $[0, GD]$ wlog. So, the expected maximum difference between the empirical and population loss of any point in P is $O\left(\frac{GD\sqrt{\log|P|}}{\sqrt{n}}\right)$. Putting it all together we get a DP-SCO expected risk bound of:

$$O\left(Gr + GD\left(\frac{d\log(D/r)}{\varepsilon n} + \frac{\sqrt{d\log(D/r)}}{\sqrt{n}}\right)\right).$$

This is approximately minimized by setting $r = Dd/\varepsilon n$. This gives a bound of:

$$O\left(GD\left(\frac{d\log(\varepsilon n/d)}{\varepsilon n}+\frac{\sqrt{d\log(\varepsilon n/d)}}{\sqrt{n}}\right)\right).$$

□

Chapter 7

ADAPTIVE BATCH SIZE FOR PRIVATELY FINDING SECOND-ORDER STATIONARY

7.1 Introduction

Privacy concerns have gained increasing attention with the rapid development of artificial intelligence and modern machine learning, particularly the widespread success of large language models. Differential privacy (DP) has become the standard notion of privacy in machine learning since it was introduced by [DMNS06]. Given two neighboring datasets, $\mathcal{D}$ and $\mathcal{D}'$, differing by a single item, a mechanism $\mathcal{M}$ is said to be (ε, δ)-differentially private if, for any event $\mathcal{X}$, it holds that:

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{X}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{X}] + \delta.$$

In this work, we focus on the stochastic optimization problem under the constraint of DP. The loss function is defined below:

$$F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{z \sim \mathcal{P}} f(x; z),$$

where the functions may be non-convex, the underlying distribution $\mathcal{P}$ is unknown, and we are given a dataset $\mathcal{D} = \{z_i\}_{i \in [n]}$ drawn i.i.d. from $\mathcal{P}$. Notably, our goal is to design a private algorithm with provable utility guarantees under the i.i.d. assumption.

Minimizing non-convex functions is generally challenging and often intractable, but most models used in practice are not guaranteed to be convex. How, then, can we explain the success of optimization methods in practice? One possible explanation is the effectiveness of Stochastic Gradient Descent (SGD), which is well-known to be able to find an α-first-order stationary point (FOSP) of a non-convex function f—that is, a point x such that $\|\nabla f(x)\| \leq \alpha$—within $O(1/\alpha^2)$ steps [Nes98]. However, FOSPs can include saddle points

or even local maxima. Thus, we focus on finding second-order stationary points (SOSP), for the non-convex function $F_{\mathcal{P}}$.

Non-convex optimization has been extensively studied in recent years due to its central role in modern machine learning, and we now have a solid understanding of the complexity involved in finding FOSPs and SOSPs [GL13b, AAZB+17, CDHS20, ZLJ+20]. Variance reduction techniques have been shown to improve the theoretical complexity, leading to the development of several promising algorithms such as Spider [FLLZ18], SARAH [NLST17], and SpiderBoost [WJZ+19]. More recently, private non-convex optimization has emerged as an active area of research [WCX19, ABG+23, GW23, GLOT23, LUW24, MUA+24].

7.1.1 Our Main Result

In this work, we study how to find the SOSP of $F_{\mathcal{P}}$ privately. Let us formally define the FOSP and the SOSP. For more on Hessian Lipschitz continuity and related background, see the preliminaries in Section 7.2.

Definition 7.1.1 (FOSP)**.** For $\alpha \geq 0$, we say a point x is an α-first-order stationary point (α-FOSP) of a function g if $\|\nabla g(x)\| \leq \alpha$.

Definition 7.1.2 (SOSP, [NP06, AAZB+17])**.** For a function $g : \mathbb{R}^d \to \mathbb{R}$ which is ρ-Hessian Lipschitz, we say a point $x \in \mathbb{R}^d$ is α-second-order stationary point (α-SOSP) of g if $\|\nabla g(x)\|_2 \leq \alpha \bigwedge \nabla^2 g(x) \succeq -\sqrt{\rho\alpha} I_d$.

Given the dataset size of n, privacy parameters ε, δ, and functions defined over d-dimension space, [GLOT23] proposed a private algorithm that can find an α_S-SOSP for $F_{\mathcal{P}}$ with

$$\alpha_S = \tilde{O}\big(\frac{1}{n^{1/3}} + (\frac{\sqrt{d}}{n\varepsilon})^{3/7}\big).$$

However, as shown in [ABG+23], the state-of-the-art bound for privately finding an

α_F-FOSP is tighter: [1]

$$\alpha_F = \tilde{O}\big(\frac{1}{n^{1/3}} + (\frac{\sqrt{d}}{n\varepsilon})^{1/2}\big)$$

When the privacy parameter ε is sufficiently small, we observe that $\alpha_F \ll \alpha_S$. This raises the question: is finding an SOSP under differential privacy constraints more difficult than finding an FOSP?

This work improves the results of [GLOT23]. Specifically, we present an algorithm that finds an α-SOSP with privacy guarantees, where:

$$\alpha = \tilde{O}(\alpha_F) = \tilde{O}\big(\frac{1}{n^{1/3}} + (\frac{\sqrt{d}}{n\varepsilon})^{1/2}\big).$$

This improved bound suggests that when we try to find the stationary point privately, we can find the SOSP for free under additional (standard) assumptions.

It is also worth noting that, our improvement primarily affects terms dependent on the privacy parameters. In the non-private setting, as $\varepsilon \to \infty$ (i.e., without privacy constraints), all the results discussed above achieve a bound of $\tilde{O}(1/n^{1/3})$, which matches the non-private lower bound established by [ACD+23] in high-dimensional settings (where $d \geq \tilde{\Omega}(1/\alpha^4)$). However, to our knowledge, whether this non-private term of $\tilde{O}(1/n^{1/3})$ can be further improved in low dimension remains an open question.

7.1.2 Overview of Techniques

In this work, we build on the SpiderBoost algorithm framework, similar to prior approaches [ABG+23, GLOT23], to find second-order stationary points (SOSP) privately. At a high level, our method leverages two types of gradient oracles: $\mathcal{O}_1(x) \approx \nabla f(x)$, which estimates the gradient at point x, and $\mathcal{O}_2(x, y) \approx \nabla f(x) - \nabla f(y)$, which estimates the gradient difference between two points, x and y. When performing gradient descent, to compute the gradient estimator ∇_t at point x_t, we can either use $\nabla_t = \mathcal{O}_1(x_t)$ for a direct estimate, or

[1] As proposed by [LUW24], allowing exponential running time enables the use of the exponential mechanism to find a warm start, which can further improve the bounds for both FOSP and SOSP.

$\nabla_t = \nabla_{t-1} + \mathcal{O}_2(x_t, x_{t-1})$ to update based on the previous gradient. In our setting, $\mathcal{O}_1$ is more accurate but incurs higher computational or privacy costs.

The approach in [ABG+23] adopts SpiderBoost by querying $\mathcal{O}_1$ periodically: they call $\mathcal{O}_1$ once and then use $\mathcal{O}_2$ for q subsequent queries, controlled by a hyperparameter q. Their method ensures the gradient estimators are sufficiently accurate on average, which works well for finding first-order stationary points (FOSP). However, finding an SOSP, where greater precision is required, presents additional challenges when relying on average-accurate gradient estimators.

To address this, [GLOT23] introduced a variable called $\text{drift}_t := \sum_{i=t_0+1}^{t} \|x_i - x_{i-1}\|^2$, where t_0 is the index of the last iteration when $\mathcal{O}_1$ was queried. If drift_t remains small, the gradient estimator stays accurate enough, allowing further queries of $\mathcal{O}_2$. However, if drift_t grows large, the gradient estimator's accuracy deteriorates, signaling the need to query $\mathcal{O}_1$ for a fresh, more accurate estimate. This modification enables the algorithm to maintain the precision necessary for privately finding an SOSP.

Our improvement introduces two new components: the use of the tree mechanism instead of using the Gaussian mechanism as in [GLOT23], and the implementation of adaptive batch sizes for constructing $\mathcal{O}_2$.

In the prior approach using the Gaussian mechanism, a noisy gradient estimator ∇_{t-1} is computed, and the next estimator is updated via $\nabla_t = \nabla_{t-1} + \mathcal{O}_2(x_t, x_{t-1}) + g_t$, where g_t is Gaussian noise added to preserve privacy. Over multiple iterations, the accumulation of noise $\sum g_t$ can severely degrade the accuracy of the gradient estimator, requiring frequent re-queries of $\mathcal{O}_1$. On the other hand, the tree mechanism mitigates this issue when frequent queries to $\mathcal{O}_2$ are needed.

However, simply replacing the Gaussian mechanism with the tree mechanism and using a fixed batch size does not yield optimal results. In worst-case scenarios, where the function's gradients are large, the drift grows quickly, necessitating frequent calls to $\mathcal{O}_1$, which diminishes the advantages of the tree mechanism.

To address this, we introduce adaptive batch sizes. In [GLOT23], the oracle $\mathcal{O}_2$ is constructed by drawing a fixed batch of size B from the unused dataset and outputting $\mathcal{O}_2(x_t, x_{t-1}) := \sum_{z \in S_t} \frac{\nabla f(x_t; z) - \nabla f(x_{t-1}; z)}{B}$. Given an upper bound on drift, they guaranteed

that $\|x_t - x_{t-1}\| \leq D$ for some parameter D, thereby bounding the sensitivity of $\mathcal{O}_2$.

In contrast, we dynamically adjust the batch size in proportion to $\|x_t - x_{t-1}\|$, setting $B_t \propto \|x_t - x_{t-1}\|$, and compute $\mathcal{O}_2(x_t, x_{t-1}) := \sum_{z \in S_t} \frac{\nabla f(x_t;z) - \nabla f(x_{t-1};z)}{B_t}$. Fixed batch sizes present two drawbacks: (i) when $\|x_t - x_{t-1}\|$ is large, the gradient estimator has higher sensitivity and variance, leading to worse estimate accuracy; (ii) when $\|x_t - x_{t-1}\|$ is small, progress in terms of function value decrease is limited. Using a fixed batch size forces us to handle both cases simultaneously: we must add noise and analyze accuracy assuming a worst-case large $\|x_t - x_{t-1}\|$, but for utility analysis, we pretend $\|x_t - x_{t-1}\|$ is small to examine the function value decrease. The adaptive batch size resolves this paradox: it allows us to control sensitivity and variance adaptively. When $\|x_t - x_{t-1}\|$ is small, we decrease the batch size but can still control the variance and sensitivity; when it is small, the function value decreases significantly, aiding in finding an SOSP.

By combining the tree mechanism with adaptive batch sizes, we improve the accuracy of gradient estimation and achieve better results for privately finding an SOSP.

7.1.3 Other Related Work

A significant body of literature on private optimization focuses on the convex setting, where it is typically assumed that each function $f(;z)$ is convex for any z in the universe (e.g., [CMS11, BST14, BFTGT19, FKT20, AFKT21, KLL21, CJJ+23]).

The tree mechanism, originally introduced by the differential privacy (DP) community [DNPR10, CSS11] for the continual observation, has inspired tree-structure private optimization algorithms like [AFKT21, BGN21, ABG+23, ZTC24]. Some prior works have explored adaptive batch size techniques in optimization. For instance, [DYJG16] introduced adaptive batch sizing for stochastic gradient descent (SGD), while [JWW+20] combined adaptive batch sizing with variance reduction techniques to modify SVRG and Spider algorithms. However, these works' motivations and approaches to setting adaptive batch sizes differ from ours. To the best of our knowledge, we are the first to propose using adaptive batch sizes in the context of optimization under differential privacy constraints.

Most of the non-convex optimization literature assumes that the functions being opti-

mized are smooth. Recent work has begun addressing non-smooth, non-convex functions as well, as seen in [ZLJ+20, KS21, DDL+22, JKL+23].

7.2 Preliminaries

Throughout the paper, we use $\|\cdot\|$ to represent both the ℓ_2 norm of a vector and the operator norm of a matrix when there is no confusion.

Definition 7.2.1 (Lipschitz, Smoothness and Hessian Lipschitz)**.** Let $\mathcal{K} \subseteq \mathbb{R}^d$. Given a twice differentiable function $f : \mathcal{K} \to \mathbb{R}$, we say f is G-Lipschitz, if for all $x_1, x_2 \in \mathcal{K}$, $|f(x_1) - f(x_2)| \leq G\|x_1 - x_2\|$; we say f is M-smooth, if for all $x_1, x_2 \in \mathcal{K}$, $\|\nabla f(x_1) - \nabla f(x_2)\| \leq M\|x_1 - x_2\|$, and we say the function f is ρ-Hessian Lipschitz, if for all $x_1, x_2 \in \mathcal{K}$, we have $\|\nabla^2 f(x_1) - \nabla^2 f(x_2)\| \leq \rho\|x_1 - x_2\|$.

7.2.1 Other Techniques

As mentioned in the introduction, we use the tree mechanism (Algorithm 20) to privatize the algorithm, whose formal guarantee is stated below:

Theorem 7.2.2 (Tree Mechanism, [DNPR10, CSS11])**.** *Let $\mathcal{Z}_1, \cdots, \mathcal{Z}_\Sigma$ be dataset spaces, and $\mathcal{X}$ be the state space. Let $\mathcal{M}_i : \mathcal{X}^{i-1} \times \mathcal{Z}_i \to \mathcal{X}$ be a sequence of algorithms for $i \in [\Sigma]$. Let $\mathcal{A} : \mathcal{Z}^{(1:\Sigma)} \to \mathcal{X}^\Sigma$ be the algorithm that given a dataset $Z_{1:\Sigma} \in \mathcal{Z}^{(1:\Sigma)}$, sequentially computes $X_i = \sum_{j=1}^{i} \mathcal{M}_i(X_{1:j-1}, Z_i) + \mathrm{TREE}(i)$ for $i \in [\Sigma]$, and then outputs $X_{1:\Sigma}$.*

Suppose for all $i \in [\Sigma]$, and neighboring $Z_{1:\Sigma}, Z'_{1:\Sigma} \in \mathcal{Z}^{(1:\Sigma)}$, $\|\mathcal{M}_i(X_{1:i-1}, Z_i) - \mathcal{M}_i(X_{1:i-1}, Z'_i)\| \leq s$ for all auxiliary inputs $X_{1:i-1} \in \mathcal{X}^{i-1}$. Then setting $\sigma = \frac{4s\sqrt{\log \Sigma \log(1/\delta)}}{\varepsilon}$, Algorithm 20 is (ε, δ)-DP. Furthermore, with probability at least $1 - \Sigma \cdot \iota$, for all $t \in [\Sigma] : \|\mathrm{TREE}(t)\| \lesssim \sqrt{d \log(1/\iota)}\sigma$.

We also need the concentration inequality for norm-subGaussian random vectors.

Definition 7.2.3 (SubGaussian, and Norm-SubGaussian)**.** We say a random vector $x \in \mathbb{R}^d$ is SubGaussian (SG(ζ)) if there exists a positive constant ζ such that $\mathbb{E}\, e^{\langle v, x - \mathbb{E}\, x\rangle} \leq e^{\|v\|^2\zeta^2/2}, \quad \forall v \in \mathbb{R}^d$. We say $x \in \mathbb{R}^d$ is norm-SubGaussian (nSG(ζ)) if there exists ζ such that $\Pr[\|x - \mathbb{E}\, x\| \geq t] \leq 2e^{-\frac{t^2}{2\zeta^2}}, \forall t \in \mathbb{R}$.

Lemma 7.2.4 (Hoeffding type inequality for norm-subGaussian, [JNG+19]). *Let $x_1, \cdots, x_k \in \mathbb{R}^d$ be random vectors, and for each $i \in [k]$, $x_i \mid \mathcal{F}_{i-1}$ is zero-mean $\mathrm{nSG}(\zeta_i)$ where $\mathcal{F}_i$ is the corresponding filtration. Then there exists an absolute constant c such that for any $\delta > 0$, with probability at least $1 - \omega$, $\| \sum_{i=1}^k x_i \| \leq c \cdot \sqrt{\sum_{i=1}^k \zeta_i^2 \log(2d/\omega)}$, which means $\sum_{i=1}^k x_i$ is $\mathrm{nSG}(\sqrt{c \log(d) \sum_{i=1}^k \zeta_i^2})$.*

Theorem 7.2.5 (Matrix Bernstein Inequality, [T+15]). *Consider a finite sequence of independent, random matrices $X_1, \cdots, X_n$ with common dimensions $d \times d$ and $\mathbb{E}[X_i] = 0, \|X_i\| \leq L$ a.s., $\forall i$. Let $\sigma^2 = \| \sum_{i=1}^n \mathbb{E}[X_i^2] \|$. Let $S = \sum_{i=1}^n X_i$. Then for any $t \geq 0$, we have*

$$\Pr[\|S\| \geq t] \leq d \cdot \exp(-\frac{-t^2}{\sigma^2 + Lt/3}).$$

Algorithm 20: Tree Mechanism

Input: Noise parameter σ, sequence length Σ;
Define
$\mathcal{T} := \{(u,v) : u = j \cdot 2^{\ell-1} + 1, v = (j+1) \cdot 2^{\ell-1}, 1 \leq \ell \leq \log \Sigma, 0 \leq j \leq \Sigma / 2^{\ell-1} - 1\}$;
Sample and store $\zeta_{(u,v)} \sim \mathcal{N}(0, \sigma^2)$ for each $(u,v) \in \mathcal{T}$;
for $t = 1, \cdots, \Sigma$ **do**
  Let $\mathrm{TREE}(t) \leftarrow \sum_{(u,v) \in \mathrm{NODE}(t)} \zeta_{(u,v)}$;
end
Return: $\mathrm{TREE}(t)$ for each $t \in [\Sigma]$;
;
Function NODE:;
Initialize $S = \{\}$ and $k = 0$;
for $i = 1, \cdots, \lceil \log \Sigma \rceil$ *while* $k < t$ **do**
  Set $k' = k + 2^{\lceil \log \Sigma \rceil - i}$;
  if $k' \leq t$ **then**
    $S \leftarrow S \cup \{(k+1, k')\}$, $k \leftarrow k'$
  end
end

7.3 SOSP

We make the following assumption for our main result.

Assumption 7.3.1. *Let $G, \rho, M, B > 0$. Any function drawn from $\mathcal{P}$ is G-Lipschitz, ρ-Hessian Lipschitz, and M-smooth, almost surely. Moreover, we are given a public initial point x_0 such that $F_{\mathcal{P}}(x_0) - \inf_x F_{\mathcal{P}}(x) \le B$.*

We modify the Stochastic SpiderBoost used in [GLOT23] and state it in Algorithm 21. The following standard lemma plays a crucial role in finding stationary points of smooth functions:

Lemma 7.3.2. *Assume F is M-smooth and let $\eta = 1/M$. Let $x_{t+1} = x_t - \eta\widetilde{\nabla}$. Then we have $F(x_{t+1}) \le F(x_t) + \eta\|\widetilde{\nabla}_t\| \cdot \|\nabla F(x_t) - \widetilde{\nabla}_t\| - \frac{\eta}{2}\|\widetilde{\nabla}_t\|^2$. Moreover, if $\|\nabla F(x_t)\| \ge \gamma$ and $\|\widetilde{\nabla}_t - \nabla F(x_t)\| \le \gamma/4$, we have*

$$F(x_{t+1}) - F(x_t) \le -\eta\|\widetilde{\nabla}_t\|^2/16.$$

Proof. By the assumption of the smoothness, we know

$$\begin{aligned} F(x_{t+1}) &\le F(x_t) + \langle \nabla F(x_t), x_{t+1} - x_t\rangle + \frac{M}{2}\|x_{t+1} - x_t\|^2 \\ &= F(x_t) - \eta/2\|\widetilde{\nabla}_t\|^2 - \langle \nabla F(x_t) - \widetilde{\nabla}_t, \eta\widetilde{\nabla}_t\rangle \\ &\le F(x_t) + \eta\|\nabla F(x_t) - \widetilde{\nabla}_t\| \cdot \|\widetilde{\nabla}_t\| - \frac{\eta}{2}\|\widetilde{\nabla}_t\|^2. \end{aligned}$$

When $\|\nabla F(x_t)\| \ge \gamma$ and $\|\widetilde{\nabla}_t - \nabla F(x_t)\| \le \gamma/4$, the conclusion follows from the calculation.

□

One of the main challenges to finding the SOSP, compared with finding the FOSP, is showing the algorithm can escape from saddle points, which was addressed by previous results (e.g.,[JGN+17]) and was established in the DP literature as well:

Lemma 7.3.3 (Essentially from [WCX19])**.** *Under Assumption 7.3.1, run SGD iterations $x_{t+1} = x_t - \eta\nabla_t$, with step size $\eta = 1/M$. Suppose x_0 is a saddle point satisfying $\|\nabla F(x_0)\| \le \alpha$ and $\mathrm{smin}(\nabla^2 F(x_0)) \le -\sqrt{\rho\alpha}$, $\alpha = \gamma\log^3(dBM/\rho\iota)$. If $\nabla_0 = \nabla F(x_0) + \zeta_1 + \zeta_2$ where $\|\zeta_1\| \le \gamma$, $\zeta_2 \sim \mathcal{N}(0, \frac{\gamma^2}{d\log(d/\iota)} I_d)$, and $\|\nabla_t - \nabla F(x_t)\| \le \gamma$ for all $t \in [\Gamma]$, with probability at least $1 - \iota$, one has $F(x_\Gamma) - F(x_0) \le -\Omega\big(\frac{\gamma^{3/2}}{\sqrt{\rho}\log^3(\frac{dMB}{\rho\gamma\iota})}\big)$, where $\Gamma = \frac{M\log(\frac{dMB}{\rho\gamma\iota})}{\sqrt{\rho\gamma}}$.*

Lemma 7.3.3 suggests that, if we meet a saddle point, then after the following $\tilde{O}(1/\sqrt{\gamma})$ steps, the function value will decrease by at least $\tilde{\Omega}(\gamma^{3/2})$. This means the function value decreases by $\tilde{\Omega}(\gamma^2)$ on average for each step. Similar to the proof in [GLOT23], we have the following guarantee of Stochastic SpiderBoost:

Proposition 7.3.4. *Under Assumption 7.3.1 and with gradient oracles such that* $\|\widetilde{\nabla}_t - \nabla F(x_t)\| \leq \gamma$ *for any* $t \in [T]$*, setting* $T = \tilde{O}(B/\eta\gamma^2)$ *and* $\zeta = \gamma/\sqrt{\log(d/\iota)}$ *and supposing it does not halt before completing all* T *iterations, with probability at least* $1 - T\iota$*, at least one point in the output set* $\{x_1, \cdots, x_T\}$ *of Algorithm 21 is* $\tilde{O}(\gamma)$*-SOSP.*

The proof intuition of Proposition 7.3.4 is, if we do not find an $O(\gamma)$-SOSP, then on average, the function value will at least decrease by $\Omega(\eta/\gamma^2)$. As we know $F_{\mathcal{P}}(x_0) \leq F^*_{\mathcal{P}} + B$, hence $O(B/\eta\gamma^2)$ steps can ensure we find an $O(\gamma)$-SOSP. See more discussion on Proposition 7.3.4 in the Appendix.

Algorithm 21: Stochastic Spider

Input: Dataset $\mathcal{D}$, privacy parameters ε, δ, parameters of objective function B, M, G, ρ, parameter κ, failure probability ω, batch size parameter b, noise parameter ζ;
set $\text{drift}_0 = \kappa, \text{frozen}_{-1} = 1, \Delta_{-1} = 0, \mathcal{D}_r \leftarrow \mathcal{D}, t = 0$;
while $t < T$ *and the number of unused functions is larger than* b **do**
  if $\text{drift}_t \geq \kappa$ **then**
    $\nabla_t = \mathcal{O}_1(x_t)$, $\text{drift}_t = 0$, $\text{frozen}_t = \text{frozen}_{t-1} - 1$;
  end
  else
    $\Delta_t = \mathcal{O}_2(x_t, x_{t-1})$, $\nabla_t = \nabla_{t-1} + \Delta_t$, $\text{frozen}_t = \text{frozen}_{t-1} - 1$;
  end
  if $\|\widetilde{\nabla}_{t-1}\| \leq \gamma \log^3(BMd/\rho\omega) \bigwedge \text{frozen}_{t-1} \leq 0$ **then**
    Set $\text{frozen}_t = \Gamma, \text{drift}_t = 0$;
    Set $\widetilde{\nabla}_t = \nabla_t + \text{TREE}(t) + g_t$, where $g_t \sim \mathcal{N}(0, \frac{\zeta^2}{d} I_d)$;
  end
  else
    Set $\widetilde{\nabla}_t = \nabla_t + \text{TREE}(t)$;
  end
  $x_{t+1} = x_t - \eta\widetilde{\nabla}_t, \text{drift}_{t+1} = \text{drift}_t + \|\widetilde{\nabla}_t\|_2$, $t = t + 1$;
end
Return: $\{x_1, \cdots, x_t\}$;

Algorithm 21 follows the SpiderBoost framework. We either query $\mathcal{O}_1$ to estimate the gradient itself or query $\mathcal{O}_2$ to estimate the gradient difference over the last consecutive points. The term drift controls the estimated error. When drift_t is small, we know Δ_t is still a good estimator, and when drift_t is large, we draw a fresh estimator from $\mathcal{O}_1$. The term frozen is used for the technical purpose of applying Lemma 7.3.3. When we meet a potential saddle point, we add Gaussian noise g_t to escape from the saddle point and set frozen to be Γ; this ensures that we won't add the Gaussian again in the following Γ steps.

7.3.1 Constructing Private Gradient Oracles

We construct the gradient oracles below in Algorithm 22.

Algorithm 22: gradient oracles

gradient $\mathcal{O}_1$;
inputs: x_t;
draw a batch size of b among unused functions;
return: $\frac{1}{b}\sum_z \nabla f(x_t; z)$;

---

gradient $\mathcal{O}_2$;
inputs: x_t, x_{t-1};
draw a batch size of b_t among unused functions;
return: $\frac{1}{b_t}\sum_z (\nabla f(x_t; z) - \nabla f(x_{t-1}; z))$;

Lemma 7.3.5 (Gradient oracles with bounded error)**.** *Under assumption 7.3.1, let $\iota > 0$ and use Algorithm 22 as instantiations of $\mathcal{O}_1$ and $\mathcal{O}_2$. If $\mathcal{D}$ is i.i.d. drawn from distribution $\mathcal{P}$, we have:*

(1) *for any x_t, we have $\mathbb{E}[\mathcal{O}_1(x_t)] = \nabla F(x_t)$ and*

$$\Pr[\|\mathcal{O}_1(x_t) - \nabla F(x_t)\| \geq \zeta_1] \leq \iota,$$

where $\zeta_1 = O((G\sqrt{\log(d/\iota)/b})$.

(2) *for any x_t, x_{t-1}, we have $\mathbb{E}[\mathcal{O}_2(x_t, x_{t-1})] = \nabla F(x_t) - \nabla F(x_{t-1})$ and*

$$\Pr[\|\mathcal{O}_2(x_t, x_{t-1}) - (\nabla F(x_t) - \nabla F(x_{t-1}))\| \geq \zeta_2] \leq \iota,$$

where $\zeta_2 = O(M\|x_t - x_{t-1}\|\sqrt{\log(d/\iota)/b_t})$.

Proof. For each data $z \sim \mathcal{P}$, we know

$$\mathbb{E}\,\nabla f(x_t; z) - \nabla F(x_t) = 0, \quad \|\nabla f(x_t; z) - \nabla F(x_t)\| \le 2G.$$

Then the conclusion follows from Lemma 7.2.4.

Similarly, for each data $z \sim \mathcal{P}$, we know

$$\mathbb{E}(\nabla f(x_t; z) - \nabla f(x_{t-1}; z)) - (\nabla F(x_t; z) - \nabla F(x_{t-1}; z)) = 0,$$
$$\|(\nabla f(x_t; z) - \nabla f(x_{t-1}; z)) - (\nabla F(x_t; z) - \nabla F(x_{t-1}; z))\| \le 2M\|x_t - x_{t-1}\|.$$

The statement (2) also follows from Lemma 7.2.4. □

From now on, we adopt Algorithm 22 as the gradient oracles for Line 5 and Line 8 respectively in Algorithm 21, and we set $\eta = 1/M$. We then bound the error between gradient estimator ∇_t and the true gradient $\nabla F(x_t)$ for Algorithm 21.

Lemma 7.3.6. *Suppose the dataset is drawn i.i.d. from the distribution* $\mathcal{P}$. *For any* $1 \le t \le T$ *and letting* $\tau_t \le t$ *be the largest integer such that* drift_{τ_t} *is set to be 0, with probability at least* $1 - T\iota$, *for some universal constant* $C > 0$, *we have*

$$\|\nabla_t - \nabla F(x_t)\|^2 \le C\Big(\frac{G^2}{b} + \sum_{i=\tau_t+1}^{t} M^2\|x_i - x_{i-1}\|^2/b_i\Big)\log(Td/\iota). \tag{7.1}$$

Proof. We consider the case when $t = \tau_t$ first, i.e., we query $\mathcal{O}_1$ to get ∇_t. Then Equation (7.1) follows from Lemma 7.3.5.

When $t > \tau_t$, then for each i such that $\tau_t < i \le t$, we know conditional on ∇_{i-1}, we have

$$\mathbb{E}[\Delta_i \mid \nabla_{i-1}] = \nabla F(x_i) - \nabla F(x_{i-1}).$$

That is $\Delta_i - (\nabla F(x_i) - \nabla F(x_{i-1}))$ is zero-mean and $\mathrm{nSG}(M\|x_i - x_{i-1}\|\sqrt{\log(dT/\iota)}/\sqrt{b_i})$ by applying Lemma 7.3.5. Then Equation (7.1) follows from applying Lemma 7.2.4. □

Now, we consider the noise added from the tree mechanism to make the algorithm private.

Lemma 7.3.7. *If we set* $\sigma = (\frac{G}{b} + \max_t \frac{\|\widetilde{\nabla}_t\|}{b_t}) \log(1/\delta)/\varepsilon$, *in the tree mechanism (Algorithm 20) and use Algorithm 22 as gradient oracles, then Algorithm 21 is* (ε, δ)*-DP.*

Proof. It suffices to consider the sensitivity of the gradient oracles.

Consider the sensitivity of $\mathcal{O}_1$ first. Let $\mathcal{O}(x_t)'$ denote the output with the neighboring dataset. Then it is obvious that

$$\|\mathcal{O}_1(x_t) - \mathcal{O}_1(x_t)'\| \leq \frac{G}{b}.$$

As for the sensitivity of $\mathcal{O}_2$, we have

$$\|\mathcal{O}_2(x_t, x_{t-1}) - \mathcal{O}_2(x_t, x_{t-1})'\| \leq \frac{M\|x_t - x_{t-1}\|}{b_t} = \frac{\|\widetilde{\nabla}_t\|}{b_t}.$$

The privacy guarantee follows from the tree mechanism (Theorem 7.2.2). □

With the noise added from the tree mechanism in mind, now we get the high-probability error bound of our gradient estimators $\widetilde{\nabla}_t$.

Lemma 7.3.8. *In Algorithm 21, setting* $b = G\sqrt{d}/\varepsilon\alpha + G^2/\alpha^2, b_t = \max\{\frac{\|\widetilde{\nabla}_t\| \cdot \sqrt{d}}{\alpha\varepsilon}, \frac{\kappa \cdot \|\widetilde{\nabla}_t\|}{\alpha^2}, 1\}$, *and* $\sigma = 2\log(1/\delta)\alpha/\sqrt{d}$ *correspondingly according to Lemma 7.3.7, for each* $t \in [T]$, *we know Algorithm 21 is* (ε, δ)*-DP, and with probability at least* $1 - \iota$, $\|\widetilde{\nabla}_t - \nabla F(x_t)\| \leq \gamma$, *where* $\gamma = \tilde{O}(\alpha)$.

Proof. By our setting of parameters, we know

$$(\frac{G}{b} + \max_t \frac{\|\widetilde{\nabla}_t\|}{b_t}) \leq 2\varepsilon\alpha/\sqrt{d}.$$

Then our choice of σ ensures the privacy guarantee by Lemma 7.3.7.

For any $t \in [T]$, we have

$$\|\widetilde{\nabla}_t - \nabla F(x_t)\| \leq \underbrace{\|\widetilde{\nabla}_t - \nabla_t\|}_{(1)} + \underbrace{\|\nabla_t - \nabla F(x_t)\|}_{(2)}.$$

By Theorem 7.2.2, we know

$$(1) \le \max_t \|\mathrm{TREE}(t)\| \le \sigma\sqrt{d\log(T)} \le \sigma\sqrt{d\log n} \lesssim \alpha\sqrt{\log n} \le \tilde{O}(\gamma).$$

By Lemma 7.3.6 and our parameter settings, we have

$$\begin{aligned}
\|\nabla_t - \nabla F(x_t)\|^2 \lesssim & (\alpha^2 + \sum_{i=\tau_t+1}^{t} \|\widetilde{\nabla}_t\|^2/b_t)\log(nd/\iota) \\
\lesssim & (\alpha^2 + \sum_{i=\tau_t+1}^{t} \|\widetilde{\nabla}_t\| \cdot \min\{\frac{\alpha\varepsilon}{\sqrt{d}}, \alpha^2/\kappa\})\log(nd/\iota) \\
\le & (\alpha^2 + \kappa \cdot \min\{\frac{\alpha\varepsilon}{\sqrt{d}}, \alpha^2/\kappa\})\log(nd/\iota) \\
\lesssim & \alpha^2\log(nd/\iota).
\end{aligned}$$

Hence we conclude that $(2) \lesssim \alpha\sqrt{\log(nd/\iota)}$, which completes the proof. □

We need to show that we can find an γ-SOSP before we use up all the functions. We need the following technical Lemma:

Lemma 7.3.9. *Consider the implementation of Algorithm 21. Suppose the size of dataset $\mathcal{D}$ can be arbitrarily large with functions drawn i.i.d. from $\mathcal{P}$, and we run the algorithm until finding an $\tilde{O}(\gamma)$-SOSP, then with probability at least $1 - T\iota$, the total number of functions we will use is bounded by*

$$\tilde{O}\Big(\frac{bBM}{\kappa\gamma} + BM(\frac{\sqrt{d}}{\gamma^2\varepsilon} + \frac{\kappa}{\gamma^3} + \frac{1}{\gamma^2})\Big).$$

Proof. By Proposition 7.3.4, setting $T = \tilde{O}(B/\eta\gamma^2)$ suffices to find an $\tilde{O}(\gamma)$-SOSP. Let $\{x_1, \cdots, x_t\}$ be the outputs of the algorithms, where $t \le T$ denotes the step we halt the algorithm. We show

$$\sum_{i=1}^{t} \|\widetilde{\nabla}_i\|_2 \lesssim \tilde{O}(BM/\gamma). \tag{7.2}$$

Denote the set $S := \{i \in [t] : \|\widetilde{\nabla}_i\| \le \gamma\}$. As $|S| \le T = \tilde{O}(B/\eta\gamma^2)$, we know

$\sum_{i\in S}\|\widetilde{\nabla}_i\| \le \tilde{O}(B/\eta\gamma)$.

Now consider the set $S^c := [t] \setminus S$ denoting the index of steps when the norm of the gradient estimator is large. It suffices to bound $\sum_{i\in S^c}\|\widetilde{\nabla}_i\|_2$.

By Lemma 7.3.2, we know when $\|\widetilde{\nabla}_i\| \ge \gamma$, $F(x_{i+1}) - F(x_i) \le -\eta\|\widetilde{\nabla}_i\|^2/16$, and when $\|\widetilde{\nabla}_i\| \le \gamma, F(x_{i+1}) \le F(x_i) + \eta\gamma^2$. Given the bound on the function values, we know

$$\sum_{i\in S^c}\|\widetilde{\nabla}_i\|^2 \le \tilde{O}(B/\eta).$$

Hence

$$\sum_{i\in S^c}\|\widetilde{\nabla}_i\| \le \frac{\sum_{i\in S^c}\|\widetilde{\nabla}_i\|^2}{\gamma} \le \tilde{O}(\frac{B}{\eta\gamma}).$$

This completes the proof of Equation (7.2).

The total number of functions we used for $\mathcal{O}_1$ is upper bounded by $b\cdot\frac{\sum_{i\in[t]}\|\widetilde{\nabla}_i\|}{\kappa} = \tilde{O}(bBM/\kappa\gamma)$. The total number of functions we used for $\mathcal{O}_2$ is upper bounded as follows:

$$\sum_{i\in[t]} b_t \lesssim (\frac{\sqrt{d}}{\alpha\varepsilon} + \frac{\kappa}{\alpha^2})\sum_{i\in[t]}\|\widetilde{\nabla}_t\| + T \le BM\cdot\tilde{O}(\frac{\sqrt{d}}{\gamma^2\varepsilon} + \frac{\kappa}{\gamma^3} + \frac{1}{\gamma^2}).$$

This completes the proof. □

Given the dataset size requirement, we can get the final bound on finding SOSP.

Lemma 7.3.10. *With $\mathcal{D}$ of size n drawn i.i.d. from $\mathcal{P}$, setting $\kappa = \max\{\frac{\alpha\sqrt{d}}{\varepsilon}, (BGM)^{1/3}\}$,*

$$\alpha = O\Big(((BGM)^{1/3} + \sqrt{BM})(\frac{\sqrt{d}}{n\varepsilon})^{1/2} + \frac{B^{\frac{2}{9}}M^{\frac{2}{9}}G^{\frac{5}{9}} + B^{\frac{4}{9}}M^{\frac{4}{9}}G^{\frac{1}{9}}}{n^{1/3}} + \frac{(MB)^{1/2}}{\sqrt{n}}\Big),$$

and other parameters as in Lemma 7.3.8, with probability at least $1-\iota$, at least one of the outputs of Algorithm 21 is γ-SOSP, with $\gamma = \tilde{O}(\alpha)$.

Proof. By Lemma 7.3.9 and our parameter settings, we need

$$n \ge \tilde{\Omega}(\frac{bBM}{\kappa\gamma} + BM(\frac{\sqrt{d}}{\gamma^2\varepsilon} + \frac{\kappa}{\gamma^3} + \frac{1}{\gamma^2})).$$

First,

$$n \geq \tilde{\Omega}(\frac{bBM}{\kappa\gamma}) = \Theta(\frac{BGM\sqrt{d}}{\kappa\varepsilon\gamma^2} + \frac{G^2BM}{\kappa\gamma^3}) \Leftarrow \gamma \geq \tilde{O}(\frac{(BGM)^{1/3}d^{1/4}}{\sqrt{n\varepsilon}} + \frac{B^{\frac{2}{9}}M^{\frac{2}{9}}G^{\frac{5}{9}}}{n^{1/3}}).$$

Secondly,

$$n \geq \tilde{\Omega}(BM\sqrt{d}/\gamma^2\varepsilon) \Leftarrow \gamma \geq \tilde{O}(\sqrt{BM}(\frac{\sqrt{d}}{n\varepsilon})^{1/2}).$$

Thirdly,

$$\begin{aligned} n \geq \tilde{\Omega}(BM\kappa/\gamma^3) \Leftrightarrow n &\geq \tilde{\Omega}(BM(\frac{\sqrt{d}}{\gamma^2\varepsilon}) + B^{4/3}M^{4/3}G^{1/3}/\gamma^3) \\ &\Leftarrow \gamma \geq \tilde{O}(\sqrt{BM}(\frac{\sqrt{d}}{n\varepsilon})^{1/2} + \frac{B^{\frac{4}{9}}M^{\frac{4}{9}}G^{\frac{1}{9}}}{n^{1/3}}). \end{aligned}$$

Finally,

$$n \geq \tilde{\Omega}(MB/\gamma^2) \Leftarrow \gamma \geq \tilde{O}(\frac{(MB)^{1/2}}{\sqrt{n}}).$$

Combining these together, we get the claimed statement.

□

Combining Lemma 7.3.7 and Lemma 7.3.10, we have the following main result for finding SOSP privately:

Theorem 7.3.11. *Given $\varepsilon, \delta > 0$, with gradient oracles in Algorithm 22, setting $b = G(\sqrt{d}/\varepsilon\alpha + 1/\alpha^2), b_t = \max\{\frac{\|\widetilde{\nabla}_t\|\cdot\sqrt{d}}{\alpha\varepsilon}, \frac{\|\widetilde{\nabla}_t\|\kappa}{\alpha^2}, 1\}$, $\kappa = \max\{\frac{\alpha\sqrt{d}}{\varepsilon}, (BGM)^{1/3}\}$ and $\sigma = \alpha/\sqrt{d}$, Algorithm 21 is (ε, δ)-DP, and if the dataset is i.i.d. drawn from the underlying distribution $\mathcal{P}$, at least one of the output of is $\tilde{O}(\alpha)$-SOSP, where*

$$\alpha = O\Big(((BGM)^{1/3} + \sqrt{BM})(\frac{\sqrt{d}}{n\varepsilon})^{1/2} + \frac{B^{\frac{2}{9}}M^{\frac{2}{9}}G^{\frac{5}{9}} + B^{\frac{4}{9}}M^{\frac{4}{9}}G^{\frac{1}{9}}}{n^{1/3}} + \frac{(MB)^{1/2}}{\sqrt{n}}\Big)$$

Remark 7.3.12. If we treat the parameters B, G, M as constants $O(1)$, then we get $\alpha = \tilde{O}((\frac{\sqrt{d}}{n\varepsilon})^{1/2} + \frac{1}{n^{1/3}})$ as claimed before in the abstract and introduction.

If we make further assumptions, like assuming the functions are defined over a constraint domain $\mathcal{X} \subset \mathbb{R}^d$ of diameter R and we allow exponential running time, we can get some other standard bounds that can be better than Theorem 7.3.11 in some regimes. See Appendix 7.5.2 for more discussions.

7.4 Discussion

We combine the concepts of adaptive batch sizes and the tree mechanism to improve the previous best results for privately finding SOSP. Our approach achieves the same bound as the state-of-the-art method for finding FOSP, suggesting that privately finding an SOSP may incur no additional cost.

Several interesting questions remain. First, what is the tight bound for privately finding FOSP and SOSP? Second, can the adaptive batch size technique be applied in other settings? Could it offer additional advantages, such as reducing runtime in practice? Finally, while we can obtain a generalization error bound of $\sqrt{d/n}$ using concentration inequalities and the union bound, can we achieve a better generalization error bound for the non-convex optimization?

7.5 Appendix

7.5.1 Discussion of Proposition 7.3.4

Proposition 7.3.4 has become a standard result in non-convex optimization. [GLOT23] assume two kinds of gradient oracles that satisfy the norm-Subgaussian definition below:

Definition 7.5.1 (SubGaussian gradient oracles)**.** For a G-Lipschitz and M-smooth function F:
(1) We say $\mathcal{O}_1$ is a first kind of ζ_1 norm-subGaussian Gradient oracle if given $x \in \mathbb{R}^d$, $\mathcal{O}(x)$ satisfies $\mathbb{E}\,\mathcal{O}_1(x) = \nabla F(x)$ and $\mathcal{O}_1(x) - \nabla F(x)$ is $\mathrm{nSG}(\zeta_1)$.
(2) We say $\mathcal{O}_2$ is a second kind of ζ_2 norm-subGaussian stochastic Gradient oracle if given $x, y \in \mathbb{R}^d$, $\mathcal{O}_2(x, y)$ satisfies that $\mathbb{E}\,\mathcal{O}_2(x, y) = \nabla F(x) - \nabla F(y)$ and $\mathcal{O}_2(x, y) - (\nabla F(x) - \nabla F(y))$ is $\mathrm{nSG}(\zeta_2\|x - y\|)$.

In Definition 7.5.1, the oracles are required to be unbiased with norm-subGaussian error. While this condition is sufficient, it is not necessary. These two types of gradient oracles are then used to construct gradient estimators whose errors are tightly controlled with high probability (Lemma3.3 in [GLOT23]). Proposition 7.3.4 can be established directly using Lemma 3.3 from [GLOT23], as demonstrated in the proof of Lemma 3.6 in the same work. We include the proof here for completeness.

Proof of Proposition 7.3.4. By Lemma 7.3.2 and the precondition that $\|\widetilde{\nabla}_t - \nabla F(x_t)\| \leq \gamma$, we know that, if $\|\nabla F(x_t)\| \geq 4\gamma$, then $F(x_{t+1}) - F(x_t) \leq -\eta\|\widetilde{\nabla}\|^2/16$. Otherwise, $\|\nabla F(x_t)\| < 4\gamma$. If $\|\nabla F(x_t)\| < 4\gamma$ but x_t is a saddle point, then by Lemma 7.3.3, we know

$$F(x_{t+\Gamma}) - F(x_t) \leq -\tilde{\Omega}(\gamma^{3/2}/\sqrt{\rho}),$$

where $\Gamma = \tilde{O}(M/\sqrt{\rho\gamma})$. Then if none of point in $\{x_i\}_{i\in[T]}$ is an $\tilde{O}(\gamma)$-SOSP, then we know $F(x_T) - F(x_0) < -B$, which is contradictory to Assumption 7.3.1. Hence at least one point in $\{x_i\}_{i\in[T]}$ should be an $\tilde{O}(\gamma)$-SOSP, and hence complete the proof.

□

7.5.2 Other results

The first result is combining the current result in finding the SOSP of the empirical function $F_{\mathcal{D}}(x) := \frac{1}{n}\sum_{\zeta\in\mathcal{D}} f(x;\zeta)$, and then apply the generalization error bound as follows:

Theorem 7.5.2. *Suppose $\mathcal{D}$ is i.i.d. drawn from the underlying distribution $\mathcal{P}$ and under Assumption 7.3.1. Additionaly assume $f(;\zeta) : \mathcal{X} \to \mathbb{R}$ for some constrained domain $\mathcal{X} \subset \mathbb{R}^d$ of diameter R. Then we know for any point $x \in \mathcal{X}$, with probability at least $1-\iota$, we have*

$$\|\nabla F_{\mathcal{P}}(x) - \nabla F_{\mathcal{D}}(x)\| \leq \tilde{O}(\sqrt{d/n}), \|\nabla^2 F_{\mathcal{P}}(x) - \nabla^2 F_{\mathcal{D}}(x)\| \leq \tilde{O}(\sqrt{d/n}).$$

Proof. We construct a maximal packing $\mathcal{Y}$ of $O((R/r)^d)$ points for $\mathcal{X}$, such that for any $x \in \mathcal{X}$, there exists a point $y \in \mathcal{Y}$ such that $\|x - y\| \leq r$.

By Union bound, the Hoeffding inequality for norm-subGaussian (Lemma 7.2.4 and the Matrix Bernstein Inequality(Theorem 7.2.5), we know with probability at least $1-\tau$, for

all point $y \in \mathcal{Y}$, we have

$$\|\nabla F_{\mathcal{P}}(y) - \nabla F_{\mathcal{D}}(y)\| \leq \tilde{O}(L\sqrt{d\log(R/r)/n}), \|\nabla^2 F_{\mathcal{P}}(y) - \nabla^2 F_{\mathcal{D}}(y)\| \leq \tilde{O}(M\sqrt{d\log(R/r)/n}). \tag{7.3}$$

Conditional on the above event Equation (7.3). Choosing $r \leq \min\{1, M/\rho\}\sqrt{d/n}$, then by the assumptions on Lipschitz and smoothness, we have for any $x \in \mathcal{X}$, there exists $y \in \mathcal{Y}$ such that $\|x - y\| \leq r$, and

$$\begin{aligned}\|\nabla F_{\mathcal{P}}(x) - \nabla F_{\mathcal{D}}(x)\| \leq & \|\nabla F_{\mathcal{P}}(x) - \nabla F_{\mathcal{P}}(y)\| + \|\nabla F_{\mathcal{P}}(y) - \nabla F_{\mathcal{D}}(y)\| + \|\nabla F_{\mathcal{D}}(y) - \nabla F_{\mathcal{D}}(x)\| \\ \leq & \tilde{O}(L\sqrt{d/n}).\end{aligned}$$

Similarly, we can show

$$\|\nabla^2 F_{\mathcal{P}}(x) - \nabla^2 F_{\mathcal{D}}(x)\| \leq \tilde{O}((M + \rho r)\sqrt{d/n}) = \tilde{O}(M\sqrt{d/n}).$$

□

The current SOTA of finding SOSP privately of $F_{\mathcal{D}}$ is from [GLOT23], where they can find an $\tilde{O}((\sqrt{d}/n\varepsilon)^{2/3})$-SOSP. Combining the SOTA and Theorem 7.5.2, we can find the α-SOSP of $F_{\mathcal{P}}$ privately with

$$\alpha = \tilde{O}(\frac{\sqrt{d}}{n} + (\frac{\sqrt{d}}{n\varepsilon})^{2/3}).$$

If we allow exponential running time, as [LUW24] suggests, we can find an initial point x_0 privately to minimize the empirical function and then use x_0 as a warm start to improve the final bound further.

Chapter 8

IMPROVED SAMPLE COMPLEXITY FOR PRIVATE NONSMOOTH NONCONVEX OPTIMIZATION

8.1 Introduction

We consider optimization problems in which the loss function is stochastic or empirical, of the form

$$F(x) := \mathop{\mathbb{E}}_{\xi\sim\mathcal{P}}[f(x;\xi)], \qquad \text{(stochastic)}$$
$$\widehat{F}^{\mathcal{D}}(x) := \frac{1}{n}\sum_{i=1}^{n} f(x;\xi_i), \qquad \text{(ERM)}$$

where $\mathcal{P}$ is the population distribution from which we sample a dataset $\mathcal{D} = (\xi_1, \ldots, \xi_n) \sim \mathcal{P}^n$, and the component functions $f(\,\cdot\,;\xi) : \mathbb{R}^d \to \mathbb{R}$ may be neither smooth nor convex. Such problems are ubiquitous throughout machine learning, where losses given by deep-learning based models give rise to highly nonsmooth nonconvex (NSNC) landscapes.

Due to its fundamental importance in modern machine learning, the field of nonconvex optimization has received substantial attention in recent years. Moving away from the classical regime of convex optimization, many works aimed at understanding the complexity of producing approximate-stationary points, namely with small gradient norm [GL13b, FLLZ18, CDHS20, ACD+23]. As it turns out, without smoothness, it is impossible to directly minimize the gradient norm without suffering from an exponential-dimension dependent runtime in the worst case [KS22b]. Nonetheless, a nuanced notion coined as Goldstein-stationarity [Gol77], has been shown in recent years to enable favorable guarantees. Roughly speaking, a point $x \in \mathbb{R}^d$ is called an (α, β)-Goldstein stationary point (or simply (α, β)-stationary) if there exists a convex combination of gradients in the α-ball around x whose norm is at most β.[1] Following the groundbreaking work of [ZLJ+20], a surge of works study

[1] Previous works typically use the notational convention (δ, ε)-stationarity instead of (α, β), namely where

NSNC optimization through the lens of Goldstein stationarity, with associated finite-time guarantees [DDL+22, LZJ22, CMO23, JKL+23, KL23, GJ23, KS24, TS24].

In this work, we study NSNC optimization problems under the additional constraint of differential privacy (DP) [DR14]. With the ever-growing deployment of ML models in various domains, the privacy of the data on which models are trained is a major concern. Accordingly, DP optimization is an extremely well-studied problem, with a vast literature focusing on functions that are assumed to be either convex or smooth [BST14, WYX17, BFTGT19, WCX19, FKT20, GLL22, ABG+23, CJJ+23, LGOGT24]. The fundamental investigation in this literature is the privacy-utility trade-off, that is, assessing the minimal dataset size n (referred to as the sample complexity) required in order to optimize the loss up to some error, using a DP algorithm.

For NSNC DP optimization, [ZTC24] recently provided a zero-order algorithm, namely that utilizes only function value evaluations of $f(\cdot\,;\xi)$, which preforms a single pass over the dataset and returns an (α,β)-stationary point of F under (ε,δ)-DP as long as

$$n = \widetilde{\Omega}\left(\frac{d}{\alpha\beta^3} + \frac{d^{3/2}}{\varepsilon\alpha\beta^2}\right). \tag{8.1}$$

To the best of our knowledge this is the only existing result in this realm.

8.1.1 Our contributions

In this paper, we provide new algorithms for NSNC DP optimization, which improve the previously best-known sample complexity for this task. For consistency with the previous result by [ZTC24], our algorithms will be zero-order, yet in Appendix 8.9 we provide first-order algorithms (i.e., gradient-based) with the same sample complexities, and better oracle complexity. Our contributions, summarized in Table 8.1, are as follows:

1. **Improved single-pass algorithm (Theorem 8.3.1):** We provide an (ε,δ)-DP algorithm that preforms a single pass over that dataset, and returns an (α,β)-stationary

δ is the radius (instead of α) and ε is the norm bound (instead of β). We depart from this notational convention in order to avoid confusion with the standard privacy notation of (ε,δ)-DP.

point as long as

$$n = \widetilde{\Omega}\left(\frac{1}{\alpha\beta^3} + \frac{d}{\varepsilon\alpha\beta^2} + \frac{d^{3/4}}{\varepsilon^{1/2}\alpha\beta^{5/2}}\right), \tag{8.2}$$

which is always at least $\Omega(\sqrt{d})$ times smaller than (8.1).[2] Notably, the "non-private" term $1/\alpha\beta^3$ is dimension-independent, as opposed to (8.1), which is the first result of this sort for NSNC DP optimization, and was erroneously claimed impossible by previous work (see Remark 8.3.2).

2. **Better multi-pass algorithm (Theorem 8.4.1):** In order to further improve the sample complexity, we move to consider ERM algorithms that go over the data multiple times (polynomially), which we will later argue generalize to the population loss. To that end, we provide an (ε, δ)-DP ERM algorithm, that returns an (α, β)-Goldstein stationary point of $\widehat{F}^{\mathcal{D}}$ as long as

$$n = \widetilde{\Omega}\left(\frac{d^{3/4}}{\varepsilon\alpha^{1/2}\beta^{3/2}}\right). \tag{8.3}$$

Notably, (8.3) substantially improves (8.2) (and thus, (8.1)) in parameter regimes of interest (small $\varepsilon, \alpha, \beta$, large d) with respect to the dimension and accuracy parameters, and in particular is the first algorithm to preform private ERM with sublinear dimension-dependent sample complexity for NSNC objectives.

In order to utilize our empirical algorithm for stochastic objectives, one must argue that Goldstein-stationarity generalizes from the ERM to the population. As no such argument is currently pointed out in the literature, we provide a result that ensures this:

- **Additional contribution: generalizing from ERM to population (Proposition 8.5.1).** We show that with high probability, any $(\alpha, \widehat{\beta})$-stationary point of $\widehat{F}^{\mathcal{D}}$ is an (α, β)-stationary point of F, for $\beta = \widehat{\beta} + \widetilde{O}(\sqrt{d/n})$. Hence, the empirical guarantee (8.3) generalizes to stochastic losses with an additional d/β^2 additive term in n (up to log terms).

---

[2] Note that $\frac{d^{3/4}}{\varepsilon^{1/2}\alpha\beta^{5/2}} \lesssim \frac{1}{\sqrt{d}}(\frac{d}{\alpha\beta^3} + \frac{d^{3/2}}{\varepsilon\alpha\beta^2})$ by the AM-GM inequality.

Sample complexity summary	empirical	stochastic
[ZTC24] (single-pass)	$\frac{d}{\alpha\beta^3} + \frac{d^{3/2}}{\varepsilon\alpha\beta^2}$	
Theorem 8.3.1 (single-pass)	$\frac{1}{\alpha\beta^3} + \frac{d}{\varepsilon\alpha\beta^2} + \frac{d^{3/4}}{\varepsilon^{1/2}\alpha\beta^{5/2}}$	
Theorem 8.4.1 (multi-pass)	$\frac{d^{3/4}}{\varepsilon\alpha^{1/2}\beta^{3/2}}$	$\frac{d}{\beta^2} + \frac{d^{3/4}}{\varepsilon\alpha^{1/2}\beta^{3/2}}$

Table 8.1: Main results (ignoring dependence on Lipschitz constant, initialization, and log terms).

8.2 Preliminaries

Notation. We denote by $\langle \cdot, \cdot \rangle, \|\cdot\|$ the standard Euclidean dot product and its induced norm. For $x \in \mathbb{R}^d$ and $\alpha > 0$, we denote by $\mathbb{B}(x, \alpha)$ the closed ball of radius α centered at x, and further denote $\mathbb{B}_\alpha := \mathbb{B}(0, \alpha)$. $\mathbb{S}^{d-1} \subset \mathbb{R}^d$ denotes the unit sphere. We make standard use of O-notation to hide absolute constants, $\widetilde{O}, \widetilde{\Omega}$ to hide poly-logarithmic factors, and also let $f \lesssim g$ denote $f = O(g)$.

Nonsmooth optimization. A function $h : \mathbb{R}^d \to \mathbb{R}$ is called L-Lipschitz if for all $x, y \in \mathbb{R}^d : |h(x) - h(y)| \leq L\|x - y\|$. We call h H-smooth, if h is differentiable and ∇h is H-Lipschitz with respect to the Euclidean norm. For Lipschitz functions, the Clarke subgradient set [Cla90] can be defined as

$$\partial h(x) := \mathrm{conv}\{g : g = \lim_{n\to\infty} \nabla h(x_n),\, x_n \to x\},$$

namely the convex hull of all limit points of $\nabla h(x_n)$ over sequences of differentiable points (which are a full Lebesgue-measure set by Rademacher's theorem), converging to x. For $\alpha \geq 0$, the Goldstein α-subdifferential [Gol77] is further defined as

$$\partial_\alpha h(x) := \mathrm{conv}(\cup_{y\in\mathbb{B}(x,\alpha)} \partial h(y)),$$

and we denote the minimum-norm element of the Goldstein α-subdifferential by

$$\overline{\partial}_\alpha h(x) := \arg\min_{g \in \partial_\alpha h(x)} \|g\| .$$

Definition 8.2.1. A point $x \in \mathbb{R}^d$ is called an (α, β)-Goldstein stationary point of h if $\left\|\overline{\partial}_\alpha h(x)\right\| \le \beta$.

Throughout the paper we impose the following standard Lipschitz assumption:

Assumption 8.2.2. *For any ξ, $f(\cdot\,;\xi) : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz (hence, so is F).*

Randomized smoothing. Given any function $h : \mathbb{R}^d \to \mathbb{R}$, we denote its randomized smoothing $h_\alpha(x) := \mathbb{E}_{y \sim \mathbb{B}_\alpha} h(x+y)$. We recall the following standard properties of randomized smoothing [FKM05, YNS12, DBW12, Sha17].

Fact 8.2.3 (Randomized smoothing)**.** *Suppose $h : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz. Then*

- *h_α is L-Lipschitz.*
- *$|h_\alpha(x) - h(x)| \le L\alpha$ for any $x \in \mathbb{R}^d$.*
- *h_α is $O(L\sqrt{d}/\alpha)$-smooth.*
- $\nabla h_\alpha(x) = \mathbb{E}_{y \sim \mathbb{B}_\alpha}[\nabla h(x+y)] = \mathbb{E}_{y \sim \mathbb{S}^{d-1}}[\frac{d}{2\alpha}(h(x+\alpha y) - h(x-\alpha y))y]$.

The following result shows that in order to find a Goldstein-stationary point of a function, it suffices to find a Goldstein-stationary point of its randomized smoothing:

Lemma 8.2.4 (KS24, Lemma 4)**.** *Any (α, β)-stationary point of h_α is a $(2\alpha, \beta)$-stationary point of h.*

Differential privacy. Two datasets $\mathcal{D}, \mathcal{D}' \in \mathrm{supp}(\mathcal{P})^n$ are said to be neighboring if they differ in only one data point. A randomized algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{R}$ is called (ε, δ) differentially private (or (ε, δ)-DP) for $\varepsilon, \delta > 0$ if for any two neighboring datasets $\mathcal{D}, \mathcal{D}'$ and measurable $E \subseteq \mathcal{R}$ in the algorithm's range, it holds that $\Pr[\mathcal{A}(\mathcal{D}) \in E] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in E] + \delta$ [DR14].

Next, we recall the well-known tree mechanism given by Algorithm 20, and its associated guarantee presented below.

Proposition 8.2.5 (Tree Mechanism DNPR10, CSS11, ZTC24)**.** *Let $\mathcal{Z}_1, \cdots, \mathcal{Z}_\Sigma$ be dataset spaces, and $\mathcal{X}$ be the state space. Let $\mathcal{M}_i : \mathcal{X}^{i-1} \times \mathcal{Z}_i \to \mathcal{X}$ be a sequence of algorithms for $i \in [\Sigma]$. Let $\mathcal{A} : \mathcal{Z}^{(1:\Sigma)} \to \mathcal{X}^\Sigma$ be the algorithm that given a dataset $Z_{1:\Sigma} \in \mathcal{Z}^{(1:\Sigma)}$, sequentially computes $X_i = \sum_{j=1}^{i} \mathcal{M}_i(X_{1:j-1}, Z_i) + \mathrm{TREE}(i)$ for $i \in [\Sigma]$, and then outputs $X_{1:\Sigma}$. Suppose for all $i \in [\Sigma]$, and neighboring $Z_{1:\Sigma}, Z'_{1:\Sigma} \in \mathcal{Z}^{(1:\Sigma)}$, $\|\mathcal{M}_i(X_{1:i-1}, Z_i) - \mathcal{M}_i(X_{1:i-1}, Z'_i)\| \leq s$ for all auxiliary inputs $X_{1:i-1} \in \mathcal{X}^{i-1}$. Then setting $\sigma = 4s\sqrt{\log \Sigma \log(1/\delta)}/\varepsilon$, Algorithm 20 is (ε, δ)-DP. Furthermore, for all $t \in [\Sigma]$: $\mathbb{E}[\mathrm{TREE}(t)] = 0$ and $\mathbb{E}\,\|\mathrm{TREE}(t)\|^2 \lesssim d \log(\Sigma)\sigma^2$.*

8.2.1 Base algorithm: O2NC

Similar to [ZTC24], our general algorithm is based on the so-called "Online-to-Non-Convex conversion" (O2NC) of [CMO23]. We slightly modify previous proofs by disentangling the role of the variance of the gradient estimator vs. its second order moment, as follows:

Proposition 8.2.6 (O2NC)**.** *Suppose that $\mathcal{O}(\cdot)$ is a stochastic gradient oracle of some differentiable function $h : \mathbb{R}^d \to \mathbb{R}$, so that for all $z \in \mathbb{R}^d$: $\mathbb{E}\,\|\mathcal{O}(z) - \nabla h(z)\|^2 \leq G_0^2$ and $\mathbb{E}\,\|\mathcal{O}(z)\|^2 \leq G_1^2$. Then running Algorithm 23 with $\eta = \frac{D}{G_1\sqrt{M}}$, $MD \leq \alpha$, uses T calls to $\mathcal{O}(\cdot)$, and satisfies*

$$\mathbb{E}\left\|\overline{\partial}_\alpha h(x^{\mathrm{out}})\right\| \leq \frac{h(x_0) - \inf h}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0.$$

We provide a proof of Proposition 8.2.6 in Appendix 8.8. Recalling that by Lemma 8.2.4 any (α, β)-stationary point of F_α is a $(2\alpha, \beta)$-stationary point of F, we see that it is enough

to design a private stochastic gradient oracle $\mathcal{O}$ of ∇F_α, while controlling its variance G_0 and second moment G_1. In the next sections, we show how to construct such private oracles and derive the corresponding guarantees through Proposition 8.2.6. As previously remarked, in the main text, our oracles will be based on zero-order queries of the component functions $f(\cdot, \xi)$, yet in Appendix 8.9, we also show we can construct oracles with the same sample complexity using first-order queries with a lower oracle complexity.

Algorithm 23: Nonsmooth Nonconvex Algorithm (based on O2NC [CMO23])

Input: Oracle $\mathcal{O} : \mathbb{R}^d \to \mathbb{R}^d$, initialization $x_0 \in \mathbb{R}^d$, clipping parameter $D > 0$, step size $\eta > 0$, averaging length $M \in \mathbb{N}$, iteration budget $T \in \mathbb{N}$;
Initialize: $\Delta_1 = \mathbf{0}$;
for $t = 1, \dots, T$ **do**
  Sample $s_t \sim \mathrm{Unif}[0, 1]$;
  $x_t = x_{t-1} + \Delta_t$;
  $z_t = x_{t-1} + s_t \Delta_t$;
  $\tilde{g}_t = \mathcal{O}(z_t)$;
  $\Delta_{t+1} = \mathrm{clip}_D\left(\Delta_t - \eta \tilde{g}_t\right)$ ▷ $\mathrm{clip}_D(z) := \min\{1, \frac{D}{\|z\|}\} \cdot z$
end
;
$K = \lfloor \frac{T}{M} \rfloor$;
for $k = 1, \dots, K$ **do**
  $\overline{x}_k = \frac{1}{M} \sum_{m=1}^{M} z_{(k-1)M+m}$;
end
Sample $x^{\mathrm{out}} \sim \mathrm{Unif}\{\overline{x}_1, \dots, \overline{x}_K\}$;
Output: x^{out};

8.3 Single-pass algorithm

In this section, we consider Algorithm 24, which provides an oracle to be used in Algorithm 23. Algorithm 24 is such that throughout T calls, it uses each data point once, and hence, privacy is maintained with no need for composition. Before getting into the details, we will provide the main underlying idea. We consider the zero-order gradient estimator

$$\tilde{\nabla} f_\alpha(x; \xi) = \frac{1}{m} \sum_{j=1}^{m} \frac{d}{2\alpha} (f(x + \alpha y_j; \xi) - f(x - \alpha y_j; \xi)), \quad y_1, \dots, y_m \overset{iid}{\sim} \mathrm{Unif}(\mathbb{S}^{d-1}), \tag{8.4}$$

296

Algorithm 24: Single-pass instantiation of $\mathcal{O}(z_t)$ in Line 7 of Algorithm 23

Input: Current iterate z_t, time $t \in \mathbb{N}$, period length $\Sigma \in \mathbb{N}$, accuracy parameter $\alpha > 0$, batch sizes $B_1, B_2 \in \mathbb{N}$, gradient validation size $m \in \mathbb{N}$, noise level $\sigma > 0$;
if $t \ mod \ \Sigma = 1$ **then**
  Sample minibatch S_t of size B_1 among unused samples;
  for *each sample* $\xi_i \in S_t$ **do**
    Sample $y_1, \dots, y_m \overset{iid}{\sim} \mathrm{Unif}(\mathbb{S}^{d-1})$;
    $\tilde{\nabla} f(z_t; \xi_i) = \frac{1}{m} \sum_{j \in [m]} \frac{d}{2\alpha} (f(z_t + \alpha y_j; \xi_i) - f(z_t - \alpha y_j; \xi_i)) y_j$;
  end
  $g_t = \frac{1}{B_1} \sum_{\xi_i \in S_t} \tilde{\nabla} f(z_t; \xi_i)$;
end
else
  Sample minibatch S_t of size B_2 among unused samples;
  for *each sample* $\xi_i \in S_t$ **do**
    Sample $y_1, \dots, y_{2m} \overset{iid}{\sim} \mathrm{Unif}(\mathbb{B}_\alpha)$;
    $\tilde{\nabla} f(z_t; \xi_i) = \frac{1}{m} \sum_{j \in [m]} \frac{d}{2\alpha} (f(z_t + \alpha y_j; \xi_i) - f(z_t - \alpha y_j; \xi_i)) y_j$;
    $\tilde{\nabla} f(z_{t-1}; \xi_i) = \frac{1}{m} \sum_{j=m+1}^{2m} \frac{d}{2\alpha} (f(z_{t-1} + \alpha y_j; \xi_i) - f(z_{t-1} - \alpha y_j; \xi_i)) y_j$;
  end
  $g_t = g_{t-1} + \frac{1}{B_2} \sum_{\xi_i \in S_t} (\tilde{\nabla} f(z_t; \xi_i) - \tilde{\nabla} f(z_{t-1}; \xi_i))$;
end
Return $\tilde{g}_t = g_t + \mathrm{TREE}(\sigma, \Sigma)(t \bmod \Sigma)$;

which is an unbiased estimator of $\nabla f_\alpha(x;\xi)$, to which we then apply variance reduction. [ZTC24] considered the oracle above specifically with $m = d$, for which it is easy to bound the sensitivity of this estimator over neighboring minibatches $\xi_{1:B}, \xi'_{1:B}$ of size B by

$$\left\| \frac{1}{B}\sum_{i=1}^{B} \tilde{\nabla} f_\alpha(x;\xi_i) - \frac{1}{B}\sum_{i=1}^{B} \tilde{\nabla} f_\alpha(x;\xi'_i) \right\| \le \frac{Ld}{B}. \tag{8.5}$$

Our key observation is that while this is indeed the worst-case sensitivity, we can get substantially lower sensitivity *with high probability*. For sufficiently large m, standard sub-Gaussian concentration properties ensure that $\tilde{\nabla} f_\alpha(x;\xi_i) \approx \nabla f_\alpha(x;\xi_i)$ with high probability, and hence under this event we show the sensitivity over a mini-batch can be decreased to an order of $\frac{L}{B}$. As this is a factor of d smaller than (8.5), we can add significantly less noise in order to privatize, therefore leading to faster convergence to stationarity.

The main theorem in this section is the following:

Theorem 8.3.1 (Single-pass algorithm)**.** *Suppose $F(x_0) - \inf_x F(x) \le \Phi$, that Assumption 8.2.2 holds, and let $\alpha, \beta, \delta, \varepsilon > 0$ such that $\alpha \le \frac{\Phi}{L}$. Then setting $B_1 = \Sigma$, $B_2 = 1$, $M = \alpha/4D$, $m = \widetilde{O}(d^2 B_1^2 + \frac{d\alpha^2 B_2^2}{D^2})$, $\sigma = \widetilde{O}(\frac{L}{B_1\varepsilon} + \frac{LD\sqrt{d}}{\alpha B_2 \varepsilon})$, $\Sigma = \widetilde{\Theta}((\frac{\alpha}{\varepsilon D})^{2/3})$, $D = \widetilde{\Theta}(\min\{(\frac{\Phi^2\alpha}{L^2T^2})^{1/3}, (\frac{\Phi\alpha\varepsilon}{dLT})^{1/2}, (\frac{\Phi^3\alpha^2\varepsilon}{d^{3/2}L^3T^3})^{1/5}\})$, $T = \Theta(n)$, and running Algorithm 23 with Algorithm 24 as the oracle subroutine, is (ε,δ)-DP. Furthermore, its output satisfies $\mathbb{E}\,\|\overline{\partial}_{2\alpha} F(x^{\text{out}})\| \le \beta$ as long as*

$$n = \widetilde{\Omega}\left(\frac{\Phi L^2}{\alpha\beta^3} + \frac{\Phi L d}{\varepsilon\alpha\beta^2} + \frac{\Phi L^{3/2} d^{3/4}}{\varepsilon^{1/2}\alpha\beta^{5/2}} \right).$$

Remark 8.3.2. It is interesting to note that the "non-private" term $\Phi L^2/\alpha\beta^3$ in Theorem 8.3.1 is independent of the dimension d. Not only is this the first result of this sort, this was even (erroneously) claimed impossible by [ZTC24]. The reason for this confusion is that while the optimal zero-order *oracle* complexity is $d/\alpha\beta^3$ [KS24], and in particular must scale with the dimension [DJWW15], the *sample* complexity might not.

In the rest of the section, we will present the basic properties of this oracle in terms of sensitivity (implying the privacy), variance and second moment. We will then plug these into Algorithm 23, which enables proving Theorem 8.3.1. Corresponding proofs are deferred to Section 8.6.

Lemma 8.3.3 (Sensitivity). *Consider the gradient oracle $\mathcal{O}(\cdot)$ in Algorithm 24 when acting on two neighboring minibatches S_t and S'_t, and correspondingly producing g_t and g'_t, respectively. If $t \bmod \Sigma = 1$, then it holds with probability at least $1 - \delta/2$ that*

$$\|g_t - g'_t\| \lesssim \frac{L}{B_1} + \frac{Ld\sqrt{\log(dB_1/\delta)}}{\sqrt{m}}.$$

Otherwise, conditioned on $g_{t-1} = g'_{t-1}$, we have with probability at least $1 - \delta/2$:

$$\|g_t - g'_t\| \lesssim \frac{L\sqrt{d}D}{\alpha B_2} + \frac{Ld\sqrt{\log(dB_1/\delta)}}{\sqrt{m}}.$$

With the sensitivity bound given by Lemma 8.3.3, we easily derive the privacy guarantee of our oracle from the Tree Mechanism (Proposition 8.2.5).

Lemma 8.3.4 (Privacy). *Running Algorithm 24 with $m = O\left(\log(dB_2/\delta)(d^2B_1^2 + \frac{d\alpha^2B_2^2}{D^2})\right)$ and $\sigma = O\left(\frac{L\sqrt{\log(1/\delta)}}{B_1\varepsilon} + \frac{LD\sqrt{d\log(1/\delta)}}{\alpha B_2\varepsilon}\right)$ is (ε, δ)-DP.*

We next analyze the variance and second moment of the gradient oracle.

Lemma 8.3.5 (Variance). *In Algorithm 24, for all $t \in [T]$ it holds that*

$$\mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha(z_t)\|^2 \lesssim \frac{L^2}{B_1} + \frac{L^2d^2}{B_1m} + \frac{L^2dD^2\Sigma}{\alpha^2B_2} + \sigma^2 d\log\Sigma + \frac{L^2d^2\Sigma}{mB_2},$$
$$\mathbb{E}\,\|\tilde{g}_t\|^2 \lesssim L^2 + \frac{L^2d^2}{B_1m} + \frac{L^2dD^2\Sigma}{\alpha^2B_2} + \sigma^2 d\log\Sigma + \frac{L^2d^2\Sigma}{mB_2}.$$

Combining the ingredients that we have set up, we can derive Theorem 8.3.1.

Proof of Theorem 8.3.1. The privacy guarantee follows directly from Lemma 8.3.4, by noting that our parameter assignment implies $B_1T/\Sigma + B_2T = O(n)$, which allows letting $T = \Theta(n)$ while never re-using samples (hence no privacy composition is required). Therefore, it remains to show the utility bound. By applying Lemma 8.2.4 and Proposition 8.2.6, we get that

$$\mathbb{E}\,\|\overline{\partial}_{2\alpha}F(x^{\text{out}})\| \le \mathbb{E}\,\|\overline{\partial}_\alpha F_\alpha(x^{\text{out}})\| \le \frac{F_\alpha(x_0) - \inf F_\alpha}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0$$

$$\leq \frac{2\Phi}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0, \tag{8.6}$$

where the last inequality used the fact that Assumption 8.2.2 and Fact 8.2.3 together imply that $F_\alpha(x_0) - \inf F_\alpha \leq F(x_0) - \inf F + L\alpha \leq \Phi + L\alpha \leq 2\Phi$. Under our parameter assignment, Lemma 8.3.5 yields

$$G_1 \lesssim G_0 + L, \tag{8.7}$$

which plugged into (8.6) gives

$$\mathbb{E}\,\|\overline{\partial}_{2\alpha} F(x^{\text{out}})\| = O\left(\frac{\Phi}{DT} + \frac{L}{\sqrt{M}} + G_0\right). \tag{8.8}$$

Moreover, under our parameter assignment Lemma 8.3.5 also gives the bound

$$G_0 \lesssim \frac{L}{B_1} + \frac{LD\sqrt{d\Sigma}}{\alpha B_2} + \sigma\sqrt{d\log\Sigma} = \widetilde{O}\left(\frac{LDd^{1/2}\Sigma^{1/2}}{\alpha} + \frac{Ld^{1/2}}{\Sigma\varepsilon} + \frac{LDd}{\alpha\varepsilon}\right), \tag{8.9}$$

which propagated into (8.8) and recalling that $M = \Theta(\alpha/D)$ shows that

$$\mathbb{E}\,\|\overline{\partial}_{2\alpha} F(x^{\text{out}})\| = \widetilde{O}\left(\frac{\Phi}{DT} + \frac{LD^{1/2}}{\alpha^{1/2}} + \frac{LDd^{1/2}\Sigma^{1/2}}{\alpha} + \frac{Ld^{1/2}}{\Sigma\varepsilon} + \frac{LDd}{\alpha\varepsilon}\right).$$

Plugging our assignments of Σ and D, and recalling that $n = \Theta(T)$, a straightforward calculation simplifies the bound above to

$$\begin{aligned}\mathbb{E}\,\|\overline{\partial}_{2\alpha} F(x^{\text{out}})\| &= \widetilde{O}\left(\left(\frac{\Phi L^2}{T\alpha}\right)^{1/3} + \left(\frac{\Phi dL}{T\alpha\varepsilon}\right)^{1/2} + \left(\frac{\Phi^2 L^3 d^{3/2}}{T^2\alpha^2\varepsilon}\right)^{1/5}\right) \\ &= \widetilde{O}\left(\left(\frac{\Phi L^2}{n\alpha}\right)^{1/3} + \left(\frac{\Phi dL}{n\alpha\varepsilon}\right)^{1/2} + \left(\frac{\Phi^2 L^3 d^{3/2}}{n^2\alpha^2\varepsilon}\right)^{1/5}\right).\end{aligned} \tag{8.10}$$

Bounding the latter by β and solving for n completes the proof.

□

8.4 *Multi-pass algorithm*

Algorithm 25: Multi-pass instantiation of $\mathcal{O}(z_t)$ in Line 7 of Algorithm 23

Input: Current iterate z_t, time $t \in \mathbb{N}$, period length $\Sigma \in \mathbb{N}$, accuracy parameter $\alpha > 0$, gradient validation size $m \in \mathbb{N}$, noise levels σ_1, $\sigma_2 > 0$;
if $t \ mod \ \Sigma = 1$ **then**
  for *each sample* $\xi_i \in \mathcal{D}$ **do**
    Sample $y_1, \dots, y_m \overset{iid}{\sim} \mathrm{Unif}(\mathbb{S}^{d-1})$;
    $\tilde{\nabla} f(z_t; \xi_i) = \frac{1}{m} \sum_{j \in [m]} \frac{d}{2\alpha} (f(z_t + \alpha y_j; \xi_i) - f(z_t - \alpha y_j; \xi_i)) y_j$;
  end
  $g_t = \frac{1}{n} \sum_{\xi_i \in \mathcal{D}} \tilde{\nabla} f(z_t; \xi_i)$;
  Return: $\tilde{g}_t = g_t + \chi_t$, where $\chi_t \sim \mathcal{N}(0, \sigma_1^2 I_d)$;
end
else
  for *each sample* $\xi_i \in \mathcal{D}$ **do**
    Sample $y_1, \dots, y_{2m} \overset{iid}{\sim} \mathrm{Unif}(\mathbb{B}_\alpha)$;
    $\tilde{\nabla} f(z_t; \xi_i) = \frac{1}{m} \sum_{j=1}^{m} \frac{d}{2\alpha} (f(z_t + \alpha y_j; \xi_i) - f(z_t - \alpha y_j; \xi_i)) y_j$;
    $\tilde{\nabla} f(z_{t-1}; \xi_i) = \frac{1}{m} \sum_{j=m+1}^{2m} \frac{d}{2\alpha} (f(z_{t-1} + \alpha y_j; \xi_i) - f(z_{t-1} - \alpha y_j; \xi_i)) y_j$;
  end
  $g_t = \tilde{g}_{t-1} + \frac{1}{n} \sum_{\xi_i \in \mathcal{D}} (\tilde{\nabla} f(z_t; \xi_i) - \tilde{\nabla} f(z_{t-1}; \xi_i))$;
  Return: $\tilde{g}_t = g_t + \chi_t$, where $\chi_t \sim \mathcal{N}(0, \sigma_2^2 I_d)$;
end

In this section, we consider a different oracle construction given by Algorithm 25, to be used in Algorithm 23. The main difference from the previous section is that this oracle reuses data points a polynomial number of times, and therefore cannot *directly* guarantee generalization to the stochastic objective. Instead, in this section we analyze the empirical objective $\widehat{F}^{\mathcal{D}}(x) := \frac{1}{n} \sum_{i=1}^{n} f(x; \xi_i)$. After establishing ERM results, in Section 8.5, we will show that any empirical Goldstein-stationarity guarantee generalizes to the population loss.

Similarly to the single-pass oracle (Algorithm 24), we use randomized smoothing and variance reduction. A difference in the oracle construction is that we replace the tree mechanism with the Gaussian mechanism and apply advanced composition for the privacy analysis (since now samples are reused). The main theorem for this section is the following:

Theorem 8.4.1 (Multi-pass ERM)**.** *Suppose* $\widehat{F}^{\mathcal{D}}(x_0) - \inf_x \widehat{F}^{\mathcal{D}}(x) \le \Phi$, *Assumption 8.2.2*

holds, and let $\alpha, \beta, \delta, \varepsilon > 0$ *such that* $\alpha \le \frac{\Phi}{L}$. *Then setting* $m = \frac{L^2 d\Sigma}{n\sigma_1^2} + \frac{L^2 d}{n\sigma_2^2}$, $\sigma_1 = O(\frac{L\sqrt{T\log(1/\delta)/\Sigma}}{n\varepsilon})$, $\sigma_2 = O(\frac{LD\sqrt{Td\log(1/\delta)}}{\alpha n\varepsilon})$, $\Sigma = \tilde{\Theta}(\frac{\alpha}{D\sqrt{d}})$, $D = \tilde{\Theta}(\frac{\alpha^2\beta^2}{L^2}), T = \tilde{\Theta}(\frac{\Phi L^2}{\alpha^2\beta^3})$, *and running Algorithm 23 with Algorithm 25 as the oracle subroutine is* (ε, δ)*-DP. Furthermore, its output satisfies* $\mathbb{E}\,\|\overline{\partial}_{2\alpha}\widehat{F}^{\mathcal{D}}(x^{\text{out}})\| \le \beta$ *as long as*

$$n = \widetilde{\Omega}\left(\frac{\sqrt{\Phi}Ld^{3/4}}{\varepsilon\alpha^{1/2}\beta^{3/2}}\right).$$

Remark 8.4.2. As we will show in Section 8.5, Theorem 8.4.1 also provides the same population guarantee for $\|\overline{\partial}_{2\alpha}F(x^{\text{out}})\|$ with an additional $L^2 d/\beta^2$ term (up to log factors) to the sample complexity.

To prove Theorem 8.4.1, we analyze the properties of the oracle given by Algorithm 25. The sensitivity of g_t in Algorithm 25 directly follows from Lemma 8.3.3.[3] By the standard composition results of the Gaussian mechanism (e.g., [Mir17]), we have the following privacy guarantee:

Lemma 8.4.3 (Privacy)**.** *Calling Algorithm 25* T *times with* $m = \frac{L^2 d\Sigma}{n\sigma_1^2} + \frac{L^2 d}{n\sigma_2^2}$, $\sigma_1 = O(\frac{L\sqrt{T\log(1/\delta)/\Sigma}}{n\varepsilon})$ *and* $\sigma_2 = O(\frac{LD\sqrt{Td\log(1/\delta)}}{\alpha n\varepsilon})$ *is* (ε, δ)*-DP.*

In terms of the oracle's variance, we show:

Lemma 8.4.4 (Variance)**.** *In Algorithm 25, for any* $t \in [T]$*, we have*

$$\mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha^{\mathcal{D}}(z_t)\|^2 \lesssim \frac{L^2 d^2\Sigma}{mn} + \sigma_1^2 d + \sigma_2^2 d\Sigma,$$
$$\mathbb{E}\,\|\tilde{g}_t\|^2 \lesssim L^2 + \frac{L^2 d^2\Sigma}{mn} + \sigma_1^2 d + \sigma_2^2 d\Sigma.$$

The proof of Theorem 8.4.1, which we defer to Section 8.6, is a combination of the two previous lemmas and Proposition 8.2.6.

[3] In this section we use full-batch size for simplicity, but using smaller batches (of arbitrary size) and applying privacy amplification by subsampling, yields the same results up to constants.

8.5 Empirical to population Goldstein-stationarity

In this section, we provide a generalization result, showing that our ERM algorithm from the previous section also guarantees Goldstein-stationarity in terms of the population loss. We prove the following more general statement:

Proposition 8.5.1. *Under Assumption 8.2.2, suppose $\mathcal{D} \sim \mathcal{P}^n$, and consider running an algorithm on $\widehat{F}^{\mathcal{D}}$ whose (possibly randomized) output $x^{\text{out}} \in \mathcal{X} \subset \mathbb{R}^d$ is supported over a set $\mathcal{X}$ of diameter $\leq R$. Then with probability at least $1-\zeta$: $\|\overline{\partial}_\alpha F(x^{\text{out}})\| \leq \|\overline{\partial}_\alpha \widehat{F}^{\mathcal{D}}(x^{\text{out}})\| + \widetilde{O}\left(L\sqrt{d\log(R/\zeta)/n}\right)$.*

We remark that in all algorithms of interest, the output is known to lie in some predefined set, such as a sufficiently large ball around the initialization. As long as the diameter R is polynomial in the problem parameters, the $\log(R)$ in the result above is therefore negligible. For instance, Algorithm 23 is easily verified to output a point $x^{\text{out}} \in \mathbb{B}(x_0, DT)$ (since $\|x_{t+1} - x_t\| \leq D$). Hence, in our use case, Proposition 8.5.1 ensures $\|\overline{\partial}_{2\alpha} F(x^{\text{out}})\| \leq \|\overline{\partial}_{2\alpha}\widehat{F}^{\mathcal{D}}(x^{\text{out}})\| + \beta$ for $n = \widetilde{O}(d/\beta^2)$.

8.6 Proofs

8.6.1 Proofs from Section 8.3

Proof of Lemma 8.3.3. Note that for any $y \in \text{Unif}(\mathbb{S}^{d-1})$: $\|\frac{d}{2\alpha}(f(z+\alpha y;\xi) - f(z-\alpha y;\xi))y\| \leq Ld$ due to the Lipschitz assumption. Hence, for any $\xi \in S_t$, by a standard sub-Gaussian bound (Theorem 8.11.2) we have

$$\Pr\left[\|\tilde{\nabla} f(z_t;\xi) - \nabla f_\alpha(z_t;\xi)\| \leq \frac{Ld\sqrt{\log(8dB_1/\delta)}}{\sqrt{m}}\right] \geq 1 - \delta/8B_1. \tag{8.11}$$

If $t \bmod \Sigma = 1$, then

$$\begin{aligned}\|g_t - g'_t\| &= \left\|\frac{1}{B_1}\Big(\sum_{\xi\in S_t}\tilde{\nabla} f(z_t;\xi) - \sum_{\xi'\in S'_t}\tilde{\nabla} f(z_t;\xi')\Big)\right\| \\ &\leq \left\|\frac{1}{B_1}\Big(\sum_{\xi\in S_t}\tilde{\nabla} f(z_t;\xi) - \nabla f_\alpha(z_t;\xi)\Big)\right\| + \left\|\frac{1}{B_1}\Big(\sum_{\xi\in S_t}\nabla f_\alpha(z_t;\xi) - \sum_{\xi'\in S'_t}\nabla f_\alpha(z_t;\xi')\Big)\right\|\end{aligned}$$

$$+ \left\| \frac{1}{B_1} \Big(\sum_{\xi' \in S_t'} \tilde{\nabla} f(z_t; \xi') - \nabla f_\alpha(z_t; \xi') \Big) \right\|.$$

Further note that $\| \frac{1}{B_1} (\sum_{\xi \in S_t} \nabla f_\alpha(z_t; \xi) - \sum_{\xi' \in S_t'} \nabla f_\alpha(z_t; \xi')) \| \leq 2L/B_1$, hence by Equation (8.11) and the union bound,

$$\Pr \left[\left\| \frac{1}{B_1} \Big(\sum_{\xi \in S_t} \tilde{\nabla} f(z_t; \xi) - \sum_{\xi' \in S_t'} \tilde{\nabla} f(z_t; \xi') \Big) \right\| \geq \frac{Ld\sqrt{\log(8dB_1/\delta)}}{\sqrt{m}} \right] \leq 1 - \delta/8,$$

which proves the claim in the case when $t \bmod \Sigma = 1$. The other case follows from the same argument. □

Proof of Lemma 8.3.4. By Lemma 8.3.3 and our assignment of m, we know that with probability at least $1 - \delta/2$, the sensitivity of all t is bounded by $O(\frac{L}{B_1} + \frac{L\sqrt{d}D}{\alpha B_2})$, namely for all t :

$$\|g_t - g_t'\| \lesssim \frac{L}{B_1} + \frac{L\sqrt{d}D}{\alpha B_2}.$$

Then the privacy guarantee follows from the Tree Mechanism (Proposition 8.2.5). □

Proof of Lemma 8.3.5. First, note that by Proposition 8.2.5 and the facts that $\mathbb{E}[g_t] = \nabla F_\alpha(z_t)$ and $\|\nabla F_\alpha(z_t)\| \leq L$, we get

$$\mathbb{E} \|\tilde{g}_t\|^2 \lesssim \mathbb{E} \|g_t\|^2 + d\sigma^2 \log \Sigma \lesssim \mathbb{E} \|g_t - \nabla F_\alpha(z_t)\|^2 + L^2 + d\sigma^2 \log \Sigma,$$

and also

$$\mathbb{E} \|\tilde{g}_t - \nabla F_\alpha(z_t)\|^2 \lesssim \mathbb{E} \|\tilde{g}_t - g_t\|^2 + \mathbb{E} \|g_t - \nabla F_\alpha(z_t)\|^2 \lesssim d\sigma^2 \log \Sigma + \mathbb{E} \|g_t - \nabla F_\alpha(z_t)\|^2.$$

Therefore, we see that in order to obtain both claimed bounds, it suffices to bound $\mathbb{E} \|g_t - \nabla F_\alpha(z_t)\|^2$. To that end, denote by $t_0 \leq t$ the largest integer such that $t_0 \bmod \Sigma = 1$,

and note that $t - t_0 < \Sigma$. Further denote $\Delta_j := g_j - g_{j-1}$. Then we have

$$\begin{aligned}\mathbb{E}\left\|g_t - \nabla F_\alpha(z_t)\right\|^2 &= \mathbb{E}\left\|g_{t_0} + \sum_{j=t_0+1}^{t} \Delta_j - \left(\sum_{j=t_0+1}^{t} (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1})) + \nabla F_\alpha(z_{t_0})\right)\right\|^2 \\ &= \underbrace{\mathbb{E}\left\|g_{t_0} - \nabla F_\alpha(z_{t_0})\right\|^2}_{(I)} + \sum_{j=t_0}^{t} \underbrace{\mathbb{E}\left\|\Delta_j - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2}_{(II)},\end{aligned} \tag{8.12}$$

where the last equality is due to the cross terms having zero mean. We further see that

$$\begin{aligned}(I) &\lesssim \mathbb{E}\left\|g_{t_0} - \frac{1}{B_1}\sum_{\xi\in S_{t_0}} \nabla f_\alpha(z_{t_0};\xi_i)\right\|^2 + \mathbb{E}\left\|\frac{1}{B_1}\sum_{\xi\in S_{t_0}} \nabla f_\alpha(z_{t_0};\xi_i) - \nabla F_\alpha(z_{t_0})\right\|^2 \\ &\lesssim \frac{L^2 d^2}{B_1 m} + \frac{L^2}{B_1},\end{aligned} \tag{8.13}$$

as well as

$$\begin{aligned}(II) &= \mathbb{E}\left\|\frac{1}{B_2}\sum_{\xi\in S_t} (\tilde{\nabla} f(z_j;\,\xi) - \tilde{\nabla} f(z_{j-1};\,\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2 \\ &= \frac{1}{B_2^2}\sum_{\xi\in S_t} \mathbb{E}\left\|(\tilde{\nabla} f(z_j;\,\xi) - \tilde{\nabla} f(z_{j-1};\,\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2 \\ &\lesssim \frac{1}{B_2^2}\sum_{\xi\in S_t} \Big(\mathbb{E}\left\|\tilde{\nabla} f(z_j;\,\xi) - \nabla f_\alpha(z_j;\,\xi)\right\|^2 + \mathbb{E}\left\|\tilde{\nabla} f(z_{j-1};\,\xi) - \nabla f_\alpha(z_{j-1};\,\xi)\right\|^2 \\ &\qquad + \mathbb{E}\left\|(\nabla f_\alpha(z_j;\,\xi) - \nabla f_\alpha(z_{j-1};\,\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2\Big) \\ &\lesssim \frac{L^2 d^2}{m B_2} + \frac{d L^2 D^2}{\alpha^2 B_2}.\end{aligned} \tag{8.14}$$

Plugging (8.13) and (8.14) into (8.12) and recalling that $t - t_0 < \Sigma$ completes the proof. □

8.6.2 Proofs from Section 8.4

Proof of Lemma 8.4.4. First, it suffices to prove the first bound, as

$$\mathbb{E}\left\|\tilde{g}_t\right\|^2 \lesssim \mathbb{E}\left\|\tilde{g}_t - \nabla F_\alpha^{\mathcal{D}}(z_t)\right\|^2 + \mathbb{E}\left\|\nabla F_\alpha^{\mathcal{D}}(z_t)\right\|^2 \le \mathbb{E}\left\|\tilde{g}_t - \nabla F_\alpha^{\mathcal{D}}(z_t)\right\|^2 + L^2.$$

To that end, let $t_0 \le t$ be the largest integer such that $t_0 \bmod \Sigma \equiv 1$, and note that $t - t_0 < \Sigma$. Define $\Delta_j := \frac{1}{n}\sum_{\xi\in\mathcal{D}}(\tilde{\nabla} f(z_j;\xi) - \tilde{\nabla} f(z_{j-1};\xi))$. It holds that

$$\mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha^{\mathcal{D}}(z_t)\|^2 \le \underbrace{\mathbb{E}\,\|g_{t_0} - \nabla F_\alpha^{\mathcal{D}}(z_{t_0})\|^2}_{(I)} + \sum_{j=t_0}^{t} \underbrace{\mathbb{E}\,\|\Delta_j - (\nabla F_\alpha^{\mathcal{D}}(z_j) - \nabla F_\alpha^{\mathcal{D}}(z_{j-1}))\|^2}_{(II)} + \underbrace{\sum_{j=t_0}^{t}\mathbb{E}\,\|\chi_j\|^2}_{(III)}.$$

Similar to the proof of Lemma 8.3.5, we have that

$$\begin{aligned}
(I) &= \mathbb{E}\left\|g_{t_0} - \frac{1}{n}\sum_{\xi\in\mathcal{D}}\nabla f_\alpha(z_{t_0};\xi_i)\right\|^2 \lesssim \frac{L^2d^2}{nm},\\
(II) &= \frac{1}{n^2}\,\mathbb{E}\,\|\sum_{\zeta\in\mathcal{D}}(\tilde{\nabla} f(z_j;\xi) - \tilde{\nabla} f(z_{j-1};\xi)) - (\nabla \widehat{F}_\alpha^{\mathcal{D}}(z_j) - \nabla \widehat{F}_\alpha^{\mathcal{D}}(z_{j-1}))\|^2\\
&\lesssim \frac{1}{n^2}\sum_{\xi\in\mathcal{D}}\Big(\mathbb{E}\,\|\tilde{\nabla} f(z_j;\xi) - \nabla f_\alpha(z_j;\xi)\|^2 + \mathbb{E}\,\|\tilde{\nabla} f(z_{j-1};\xi) - \nabla f_\alpha(z_{j-1};\xi)\|^2\Big)\\
&\lesssim \frac{L^2d^2}{mn},\\
(III) &\le \sigma_1^2 d + \sigma_2^2(\Sigma - 1),
\end{aligned}$$

overall completing the proof.

□

Proof of Theorem 8.4.1. Setting $m = \frac{L^2d\Sigma}{n\sigma_1^2} + \frac{L^2d}{n\sigma_2^2}$, $\sigma_1 = O(\frac{L\sqrt{T\log(1/\delta)/\Sigma}}{n\varepsilon})$ and $\sigma_2 = O(\frac{LD\sqrt{Td\log(1/\delta)}}{\alpha n\varepsilon})$, the privacy guarantee follows from Lemma 8.4.3. Moreover, by our parameter settings, we have

$$\begin{aligned}
G_0^2 := \mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha^{\mathcal{D}}(z_t)\|^2 &\lesssim \frac{L^2dT\log(1/\delta)/\Sigma}{n^2\varepsilon^2} + \frac{L^2D^2Td^2\Sigma\log(1/\delta)}{\alpha^2n^2\varepsilon^2},\\
G_1^2 := \mathbb{E}\,\|\tilde{g}_t\|^2 &\lesssim L^2 + \frac{L^2dT\log(1/\delta)/\Sigma}{n^2\varepsilon^2} + \frac{L^2D^2Td^2\Sigma\log(1/\delta)}{\alpha^2n^2\varepsilon^2}.
\end{aligned}$$

Therefore, setting $\Sigma = \widetilde{\Theta}(\frac{\alpha}{D\sqrt{d}})$, we see that $G_0 = \widetilde{O}(\frac{L\sqrt{DT}d^{3/4}}{n\varepsilon\sqrt{\alpha}})$ and $G_1 \lesssim L + G_0$. By Proposition 8.2.6, we also know that

$$\mathbb{E}\,\|\overline{\partial}_{2\alpha}\widehat{F}^{\mathcal{D}}(x^{\text{out}})\| \le \mathbb{E}\,\|\overline{\partial}_{\alpha}\widehat{F}_\alpha^{\mathcal{D}}(x^{\text{out}})\| \le \frac{F_\alpha(x_0) - \inf F_\alpha}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0$$

$$\leq \frac{2\Phi}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0.$$

Recalling that $M = \Theta(\alpha/D)$ and setting $D = \tilde{\Theta}(\frac{\alpha^2\beta^2}{L^2})$, $T = \tilde{\Theta}(\frac{\Phi L^2}{\alpha^2\beta^3})$, we have

$$\begin{aligned}\mathbb{E}\left\|\overline{\partial}_\alpha \widehat{F}^{\mathcal{D}}_\alpha(x^{\text{out}})\right\| &= \widetilde{O}\left(\frac{\Phi}{DT} + \frac{L\sqrt{D}}{\sqrt{\alpha}} + \frac{L\sqrt{DT}d^{3/4}}{n\varepsilon\sqrt{\alpha}}\right) \\ &= \frac{\beta}{2} + \widetilde{O}\left(\frac{Ld^{3/4}\sqrt{\Phi}}{n\varepsilon\sqrt{\alpha\beta}}\right).\end{aligned}$$

The latter is bounded by β for $n = \widetilde{\Omega}\left(\frac{L\sqrt{\Phi}d^{3/4}}{\varepsilon\alpha^{1/2}\beta^{3/2}}\right)$, hence completing the proof.

□

8.6.3 Proofs from Section 8.5

Proof of Proposition 8.5.1. Applying a gradient uniform convergence bound for Lipschitz objectives over a bounded domain [MBM18, Theorem 1], shows that with probability at least $1-\zeta$, for any differentiable $x \in \mathcal{X}$:

$$\left\|\nabla \widehat{F}^{\mathcal{D}}(x) - \nabla F(x)\right\| = \widetilde{O}\left(L\sqrt{\frac{d\log(R/\zeta)}{n}}\right). \tag{8.15}$$

Therefore, given any $x \in \mathcal{X}$, let $y_1, \ldots, y_k \in \mathbb{B}(x, \alpha)$ be points satisfying $\overline{\partial}_\alpha \widehat{F}^{\mathcal{D}}(x) = \sum_{i=1}^k \lambda_i \nabla \widehat{F}^{\mathcal{D}}(y_i)$ for coefficients $(\lambda_i)_{i=1}^k \geq 0, \sum_{i=1}^k \lambda_i = 1$ — note that such points exist by definition of the Goldstein subdifferential. Noting that $\sum_{i=1}^k \lambda_i \nabla F(y_i) \in \partial_\alpha F(x)$, and recalling that $\overline{\partial}_\alpha F(x)$ is the minimal norm element of $\partial_\alpha F(x)$, we get that

$$\left\|\overline{\partial}_\alpha F(x)\right\| \leq \left\|\sum_{i=1}^k \lambda_i \nabla F(y_i)\right\| = \left\|\sum_{i=1}^k \lambda_i (\nabla \widehat{F}^{\mathcal{D}}(y_i) + \upsilon_i)\right\| = (\star)$$

where $\upsilon_i := \nabla F(y_i) - \nabla \widehat{F}^{\mathcal{D}}(y_i)$ satisfy $\|\upsilon_i\| = \widetilde{O}\left(L\sqrt{\frac{d\log(R/\zeta)}{n}}\right)$ for all $i \in [k]$ by (8.15). Hence

$$(\star) \leq \left\|\sum_{i=1}^k \lambda_i \nabla \widehat{F}^{\mathcal{D}}(y_i)\right\| + \left\|\sum_{i=1}^k \lambda_i \upsilon_i\right\|$$

$$\leq \left\|\overline{\partial}_\alpha \widehat{F}^{\mathcal{D}}(x)\right\| + \sum_{i=1}^{k} \lambda_i \left\|\upsilon_i\right\|$$
$$\leq \left\|\overline{\partial}_\alpha \widehat{F}^{\mathcal{D}}(x)\right\| + \widetilde{O}\left(L\sqrt{\frac{d\log(R/\zeta)}{n}}\right).$$

□

8.7 Discussion

In this paper, we studied nonsmooth nonconvex optimization, and proposed differentially private algorithms for this task which return Goldstein-stationary points, improving the previously known sample complexity for this task.

Our single-pass algorithm reduces the sample complexity by at least a $\Omega(\sqrt{d})$ factor (and sometimes more, depending on the parameter regime of interest), compared to the previous such result by [ZTC24]. Furthermore, our result has a dimension-independent "non-private" term, which was previously claimed impossible. Moreover, we propose a multi-pass algorithm which preforms sample-efficient ERM, and show that it further generalizes to the population.

It is interesting to note that our guarantees are in terms of so-called "approximate" (ε, δ)-DP, whereas [ZTC24] derive a Rényi-DP guarantee [Mir17]. This is in fact inherent to our techniques, since we condition on a highly probable event in order to substantially decrease the effective sensitivity of our gradient estimators. Further examining this potential gap in terms of sample complexity between approximate- and Rényi-DP for nonsmooth nonconvex optimization is an interesting direction for future research.

Another important problem that remains open is establishing tight lower bounds for DP nonconvex optimization and perhaps further improving the sample complexities obtained in this paper. We note that the current upper and lower bounds do not fully match even in the smooth setting. In Appendix 8.10, we provide evidence that our upper bound can be further improved, by proposing a computationally-*inefficient* algorithm, which converges to a relaxed notion of stationarity, using even fewer samples than the algorithms we presented in this work.

8.8 *Proof of Proposition 8.2.6 (O2NC)*

We start by noting that the update rule for Δ_t which is given by

$$\Delta_{t+1} = \mathrm{clip}_D\left(\Delta_t - \eta\tilde{g}_t\right) = \min\left\{1, \tfrac{D}{\|\Delta_t - \eta\tilde{g}_t\|}\right\} \cdot \left(\Delta_t - \eta\tilde{g}_t\right)$$

is precisely the online project gradient descent update rule, with respect to linear losses of the form $\ell_t(\cdot) = \langle \tilde{g}_t, \cdot \rangle$, over the ball of radius D around the origin. Accordingly, recalling that $\mathbb{E}\,\|\tilde{g}_t - \nabla h(z_t)\|^2 \le G_1^2$, combining the linearity of expectation with the standard regret analysis of online linear optimization (cf. Haz16) gives the following:

Lemma 8.8.1. *By setting $\eta = \frac{D}{G_1\sqrt{M}}$, for any $u \in \mathbb{R}^d$ with $\|u\| \le D$ it holds that*

$$\mathop{\mathbb{E}}_{\tilde{g}_1,\ldots,\tilde{g}_M}\left[\sum_{m=1}^{M}\langle \tilde{g}_m, \Delta_m - u\rangle\right] \le \tfrac{3}{2}DG_1\sqrt{M}.$$

Back to analyzing Algorithm 23, since $x_t = x_{t-1} + \Delta_t$ it holds that

$$\begin{aligned} h(x_t) - h(x_{t-1}) &= \int_0^1 \langle \nabla h(x_{t-1} + s\Delta_t), \Delta_t\rangle\, ds \\ &= \mathop{\mathbb{E}}_{s_t \sim \mathrm{Unif}[0,1]}\left[\langle \nabla h(x_{t-1} + s_t\Delta_t), \Delta_t\rangle\right] = \mathop{\mathbb{E}}_{s_t}\left[\langle \nabla h(z_t), \Delta_t\rangle\right]. \end{aligned}$$

Note that $\langle \nabla h(z_t), \Delta_t\rangle = \langle \nabla h(z_t), u\rangle + \langle \tilde{g}_t, \Delta_t - u\rangle + \langle \nabla h(z_t) - \tilde{g}_t, \Delta_t - u\rangle$, so by summing over $t \in [T] = [K \times M]$, we get for any fixed sequence $u_1, \ldots, u_K \in \mathbb{R}^d$:

$$\begin{aligned} \inf h \le h(x_T) &\le h(x_0) + \sum_{t=1}^{T}\mathbb{E}\left[\langle \nabla h(z_t), \Delta_t\rangle\right] \\ &= h(x_0) + \sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\left[\left\langle \tilde{g}_{(k-1)M+m}, \Delta_{(k-1)M+m} - u_k\right\rangle\right] \\ &\quad + \sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\left[\left\langle \nabla h(z_{(k-1)M+m}), u_k\right\rangle\right] + \sum_{t=1}^{T}\mathbb{E}[\langle \nabla h(z_t) - \tilde{g}_t, \Delta_t - u\rangle] \\ &\le h(x_0) + \tfrac{3}{2}KDG_1\sqrt{M} + \sum_{k=1}^{K}\sum_{m=1}^{M}\mathbb{E}\left[\left\langle \nabla h(z_{(k-1)M+m}), u_k\right\rangle\right] + G_0DT, \end{aligned}$$

where the last inequality follows from applying Lemma 8.8.1 to each M consecutive iterates, and combining the bias bound $\mathbb{E}\,\|\tilde{g}_t - \nabla h(z_t)\| \le G_0$ with Cauchy-Schwarz.

Letting $u_k := -D\frac{\sum_{m=1}^{M}\nabla h(z_{(k-1)M+m})}{\left\|\sum_{m=1}^{M}\nabla h(z_{(k-1)M+m})\right\|}$, rearranging and dividing by $DT = DKM$, we obtain

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla h(z_{(k-1)M+m})\right\| \le \frac{h(x_0)-\inf h}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0. \tag{8.16}$$

Finally, note that for all $k \in [K], m \in [M] : \left\|z_{(k-1)M+m} - \overline{x}_k\right\| \le MD \le \alpha$ since the clipping operation ensures each iterate is at most of distance D to its predecessor, and therefore $\nabla h(z_{(k-1)M+m}) \in \partial_\alpha h(\overline{x}_k)$. Since the set $\partial_\alpha h(\cdot)$ is convex by definition, we further see that

$$\frac{1}{M}\sum_{m=1}^{M}\nabla h(z_{(k-1)M+m}) \in \partial_\alpha h(\overline{x}_k)\ ,$$

and hence by (8.16) we get

$$\mathbb{E}\left\|\overline{\partial}_\alpha h(x^{\mathrm{out}})\right\| = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\overline{\partial}_\alpha h(\overline{x}_k)\right\| \le \frac{h(x_0)-\inf h}{DT} + \frac{3G_1}{2\sqrt{M}} + G_0.$$

8.9 First-order algorithm

In this appendix, our goal is to show that the zero-order algorithms presented in the main text can be replaced by first-order algorithms with the sample complexity, and reduced oracle complexity.

The simple idea is to replace the zero-order gradient estimator from Eq. (8.4) by the first-order estimator

$$\tilde{\nabla} f_\alpha(x;\xi) = \frac{1}{m}\sum_{j=1}^{m}\nabla f(x+\alpha y_j;\xi), \quad y_1,\dots,y_m \overset{iid}{\sim} \mathrm{Unif}(\mathbb{S}^{d-1}). \tag{8.17}$$

While this estimator has the same expectation as the zero-order variant, the key difference lies in the fact that its subGaussian norm is substantially smaller (as it does not depend on d), hence smaller m suffices for concentration. This observation enables reducing the oracle

complexity, while ensuring the same sample complexity guarantee as in the main text.

We fully analyze here a single-pass first-order oracle, presented in Algorithm 26, which can be used in Algorithm 23, similarly to Section 8.3. We note that a similar analysis can be applied to the multi-pass oracle of Section 8.4, once again by replacing (8.4) by (8.17).

As in Section 8.3, we will present the basic properties of this oracle. We will then plug these into Algorithm 23, leading to the main result of this section, Theorem 8.9.4.

Algorithm 26: First-order instantiation of $\mathcal{O}(z_t)$ in Line 7 of Algorithm 23

Input: Current iterate z_t, time $t \in \mathbb{N}$, period length $\Sigma \in \mathbb{N}$, accuracy parameter $\alpha > 0$, batch sizes $B_1, B_2 \in \mathbb{N}$, gradient validation size $m \in \mathbb{N}$, noise level $\sigma > 0$.;
if $t \ mod \ \Sigma = 1$ **then**
  Sample minibatch S_t of size B_1 among unused samples;
  Sample $y_1, \dots, y_{B_1} \overset{iid}{\sim} \mathrm{Unif}(\mathbb{B}_\alpha)$;
  $g_t = \frac{1}{B_1} \sum_{\xi_i \in S_t} \nabla f(z_t + y_i; \xi_i)$;
end
else
  Sample minibatch S_t of size B_2 among unused samples;
  for *each sample* $\xi_i \in S_t$ **do**
    Sample $y_1, \dots, y_{2m} \overset{iid}{\sim} \mathrm{Unif}(\mathbb{B}_\alpha)$;
    $\tilde{\nabla} f(z_t; \xi_i) = \frac{1}{m} \sum_{j=1}^{m} \nabla f(z_t + y_j; \xi_i)$;
    $\tilde{\nabla} f(z_{t-1}; \xi_i) = \frac{1}{m} \sum_{j=m+1}^{2m} \nabla f(z_{t-1} + y_j; \xi_i)$;
  end
  $g_t = g_{t-1} + \frac{1}{B_2} \sum_{\xi_i \in S_t} (\tilde{\nabla} f(z_t; \xi_i) - \tilde{\nabla} f(z_{t-1}; \xi_i))$
end
Return $\tilde{g}_t = g_t + \mathrm{TREE}(\sigma, \Sigma)(t \bmod \Sigma)$;

Lemma 8.9.1 (Sensitivity)**.** *Consider the gradient oracle $\mathcal{O}(\cdot)$ in Algorithm 26 when acting on two neighboring minibatches S_t and S_t', and correspondingly producing g_t and g_t', respectively. If $t \bmod \Sigma = 1$, then*

$$\|g_t - g_t'\| \leq \frac{L}{B_1}.$$

Otherwise, conditioned on $g_{t-1} = g'_{t-1}$, we have with probability at least $1 - \delta/2$:

$$\|g_t - g'_t\| \lesssim \frac{L\sqrt{d}D}{\alpha B_2} + \frac{L\sqrt{\log(dB_2/\delta)}}{\sqrt{m}}.$$

With the sensitivity bound given by Lemma 8.9.1, we easily derive the privacy guarantee of our algorithm from the Tree Mechanism (Proposition 8.2.5).

Lemma 8.9.2 (Privacy)**.** *Running Algorithm 26 with $m = O(\log(dB_2/\delta)\frac{B_2^2\alpha^2}{D^2 d})$ and $\sigma = O(\frac{L\sqrt{\log(1/\delta)}}{B_1\varepsilon} + \frac{LD\sqrt{d\log(1/\delta)}}{\alpha B_2\varepsilon})$ is (ε, δ)-DP.*

Proof. By Lemma 8.9.1 and our assignment of m, we know that with probability at least $1 - \delta/2$, for any t, we have

$$\|g_t - g'_t\| \lesssim \frac{L}{B_1} + \frac{L\sqrt{d}D}{\alpha B_2}.$$

Then the privacy guarantee follows from the Tree Mechanism (Proposition 8.2.5).

□

We next provide the required variance bound on the gradient oracle.

Lemma 8.9.3 (Variance)**.** *In Algorithm 26, for all t it holds that*

$$\mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha(z_t)\|^2 \lesssim \frac{L^2}{B_1} + \frac{L^2 dD^2\Sigma}{\alpha^2 B_2} + \sigma^2 d\log\Sigma + \frac{L^2\Sigma}{mB_2},$$
$$\mathbb{E}\,\|\tilde{g}_t\|^2 \lesssim L^2 + \frac{L^2 dD^2\Sigma}{\alpha^2 B_2} + \sigma^2 d\log\Sigma + \frac{L^2\Sigma}{mB_2}.$$

Having set up the required bounds, we can prove our main result for the first-order setting.

Theorem 8.9.4 (First-order)**.** *Suppose $F(x_0) - \inf_x F(x) \le \Phi$, that Assumption 8.2.2 holds, and let $\alpha, \beta, \delta, \varepsilon > 0$ such that $\alpha \le \frac{\Phi}{L}$. Then setting $B_1 = \Sigma$, $B_2 = 1$, $M = \alpha/4D$, $m = \widetilde{O}(\frac{B_2^2\alpha^2}{D^2 d})$, $\sigma = \widetilde{O}(\frac{L}{B_1\varepsilon} + \frac{LD\sqrt{d}}{\alpha B_2\varepsilon})$, $\Sigma = \widetilde{\Theta}((\frac{\alpha}{\varepsilon D})^{2/3})$, $D = \widetilde{\Theta}(\min\{(\frac{\Phi^2\alpha}{L^2T^2})^{1/3}, (\frac{\Phi\alpha\varepsilon}{dLT})^{1/2}, (\frac{\Phi^3\alpha^2\varepsilon}{d^{3/2}L^3T^3})^{1/5}\})$, $T = \Theta(n)$, and running Algorithm 23 with Algorithm 26 as the oracle subroutine, is (ε, δ)-DP.*

Furthermore, its output satisfies $\mathbb{E}\,\|\overline{\partial}_{2\alpha}F(x^{\mathrm{out}})\| \le \beta$ *as long as*

$$n = \widetilde{\Omega}\left(\frac{\Phi L^2}{\alpha\beta^3} + \frac{\Phi L d}{\varepsilon\alpha\beta^2} + \frac{\Phi L^{3/2} d^{3/4}}{\varepsilon^{1/2}\alpha\beta^{5/2}}\right).$$

Remark 8.9.5 (Oracle complexity). Compared to the zero-order result given by Theorem 8.3.1, we see that the number of calls to $\mathcal{O}(\,\cdot\,)$, namely T, is on the same order, and that in both cases the amortized oracle complexity of $\mathcal{O}(\,\cdot\,)$ is $O(m)$. The difference between the settings is that the first-order oracle instantiation sets m to be $\widetilde{\Omega}(d^2)$ times smaller than its zero-order counterpart, and hence we gain this multiplicative factor in the overall oracle complexity.

Proof of Theorem 8.9.4. The privacy guarantee follows directly from Lemma 8.9.2, by noting that our parameter assignment implies $B_1 T/\Sigma + B_2 T = O(n)$, hence it allows letting $T = \Theta(n)$ while never re-using samples.

As to the sample complexity, note that our parameter assignment ensures that

$$\begin{aligned} G_1 &= O\left(G_0 + L\right), \\ G_0 &= \widetilde{O}\left(\frac{LDd^{1/2}\Sigma^{1/2}}{\alpha} + \frac{Ld^{1/2}}{\Sigma\varepsilon} + \frac{LDd}{\alpha\varepsilon}\right), \end{aligned}$$

similarly to (8.7) and (8.9) in the proof of Theorem 8.3.1. The rest of the proof is therefore exactly the same as for Theorem 8.3.1. □

8.9.1 Proofs from Appendix 8.9

Proof of Lemma 8.9.1. The case when $t \bmod \Sigma = 1$ trivially follows the Lipschitz assumption. Thus we will consider the more involved case. For any $\xi \in S_t$, by a standard sub-Gaussian bound (Theorem 8.11.2) we have

$$\Pr\left[\|\tilde{\nabla} f(z_t;\,\xi) - \nabla f_\alpha(z_t;\,\xi)\| \le \frac{L\sqrt{\log(8dB_2/\delta)}}{\sqrt{m}}\right] \ge 1 - \delta/8B_2,$$

so by the union bound, we get that with probability at least $1 - \delta/8$, for all $\xi_i \in S_t$:

$$\|\tilde{\nabla} f(z_t; \xi) - \nabla f_\alpha(z_t; \xi)\| \leq \frac{L\sqrt{\log(8dB_2/\delta)}}{\sqrt{m}}. \tag{8.18}$$

Hence,

$$\begin{aligned}
\|g_t - g_t'\| \leq & \left\| \frac{1}{B_2} \sum_{\xi \in S_t} \Big((\tilde{\nabla} f(z_t; \xi) - \tilde{\nabla} f(z_{t-1}; \xi_i)) - (\nabla f_\alpha(z_t; \xi)) - \nabla f_\alpha(z_{t-1}; \xi)) \Big) \right\| \\
& + \left\| \frac{1}{B_2} \sum_{\xi \in S_t} \Big((\nabla f_\alpha(z_t; \xi) - \nabla f_\alpha(z_{t-1}; \xi)) - \sum_{\xi' \in S_t'} (\nabla f_\alpha(z_t; \xi') - \nabla f_\alpha(z_t; \xi')) \Big) \right\| \\
& + \left\| \frac{1}{B_2} \sum_{\xi' \in S_t'} \Big((\tilde{\nabla} f(z_t; \xi') - \tilde{\nabla} f(z_{t-1}; \xi')) - (\nabla f_\alpha(z_t; \xi') - \nabla f_\alpha(z_{t-1}; \xi')) \Big) \right\| \\
\lesssim & \frac{L\sqrt{d}D}{\alpha B_2} + \frac{L\sqrt{\log(dB_2/\delta)}}{\sqrt{m}},
\end{aligned}$$

where the last inequality step is due to the smoothness of f_α (Fact 8.2.3) combined with the fact that $\|z_t - z_{t-1}\| \leq 2D$, and (8.18).

□

Proof of Lemma 8.9.3. Applying by Proposition 8.2.5, we have

$$\mathbb{E}\,\|\tilde{g}_t - \nabla F_\alpha(z_t)\|^2 \lesssim \mathbb{E}\,\|\tilde{g}_t - g_t\|^2 + \mathbb{E}\,\|g_t - \nabla F_\alpha(z_t)\|^2 \lesssim d\sigma^2 \log \Sigma + \mathbb{E}\,\|g_t - \nabla F_\alpha(z_t)\|^2,$$

and also since $\mathbb{E}[g_t] = \nabla F_\alpha(z_t)$ and $\|\nabla F_\alpha(z_t)\| \leq L$, we have

$$\mathbb{E}\,\|\tilde{g}_t\|^2 \lesssim \mathbb{E}\,\|g_t\|^2 + d\sigma^2 \log \Sigma \lesssim \mathbb{E}\,\|g_t - \nabla F_\alpha(z_t)\|^2 + L^2 + d\sigma^2 \log \Sigma.$$

We therefore see that both claimed bounds will follow from bounding $\mathbb{E}\,\|g_t - \nabla F_\alpha(z_t)\|^2$.

To that end, denote by $t_0 \leq t$ the largest integer such that $t_0 \bmod \Sigma = 1$, and note that

$t - t_0 < \Sigma$. Further denote $\Delta_j := g_j - g_{j-1}$. Then we have

$$\begin{aligned}
\mathbb{E}\left\|g_t - \nabla F_\alpha(z_t)\right\|^2 &= \mathbb{E}\left\|g_{t_0} + \sum_{j=t_0+1}^{t} \Delta_j - \Big(\sum_{j=t_0+1}^{t} (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1})) + \nabla F_\alpha(z_{t_0})\Big)\right\|^2 \\
&= \mathbb{E}\left\|g_{t_0} - \nabla F_\alpha(z_{t_0})\right\|^2 + \sum_{j=t_0}^{t} \mathbb{E}\left\|\Delta_j - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2, \\
&\lesssim \frac{L^2}{B_1} + \sum_{j=t_0}^{t} \underbrace{\mathbb{E}\left\|\Delta_j - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2}_{(\star)} \qquad (8.19)
\end{aligned}$$

where the second equality is due to the cross terms having zero mean. Moreover, we have

$$\begin{aligned}
(\star) &= \mathbb{E}\left\|\frac{1}{B_2}\sum_{\xi\in S_t} (\tilde{\nabla} f(z_j;\xi) - \tilde{\nabla} f(z_{j-1};\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2 \\
&= \frac{1}{B_2^2}\sum_{\xi\in S_t} \mathbb{E}\left\|(\tilde{\nabla} f(z_j;\xi) - \tilde{\nabla} f(z_{j-1};\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|^2 \\
&\lesssim \frac{1}{B_2^2}\sum_{\xi\in S_t} \Big(\mathbb{E}\left\|\tilde{\nabla} f(z_j;\xi) - \nabla f_\alpha(z_j;\xi)\right\|^2 + \mathbb{E}\left\|\tilde{\nabla} f(z_{j-1};\xi) - \nabla f_\alpha(z_{j-1};\xi)\right\|^2 \\
&\qquad + \mathbb{E}\left\|(\nabla f_\alpha(z_j;\xi) - \nabla f_\alpha(z_{j-1};\xi)) - (\nabla F_\alpha(z_j) - \nabla F_\alpha(z_{j-1}))\right\|\Big)^2 \\
&\lesssim \frac{L^2}{mB_2} + \frac{dL^2D^2}{\alpha^2 B_2},
\end{aligned}$$

which plugged into (8.19) completes the proof by recalling that $t - t_0 \le \Sigma$.

□

8.10 Even better sample complexity via optimal smoothing

In this Appendix, our aim is to provide evidence that the sample complexities of NSNC DP optimization obtained in our work are likely improvable, at least with a computationally inefficient method. This approach is inspired by [LUW24], which in the context of smooth optimization, showed significant sample complexity gains using algorithms with exponential runtime. As as we will show, a similar phenomenon might hold for nonsmooth optimization. To that end, we propose a slight relaxation of Goldstein-stationarity, and show it can be achieved using less samples via an exponential time algorithm.

8.10.1 Relaxation of Goldstein-stationarity

Recall that $x \in \mathbb{R}^d$ is called an (α, β)-Goldstein stationary point of an objective $F(x) = \mathbb{E}_\xi[f(x;\xi)]$ if there exist $y_1, \ldots, y_k \in \mathbb{B}(x, \alpha)$ and convex coefficients $(\lambda_i)_{i=1}^k$ so that $\| \sum_{i\in[k]} \lambda_i \mathbb{E}_\xi[\nabla f(y_i;\xi)]\|$ β. Arguably, the two most important properties satisfied by this definition are that

(i) If $f(x;\xi)$ are L-smooth, any (α, β)-stationary point is $O(\alpha + \beta)$-stationary.

(ii) If $\left\|\overline{\partial}_\alpha F(x)\right\| \neq 0$, then $F\left(x - \frac{\alpha}{\left\|\overline{\partial}_\alpha F(x)\right\|}\overline{\partial}_\alpha F(x)\right) \leq F(x) - \alpha \left\|\overline{\partial}_\alpha F(x)\right\|$.

The first property shows that Goldstein-stationarity reduces to ("classic") stationarity under smoothness. The second, known as Goldstein's descent lemma [Gol77], is a generalization of the classic descent lemma for smooth functions.

It is easy to see that Goldstein-stationarity is equivalent to the existence of a distribution P supported over $\mathbb{B}(x, \alpha)$, such that $\| \mathbb{E}_{\xi,\, y\sim P}[\nabla f(y;\xi)]\| \leq \beta$. We will now define a relaxation of Goldstein-stationarity that is easily verified to satisfy both of the aforementioned properties.

Definition 8.10.1. We call a point $x \in \mathbb{R}^d$ an (α, β)-*component-wise* Goldstein-stationary point of $F(x) = \mathbb{E}_\xi[f(x;\xi)]$ if there exist distributions P_ξ supported over $\mathbb{B}(x, \alpha)$, such that $\| \mathbb{E}_{\xi,\, y\sim P_\xi}[\nabla f(y;\xi)]\| \leq \beta$.

In other words, the definition above allows the sampled points $y_1, \ldots, y_k$ in the vicinity of x to vary for different components, and as before, the sampled gradient must have small expected norm. We next show that this relaxed stationarity notion allows improving the sample complexity of DP NSNC optimization.

8.10.2 Optimal smoothing and faster algorithm

In the previous sections, given an objective f, we used the fact that Goldstein-stationary points of the randomized smoothing f_α correspond to Goldstein-stationary point of f, and therefore constructed private gradient oracles of f_α, which is $O(\sqrt{d}/\alpha)$-smooth. Consequently, the sensitivity of the gradient oracle had a $\sqrt{d}$ dimension dependence (as seen in Lemma 8.3.3), thus affecting the overall sample complexity.

Instead of randomized smoothing, we now consider the Lasry-Lions (LL) smoothing [LL86], a method that smooths Lipschitz functions in a dimension independent manner, which we now recall. Given $h : \mathbb{R}^d \to \mathbb{R}$, denote the so-called Moreau envelope

$$M_\lambda(h)(x) := \min_y \left[h(y) + \frac{1}{2\lambda} \|y - x\|^2 \right],$$

and the Lasry-Lions smoothing:

$$\tilde{h}_{\lambda\mathrm{LL}}(x) := -M_\lambda(-M_{2\lambda}(h))(x) = \max_z \min_y \left[h(z) + \frac{1}{4\lambda} \|z - y\|^2 - \frac{1}{2\lambda} \|y - x\|^2 \right]. \quad (8.20)$$

Fact 8.10.2. *[LL86, AA93] Suppose $h : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz. Then: (i) $\tilde{h}_{\lambda\mathrm{LL}}$ is L-Lipschitz; (ii) $|\tilde{h}_{\lambda\mathrm{LL}}(x) - h(x)| \le L\lambda$ for any $x \in \mathbb{R}^d$; (iii) $\arg\min \tilde{h}_{\lambda\mathrm{LL}} = \arg\min h$; (iv) $\tilde{h}_{\lambda\mathrm{LL}}$ is $O(L/\lambda)$-smooth.*

The key difference between LL-smoothing and randomized smoothing is that the smoothness constant of LL-smoothing is dimension independent. By solving the optimization problem in Eq. (8.20), it is clear that the values, and therefore gradients, of $\tilde{f}_{\lambda\mathrm{LL}}(x; \xi_i)$ can be obtained up to arbitrarily high accuracy. Notably, it was shown by [KS22b] that solving this problem requires, in general, an exponential number of oracle calls to the original function.

Nonetheless, computational considerations aside, it is not even clear that the LL smoothing helps in terms of finding Goldstein-stationary points of the original function, which was previously shown for randomized smoothing (Lemma 8.2.4). This is the purpose of the following result, which we prove:

Lemma 8.10.3. *If h is L-Lipschitz, then any β-stationary point of $\tilde{h}_{\lambda\mathrm{LL}}$ is a $(3\lambda L, \beta)$-Goldstein stationary point of h.*

Given the lemma above, we are able to utilize smooth algorithms for finding stationary points, and convert the guarantee to Goldstein-stationary points of our objective of interest. Specifically, we will invoke the following result.

Proposition 8.10.4 (LUW24)**.** *Given an ERM objective $\tilde{F}(x) = \frac{1}{n}\sum_{i=1}^n \tilde{f}(x; \xi_i)$ with L_0-Lipschitz and L_1-smooth components, and an initial point $x_0 \in \mathbb{R}^d$ such that $\mathrm{dist}(x_0, \arg\min \tilde{F}) \le$*

R, there's an (ε, δ)-DP algorithm that returns $\tilde{x}^{\text{out}}$ with $\mathbb{E}\left\|\nabla \tilde{F}(\tilde{x}^{\text{out}})\right\| = \widetilde{O}\left(\frac{R^{1/3} L_0^{2/3} L_1^{1/3} d^{2/3}}{n\varepsilon} + \frac{L_0\sqrt{d}}{n\varepsilon}\right)$.

We remark that we assume for simplicity that $\operatorname{dist}(x_0, \arg\min \widehat{F}^{\mathcal{D}}) = \operatorname{dist}(x_0, \arg\min \tilde{F}) \leq R$, though the analysis extends to that case where R is the initial distance to a point with sufficiently small loss (e.g., if the infimum is not attained). Overall, by setting $\lambda = \alpha/3L$, and combining Fact 8.10.2, Lemma 8.10.3 and Proposition 8.10.4, we get the following:

Theorem 8.10.5. *Under Assumption 8.2.2, suppose* $\operatorname{dist}(x_0, \arg\min \widehat{F}^{\mathcal{D}}) \leq R$. *Then there is an (ε, δ)-DP algorithm that outputs x^{out} satisfying (α, β)-component-wise Goldstein-stationarity (in expectation) as long as*

$$n = \widetilde{\Omega}\left(\frac{R^{1/3} L^{4/3} d^{2/3}}{\varepsilon \alpha^{1/3} \beta}\right).$$

8.10.3 Proofs from Appendix 8.10

Proof of Lemma 8.10.3. Suppose x is a β-stationary point of $\tilde{h}_{\lambda\text{LL}}$. Let $z^* \in \mathbb{R}^d$ be the solution of the maximization problem defining the LL smoothing. By [AA93, Remark 4.3.e], z^* is uniquely defined, and satisfies

$$\nabla \tilde{h}_{\lambda\text{LL}}(x) \in \partial(M_{2\lambda}(h))(z^*). \tag{8.21}$$

Further denote $\mathcal{Y}^* := \arg\min_y \left[h(y) + \frac{1}{4\lambda}\left\|z^* - y\right\|^2\right] \subseteq \mathbb{R}^d$. Rearranging the definition of the Moreau envelope by expanding the square, we see that

$$M_{2\lambda}(h)(z^*) = \frac{1}{4\lambda}\left\|z^*\right\|^2 - \frac{1}{2\lambda}\max_y \left[\langle z^*, y\rangle - 2\lambda h(y) - \frac{1}{2}\left\|y\right\|^2\right],$$

from which we get

$$\partial M_{2\lambda} h(z^*) = \frac{1}{2\lambda} z^* - \frac{1}{2\lambda}\operatorname{conv}\left\{y^* : y^* \in \mathcal{Y}^*\right\} = \operatorname{conv}\left\{\frac{1}{2\lambda}(z^* - y^*) : y^* \in \mathcal{Y}^*\right\}. \tag{8.22}$$

Furthermore, for any $y^* \in \mathcal{Y}^*$, by first-order optimality it holds that

$$0 \in \partial\left[h(y^*) + \frac{1}{4\lambda}\left\|y^* - z^*\right\|^2\right] \subseteq \partial h(y^*) + \frac{1}{2\lambda}(y^* - z^*),$$

and therefore

$$\frac{1}{2\lambda}(z^* - y^*) \in \partial h(y^*). \tag{8.23}$$

By combining (8.21), (8.22) and (8.23) we conclude that

$$\nabla \tilde{h}_{\lambda\mathrm{LL}}(x) \in \partial M_{2\lambda} h(z^*) \subseteq \operatorname{conv}\{\partial h(y^*) : y^* \in \mathcal{Y}^*\} \subseteq \partial_r h(x),$$

where the last holds for $r := \max_{y^* \in \mathcal{Y}^*} \|x - y^*\|$. Therefore, recalling that $\|\nabla \tilde{h}_{\lambda\mathrm{LL}}(x)\| \leq \beta$, all that remains is to bound r.

To that end, it clearly holds that $r \leq \|x - z^*\| + \max_{y^* \in \mathcal{Y}^*} \|z^* - y^*\|$. Furthermore, by [AA93, Remark 4.3.e] it holds that $z^* - x = \lambda \nabla \tilde{h}_{\lambda\mathrm{LL}}(x)$ which implies $\|x - z^*\| = \lambda\beta$. As to the second summand, by (8.22) it holds that $\max_{y^* \in \mathcal{Y}^*} \|z^* - y^*\| \leq 2\lambda \cdot \max_{g \in \partial M_{2\lambda} h(z^*)} \|g\| \leq 2\lambda L$, by the fact that $M_{2\lambda}(h)$ is L-Lipschitz. Overall $r \leq \lambda\beta + 2\lambda L$, and as we can assume without loss of generality that $\beta \leq L$ since otherwise the claim is trivially true (note that all points are L stationary), this completes the proof.

□

8.11 Concentration lemma for vectors with sub-Gaussian norm

Here we recall a standard concentration bound for vectors with sub-Gaussian norm, which notably applies in particular to bounded random vectors.

Definition 8.11.1 (Norm-sub-Gaussian)**.** We say a random vector $X \in \mathbb{R}^d$ is ζ-norm-sub-Gaussian for $\zeta > 0$, if $\Pr[\|X - \mathbb{E}\,X\| \geq t] \leq 2e^{-t^2/2\zeta^2}$ for all $t \geq 0$.

Theorem 8.11.2 (Hoeffding-type inequality for norm-subGaussian, JNG+19)**.** *Let $X_1, \cdots, X_k \in \mathbb{R}^d$ be random vectors, and let $\mathcal{F}_i = \sigma(X_1, \cdots, X_i)$ for $i \in [k]$ be the corresponding filtration. Suppose for each $i \in [k]$, $X_i \mid \mathcal{F}_{i-1}$ is zero-mean ζ_i-norm-sub-Gaussian. Then, there exists an absolute constant $c > 0$, such that for any $\gamma > 0$:*

$$\Pr\left[\left\|\sum_{i\in[k]} X_i\right\| \geq c\sqrt{\log(d/\gamma)\sum_{i\in[k]} \zeta_i^2}\right] \leq \gamma.$$

Part IV

USER LEVEL

Chapter 9

USER-LEVEL DIFFERENTIALLY PRIVATE STOCHASTIC CONVEX OPTIMIZATION: EFFICIENT ALGORITHMS WITH OPTIMAL RATES

9.1 Introduction

Differentially private stochastic convex optimization (DP-SCO) is a central problem in privacy-preserving machine learning, whose aim is to minimize a convex function

$$\begin{aligned} &\text{minimize } L_{\mathcal{P}}(\theta) := \mathop{\mathbb{E}}_{z\sim\mathcal{P}}[\ell(\theta; z)] \\ &\text{subject to } \theta \in \Theta \subset \mathbb{R}^d, \end{aligned} \tag{9.1}$$

under the constraint of differential privacy, given n users each holding a single sample $z_i \in \mathcal{Z}$ from the distribution $\mathcal{P}$. Numerous works have studied this problem, known as item-level DP-SCO, and it is by now relatively well understood [BST14, BFTGT19, FKT20, AFKT21, ADF+21, KLL21].

A significant concern about item-level DP-SCO in practice is that each user may hold and contribute multiple items to the dataset, significantly degrading the actual privacy protection provided by the item-level differentially private to users. This is the case in many machine learning applications in practice, such as training language and vision models on users' data in federated learning. To address this problem, prior work has studied user-level versions of differential privacy, where the algorithm preserves privacy for users that may contribute $m \geq 1$ items [LSY+20, BRP21, LSA+21]. This definition is stronger than item-level DP as it forces the algorithm not to be sensitive to changes of a single user or equivalently m items.

Motivated by the realistic and strong privacy protections guaranteed by user-level privacy, many papers have studied DP-SCO under this notion of privacy. [LSA+21] has initiated the study of this problem and proposed new algorithms based on localized SGD. The

main observation in [LSA+21] is that averaging the gradients of users in SGD results in gradients that are concentrated in a ball of small radius of roughly $1/\sqrt{m}$, yielding a final excess risk of $1/\sqrt{nm} + d/n\sqrt{m}\varepsilon$. However, as the optimal rates for item-level DP-SCO ($m = 1$) are known to be $1/\sqrt{n} + \sqrt{d}/n\varepsilon$, it is evident that the rates of [LSA+21] are sub-optimal. Moreover, their algorithms are applicable only to smooth functions.

Two recent works of [BS23, GKK+23b] have resolved some of these issues. [BS23] developed new algorithms based on DP-SGD with improved mean estimation procedures to obtain an optimal rate $1/\sqrt{nm}+\sqrt{d}/n\sqrt{m}\varepsilon$. However, their algorithms also require smoothness of the function and require a stringent lower bound on the number of users $n \geq \sqrt{d}/\varepsilon$. Moreover, their algorithm cannot work for large m and requires $m \leq \max\{\sqrt{d}, n\varepsilon^2/\sqrt{d}\}$. On the other hand, [GKK+23b] observes that user-level DP-SCO has small local sensitivity to deletions and uses propose-test-release to design new algorithms. Their algorithm requires only $n \leq \log(d)/\varepsilon$ users and is also applicable to non-smooth functions. However, it runs in super-polynomial time and achieves sub-optimal error $1/\sqrt{nm} + \sqrt{d}/n\sqrt{m}\varepsilon^{2.5}$.

As a result, existing algorithms for user-level DP-SCO are not satisfactory: they either require smoothness and a large number of users that grow polynomially with the dimension [BS23], or run in super-polynomial time [GKK+23b].

9.1.1 Contributions and Technical Overview

In this work, we develop new algorithms for user-level DP-SCO that resolve the abovementioned issues. In particular, our algorithms obtain optimal rates in polynomial time, are applicable for non-smooth functions, and require the number of users to grow only logarithmically in the dimension $n \leq \frac{\log(d)}{\varepsilon}$. We summarize our results for the convex case and compare them to prior work in Table 9.1. Additionally, building on our algorithm for the convex case, we propose a new algorithm that obtains optimal rates for user-level DP-SCO in the strongly convex case.

Our algorithm follows a similar recipe to that of [BS23]: as it is well known that DP-SGD is optimal in the item-level setting, we wish to extend it to user-level DP using new mean estimation procedures that add less noise to estimate the gradients at each iteration.

	Excess Risk	Polynomial Runtime	Number of Users
[BS23]	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon}$	Yes	$n \geq \frac{\sqrt{d}}{\varepsilon}$
[GKK+23b]	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon^{2.5}}$	No	$n \geq \frac{1}{\varepsilon}$
This work	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon}$	Yes	$n \geq \frac{1}{\varepsilon}$

Table 9.1: Comparison of excess risk bounds for user-level DP-SCO with prior work, with logarithmic terms omitted. The work of [BS23] additionally requires smoothness of the loss function and $m \leq \max\{\sqrt{d}, n\varepsilon^2/\sqrt{d}\}$.

To this end, note that if we average the gradients of each user using their m samples, this guarantees that the resulting averaged gradients of all users will lie in a ball of radius roughly $\tau = 1/\sqrt{m}$. This concentration allows to design algorithms for mean estimation with sensitivity τ/n (instead of $1/n$), hence obtaining error (e.g., [BS23]) $\tau\sqrt{d}/n\varepsilon_i$ for estimating the gradients at iteration i, where ε_i is the privacy budget at iteration i. As we have T iterations, this requires $\varepsilon_i = \varepsilon/\sqrt{T}$. The key challenge here is that private mean estimation procedures for τ-concentrated data (e.g. [LSA+21, BS23]) require $n \geq 1/\varepsilon_i = \sqrt{T}/\varepsilon$, which results in a strong restriction on the number of rounds T that we can run.

Our main challenge is then to design a private mean estimation procedure with T iterations. Each iteration we wish to estimate the mean of τ-concentrated data with privacy budget $\varepsilon_i = \varepsilon/\sqrt{T}$ such that the error at each iteration is $\tau\sqrt{d}/n\varepsilon_i$, and the algorithm uses only $n \leq \log(T)/\varepsilon$ samples. We develop a new private mean estimation algorithm for τ-concentrated data that satisfies these properties.

Our approach draws inspiration from the FriendlyCore framework [TCK+22], which we use for removing outliers from the dataset. Our methodology has two distinct phases: in the initial stage, we employ an outlier-elimination process that yields a subset of data samples exhibiting τ-concentration. Subsequently, we privatize the mean of the concentrated sample by adding Gaussian noise proportional to τ.

Our outlier-detection phase is based on a score we give to each sample to measure how likely it is to be an outlier; the score measures how many samples in the dataset are in a ball of size τ around the sample. We then keep each sample in the dataset with probability

proportional to its score, hence removing outliers that have low scores. To guarantee that our final algorithm is private, we have to upper bound the sensitivity of the mean of the sub-sampled dataset is minor. To this end, we apply an extra step via AboveThreshold [DR14] to verify that the input dataset is nearly τ-concentrated, hence limiting the number of outliers that can be detected.

This improved mean estimation procedure is the building block of all of our results: it allows us to use DP-SGD with a small number of users and run it for large number of rounds to get the optimal rate. Moreover, the large number of rounds made possible by our mean estimation procedure allows us to use randomized smoothing in order to obtain optimal results in the non-smooth case as well, in contrast to prior work where randomized smoothing would not result in optimal rates in the non-smooth setting.

9.1.2 Related Work

User-level differential privacy (DP) is a relatively recent and less-explored area compared to the more established item-level DP setting. It has gained increased attention lately due to its significance in machine learning applications, particularly in the context of federated learning. Several works have studied user-level DP for several applications, including DP-SCO [LSA+21, BS23], PAC learning [BRP21], and discrete distribution estimation [LSY+20, ALS23]. In recent work, [GKK+23a] proposed a generic transformation of any item-level DP algorithm to a user-level DP algorithm. However, it is inefficient, and the dependence on ε may not be optimal.

DP-SCO has been studied in the item-level DP setting extensively [BFTGT19, FKT20, AFKT21, ADF+21, KLL21, GLL+23]. The rates of DP-SCO in the item-level setting are well understood and [BFTGT19] obtained the optimal $1/\sqrt{n} + \sqrt{d\log(1/\delta)}/n\varepsilon$ rate using stability based analysis of DP-SGD with a large batch size. These algorithms are not efficient, leading [FKT20] to develop new optimal algorithms for the smooth case that run in linear time. However, the best runtime for the non-smooth setting is super-linear, and this is an ongoing research direction which is still open [AFKT21, KLL21, CJJ+23]. Item-level DP-SCO has also been studied in various other settings, such as the stronger pure DP

model [ALD21], heavy-tailed data distributions [LR23], non-euclidean geometries [AFKT21, BGN21], and non-convex loss functions [GLOT23, ABG+23].

9.2 Preliminaries

Let $[k] = \{1, \cdots, k\}$ be the set of positive integers no larger than k. Throughout the paper, we assume that the loss function $\ell(:, z) : \Theta \to \mathbb{R}$ is convex and G-Lipschitz for any $z \in \mathcal{Z}$, and $\Theta \subset \mathbb{R}^d$ is a closed convex domain of diameter R. There are n users, each holding m i.i.d. samples from the underlying distribution $\mathcal{P}$; we denote the samples of the i-th user by $Z_i = \{z_{i,j}\}_{j\in[m]}$. We use capital Z to denote one user and z to denote one item. The dataset $\mathcal{D} = \{Z_i\}_{i\in[n]}$ contains all the users along with all the items.

The objective is to design efficient algorithms for minimizing $L_{\mathcal{P}}(\theta) := \mathbb{E}_{z\sim\mathcal{P}}\, \ell(\theta, z)$, which is differentially private at the user level. For a user $Z_i = \{z_{i,j}\}_{j\in[m]}$, we let $\nabla L(\theta; Z_i) := \frac{1}{m}\sum_{j\in[m]} \nabla\ell(\theta; Z_{i,j})$ denote the average of the gradients for the user's samples. We denote the empirical function $L_{\mathcal{D}}(\theta) := \frac{1}{nm}\sum_{z\in Z_i}\sum_{Z_i\in\mathcal{D}} \ell(\theta, z)$. For a distribution X, we let $\mathrm{supp}(X)$ be the support of the distribution X.

9.2.1 Differential Privacy

In this work, we use the notion of user-level differential privacy where each user has a sample $z \in \mathcal{Z}^m$.

Definition 9.2.1 (User-Level Differential Privacy)**.** A mechanism $\mathcal{M} : (\mathcal{Z}^m)^n \to \mathbb{R}^d$ is (ε, δ) user-level differentially private, if for any neighboring datasets $\mathcal{D}, \mathcal{D}' \in (\mathcal{Z}^m)^n$ that differ in one user, and for any event O in the range of $\mathcal{M}$, we have

$$\Pr[M(\mathcal{D}) \in O] \le e^{\varepsilon} \Pr[M(\mathcal{D}') \in O] + \delta.$$

Note that item-level differential privacy is a specific case of this definition where $m = 1$.

Additionally, our analysis requires the notion of indistinguishability between two random variables.

Definition 9.2.2 (Indistinguishablity)**.** Two random variables X and Y are (ε, δ)-Indistinguishable if for any event $\mathcal{O}$, we have

$$\Pr[X \in \mathcal{O}] \leq e^{\varepsilon} \Pr[Y \in \mathcal{O}] + \delta,$$
$$\text{and } \Pr[Y \in \mathcal{O}] \leq e^{\varepsilon} \Pr[X \in \mathcal{O}] + \delta.$$

Moreover, for two distributions X and Y, we use the notation $X \sim_{\gamma} Y$ to denote that the total variation distance between X and Y is bounded by γ. We also define the following divergence.

Definition 9.2.3. Given two distributions X and Y, the δ-approximate max divergence between X and Y is defined as

$$D_{\infty}^{\delta}(X \| Y) = \sup_{Z \in \text{supp}(X): \Pr[X \in Z] \geq \delta} \log \frac{\Pr[X \in Z] - \delta}{\Pr[Y \in Z]}$$

AboveThreshold

Our algorithms use the AboveThreshold algorithm [DR14] which is a key tool in differential privacy to identify whether there is a query $q_i : \mathcal{Z} \to \mathbb{R}$ in a stream of queries $q_1, \ldots, q_T$ that is above a certain threshold Δ. The AboveThreshold algorithm (presented in appendix) has the following guarantees.

Lemma 9.2.4 ([DR14], Theorem 3.24)**.** AboveThreshold *is* $(\varepsilon, 0)$*-DP. Moreover, let* $\alpha = \frac{8 \log(2T/\gamma)}{\varepsilon}$ *and* $\mathcal{D} \in \mathcal{Z}^n$*. For any sequence of* T *queries* $q_1, \cdots, q_T : \mathcal{Z}^n \to \mathbb{R}$ *each of sensitivity* 1*,* AboveThreshold *halts at time* $k \in [T+1]$ *such that with probability at least* $1 - \gamma$*,*

- *For all* $t < k$*,* $a_t = \top$ *and* $q_t(\mathcal{D}) \geq \Delta - \alpha$*;*
- $a_k = \perp$ *and* $q_k(\mathcal{D}) \leq \Delta + \alpha$ *or* $k = T + 1$*.*

Algorithm 27: AboveThreshold

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, threshold $\Delta \in \mathbb{R}$, privacy parameter ε;
Let $\widehat{\Delta} := \Delta - \mathrm{Lap}(\frac{2}{\varepsilon})$;
for $t = 1$ *to* T **do**
  Receive a new query $q_t : \mathcal{Z}^n \to \mathbb{R}$;
  Sample $\nu_i \sim \mathrm{Lap}(\frac{4}{\varepsilon})$;
  if $q_t(\mathcal{D}) + \nu_i < \widehat{\Delta}$ **then**
    Output: $a_i = \perp$;
    Halt;
    else
      Output: $a_i = \top$;
    end
  end
end

9.2.2 Randomized Smoothing

To develop optimal algorithms in the non-smooth setting, our algorithm use randomized smoothing [YNS12, DBW12] to make the functions smooth. To this end, for a convex function $\ell(:; Z)$, we denote the convolution function $\widehat{\ell}(:; Z) := \ell(:; Z) * n_r.$, where n_r is the uniform density in the ℓ_2 ball of radius r centered at the origin in $\mathbb{R}^d$. Specifically, $n_r(y) = \frac{\Gamma(\frac{d}{2}+1)}{\pi^{\frac{d}{2}} r^d}$ for $\|y\| \le r$, and $n_r(y) = 0$ otherwise. For simplicity, we may omit the dependence on z, and write the function as $\widehat{\ell}$ and ℓ. Denote $\widehat{L}_{\mathcal{P}}(\theta) := \mathbb{E}_{z\sim P, y\sim n_r} \ell(\theta + y; z)$ and $\widehat{L}_{\mathcal{D}}(\theta) := \frac{1}{|\mathcal{D}|} \sum_{z\in\mathcal{D}} \mathbb{E}_{y\sim n_r} \ell(\theta + y; z)$.

Lemma 9.2.5 (Randomized Smoothing, [YNS12, DBW12])**.** *The convolution function has the following properties:*

- $\widehat{\ell}(\theta) \le \ell(\theta) \le \widehat{\ell}(\theta) + Gr.$
- $\widehat{\ell}$ *is* G*-Lipschitz and convex.*
- $\widehat{\ell}$ *is* $\frac{G\sqrt{d}}{r}$*-smooth.*
- *For random variables* $y \sim n_r$*, and* $z \in \mathcal{D}$*, we have* $\mathbb{E}_{y,z}[\nabla \ell(\theta + y; z)] = \nabla \widehat{L}_{\mathcal{D}}(\theta)$.

9.2.3 Norm-Subgaussian Concentration

Our analysis also uses a notion named concentration properties for norm-Subgaussian random variables.

Definition 9.2.6 (norm-Subgaussian)**.** A random vector $X \in \mathbb{R}^d$ is norm-SubGaussian with parameter σ, denoted nSG(σ), if for all $t \in \mathbb{R}$

$$\Pr[\|X - \mathbb{E}\, X\| \geq t] \leq 2\exp(-\frac{t^2}{2\sigma^2}).$$

The following concentration result holds for norm-Subgaussian random variables.

Lemma 9.2.7 ([JNG+19], concentration of NormSubgaussian)**.** *There exists a constant $c > 0$, such that for zero-mean independent random vectors $X_1, \cdots, X_n \in \mathbb{R}^d$ where X_i is* nSG(σ_i) *for all $i \in [n]$, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\|\sum_{i\in[n]} X_i\| \leq c\sqrt{\sum_{i\in[n]} \sigma_i^2 \log\frac{2d}{\delta}}.$$

We also use the following standard Chernoff-Hoeffding bound.

Lemma 9.2.8 (Chernoff-Hoeffding Bound)**.** *Let $X_1, \cdots, X_n$ be independent Bernoulli random variables such that $\mathbb{E}[X_i] = p_i$. Let $X = \sum_{i\in[n]} X_i$ and $\mu = \mathbb{E}[X]$. Then we know for any $\lambda > 0$, we have*

$$\Pr[X \geq (1+\lambda)\mu] \leq \exp(-\frac{\lambda^2\mu}{2+\lambda}).$$

9.3 Adaptive Mean Estimation for Concentrated Samples

The main component in our algorithms is a novel mean estimation procedure for adaptive queries for τ-concentrated samples where the samples lie in a ball of radius τ (see Definition 9.3.1). This algorithm will be used to estimate the gradients in our optimization procedure, as the user-level setting will guarantee that $\tau \approx 1/\sqrt{m}$ for an i.i.d. input. We add Gaussian noise scales with τ; hence, the final loss bound benefits from small τ.

Definition 9.3.1. A random samples $\{X_i\}_{i\in[n]}$ is (τ,γ)-concentrated if there exists a point $x \in \mathbb{R}^d$ such that with probability at least $1-\gamma$,

$$\max_{i\in[n]} \|X_i - x\| \le \tau.$$

Given T adaptive mean estimation queries $q_1,\dots,q_T : (\mathcal{Z}^m)^n \to \mathbb{R}^d$ such that the n users are τ-concentrated with respect to these queries, our goal is to get a nearly unbiased estimate of the mean of each query with variance $\frac{\tau^2 T d}{n^2\varepsilon^2}$ under (ε,δ)-DP. The standard approach for solving this task, as done in [BS23], is to assign a privacy budget $\varepsilon_i = \varepsilon/\sqrt{T}$ for each query, hence resulting in variance $\frac{\tau^2 T d}{n^2\varepsilon^2}$. However, this procedure requires $n \ge \frac{1}{\varepsilon_i} = \sqrt{T}/\varepsilon$ to guarantee the desired utility bounds, which is too prohibitive for our purposes.

In this section, we design a new algorithm for adaptive mean estimation that achieves the desired variance with only $n \ge 1/\varepsilon$. Our algorithm is inspired by the FriendlyCore framework [TCK+22], where we use the basic filter to identify outliers in the dataset. Our procedure consists of two stages: first, we apply an outlier-removal procedure, which returns a subset of the samples that is τ-concentrated. Then, we add Gaussian noise proportional to τ to privatize the mean of the concentrated sample.

To identify outliers, we give a score to each sample, which measures how many samples in the dataset are in a ball of size τ around the sample. As outlier samples will have a low score, we then keep each sample in the dataset with probability proportional to its score. This will preserve privacy for samples that are nearly τ-concentrated, whereas we aim to preserve privacy for all input datasets. Therefore, we add an initial check to the algorithm which verifies that the algorithm is nearly τ-concentrated. To this end, we define a τ-concentration score of the dataset for a query q_i to be

$$s_i^{\mathsf{conc}}(\mathcal{D},\tau) := \frac{1}{n}\sum_{z\in\mathcal{D}}\sum_{z'\in\mathcal{D}} \mathcal{I}(\|q_i(z) - q_i(z')\| \le \tau). \tag{9.2}$$

and check via AboveThreshold that this score is above the desired threshold for all queries. The following procedure will be processed only if the dataset and the queries pass the check, which means our samples are nearly concentrated and ensures the privacy guarantee of the

following procedure. We describe the full details of our algorithm in Algorithm 28.

Algorithm 28: Outlier-Removal Based Mean Estimation for Concentrated Data

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, privacy parameters (ε, δ), parameters τ ;
for $i = 1$ *to* T **do**
  Receive a new mean estimation query $q_i : \mathcal{Z} \to \mathbb{R}^d$;
  Define concentration score

$$s_i^{\mathsf{conc}}(\mathcal{D}, \tau) := \frac{1}{n} \sum_{Z \in \mathcal{D}} \sum_{Z' \in \mathcal{D}} \mathcal{I}(\|q_i(Z) - q_i(Z')\| \le \tau)$$

  if $\mathsf{AboveThreshold}(s_i^{\mathsf{conc}}, \varepsilon/2, 4n/5) = \top$ **then**
    Set $S_i = \emptyset$;
    for *Each User* $Z_j \in \mathcal{D}$ **do**
      Set $f_{i,j} = \sum_{Z \in \mathcal{D}} \mathcal{I}(\|q_i(Z_j) - q_i(Z)\| \le 2\tau)$;
      Add Z_j to S_i with probability $p_{i,j}$ for $p_{i,j} = \begin{cases} 0 & f_{i,j} < n/2 \\ 1 & f_{i,j} \ge 2n/3 \\ \frac{f_{i,j} - n/2}{n/6} & o.w. \end{cases}$
    end
    Let $g_i = \frac{1}{|S_i|} \sum_{Z \in S_i} q_i(Z)$ if S_i is not empty, and 0 otherwise ;
    Output: $\widehat{g}_i \leftarrow g_i + \nu_i$, where $\nu_i \sim \mathcal{N}(0, \frac{8\tau^2 T \log(e^\varepsilon T/\delta) \log(e^{\varepsilon/2}/\delta)}{n^2 \varepsilon^2} I_d)$
  end
  else
    Output: $g_i = \mathbf{0}$;
    Halt;
  end
end

The following theorem summarizes the main guarantees of our algorithm.

Theorem 9.3.2. *For $0 < \varepsilon < 10, 0 < \delta < 1$. Let $\mathcal{D} = (Z_1, \ldots, Z_n) \in (\mathcal{Z}^m)^n$ be a dataset with $n \ge \frac{8 \log(T/\gamma) + 8 \log(T/\delta)}{\varepsilon}$ users. Algorithm 28 is (ε, δ)-DP. Moreover, if $(q_i(Z_1), \ldots, q_i(Z_n))$ is (τ, γ)-concentrated for all $i \in [T]$ and let $\{\widehat{g}_i\}_{i \in [T]}$ be the outputs of Algorithm 28, then there exists random variables $\widehat{g}'_1, \ldots, \widehat{g}'_T$ such that the joint distributions $\{\widehat{g}_i\}_{i \in [T]} \sim_{(1+T)\gamma} \{\widehat{g}'_i\}_{i \in [T]}$. Moreover, for each $i \in [T]$, given $\{g'_j\}_{j \le i-1}$ and q_i, $\widehat{g}'_i$ satisfy that*

$$\mathbb{E}\, \widehat{g}'_i = \frac{1}{n} \sum_{j=1}^{n} q_i(Z_j),$$

$$\mathbb{E}\left\|\widehat{g}_i' - \frac{1}{n}\sum_{j=1}^{n} q_i(Z_j)\right\|^2 \lesssim \frac{\tau^2 T \log(T/\delta)\log(1/\delta)}{n^2\varepsilon^2}.$$

To prove Theorem 9.3.2, we consider the privacy and utility guarantees separately. We argue about privacy first. The following lemma upper bounds the sensitivity of the probability distribution p_i for adding users to S_i.

Lemma 9.3.3. *For any neighboring dataset $\mathcal{D}, \mathcal{D}'$ that differs in one user, let $p_i = (p_{i,1}, \cdots, p_{i,n})$ be the probability for users to be selected into S_i for $\mathcal{D}$, and let p_i' be the corresponding probability for $\mathcal{D}'$. Then*

$$\|p_i - p_i'\|_1 \leq 2.$$

In the following lemma, we show when $|p_i - p_i'| \leq 2$, then the hamming distance between the selected sets S_i and S_i' cannot be large. Thus, given the low sensitivity of p_i for two neighboring datasets, this shows that sub-sampled datasets at round i will not be too far from each other.

Lemma 9.3.4. *Let $p, p' \in [0,1]^n$ such that $\|p - p'\|_1 \leq 2$, and let V and V' be drawn from* $\mathrm{Ber}(p)$ *and* $\mathrm{Ber}(p')$ *respectively. For any $\zeta \in (0,1)$, there exists a coupling Γ over V and V' such that for (x,y) drawn from Γ, with probability at least $1-\zeta$,*

$$\|x - y\|_1 \leq O(\log(1/\zeta)).$$

Now, we analyze the privacy guarantee. Since AboveThreshold is private, it suffices to prove privacy for the case where AboveThreshold always outputs "$\top$", as otherwise the output is $\mathbf{0}$. Note that when AboveThreshold outputs "$\top$", the dataset is well concentrated with respect to the queries. This concentration, together with the fact that the sub-sampled datasets are not too far from each other (Lemma 9.3.4), allows us to upper bound the sensitivity of the mean of the sub-sampled datasets, that is g_i. Hence, the privacy guarantee of the outputs $\{\widehat{g}_i\}$ will follow from the guarantees of the Gaussian mechanism.

To formalize the above intuition, let $a_i \in \{\top, \perp\}$ be the output of AboveThreshold for

i-th query. Recall that in Algorithm 27, we draw one random variable from $\mathrm{Lap}(\frac{2}{\varepsilon})$ and T independent random variables from $\mathrm{Lap}(\frac{4}{\varepsilon})$. Let E be the event that the absolute values of these random variables are no more than $\frac{4\log(2T/\zeta)}{\varepsilon}$. Then, we know the probability of E is at least $1-\zeta/2$. Conditional on E, for all $a_i = \top$, we have $q_i \geq \frac{4n}{5} - \alpha$ and for all $a_i = \perp$ we have $q_i \leq \frac{4n}{5} + \alpha$. Note that $\frac{4n}{5} - \alpha \geq \frac{2n}{3}$ by the value of α and the precondition that $n \geq \frac{40\log(2T/\zeta)}{\varepsilon}$. The guarantees of AboveThreshold (Lemma 9.2.4) also imply that the measure of E is at least $1-\zeta$. Define E' to be the event w.r.t. input $\mathcal{D}'$.

The following lemma upper bounds the sensitivity of the mean of the sub-sampled datasets.

Lemma 9.3.5. *For any i-th iteration and any neighboring datasets $\mathcal{D}, \mathcal{D}'$, conditional on E and E' and conditional on $a_i = a_i'$, there exists a coupling Γ_i over g_i and g_i', such that for (x, y) drawn from Γ_i, with probability at least $1-\zeta$,*

$$\|x - y\|_2 \lesssim \frac{\tau \log(1/\zeta)}{n}.$$

Given the sensitivity bound of Lemma 9.3.5, we can argue for indistinguishability of the outputs using advanced composition and standard guarantees of the Gaussian mechanism.

Proposition 9.3.6. *For any dataset $\mathcal{D}$, if $n \geq \frac{40\log(4T/\delta)}{\varepsilon}$, then for any neighboring dataset $\mathcal{D}'$, the outputs of Algorithm 28 with $\mathcal{D}$ and $\mathcal{D}'$ as inputs are (ε, δ)-indistinguishable.*

Proof. (sketch) We only provide a sketch of the proof here and defer the full proof to Section 9.6.4. First, note that $a_1, \ldots, a_T \in \{\top, \perp\}$ are $\varepsilon/2$-DP using the guarantees of AboveThreshold. Moreover, if there exists an $a_i = \perp$ then g_i is post-processing of a_i hence private as well. Thus, we prove privacy of $\{\widehat{g}_1, \ldots, \widehat{g}_T\}$ assuming $a_1 = a_2 = \cdots = a_T = \top$. First, we condition on the high-probability event E which indicates the success of AboveThreshold (the failure probability will be added to the δ term). Under this event, Lemma 9.3.5 implies that the sensitivity of g_i is bounded by $\frac{\tau\log(1/\zeta)}{n}$. Thus, advanced composition and the guarantees of the Gaussian mechanism imply that $\{\widehat{g}_1, \ldots, \widehat{g}_T\}$ are $(\varepsilon/2, \delta)$-DP. The claim follows. □

Having estabilished the privacy guarantee of Algorithm 28, we now proceed to prove

its utility. The following proposition shows that if the dataset is well concentrated with respect to the query, then no user will be removed in the outlier-removal stage with high probability, hence the estimate is nearly unbiased.

Proposition 9.3.7. *For all $i \in [T]$, if $(q_i(Z_1), \ldots, q_i(Z_n))$ is (τ, γ)-concentrated and $n \geq \frac{8\log(T/\gamma)}{\varepsilon}$, then with probability at least $1-(T+1)\gamma$, we have $S_i = \mathcal{D}$ for all $i \in [T]$. In particular, it holds that $g_i = \frac{1}{n}\sum_{Z\in\mathcal{D}} q_i(Z)$ with probability at least $1-(T+1)\gamma$.*

Proof. To prove the lemma, we have to show that AboveThreshold will succeed (output $\top$) for each $i \in [T]$, and that the outlier-removal stage will not remove any item from the set. This will imply that $S_i = \mathcal{D}$ for all $i \in [T]$, hence $g_i = \frac{1}{n}\sum_{Z\in\mathcal{D}} q_i(Z)$.

To this end, fix any $i \in [T]$. Under the precondition that $(q_i(Z_1), \ldots, q_i(Z_n))$ is (τ, γ)-concentrated, we know that $s_i^{\mathsf{conc}}(\mathcal{D}, \tau) = n$ with probability $1-\gamma$ for each $i \in [T]$. Moreover, the guarantees of AboveThreshold (Lemma 9.2.4) imply that it will output "$\top$" with probability at least $1-\gamma/T$ for each $i \in [T]$ when $s_i^{\mathsf{conc}}(\mathcal{D}, \tau) = n$. Finally, under the event that $(q_i(Z_1), \ldots, q_i(Z_n))$ is τ-concentrated, we have that $f_{i,j} = n$ for each user $Z_j \in \mathcal{D}$, and hence Z_j will be added into S_i. The statement follows by applying a union bound. □

The utility guarantees of Theorem 9.3.2 now follows from Proposition 9.3.7 by setting $\widehat{g}'_i = \frac{1}{n}\sum_{Z\in\mathcal{D}} q_i(Z) + \nu_i$, where $\nu_i \sim \mathcal{N}(0, \frac{8\tau^2 T\log(e^\varepsilon T/\delta)\log(e^{\varepsilon/2}/\delta)}{n^2\varepsilon^2} I_d)$.

9.4 Optimal Rates for User-Level DP-SCO

In this section, we present our main algorithm for user-level DP-SCO based on the gradient estimation procedure constructed above. Our algorithm leverages the Stochastic Gradient Descent (SGD) over a smoothed version of the loss function using randomized smoothing by applying the gradient estimation procedure to get (nearly) unbiased stochastic gradients. We present the full details of the algorithm in Algorithm 29.

Three key techniques are crucial for our algorithm and its analysis: first, for a fixed $\theta \in \Theta$, a simple concentration argument shows that the average gradient of each user will lie with high probability in a ball of small radius around the population gradient (see Lemma 9.4.4)

$$\|\nabla L(\theta; Z_i) - \nabla L_{\mathcal{P}}(\theta)\| \leq \frac{G\log(nd/\gamma)}{\sqrt{m}}.$$

This is not sufficient for our algorithms as we need this property to hold for data-dependent θ_t. To this end, similarly to [BS23], we use the generalization properties of differential privacy to show in Lemma 9.4.6 that a similar concentration holds for $\nabla L(\theta_t; Z_i)$. Given this concentration, our mean estimation procedure (Algorithm 28) adds lower noise to estimate of the gradients.

Our second technique is based on the observation that smoothness is necessary to obtain the full potential of DP-SGD in user-level DP-SCO (similarly to existing work that used SGD-based algorithms for user-level DP-SCO [LSA+21, BS23]). Convergence rates of SGD cause the limitation for non-smooth functions, which depend on the second moment of the gradients, whereas it depends on the variance for smooth functions (Proposition 9.4.10). As averaging the gradients of m samples reduces the variance while keeping the second moment the same, this yields better performance for smooth functions. To address this, we adopt randomized smoothing to smooth the loss functions and apply SGD over the smoothed functions. This is made possible due to our mean estimation procedure, which only requires $n \geq \log(mnd/\delta)/\varepsilon$, in contrast to prior work, which required $n \geq \sqrt{T}/\varepsilon$; this strict bound on the number of rounds is not sufficient to obtain optimal rates with randomized smoothing.

Finally, as we are using multi-pass SGD, an additional argument is needed to guarantee a low risk for population error. To this end, we analyze the stability of our algorithm for non-smooth functions using [BFGT20], which implies that our algorithm has low generalization error.

Let $\Theta_r = \{\theta + y : \theta \in \Theta, \|y\| \leq r\}$. The following theorem summarizes our main result.

Theorem 9.4.1 (User-level DP-SCO)**.** *Let $0 < \varepsilon < 10$ and $0 < \delta < 1$. Algorithm 29 is user-level (ε, δ)-DP. Setting $\widehat{R} = R, r = \frac{d^{1/4}\widehat{R}}{\sqrt{T}}, \eta = \frac{\widehat{R}}{G} \cdot \min\{\frac{\sqrt{m}n\varepsilon}{T\sqrt{d\log^2(mnd/\delta)}}, \frac{1}{T^{3/4}}, \frac{\sqrt{nm}}{T}\}$, $\tau = \frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $T = O(m^2n^2 + mn\sqrt{d})$, if $\Theta \subset \mathbb{R}^d$ is a convex set of diameter R, $\{\ell(:, z)\}_{z\in\mathcal{Z}}$ is a family of G-Lipschitz convex function over Θ_r, each item in $\mathcal{D}$ is drawn i.i.d. from the underlying distribution P, and $n \gtrsim \frac{\log(mdn/\delta)}{\varepsilon}$, then the output $\widehat{\theta}$ of Algorithm 29 satisfies*

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star\in\Theta} L_{\mathcal{P}}(\theta^\star)\right] \leq O\left(GR \cdot \left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d\log^2(ndm/\delta)}}{n\sqrt{m}\varepsilon}\right)\right).$$

Algorithm 29: DP-SGD for user-level DP

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n) \in (\mathcal{Z}^m)^n$, private parameters (ε, δ), initial point θ_0, convolution parameter r, number of rounds T, stepsize η, concentration parameter τ, initial distance $\widehat{R}$;
for $t = 1, \cdots, T$ **do**
  Define a query $q_t(Z) = \frac{1}{m}\sum_{j=1}^{m} \nabla\widehat{\ell}(\theta_t; z_{i,j})$ for $Z \in \mathcal{Z}^m$, See Equation (9.3) for the definition ;
  Run Algorithm 28 with query q_t and parameters $\mathcal{D}, \varepsilon, \frac{\delta}{2Tmnd}, \tau$;
  Let $\overline{g}_t$ be the output of Algorithm 28 ;
  if $\overline{g}_t \neq \perp$ **then**
    Update $\theta_{t+1} \leftarrow \Pi(\theta_t - \eta\overline{g}_t)$;
  end
  else
    Output: Initial point θ_0;
    Halt
  end
end
Return: $\widehat{\theta} = \frac{1}{T}\sum_{t\in[T]} \theta_t$

Remark 9.4.2. If we have a random initial point θ_0 such that $\mathbb{E}[\|\theta_0 - \theta^*\|^2] \leq R'^2$ for $\theta^* = \arg\min L_{\mathcal{D}}(\theta)$ and some $R' < R$, then we can replace the parameter setting $\widehat{R} = R$ by $\widehat{R} = R'$ in the population loss bound and the dependence on R can be reduced to R' in the loss bound.

Remark 9.4.3. We define the functions on Θ_r rather than Θ to make use of the randomized smoothing technique. As r is much smaller than R, this impact can be minimal. One can eliminate this domain extension by applying other smoothing techniques, such as the Moreau envelope smoothing method, but this method will increase the gradient computation cost.

In some regimes, when d is large, we can set T in the theorem statement smaller. But we omit those terms to avoid complexity, as getting smaller T is not the primary goal of this work.

We begin by showing that the gradients are concentrated. For any user Z_i who holds m

items denoted by $\{z_{i,j}\}_{j\in[m]}$ and any point $\theta \in \Theta$, we denote

$$\nabla\widehat{\ell}(\theta; Z_i) := \frac{1}{m}\sum_{j\in[m]} \nabla\ell(\theta + y_j; z_{i,j}), \tag{9.3}$$

the average stochastic gradients of all items owned by Z_i, where $y_j \sim n_r$ is drawn independently of θ and $z_{i,j}$ for the randomized smoothing.

Our goal is to eventually prove that $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$ are concentrated. To this end, we start with proving concentration for $\{\nabla\widehat{\ell}(\theta; Z_i)\}_{Z_i\in\mathcal{D}}$ for a fixed $\theta \in \Theta$.

Lemma 9.4.4. *For any fixed θ and for each Z_i, if each item in Z_i is drawn i.i.d. from $\mathcal{P}$, with probability at least $1 - \gamma/n$, we have*

$$\|\nabla\widehat{\ell}(\theta; Z_i) - \nabla\widehat{L}_{\mathcal{P}}(\theta)\| \le \frac{G\log(nd/\gamma)}{\sqrt{m}},$$

One issue with applying Lemma 9.4.4 to demonstrate the concentration property of the stochastic gradients is that the dataset $\mathcal{D}$ and the points $\{\theta_i\}_{i\in[T]}$ are not independent. To tackle this, similarly to [BS23], we make use of the generalization properties of private mechanisms. We need the following lemma.

Lemma 9.4.5 (Lemma 3.7 in [FMT22])**.** *Let $\mathcal{A}$ be an (ε, δ)-DP algorithm with respect the input $\mathcal{D}$. Then there exists an $(2\varepsilon, 0)$-DP algorithm $\mathcal{A}'$, such that*

$$d_{TV}(\mathcal{A}(\mathcal{D}), \mathcal{A}'(\mathcal{D})) \le \delta.$$

Lemma 9.4.6 (Similar to Theorem 3.4 in [BS23])**.** *Suppose $\mathcal{D} = \{z_{i,j}\}_{i\in[n],j\in[m]}$ are drawn i.i.d. from the distribution $\mathcal{P}$. In Algorithm 29, for all $t \in [T]$, $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$ is (τ, γ')-concentrated for*

$$\tau = \frac{G\log(nd/\gamma)}{\sqrt{m}}, \gamma' = T(e^{2\varepsilon}\gamma + \frac{\delta}{2Tmnd}).$$

Having established the concentration property of $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$, we can bound the utility of our procedure for the empirical function $\widehat{L}_{\mathcal{D}}$. Now, we turn to prove the upper

bounds for the generalization error, which needs the following well-known Lemma.

Lemma 9.4.7 ([BE02]). *For an algorithm $\mathcal{A}$, a dataset $\mathcal{D} = \{z_{i,j}\}_{i\in[n],j\in[m]}$ drawn i.i.d. from the distribution $\mathcal{P}$. If we replace one random data $z_{i,j}$ in $\mathcal{D}$ by a fresh new sample $z'_{i,j}$ from $\mathcal{P}$ and get the dataset $\mathcal{D}'$ and let $\mathcal{A}(\mathcal{D})$ be the (random) output of the algorithm, one has*

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathcal{D},\mathcal{A}}\left[L_{\mathcal{P}}(\mathcal{A}(\mathcal{D})) - L_{\mathcal{D}}(\mathcal{A}(\mathcal{D}))\right]\\ =&\mathop{\mathbb{E}}_{\mathcal{D},z'_{i,j},\mathcal{A}}\left[\ell(\mathcal{A}(\mathcal{D}); z'_{i,j}) - \ell(\mathcal{A}(\mathcal{D}'); z'_{i,j})\right].\end{aligned}$$

As we are considering Lipschitz functions, if we can bound the total variation distance between $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$ where $\mathcal{D}$ and $\mathcal{D}'$ differs from one single item, named by algorithmic stability, then we can bound the generalization error. Formally, we define the algorithmic stability of $\mathcal{A}$ as follows:

$$\Lambda(\mathcal{A}) := d_{TV}(\mathcal{A}(\mathcal{D}), \mathcal{A}(\mathcal{D}')),$$

where $d_{TV}(\mathcal{A}(\mathcal{D}), \mathcal{A}(\mathcal{D}'))$ denotes the total variation distance between $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$. Notably, the user-level differential privacy concerns replace m data of one user, while the algorithmic stability only concerns replacing one single item of a user. We have the following Lemma.

Lemma 9.4.8 (Lemma 3.1 in [BFGT20]). *Let $(x^t)_{t\in[T]}$ and $(y^t)_{t\in[T]}$ be two trajectories of running SGD for G-Lipschitz convex function f, that is $x^t = \Pi(x^{t-1} - \eta\nabla f(x^{t-1}))$ and $y^t = \Pi(y^{t-1} - \eta\nabla f'(y^{t-1}))$. Suppose $\|\nabla f(x^t) - \nabla f'(x^t)\| \le a_t \le 2G$ for all $t \in [T]$, then*

$$\|x^T - y^T\| \le 2G\sqrt{\sum_{t\in[T-1]}\eta_t^2} + 2\sum_{t\in[T-1]}\eta_t a_t.$$

We use $\mathcal{A}$ to represent Algorithm 29. Then, we can bound the algorithmic stability of $\mathcal{A}$ based on the unbiased property of our mean estimate procedure (Lemma 9.3.7) constructed in the previous section.

Lemma 9.4.9 (Algorithmic stability bound)**.** *Suppose $\{Z_i\}$ are drawn i.i.d. from the underlying distribution $\mathcal{P}$. Suppose $\tau \geq \frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $n \gtrsim \frac{\log(mdn/\delta)}{\varepsilon}$, with probability at least $1 - \frac{\delta}{mnd}$, the stability of Algorithm 29 is bounded as follows:*

$$\Lambda(\mathcal{A}) \leq G\eta\sqrt{T} + \frac{G\eta T}{nm}.$$

Finally, to prove our main result, we need the following convergence rates for SGD.

Proposition 9.4.10 (SGD, [Bub15])**.** *Consider a convex function f over a convex domain X. Suppose the random initial point x_0 satisfies $\mathbb{E}[\|x_0 - x^*\|] \leq R^2$ where $x^* = \arg\min_{x\in X} f(x)$. Assume the unbiased stochastic oracle is such that $\mathbb{E}[\|\tilde{g}(x)\|^2] \leq \sigma^2$. Running gradient descent with step size η satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f(x_{t+1}) - \min_{x^*} f(x^*)\right] \leq \frac{R^2}{\eta T} + \eta\sigma^2.$$

Moreover, if the function f is β-smooth and the unbiased stochastic oracle is such that $\mathbb{E}[\|\tilde{g}(x) - \nabla f(x)\|^2] \leq \sigma^2$, then running SGD for T steps with step size η satisfies that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f(x_{t+1}) - \min_{x^*} f(x^*)\right] \leq (\beta + \frac{1}{\eta})\frac{R^2}{T} + \frac{\eta\sigma^2}{2}.$$

Combining these lemmas, we are now ready to prove Theorem 9.4.1.

Proof of Theorem 9.4.1. The privacy guarantee of Algorithm 29 follows from the privacy guarantee of our mean estimation procedure (Algorithm 28), as Algorithm 29 is post processing of the outputs of Algorithm 28.

Now, we prove utility. Let $\widehat{\theta} = \frac{1}{T}\sum_{t\in[T]}\theta_t$ denote the output of the algorithm. We upper bound the error by splitting it to two terms: one for generalization error and empirical error,

$$\begin{aligned}\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] &= \mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})\right] + \mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta\in\Theta} L_{\mathcal{D}}(\theta)\right] + \mathbb{E}\left[\min_{\theta\in\Theta} L_{\mathcal{D}}(\theta) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] \\ &\leq \mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})\right] + \mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta\in\Theta} L_{\mathcal{D}}(\theta)\right]. \qquad (9.4)\end{aligned}$$

where the second inequality holds since $\mathbb{E}[\min_{\theta\in\Theta} L_{\mathcal{D}}(\theta)] \le \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)$.

For the empirical quantity (the second quantity in Equation (9.4)), first note that the error caused by randomized smoothing is Gr (Lemma 9.2.5), hence

$$\mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta\in\Theta} L_{\mathcal{D}}(\theta)\right] \le \mathbb{E}\left[\widehat{L}_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta\in\Theta} \widehat{L}_{\mathcal{D}}(\theta)\right] + 2Gr.$$

As our algorithm basically applies noisy SGD over $\widehat{L}_{\mathcal{D}}$, we now use Proposition 9.4.10 to bound the empirical error. By Lemma 9.3.7 and Theorem 9.3.2, we have

$$\overline{g}_t \sim_{\delta/Tnmd} \nabla\widehat{L}_{\mathcal{D}}(\theta_{t-1}) + \zeta,$$

where $\zeta \sim \mathcal{N}(0, \frac{G^2T\log^2(Tmnd/\delta)}{mn^2\varepsilon^2})$. Hence we know the variance of the stochastic (sub)gradients we get is bounded by $O(\frac{G^2Td\log^2(Tmnd/\delta)}{mn^2\varepsilon^2})$. Moreover, we know that $\widehat{\ell}$ is $\frac{G\sqrt{d}}{r}$-smooth by Lemma 9.2.5. Thus, Proposition 9.4.10 now implies that

$$\mathbb{E}[\widehat{L}_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta} \widehat{L}_{\mathcal{D}}(\theta)] \lesssim \left(\frac{G\sqrt{d}}{r} + \frac{1}{\eta}\right)\frac{R^2}{T} + \frac{\eta G^2 Td\log^2(Tmnd/\delta)}{mn^2\varepsilon^2} + \frac{GR\delta}{mnd},$$

where the term $\frac{GR\delta}{mnd}$ comes from the failure probability.

Now we proceed to upper bound the generalization error (first quantity in Equation (9.4)). Combining Lemma 9.4.7 and Lemma 9.4.9, and the assumption that the functions are G-Lipschitz, we get

$$\mathbb{E}[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})] \le G^2\eta\sqrt{T} + \frac{G^2\eta T}{nm} + \frac{GR\delta}{mnd}.$$

Overall, combining these together and putting them back into Equation (9.4), we get

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim \frac{G\sqrt{d}R^2}{rT} + \frac{R^2}{\eta T} + \frac{\eta G^2 Td\log^2(Tmnd/\delta)}{mn^2\varepsilon^2} + Gr + G^2\eta\sqrt{T} + \frac{G^2\eta T}{nm} + \frac{GR\delta}{mnd}.$$

Optimizing the above parameters by setting $r = \frac{d^{1/4}R}{\sqrt{T}}$, $\eta = \frac{R}{G}\cdot\min\{\frac{\sqrt{mn}\varepsilon}{T\sqrt{d\log^2(Tmnd/\delta)}}, \frac{1}{T^{3/4}}, \sqrt{nm}/T\}$,

we get

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim GR \cdot \left(\frac{d^{1/4}}{\sqrt{T}} + \frac{1}{T^{1/4}} + \frac{\sqrt{d\log^2(Tmnd/\delta)}}{\sqrt{mn}\varepsilon} + \frac{1}{\sqrt{nm}}\right).$$

By setting $T = O(m^2n^2 + mn\sqrt{d})$, we have

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim GR \cdot \left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d\log^2(ndm/\delta)}}{n\varepsilon\sqrt{m}}\right),$$

which completes the proof. □

9.4.1 Implication for Strongly convex functions

Building on our optimal algorithm for the convex setting, in the section, we proceed to obtain optimal rates for the strongly convex case using the localization framework [FKT20]. The idea is to iteratively run Algorithm 29 for $\log\log(mn)$ rounds, where at each round, we run it with improved parameters. We present the details in Algorithm 30, and defer the full proof to the supplement with detailed parameter settings therein.

Algorithm 30: User-level DP-SCO for strongly convex functions

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n) \in (\mathcal{Z}^m)^n$, privacy parameters (ε, δ), initial point θ_0;
Set $k = \lceil \log\log mn \rceil$;
Divide $\mathcal{D}$ into k disjoint datasets $\{\mathcal{D}_i\}_{i\in[k]}$, where $\mathcal{D}_i$ is of size $n_i := n/2^{k+1-i}$;
for $i = 1, \cdots, k$ **do**
    Run Algorithm 29 with $\mathcal{D}_i, \varepsilon, \delta, \theta_{i-1}, r_i, T_i, \eta_i, \tau_i, \widehat{R}_i$ as inputs, and get its output θ_i;
end
Output: $\widehat{\theta} = \theta_k$;

Theorem 9.4.11 (Strongly convex case)**.** *For $0 < \varepsilon < 10, 0 < \delta < 1$, Algorithm 30 is user-level (ε, δ)-DP. Under the same assumptions as in Theorem 9.4.1, additionally assuming that $n > \frac{\log(mdn)\log(mdn/\delta)}{\varepsilon}$ and the functions are μ-strongly convex, then with proper parameter*

settings, Algorithm 30 outputs $\widehat{\theta}$ *such that*

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star \in \Theta} L_{\mathcal{P}}(\theta^\star)\right] \leq O\left(\frac{G^2}{\mu} \cdot \left(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2 m\varepsilon^2}\right)\right).$$

9.5 Conclusion

In this work, we have studied user-level DP-SCO and proposed new efficient algorithms that obtain near-optimal rates even in the non-smooth setting. There remain open questions in this domain. First, our rates are optimal up to logarithmic factors and we leave it for future work to improve these factors. Moreover, our algorithms require the number of rounds $T \geq n^2m^2 \cdot \min(1, n^2/d)$, and it remains open whether there is a more efficient algorithm. In particular, are there linear time algorithms for user-level DP-SCO in the smooth setting, similar to the item-level setting where such results are known [FKT20]?

9.6 Missing Proofs in Section 9.3

9.6.1 Proof of Lemma 9.3.3

Lemma 9.3.3. *For any neighboring dataset* $\mathcal{D}, \mathcal{D}'$ *that differs in one user, let* $p_i = (p_{i,1}, \cdots, p_{i,n})$ *be the probability for users to be selected into* S_i *for* $\mathcal{D}$*, and let* p'_i *be the corresponding probability for* $\mathcal{D}'$*. Then*

$$\|p_i - p'_i\|_1 \leq 2.$$

Proof. Without loss of generality, let $\mathcal{D} = (Z_1, Z_2, \ldots, Z_n)$ and $\mathcal{D}' = (Z'_1, Z_2, \ldots, Z_n)$ differ in the first user. Note that $f_{i,j}$ has sensitivity 1 for $j \neq 1$, hence $|p_{i,j} - p'_{j,j}| \leq 1/n$ for all $j \neq 1$. Moreover, $|p_{i,1} - p'_{i,1}| \leq 1$. Therefore, $\|p_i - p'_i\|_1 \leq 2$. □

9.6.2 Proof of Lemma 9.3.4

Lemma 9.3.4. *Let* $p, p' \in [0,1]^n$ *such that* $\|p - p'\|_1 \leq 2$*, and let* V *and* V' *be drawn from* $\mathrm{Ber}(p)$ *and* $\mathrm{Ber}(p')$ *respectively. For any* $\zeta \in (0,1)$*, there exists a coupling* Γ *over* V *and* V'

such that for (x, y) *drawn from* Γ*, with probability at least* $1 - \zeta$*,*

$$\|x - y\|_1 \leq O(\log(1/\zeta)).$$

Proof. We construct the coupling by considering each coordinate separately. Let p_i and p'_i be the i-th coordinate of p and p' respectively. Consider i-th coordinate, without losing generality, let $p_i \geq p'_i$. Then, we set

$$(x_i, y_i) = \begin{cases} (1,1), & \text{w.p. } p'_i \\ (1,0), & \text{w.p. } p_i - p'_i \\ (0,0), & \text{w.p. } 1 - p_i \end{cases}$$

And coordinates are independent of each other. We draw (x, y) from the coupling Γ, and set $X_i = 1$ if $x_i = y_i$ and, $X_i = 0$ otherwise. Hence we know $\{X_i\}$ are independent Bernoulli random variables such that $\mathbb{E}[X_i] = |p_i - p'_i|$. By Lemma 9.2.8, we know

$$\Pr[\|x - y\|_1 \geq O(\log(1/\zeta))] = \Pr[\sum_i X_i \geq O(\log(1/\zeta))] \leq \zeta.$$

This completes the proof. □

9.6.3 Proof of Lemma 9.3.5

Recall that E corresponds to the absolute values of the Laplacian noise used in AboveThreshold are bounded. Define E' to be the event w.r.t. input $\mathcal{D}'$.

Lemma 9.3.5. *For any i-th iteration and any neighboring datasets $\mathcal{D}, \mathcal{D}'$, conditional on E and E' and conditional on $a_i = a'_i$, there exists a coupling Γ_i over g_i and g'_i, such that for (x, y) drawn from Γ_i, with probability at least $1 - \zeta$,*

$$\|x - y\|_2 \lesssim \frac{\tau \log(1/\zeta)}{n}.$$

Proof. If $a_i = a'_i = \perp$, then both g_i and g'_i will be $\mathbf{0}$.

Consider the non-trivial case when $a_i = a_i' = \top$. As $s_i^{\mathsf{conc}}(\mathcal{D}, \tau) > \frac{2n}{3}$, we know there exists $Z^* \in \mathcal{D}$ such that $\sum_{Z \in \mathcal{D}} \mathcal{I}(\|q_i(Z^*) - q_i(Z)\| \leq \tau) \geq \frac{2n}{3}$. Let $H_i = \{Z \in \mathcal{D} : \|q_i(Z) - q_i(Z^*)\| \leq \tau\}$ be the set of users whose queried values are close to Z^*. We know $H_i \subset S_i$. Moreover, we can argue for any $Z \in S_i$, $\|q_i(Z) - q_i(Z^*)\| \leq 4\tau$. The same argument holds for $\mathcal{D}'$, that is there exists $Z'^* \in \mathcal{D}'$, such that $H_i' \subset S_i'$ and for any $Z \in S_i', \|q_i(Z) - q_i(Z'^*)\| \leq 4\tau$.

We know $\|q_i(Z^*) - q_i(Z'^*)\| \leq 2\tau$, as there exists Z in $\mathcal{D} \cap \mathcal{D}'$ such that $\|q_i(Z^*) - q_i(Z)\| \leq \tau$ and $\|q_i(Z'^*) - q_i(Z)\| \leq \tau$. Hence for any point $Z_1, Z_2 \in S_i \cup S_i'$, $\|q_i(Z_1) - q_i(Z_2)\| \leq 10\tau$.

Note that $g_i = \frac{1}{|S_i|} \sum_{Z \in S_i} q_i(Z)$ and $g_i' = \frac{1}{|S_i'|} \sum_{Z \in S_i'} q_i(Z)$. By Lemma 9.3.3 and Lemma 9.3.4, we know there exists a Coupling Γ_i over S_i and S_i' such that if we draw (S, S') from Γ_i, with probability at least $1 - \zeta$, we have

$$\|S - S'\|_0 \lesssim \log(1/\zeta).$$

Assume $|S'| \geq |S|$ without loss of generality and let $Z_0 \in S$. Note that we have

$$\begin{aligned}
&\|g_i - g_i'\|_2 \\
&= \left\| \frac{1}{|S|} \sum_{Z \in S} q_i(Z) - \frac{1}{|S'|} \sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|} \left\| \frac{|S'|}{|S|} \sum_{Z \in S} q_i(Z) - \sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|} \left\| \frac{|S'| - |S|}{|S|} \sum_{Z \in S} q_i(Z) + \sum_{Z \in S} q_i(Z) - \sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|} \left\| \frac{|S'| - |S|}{|S|} \sum_{Z \in S} q_i(Z) + \sum_{Z \in S \setminus S'} q_i(Z) - \sum_{Z \in S' \setminus S} q_i(Z) \right\|_2 \\
&\leq \frac{1}{|S'|} \left\| \frac{|S'| - |S|}{|S|} \sum_{Z \in S} q_i(Z) + \sum_{Z \in S \setminus S'} q_i(Z) - |S' \setminus S| \cdot q_i(Z_0) \right\|_2 + \frac{1}{|S'|} \left\| |S' \setminus S| \cdot q_i(Z_0) - \sum_{Z \in S' \setminus S} q_i(Z) \right\|_2 \\
&\overset{(i)}{=} \frac{1}{|S'|} \left\| \frac{|S'| - |S|}{|S|} \sum_{Z \in S} (q_i(Z) - q_i(Z_0)) + \sum_{Z \in S \setminus S'} (q_i(Z) - q_i(Z_0)) \right\|_2 + \frac{1}{|S'|} \left\| \sum_{Z \in S' \setminus S} (q_i(Z_0) - q_i(Z)) \right\|_2
\end{aligned}$$

$$\overset{(ii)}{\leq} \frac{10\tau}{|S'|} \cdot \big((|S'| - |S|) + |S \setminus S'| + |S' \setminus S|\big)$$
$$\overset{(iii)}{\lesssim} \frac{\tau \log(1/\zeta)}{n}.$$

where (i) follows since $|S'|-|S|+|S\setminus S'| = |S'\setminus S|$, and (ii) follows since $\max_{Z_1,Z_2\in S\cup S'} \|q_i(Z_1)-q_i(Z_2)\|_2 \leq 10\tau$, and (iii) follows since $\|S - S'\|_0 \lesssim \log(1/\zeta)$ and hence $|S'| - |S| + |S' \setminus S| + |S \setminus S'| \lesssim \log(1/\zeta)$.

This completes the proof.

□

9.6.4 Proof of Proposition 9.3.6

Proposition 9.3.6. *For any dataset $\mathcal{D}$, if $n \geq \frac{40\log(4T/\delta)}{\varepsilon}$, then for any neighboring dataset $\mathcal{D}'$, the outputs of Algorithm 28 with $\mathcal{D}$ and $\mathcal{D}'$ as inputs are (ε, δ)-indistinguishable.*

Proof. Let $\{a_i\}_{i\in T} = \{\top, \perp\}^T$ be the outputs of Algorithm 27 with input $\mathcal{D}$, where if $a_i = \perp$ we set $a_j = \perp$ for all $j \geq i$. Define the $\{a_i'\}$ correspondingly with input $\mathcal{D}'$. Then by Theorem 9.2.4, we know $\{a_i\}$ and $\{a_i'\}$ are $(\varepsilon/2, 0)$-indistinguishable.

Now conditional on that E and E' hold. If $a_i = a_i' = \perp$, then the algorithm halts and outputs the initial point, hence no privacy leakage.

Our proof proceeds by assuming the Gaussian noise v_i we add is drawn from $\mathcal{N}(0, \frac{4\tau^2 T \log(1/\zeta')\log(1/\delta')}{n^2\varepsilon^2})$. Then the statement follows from setting ζ' and δ'.

Under the assumption on $n \geq \frac{40\log(2T/\zeta')}{\varepsilon}$, for any $b \in \{\top, \perp\}^T$, by Lemma 9.3.5, the Union Bound, we know there exists a coupling over $\{g_i\}_{i\in T}$ and $\{g_i'\}_{i\in T}$, such that for $(\{x_i\}, \{y_i\})$ drawn from Γ, with probability at least $1 - T\zeta'$,

$$\text{for all } i \in [T], \|x_i - y_i\| \lesssim \frac{\tau \log(1/\zeta')}{n}.$$

By the guarantee of the Gaussian Mechanism and the composition [BS16], we know

$$\Pr[\{g_i + \nu_i\} \in \mathcal{O} \mid E, \{a_i\} = b] \leq e^{\varepsilon/2} \Pr[\{g_i' + \nu_i'\} \in \mathcal{O} \mid E', \{a_i'\} = b] + \delta' + T\zeta',$$

where we note that the Gaussian noise of $\{\nu_i\}$ and $\{\nu'_i\}$ are independent of the Laplacian noise we add in Algorithm 27.

To conclude, letting $\{g_i + \nu_i\}$ be the sequence of output, we have for any event $\mathcal{O}$,

$$\begin{aligned}
\Pr[\{g_i+\nu_i\} \in \mathcal{O}] &= \Pr[\{g_i+\nu_i\} \in \mathcal{O} \mid E] \Pr[E] + \Pr[\{g_i+\nu_i\} \in \mathcal{O} \mid \neg E] \Pr[\neg E] \\
&\leq \Pr[\{g_i+\nu_i\} \in \mathcal{O} \mid E] \Pr[E] + \zeta' \\
&= \sum_{b \in \{\top,\bot\}^T} \Pr[\{g_i+\nu_i\} \in \mathcal{O} \mid E, \{a_i\} = b] \Pr[E, \{a_i\} = b] + \zeta' \\
&\leq \sum_{b \in \{\top,\bot\}^T} e^{\varepsilon/2} (\Pr[\{g'_i+\nu'_i\} \in \mathcal{O} \mid E', \{a'_i\} = b] + \delta') \Pr[E, \{a_i\} = b] + (T+1)\zeta' \\
&\leq \sum_{b \in \{\top,\bot\}^T} e^{\varepsilon/2} \Pr[\{g'_i+\nu'_i\} \in \mathcal{O} \mid E', \{a'_i\} = b] \Pr[E, \{a_i\} = b] + (T+1)\zeta' + e^{\varepsilon/2}\delta'.
\end{aligned}$$

Note that the randomness of $\{a_i\}$ and whether E holds comes from the Laplacian variables we draw. By the privacy guarantee of AboveThreshold, for any $b \in \{\top,\bot\}^T$, we have

$$\Pr[\{a_i\} = b] \leq e^{\varepsilon/2} \Pr[\{a'_i\} = b].$$

It is not hard to observe that

$$\Pr[\{a_i\} = b, E] \leq e^{\varepsilon/2} \Pr[\{a'_i\} = b, E'] + e^{\varepsilon/2}\zeta'.$$

Hence

$$\begin{aligned}
&\Pr[\{g_i+\nu_i\} \in \mathcal{O}] \\
\leq &\sum_{b \in \{\top,\bot\}^T} e^{\varepsilon/2} \Pr[\{g'_i+\nu'_i\} \in \mathcal{O} \mid E', \{a'_i\} = b] \Pr[E, \{a_i\} = b] + (T+1)\zeta' + e^{\varepsilon/2}\delta' \\
\leq &\sum_{b \in \{\top,\bot\}^T} e^{\varepsilon} \Pr[\{g'_i+\nu'_i\} \in \mathcal{O} \mid E', \{a'_i\} = b] \Pr[E', \{a'_i\} = b] + (T+1+e^{\varepsilon})\zeta' + e^{\varepsilon/2}\delta'
\end{aligned}$$

Setting $\zeta' = \frac{\delta}{2(e^{\varepsilon}+1+T)}$ and $\delta' = \frac{\delta}{2e^{\varepsilon/2}}$, we get the Noise scale as stated in the pseudo-code of Algorithm 28 and complete the proof. □

9.7 Missing Proof in Section 4

9.7.1 Proof of Lemma 9.4.4

Lemma 9.4.4. *For any fixed θ and for each Z_i, if each item in Z_i is drawn i.i.d. from $\mathcal{P}$, with probability at least $1-\gamma/n$, we have*

$$\|\nabla\widehat{\ell}(\theta;Z_i)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\le\frac{G\log(nd/\gamma)}{\sqrt{m}},$$

Proof. The lemma follows from the concentration of Norm Subgaussian random variables (Lemma 9.2.7). Specifically, we know for each $z_{i,j}\in Z_i$, $\mathbb{E}\,\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)=0$, and $\|\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\le 2G$, which implies $\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)$ is zero-mean and nSG$(2G)$. The statement follows. □

9.7.2 Proof of Lemma 9.4.6

Lemma 9.4.6 (Similar to Theorem 3.4 in [BS23])**.** *Suppose $\mathcal{D}=\{z_{i,j}\}_{i\in[n],j\in[m]}$ are drawn i.i.d. from the distribution $\mathcal{P}$. In Algorithm 29, for all $t\in[T]$, $\{\nabla\widehat{\ell}(\theta_t;Z_i)\}_{Z_i\in\mathcal{D}}$ is (τ,γ')-concentrated for*

$$\tau=\frac{G\log(nd/\gamma)}{\sqrt{m}},\gamma'=T(e^{2\varepsilon}\gamma+\frac{\delta}{2Tmnd}).$$

Proof. It suffices to prove that for each $t\in[T]$, $\{\nabla\widehat{\ell}(\theta_t;Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau,e^{2\varepsilon}\gamma+\frac{\delta}{2Tmnd})$-concentrated. Note that by Theorem 9.3.2 and the parameter settings in the precondition, Algorithm 29 is user-level $(\varepsilon,\frac{\delta}{2Tnmd})$-DP. Then there exists an $(2\varepsilon,0)$-DP $\mathcal{A}'$ such that $d_{TV}(\mathcal{A}(\mathcal{D}),\mathcal{A}'(\mathcal{D}))\le\delta/2Tmnd$ by Lemma 9.4.5. Let $\{\theta'_t\}_{t\in[T]}$ be the output of $\mathcal{A}'(\mathcal{D})$. It suffices to show for any $t\in[T]$, $\{\nabla\widehat{\ell}(\theta'_t;Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau,e^{2\varepsilon}\gamma)$-concentrated.

Let $f_{Z_i}(Z)$ be the density of $Z_i=Z$ and and $f_{Z_i}(Z\mid\theta'_t=\theta)$ be the density conditional on $\theta'_t=\theta$. Similarly, we let $f_{\theta'_t}(\theta)$ and $f_{\theta'_t}(\theta\mid Z_i=Z)$ be the (conditional) density of θ'_t. For any θ,Z, we have

$$\frac{f_{Z_i}(Z\mid\theta'_t=\theta)}{f_{Z_i}(Z)}=\frac{f_{\theta'_t}(\theta\mid Z_i=Z)}{f_{\theta'_t}(\theta)}\le e^{2\varepsilon},$$

where the last inequality comes from the privacy guarantee of $\mathcal{A}'$.

One has

$$\begin{aligned}
&\Pr_{Z_i,\theta'_t}\left[\|\nabla\widehat{\ell}(\theta'_t;Z_i)-\nabla\widehat{L}_{\mathcal{P}}(\theta'_t)\|\geq\tau\right]\\
=&\int\int f_{\theta'_t}(\theta)f_{Z_i}(Z\mid\theta'_t=\theta)\mathcal{I}(\|\nabla\widehat{\ell}(\theta;Z)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\geq\tau)\mathrm{d}Z\mathrm{d}\theta\\
\leq& e^{2\varepsilon}\int\int f_{\theta'_t}(\theta)f_{Z_i}(Z)\mathcal{I}(\|\nabla\widehat{\ell}(\theta;Z)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\geq\tau)\mathrm{d}Z\mathrm{d}\theta.
\end{aligned}$$

Note that for any θ, we have

$$\int f_{Z_i}(Z)\mathcal{I}(\|\nabla\widehat{\ell}(\theta;Z)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\geq\tau)\mathrm{d}Z\leq\gamma/n.$$

Then by union bound, we know $\{\nabla\widehat{\ell}(\theta'_t;Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau,e^{2\varepsilon}\gamma)$-concentrated which completes the proof as $d_{TV}(\mathcal{A}(\mathcal{D}),\mathcal{A}'(\mathcal{D}))\leq\delta/2Tmnd$. □

9.7.3 Proof of Lemma 9.4.9

Lemma 9.4.9 (Algorithmic stability bound)**.** *Suppose $\{Z_i\}$ are drawn i.i.d. from the underlying distribution $\mathcal{P}$. Suppose $\tau\geq\frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $n\gtrsim\frac{\log(mdn/\delta)}{\varepsilon}$, with probability at least $1-\frac{\delta}{mnd}$, the stability of Algorithm 29 is bounded as follows:*

$$\Lambda(\mathcal{A})\leq G\eta\sqrt{T}+\frac{G\eta T}{nm}.$$

Proof. We use Lemma 9.4.8 to upper bound the stability of our algorithm. As we are using fixed step sizes $\eta_t=\eta$, Lemma 9.4.8 implies that

$$\begin{aligned}
\Lambda(\mathcal{A})&\leq 2G\sqrt{\sum_{t\in[T-1]}\eta_t^2}+2\sum_{t\in[T-1]}\eta_t a_t\\
&\leq 2G\eta\sqrt{T}+2\eta\sum_{t\in[T-1]}a_t
\end{aligned}$$

Thus it suffices to upper bound a_t for all $t\in[T]$.

By Lemma 9.4.6, we know for all $t \in [T]$, $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i \in \mathcal{D}}$ is (τ, γ')-concentrated for

$$\tau = \frac{G\log(nd/\gamma)}{\sqrt{m}}, \gamma' = T(e^{2\varepsilon}\gamma + \frac{\delta}{2Tmnd}).$$

Then by Theorem 9.3.2 and Lemma 9.3.7, we know

$$\overline{g}_t \sim_{2\gamma'} \frac{1}{nm}\sum_{Z_i \in \mathcal{D}}\sum_{z_{i,j} \in Z_i} \nabla\widehat{\ell}(\theta_t + y_j; z_{i,j}) + \nu,$$

where ν is Gaussian noise independent of the data. Thus we have $a_t \leq \frac{G}{nm}$. Setting $\gamma = \frac{\delta}{2Te^{2\varepsilon}}$ completes the proof. □

9.7.4 Proof of Theorem 9.4.11

Theorem 9.4.11 (Strongly convex case)**.** *For $0 < \varepsilon < 10, 0 < \delta < 1$, Algorithm 30 is user-level (ε, δ)-DP. Under the same assumptions as in Theorem 9.4.1, additionally assuming that $n > \frac{\log(mdn)\log(mdn/\delta)}{\varepsilon}$ and the functions are μ-strongly convex, then with proper parameter settings, Algorithm 30 outputs $\widehat{\theta}$ such that*

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star \in \Theta} L_{\mathcal{P}}(\theta^\star)\right] \leq O\left(\frac{G^2}{\mu}\cdot\left(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2m\varepsilon^2}\right)\right).$$

Proof. Let $L^*_{\mathcal{P}} = \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)$, $\Delta_i := \mathbb{E}[L_{\mathcal{P}}(\theta_i) - L^*_{\mathcal{P}}]$ and $R_i^2 := \mathbb{E}[\|\theta_i - \theta^*\|^2]$. Due to the strong convexity, we know $\frac{1}{2}\mu R_i^2 \leq \Delta_i$.

Let $C > 2$ be the constant hidden in the population loss bound in Theorem 9.4.1. For $i \geq 0$, define $E_i := \frac{4C^2G^2}{\mu}(\frac{1}{n_i m} + \frac{d\log^2(n_i dm/\delta)}{n_i^2\varepsilon^2 m})$ and we know $E_i/E_{i+1} \leq 4$. Define $D_i = 16E_i\sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{16E_0}}$. By the definition, we know

$$\frac{D_{i+1}}{16E_{i+1}} = \sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{16E_0}} \leq \sqrt{\frac{D_i}{16E_i}},$$

$$\sqrt{D_iE_{i+1}} = 4\sqrt{E_iE_{i+1}}\sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{4E_1}} \leq 16E_{i+1}\sqrt[2^{i+1}]{\frac{2G^2}{\mu}\cdot\frac{1}{4E_1}} = D_{i+1}.$$

Hence by setting $k \geq \log\log(D_1/(16E_1))$, then $\frac{D_k}{16E_k} \leq 2$. Note that $E_0 \geq \frac{4C^2G^2}{\mu nm}$,

and $D_0 = \frac{2G^2}{\mu}$. We get $\frac{D_0}{16E_0} \leq mn$ and setting $k = \log\log(mn)$ is large enough to get $D_k \leq 32E_k$. Note that $\Delta_0 \leq \frac{2G^2}{\mu}$ and $R_0 \leq \frac{2G}{\mu}$ by the strong convexity and assumption on being Lipschitz.

For $j \geq 1$, set $\widehat{R}_j = \sqrt{2D_{j-1}/\mu}$, $r_j = \frac{d^{1/4}\widehat{R}_j}{\sqrt{T_j}}, \eta_j = \frac{\widehat{R}_j}{G} \cdot \min\{\frac{\sqrt{m}n_j\varepsilon}{T_j\sqrt{d\log^2(mn_jd/\delta)}}, \frac{1}{T_j^{3/4}}, \frac{\sqrt{n_jm}}{T_j}\}$, $\tau = \frac{G\log(n_jdme^{\varepsilon}T_j/\delta)}{\sqrt{m}}$ and $T_j = O(m^2n_j^2 + mn_j\sqrt{d})$. As $n_j \geq n/\log(nm) \geq \frac{\log(mdn/\delta)}{\varepsilon}$ by the precondition, $R_0 \leq \widehat{R}_1 = \frac{2G}{\mu}$, by Theorem 9.4.1 and our parameter setting, recursively we know

$$\begin{aligned}
\Delta_j \leq & CG\widehat{R}_j \cdot (\frac{1}{\sqrt{n_jm}} + \frac{\sqrt{d\log^2(n_jdm/\delta)}}{n_j\varepsilon\sqrt{m}}) \\
\leq & CG\sqrt{2D_{j-1}/\mu} \cdot (\frac{1}{\sqrt{n_jm}} + \frac{\sqrt{d\log^2(n_jdm/\delta)}}{n_j\varepsilon\sqrt{m}}) \\
\leq & CG\sqrt{2D_{j-1}/\mu} \cdot \sqrt{\frac{\mu E_j}{2C^2G^2}} \\
\leq & \sqrt{D_{j-1}E_j} \leq D_j,
\end{aligned}$$

where we used $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a+b)}$ for $a, b > 0$. We know $R_i \leq \sqrt{\frac{2\Delta_i}{\mu}} \leq \sqrt{\frac{2D_i}{\mu}} = \widehat{R}_{i+1}$ recursively as well.

After k-iteration, we have

$$\mathbb{E}[L_{\mathcal{P}}(\theta_k) - L_{\mathcal{P}}^*] = \Delta_k \leq D_k \leq \ 32E_k = O(\frac{G^2}{\mu}(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2\varepsilon^2m})).$$

The statement follows. □

Chapter 10

FASTER ALGORITHMS FOR USER-LEVEL PRIVATE STOCHASTIC CONVEX OPTIMIZATION

10.1 Introduction

The increasing ubiquity of machine learning (ML) systems in industry and society has sparked serious concerns about the privacy of the personal data used to train these systems. Much work has shown that ML models may violate individuals' privacy by leaking their sensitive training data [SSSS17, LLL+24a, LLL+24b]. For instance, large language models (LLMs) are vulnerable to black-box attacks that extract individual training examples [CTW+21]. *Differential privacy* (DP) [DMNS06] prevents ML models from leaking their training data.

The classical definition of differential privacy—*item-level differential privacy* [DR14]—is ill-suited for many modern applications. Item-level DP ensures that the inclusion or exclusion of any *one training example* has a negligible impact on the model's outputs. *If each person (a.k.a. user) contributes only one piece of training data*, then item-level DP provides a strong guarantee that each user's data cannot be leaked. However, in many modern ML applications, such as training LLMs on users' data in federated learning, each user contributes a large number of training examples [XZ24]. In such scenarios, the privacy protection that item-level DP provides for each user is insufficiently weak.

User-level differential privacy is a stronger privacy notion that addresses the above shortcoming of item-level DP. Informally, user-level DP ensures that the inclusion or exclusion of any *one user's entire training data* (m samples) has a negligible impact on the model's outputs. Thus, user-level DP provides a strong guarantee that no user's data can be leaked, even when users contribute many training examples.

A fundamental problem in (private) machine learning is *stochastic convex optimization*

(SCO): given a data set $\mathcal{D} = (Z_1, \ldots, Z_n)$ from n i.i.d. users, each possessing m i.i.d. samples from an unknown distribution $Z_i \sim P^m$ our goal is to approximately minimize the expected population loss

$$F(x) := \mathbb{E}_{z\sim P}[f(x, z)].$$

Here, $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is a loss function (e.g., cross-entropy loss), $\mathcal{X} \subset \mathbb{R}^d$ is the parameter domain, and $\mathcal{Z}$ is the data universe. We require that the output of the optimization algorithm $\mathcal{A} : \mathcal{Z}^{mn} \to \mathcal{X}$ satisfies user-level DP (Definition 10.1.3). We measure the accuracy of $\mathcal{A}$ by its *excess (population) risk*

$$\mathbb{E}F(\mathcal{A}(\mathcal{D})) - F^* := \mathbb{E}_{\mathcal{A},\mathcal{D}\sim P^{nm}} F(\mathcal{A}(\mathcal{D})) - \min_{x\in\mathcal{X}} F(x).$$

Given the practical importance of user-level DP SCO, it is unsurprising that many prior works have studied this problem. The work of [LSA+21] initiated this line of work, and provided an excess risk lower bound of $\Omega(1/\sqrt{nm} + \sqrt{d}/(\varepsilon n\sqrt{m}))$, where ε is the privacy parameter. However, their upper bound was suboptimal and required strong assumptions. The work of [BS23] gave an algorithm that achieves optimal risk for β-smooth losses with $\beta < (n/\sqrt{md} \wedge n^{3/2}/(d\sqrt{m}))$, provided that $n \geq \sqrt{d}/\varepsilon$ and $m \leq \max(\sqrt{d}, n\varepsilon^2/\sqrt{d})$. *These assumptions are restrictive* in large-scale applications with a large number of examples per user m or when the number of model parameters d is large. For example, in deep learning, we often have $d \gg n$ and an enormous smoothness parameter $\beta \gg 1$. Moreover, their algorithm requires $mn^{3/2}$ gradient evaluations, making it slow when the number of users n is large.[1] The work of [GKK+23a] gave another user-level DP algorithm that only requires $n \geq \log(d)/\varepsilon$, but unfortunately their algorithm does not run in polynomial-time.

To address the deficiencies of previous works on user-level DP SCO, the recent work [AL24] provided an algorithm that achieves optimal excess risk in polynomial-time, while also only requiring $n \geq \log(md)/\varepsilon$ users. Moreover, their algorithm also works for non-smooth losses. The drawback of [AL24] is that it is even *slower* than the algorithm of [BS23]: for β-smooth

[1] In the introduction, whenever ε does not appear, we are assuming $\varepsilon = 1$ to ease readability. For runtime bounds, we also assume $n = d$ to further simplify.

losses, their algorithm requires $\beta \cdot (nm)^{3/2}$ gradient evaluations; for non-smooth losses, their algorithm requires $(nm)^3$ evaluations.

Evidently, *the runtime requirements and parameter restrictions of existing algorithms for user-level DP SCO are prohibitive in many important ML applications.* Thus, an important question is:

> **Question 1.** Can we develop *faster* user-level DP algorithms that achieve *optimal* excess risk *without restrictive assumptions*?

Contribution 1. We give a positive answer to Question 1, providing a novel algorithm that achieves optimal excess risk using $\max\{\beta^{1/4}(nm)^{9/8}, \beta^{1/2}n^{1/4}m^{5/4}\}$ gradient computations for β-smooth loss functions, with any $\beta < \infty$ (theorem 10.3.2). For non-smooth loss functions, our algorithm achieves optimal excess risk using $n^{11/8}m^{5/4}$ gradient evaluations for non-smooth loss functions (theorem 10.4.1). *Our runtime bounds dominate those of all prior works* in every applicable parameter regime, by polynomial factors in n, m, and d. Moreover, our results only require $n^{1-o(1)} \geq \log(d)/\varepsilon$ users. See Table 10.1 for a comparison of our results vs. prior works. For example, *for non-smooth loss functions, our optimal algorithm is faster than the previous state-of-the-art [AL24] by a multiplicative factor of* $n^{13/8}m^{7/4}$. For smooth loss functions, our optimal algorithm is faster than [AL24] by a factor of $(nm)^{3/8}\beta^{3/4}$ (in the typical parameter regime when $n^7 \geq m$).

Loss Function	Reference	Gradient complexity	Assumptions
β-Smooth	[BS23]	$mn^{3/2}$	$\beta \leq \sqrt{n/m}$ & $m \leq \sqrt{d} \leq n$
	[AL24]	$\beta \cdot (mn)^{3/2}$	None
	Our Algorithm 33	$\beta^{1/4} \cdot (mn)^{9/8} + \beta^{1/4}n^{1/4}m^{5/4}$	None
Non-Smooth	[AL24]	$(mn)^3$	None
	Our Algorithm 33 (smoothed)	$n^{11/8}m^{5/4}$	None

Table 10.1: Optimal algorithms for user-level DP SCO. We omit logarithms, fix $L = R = 1 = \varepsilon$ and $n = d$.

Linear-Time Algorithms The "holy grail" of DP SCO is a *linear-time* algorithm with optimal excess risk, which is unimprovable both in terms of runtime and accuracy. In the

item-level DP setting, such algorithms are known to exist for smooth loss functions [FKT20, ZTOH22]. [AL24] posed an interesting open question: is there a *user-level* DP algorithm that achieves optimal excess risk in linear time for smooth functions? For our second contribution, we make progress towards answering this question.

Existing techniques for user-level DP SCO are not well-suited for linear-time algorithms. Indeed, the only prior non-trivial linear-time algorithm is the user-level LDP algorithm of [BS23, Algorithm 5].[2] Their algorithm can achieve excess risk $\approx 1/\sqrt{nm\varepsilon} + \sqrt{d}/(\sqrt{nm}\varepsilon)$. Unfortunately, however, their algorithm requires a very stringent assumption on the smoothness parameter $\beta < \sqrt{n^3/(md^3)}$, which is unlikely to hold for large-scale ML problems. Further, the result of [BS23] requires the number of users queried in each round to grow polynomially with the dimension d, and it assumes $m < d < n$. *These assumptions severely limit the applicability of [BS23, Algorithm 5] in practical ML scenarios.* This leads us to:

> **Question 2.** Can we develop a *linear-time* user-level DP algorithm with state-of-the-art excess risk, *without restrictive assumptions*?

Contribution 2. We answer Question 2 affirmatively in theorem 10.2.1: under a very mild requirement on the smoothness parameter $\beta < \sqrt{nmd}$, our novel linear-time algorithm achieves excess risk of $\approx 1/\sqrt{nm\varepsilon} + \sqrt{d}/(\sqrt{nm}\varepsilon)$. Moreover, our algorithm does not require the number of users to grow polynomially in the dimension d, and our result holds for any values of m, d, and n. Thus, our algorithm has excess risk matching that of [BS23], but is much more widely applicable.

10.1.1 Techniques

We develop novel techniques and algorithms to achieve new state-of-the-art results in user-level DP SCO. Before discussing our techniques, let us review the key ideas from prior works that we build on.

[2] It is trivial to achieve excess risk $\approx 1/\sqrt{nm} + \sqrt{d}/(\varepsilon n)$ with (ε, δ)-user-level, e.g. by applying *group privacy* to an optimal item-level DP algorithm such as [FKT20]. The error due to privacy in this bound does not decrease with m.

The goal of prior works [BS23, AL24] was to develop user-level analogs of DP-SGD [BFTGT19], which is optimal in the item-level setting. To do so, they observed that each user i's gradient $\frac{1}{m}\sum_{j=1}^{m}\nabla f(x, Z_{i,j})$ lies in a ball of radius $\approx 1/\sqrt{m}$ around the population gradient $\nabla F(x)$ with high probability, if the data is i.i.d ($Z_i \sim P^m$). Consequently, if the data is i.i.d., then replacing one user $Z_i \in \mathcal{D}$ by another user $Z_i' \in \mathcal{D}'$ will not change the empirical gradient $\nabla F_{\mathcal{D}}(x)$ by too much: $\|\nabla F_{\mathcal{D}}(x) - \nabla F_{\mathcal{D}'}(x)\| \lesssim 1/(n\sqrt{m})$ with high probability. Thus, one would hope for a method to privatize $\nabla F_{\mathcal{D}}(x)$ by adding noise that scales with $1/(n\sqrt{m})$—rather than $1/n$—which would allow for optimal excess risk. [AL24] devised such a method, which was inspired by FriendlyCore [TCK+22]. Their method privately detects and removes "outlier" user gradients, and then adds noise to the average of the "inlier" user gradients. This outlier-removal procedure ensures privacy with noise scaling with $1/(n\sqrt{m})$, provided $n \gtrsim 1/\varepsilon$. Moreover, when the data is i.i.d., no outliers will be removed with high probability, leading to a nearly unbiased estimator of the empirical gradient.

Our algorithms apply variations of the outlier-removal idea of [AL24] in novel ways.

Our linear-time Algorithm 31 takes a different approach to outlier removal, compared to prior works. Instead of removing outlier *gradients*, we aim to detect and remove outlier SGD *iterates*.[3] The high-level idea of our algorithm is to partition the n users into $C \approx 1/\varepsilon$ groups, with each group containing $\approx n\varepsilon$ users. For each group of users, we run $T \approx mn\varepsilon$ steps of online SGD using the samples in this group and obtain the average iterate of each group: $\{\tilde{x}_j\}_{j=1}^{C}$. We then *privately identify and remove the outlier iterates* from $\{\tilde{x}_j\}_{j=1}^{C}$. In order to successfully do so, we need to argue that if we run online SGD independently on user Z and user Z' to obtain $\tilde{x}$ and $\tilde{x}'$ respectively, then $\|\tilde{x} - \tilde{x}'\| \lesssim \eta\sqrt{T}$ with high probability, where η is the SGD step size. We prove such a stability bound in Lemma 10.2.3, which we hope will be of independent interest. By repeating the above process $\log(n)$ times and using iterative *localization* [FKT20], we obtain our state-of-the-art linear-time result.

Our second algorithm, Algorithm 33, builds on [AL24] in a different way. In Algorithm 33, we apply an outlier-removal procedure to users' gradients. However, un-

[3]The reason that this innovation is necessary is discussed in the last paragraph of Section 10.2.

like [AL24], we draw random *minibatches* of users in each iteration and apply outlier-removal to these minibatches. To make this procedure private while also achieving optimal excess risk, we combine *AboveThreshold* [DR14] with *privacy amplification by subsampling* [BBG18]. We then develop an *accelerated* [GL12] user-level DP algorithm that solves a carefully chosen sequence of regularized ERM problems, and applies localization in the spirit of [KLL21, AFKT21]. An obstacle that arises when we try to extend the ERM-based localization framework to the user-level DP setting is getting a tight bound on the variance of our minibatch stochastic gradient estimator that scales with $1/m$. We overcome this obstacle in Lemma 10.3.5, by appealing to the *stability of user-level DP* [BS23]. To handle non-smooth loss functions, we apply randomized smoothing to our accelerated algorithm.

10.1.2 Preliminaries

We consider loss functions $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$, where $\mathcal{X}$ is a convex parameter domain and $\mathcal{Z}$ is a data universe. Let P be an unknown data distribution and $F(x) := \mathbb{E}_{z\sim P}[f(x, z)]$ be the population loss function. Denote $F^* := \min_{x\in\mathcal{X}} F(x)$. The SCO problem is $\min_{x\in\mathcal{X}} F(x)$. Let $\|\cdot\|$ denote the ℓ_2 norm. $\Pi_{\mathcal{X}}(u) := \operatorname{argmin}_{x\in\mathcal{X}}\|u - x\|^2$ denotes projection onto $\mathcal{X}$.

Assumptions and Notation. Function $g : \mathcal{X} \to \mathbb{R}$ is *L-Lipschitz* if $|g(x) - g(x')| \le L\|x - x'\|_2$ for all $x, x' \in \mathcal{X}$. Function $g : \mathcal{X} \to \mathbb{R}$ is *β-smooth* if g is differentiable and has β-Lipschitz gradient: $\|\nabla g(x) - \nabla g(x')\|_2 \le \beta\|x - x'\|_2$. Function $g : \mathcal{X} \to \mathbb{R}$ is *μ-strongly convex* if $g(\alpha x + (1-\alpha)x') \le \alpha g(x) + (1-\alpha)g(x') - \frac{\alpha(1-\alpha)\mu}{2}\|x - x'\|^2$ for all $\alpha \in [0, 1]$ and all $x, x' \in \mathcal{X}$. If $\mu = 0$, we say g is *convex*.

Assumption 10.1.1. *1. The convex set $\mathcal{X}$ is compact with $\|x-x'\| \le R$ for all $x, x' \in \mathcal{X}$.*

2. The loss function $f(\cdot, z)$ is L-Lipschitz and convex for all $z \in \mathcal{Z}$.

In all of the paper *except for section 10.4*, we will also assume:

Assumption 10.1.2. *The loss function $f(\cdot, z)$ is β-smooth for all $z \in \mathcal{Z}$.*

Denote $a \wedge b := \min(a, b)$. For functions f and g of input parameters θ, we write $f \lesssim g$ if there is an absolute constant $C > 0$ such that $f(\theta) \le Cg(\theta)$ for all permissible values of

θ. We use $\widetilde{O}$ to hide logarithmic factors. Write $a \leq \text{poly}(b)$ if there exists some large $J > 1$ for which $a \leq b^J$.

Differential Privacy.

Definition 10.1.3 (User-Level Differential Privacy)**.** Let $\varepsilon \geq 0,\ \delta \in [0, 1)$. Randomized algorithm $\mathcal{A} : \mathcal{Z}^{nm} \to \mathcal{X}$ is (ε, δ)-*user-level differentially private* (DP) if for any two datasets $\mathcal{D} = (Z_1, \ldots, Z_n)$ and $\mathcal{D}' = (Z'_1, \ldots, Z'_n)$ that differ in one user's data (say $Z_i \neq Z'_i$ but $Z_j = Z'_j$ for $j \neq i$), we have

$$\mathbb{P}(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(\mathcal{D}') \in S) + \delta,$$

for all measurable subsets $S \subset \mathcal{X}$.

Definition 10.1.3 prevents any adversary from learning much more about an individual's data set than if that data had not been used for training. Appendix 10.6 contains the necessary background on DP.

10.1.3 Roadmap

We begin with our state-of-the-art linear-time algorithm in section 10.2. In section 10.3, we present our error-optimal algorithm with state-of-the-art runtime for smooth loss functions. section 10.4 extends our fast optimal algorithm to non-smooth loss functions. We conclude in section 10.5 with a discussion and guidance on future research directions stemming from our work.

10.2 A state-of-the-art linear-time algorithm for user-level DP SCO

In this section, we develop a new algorithm (Algorithm 31) for user-level DP SCO that runs in linear time and has state-of-the-art excess risk, without requiring any impractical assumptions. The algorithm can be seen as a user-level DP variation of the localized phased SGD of [FKT20]: we execute a sequence of SGD trajectories with geometrically decaying step sizes, shrinking both the expected distance to the population minimizer and the privacy noise over a logarithmic number of phases.

In each phase i, we first re-set algorithmic parameters and draw a disjoint set of n_i users $D_i \subset \mathcal{D}$ (lines 4-5). We further partition D_i into C disjoint subsets $\{D_{i,j}\}_{j=1}^{C}$. For each $j \in [C]$, we pool together all of the $n_i m$ samples in $D_{i,j}$ and run one-pass online SGD on $D_{i,j}$ with initial point x_{i-1} given to us from the previous phase. Next, in lines 10-20, we privately detect and remove "outliers" from $\{\tilde{x}_{i,j}\}_{j=1}^{C}$. That is, our goal is to privately select a subset $\mathcal{S}_i \subset \{\tilde{x}_{i,j}\}_{j=1}^{C}$, such that for any two points $\tilde{x}_{i,j}, \tilde{x}_{i,j'} \in \mathcal{S}_i$, $\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \le \tau_i = \widetilde{O}(\eta_i L\sqrt{T_i})$. This will enable us to add noise scaling with τ_i in line 22, rather than with the much larger worst-case sensitivity (that scales linearly with T_i). In order to privately select such a subset $\mathcal{S}_i$, we first compute (and privatize) the *concentration score* for $\{\tilde{x}_{i,j}\}_{j=1}^{C}$ in line 10. A small concentration score indicates that outlier removal is doomed to fail and we must halt the algorithm to avoid breaching the privacy constraint. A large concentration score indicates that $\{\tilde{x}_{i,j}\}_{j=1}^{C}$ is nearly τ_i-concentrated and we may proceed with outlier removal in lines 12-15.

Theorem 10.2.1 (Privacy and utility of Algorithm 31 - Informal)**.** *Let $\varepsilon \le 10$, $n^{1-o(1)} \gtrsim \frac{\log(n/\delta)}{\varepsilon}$, $\beta \le (L/R)\sqrt{dmn\varepsilon}$, and $m \lesssim poly(n)$. Then, Algorithm 31 is (ε, δ)-user-level DP. Further,*

$$\mathbb{E}F(x_l) - F^* \le LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{nm\varepsilon}} + \frac{\sqrt{d\log(1/\delta)}}{\sqrt{n}\varepsilon\sqrt{m}}\right).$$

The gradient complexity of Algorithm 31 is $\le nm$.

Remark 10.2.2 (State-of-the-art excess risk in linear time, without the restrictive assumptions). Under the assumptions that $\beta < (L\varepsilon^3/R)\sqrt{n^3/md^3}$ and $m \le d/\varepsilon^2 \le n$, [BS23] gave a linear-time algorithm with similar excess risk to Algorithm 31. However, their assumptions are very restrictive in practice: For example, in the canonical regime $n \approx d$, their assumption on β rules out essentially every (non-linear) loss function. By contrast, our result holds even if the smoothness parameter is huge ($\beta \approx \sqrt{nmd}$) and we only require a logarithmic number of users. Thus, our algorithm and result is applicable to many practical ML problems.

To prove that Algorithm 31 is private, we essentially argue that for any phase i, the ℓ_2-sensitivity of $\tilde{x}_i$ is upper bounded by $\widetilde{O}(\tau_i/C)$ with probability at least $1 - \delta/2$. The

Algorithm 31: User-Level DP Phased SGD with Outlier Iterate Removal and Output Perturbation

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, privacy parameters (ε, δ), parameters $p, q > 0$, stepsize η;
Set $l = \lfloor \log_2(n) \rfloor$, $C := 100 \log(20nme^{\varepsilon}/\delta)/\varepsilon$;
for $i = 1, \cdots, l$ **do**
  Set $n_i = (1 - (1/2)^q)n/2^{iq}$, $\eta_i = \eta/2^{pi}$, $N_i = n_i/C$, $T_i = N_i m$, $\tau_i = 1000\eta_i L\sqrt{T_i}\log(ndm)$;
  Draw disjoint users D_i of size n_i from $\mathcal{D}$;
  Divide D_i into C disjoint subsets $\{D_{i,j}\}_{j=1}^{C}$, each containing $|D_{i,j}| = N_i$ users;
  for $j = 1, \cdots, C$ **do**
    $\tilde{x}_{i,j} \leftarrow SGD(D_{i,j}, \eta_i, T_i, x_{i-1}) =$ average iterate of T_i steps of one-pass projected SGD with data $D_{i,j}$, stepsize η_i, and initial point x_{i-1} ;
  end
  Compute the concentration score for D_i:

$$s_i(\tau_i) := \frac{1}{C} \sum_{j,j' \in [C]} \mathbf{1}(\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \le \tau_i)$$

  Let $\widehat{s}_i(\tau_i) = s_i(\tau_i) + \mathrm{Lap}(20/\varepsilon)$;
  if $\widehat{s}_i(\tau_i) \ge \frac{4C}{5}$ **then**
    $\mathcal{S}_i = \emptyset$;
    for $j = 1, \cdots, C$ **do**
      Compute the score function of $\tilde{x}_{i,j}$: $h_{i,j} = \sum_{j'=1}^{C} \mathcal{I}(\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \le 2\tau_i)$;
      Add $\tilde{x}_{i,j}$ to $\mathcal{S}_i$ with probability $p_{i,j}$ for $p_{i,j} = \begin{cases} 0 & h_{i,j} < C/2 \\ 1 & h_{i,j} \ge 2C/3 \\ \frac{h_{i,j} - C/2}{C/6} & o.w. \end{cases}$
    end
  end
  else
    Halt; Output 0
  end
  Let $\tilde{x}_i = \frac{1}{|\mathcal{S}_i|} \sum_{\tilde{x}_{i,j} \in \mathcal{S}_i} \tilde{x}_{i,j}$;
  $x_i \leftarrow \tilde{x}_i + \zeta_i$, where $\zeta_i \sim \mathcal{N}(0, \sigma_i^2 I_d)$ with $\sigma_i = \frac{100\tau_i \log^2(n/\delta)}{\varepsilon C}$;
end
Output: x_l.

argument goes roughly as follows: First, the Laplace noise added to $s_i(\tau_i)$ ensures that $s_i(\tau_i)$ is $\varepsilon/4$-user-level DP. Now, it suffices to assume $\widehat{s}_i(\tau_i) \geq 4C/5$, since otherwise the algorithm halts and outputs 0 independently of the data. Next, conditional on the high probability event that the Laplace noise is smaller than $\widetilde{O}(1/\varepsilon)$, we know that $\widehat{s}_i(\tau_i) \geq 4C/5 \implies s_i(\tau_i) \geq 2C/3$ with high probability by our choice of C. In this case, an argument along the lines of [AL24, Lemma 3.5] shows that $\tilde{x}_i$ has sensitivity bounded by $\widetilde{O}(\tau_i/C)$ with probability at least $1 - \delta/2$. See Appendix 10.7 for the detailed proof.

To prove the excess risk bound in Algorithm 31, the key step is to show that if the data is i.i.d., then with high probability, no points are removed from $\{\tilde{x}_{i,j}\}_{j=1}^{C}$ during the outlier-removal phase (i.e. $|\mathcal{S}_i| = C$). If $|\mathcal{S}_i| = C$ holds, then we can use the convergence guarantees of SGD and the localization arguments in [FKT20] to establish the excess risk guarantee. In order to prove that $|\mathcal{S}_i| = C$ with high probability, we need the following novel *stability* lemma:

Lemma 10.2.3. *Assume $f(\cdot, z)$ is convex, L-Lipschitz, and β-smooth on $\mathcal{X}$ with $\eta \leq 1/\beta$. Let $\tilde{x} \leftarrow SGD(D, \eta, T, x_0)$ and $\tilde{y} \leftarrow SGD(D', \eta, T, x_0)$ be two independent runs of projected SGD, where $D, D' \sim P^N$ are i.i.d. Then, with probability at least $1 - \zeta$, we have*

$$\|\tilde{x} - \tilde{y}\| \lesssim \eta L \sqrt{T \log(dT/\zeta)}.$$

We prove Lemma 10.2.3 via induction on t, using non-expansiveness of gradient descent on smooth losses [HRS16], subgaussian concentration bounds, and a union bound.

Lemma 10.2.3 implies that if the data is i.i.d., then the following events hold with high probability: $\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \leq \tau_i$ for all $j, j' \in [C_i]$ and hence $s_i(\tau_i) = C$. Further, conditional on $s_i(\tau_i) = C$, we know that $\widehat{s}_i(\tau_i) \geq 4C/5$ with high probability, so that the algorithm does not halt. Also, $\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \leq \tau_i$ for all j, j' implies $p_{i,j} = 1$ for all j and hence $|\mathcal{S}_i| = C$ for all j. The detailed excess risk proof is provided in Appendix 10.7.

Challenges of getting optimal excess risk in linear time: In the item-level DP setting, there are several (nearly) linear time algorithms that achieve optimal excess risk for

smooth DP SCO under mild smoothness conditions, such as snowball SGD [FKT20], phased SGD [FKT20], and phased ERM with output perturbation [ZTOH22]. Extending these approaches into optimal nearly linear-time user-level DP algorithms is challenging. First, the user-level DP implementation of output perturbation in [GKK+23a] is computationally inefficient. Second, snowball SGD relies on *privacy amplification by iteration*, which does not extend nicely to the user-level DP case due to instability of the outlier-detection procedure in [AL24]. Specifically, since amplification by iteration intermediate only provides DP for the last iterate x_T but not the intermediate iterates x_t $(t < T)$, the sensitivity of the concentration score function is not $O(1)$, which impairs DP outlier-detection. A similar instability issue arises if one tries to naively extend phased SGD to be user-level DP by applying [AL24] to user gradients. This issue motivates our Algorithm 31, which extends phased SGD in an alternative way: by applying outlier-detection/removal to the SGD *iterates* instead of the gradients, we can control the sensitivity of the concentration score and thus prove that our algorithm is DP. However, since the bound in Lemma 10.2.3 scales polynomially with T (and we believe this dependence on T is necessary), Algorithm 31 adds excessive noise and has suboptimal excess risk. We believe that obtaining optimal risk in linear time will require a fundamentally different user-level DP mean estimation procedure that does not suffer from the instability issue.

10.3 An optimal algorithm with $\approx (mn)^{9/8}$ gradient complexity for smooth losses

In this section, we provide an algorithm that achieves optimal excess risk using $\approx (mn)^{9/8}$ stochastic gradient evaluations. Our Algorithm 33 is inspired by the item-level accelerated phased ERM algorithm of [KLL21]. It applies iterative localization [FKT20] to the user-level DP accelerated minibatch SGD Algorithm 32. Algorithm 32 is a user-level DP variation of the accelerated minibatch SGD of [GL12, GL13a].

Our Algorithm 32 applies a DP outlier-removal procedure to the users' gradients in each iteration. We use *Above Threshold* [DR14] to privatize the concentration scores $s_i^{(t)}$ and determine whether or not most of the gradients of users in minibatch D_i^t are 2τ-close to each other. If $\widehat{s}_i^t \geq \widehat{\Delta}_i$, indicating that the gradients of users in D_i^t are nearly 2τ-

Algorithm 32: `User-Level DP Accelerated Minibatch SGD`$(\widehat{F}_i, T_i, K_i, x_{i-1}, \tau, \varepsilon, \delta)$

Initialize $x_{i-1}^1 \leftarrow x_{i-1}$;
for $t = 1, \cdots, T_i$ **do**
  Draw K_i random users $D_i^t = \{Z_{i,j}^t\}_{j=1}^{K_i}$ from D_i uniformly with replacement;
  Set noisy threshold $\widehat{\Delta}_i := \frac{4K_i}{5} + \xi_i$, where $\xi_i \sim \mathrm{Lap}\left(\frac{8}{\varepsilon}\right)$;
  Let $q_t(Z) := \frac{1}{m}\sum_{z \in Z} \nabla f(x_{i-1}^t, z)$ for user Z;
  Compute the concentration score of D_i^t:

$$s_i^t(\tau) := \frac{1}{K_i} \sum_{Z, Z' \in D_i^t} \mathbf{1}(\|q_t(Z) - q_t(Z')\| \le 2\tau)$$

  Let $\widehat{s}_i^t(\tau) = s_i^t(\tau_i) + v_i^t$, where $v_i^t \sim \mathrm{Lap}\left(\frac{16}{\varepsilon}\right)$;
  if $\widehat{s}_i^t(\tau) \ge \widehat{\Delta}_i$ **then**
    $\mathcal{S}_i^t = \emptyset$;
    for *Each User* $Z \in D_i^t$ **do**
      Set $h_i^t(Z) = \sum_{Z' \in D_i^t} \mathcal{I}(\|q_t(Z) - q_t(Z')\| \le 2\tau)$;
      Add Z to $\mathcal{S}_i^t$ with probability $p_i^t(Z) := \begin{cases} 0 & h_i^t(Z) < K_i/2 \\ 1 & h_i^t(Z) \ge 2K_i/3 \\ \frac{h_i^t(Z) - K_i/2}{K_i/6} & o.w. \end{cases}$
    end
    $g_i^t = \frac{1}{|\mathcal{S}_i^t|}\sum_{Z \in \mathcal{S}_i^t} \nabla \widehat{F}(x_{i-1}^t, Z)$;
    $\widehat{g}_i^t = g_i^t + \zeta_i^t$, where $\zeta_i^t \sim \mathcal{N}(0, \sigma_i^2)$ with $\sigma_i = \frac{1000\tau\sqrt{T_i}\log(nde^{\varepsilon}/\delta)}{\varepsilon n_i}$;
    Do 1 iteration of Accelerated Minibatch SGD (AC-SA) [GL12] on $\widehat{F}_i$, using gradient estimator $\widehat{g}_i^t + \lambda_i(x_{i-1}^t - x_{i-1})$ to obtain x_{i-1}^{t+1}.
  end
  else
    Halt; Return 0
  end
end
Output $x_{i-1}^{T_i}$.

Algorithm 33: User-Level DP Accelerated Phased ERM with Outlier Gradient Removal

1 **Input:** Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, privacy parameters (ε, δ), parameters $p, q, \lambda > 0$;
2 Set $l = \lfloor \log_2(n) \rfloor$ and $\tau = O(L \log(ndm)/\sqrt{m})$, choose any initial point $x_0 \in \mathcal{X}$;
3 **for** $i = 1, \cdots, l$ **do**
4 Set $n_i = (1 - (1/2)^q) n / 2^{iq}$, $\lambda_i = \lambda \cdot 2^{pi}$, $T_i = \widetilde{\Theta}(1 + \sqrt{\beta/\lambda_i})$, $K_i = 500 \log(n_i^2 m^2 e^{\varepsilon}/\delta) \left(\frac{1}{\varepsilon} + \frac{n_i \varepsilon}{\sqrt{T_i \log(1/\delta)}} \right)$;
5 Draw disjoint users D_i of size n_i from $\mathcal{D}$;
6 Let $\widehat{F}_i(x) := \frac{1}{n_i} \sum_{Z_{i,j} \in D_i} \widehat{F}(x, Z_{i,j}) + \frac{\lambda_i}{2} \|x - x_{i-1}\|^2$, where $\widehat{F}(x, Z_{i,j})$ is user $Z_{i,j}$'s empirical loss;
7 $x_i \leftarrow$ `User-Level DP Accelerated Minibatch SGD`$(\widehat{F}_i, T_i, K_i, x_{i-1}, \tau, \varepsilon, \delta)$. ;
8 **end**
9 **Output** x_l.

concentrated, then we proceed with outlier removal in lines 8-12. We then invoke *privacy amplification by subsampling* [BBG18] and the *advanced composition theorem* [KOV15] to privatize the average of the "inlier" gradients with additive Gaussian noise. By properly choosing algorithmic parameters, we obtain the following results, proved in Appendix 10.8:

Theorem 10.3.1 (Privacy of Algorithm 33)**.** *Let $\varepsilon \leq 10$, $q > 0$ such that $n^{1-q} > \frac{100 \log(20nmde^{\varepsilon}/\delta)}{\varepsilon(1-(1/2)^q)}$. Then, Algorithm 33 is (ε, δ)-DP.*

Theorem 10.3.2 (Utility & runtime of Algorithm 33 - Informal)**.** *Let $\varepsilon \leq 10$ and $\delta < 1/(mn)$. Then, Algorithm 33 yields optimal excess risk:*

$$\mathbb{E}F(x_l) - F^* \leq LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{mn}} + \frac{\sqrt{d \log(1/\delta)}}{\varepsilon n \sqrt{m}} \right).$$

The gradient complexity of this algorithm is upper bounded by

$$mn \left(1 + \varepsilon \left(\frac{\beta R}{L} \right)^{1/4} \left((mn)^{1/8} \wedge \left(\frac{\varepsilon^2 n^2 m}{d} \right)^{1/8} \right) \right) + \sqrt{\frac{\beta R}{L}} \left(\frac{n^{1/4} m^{5/4}}{\varepsilon} + \left(\frac{n^{1/2} m^{5/4}}{d^{1/4} \varepsilon^{1/2}} \right) \right).$$

If $n = d$, $\varepsilon = 1$, and $\beta R = L$ then the gradient complexity bound in theorem 10.3.2 simplifies to $(mn)^{9/8} + n^{1/4} m^{5/4}$. Typically, $n^7 \geq m$, so that the dominant term in this bound is $(mn)^{9/8}$.

Remark 10.3.3 (State-of-the-art runtime). The gradient complexity bound in theorem 10.3.2 is *superior to the runtime bounds of all existing near-optimal algorithms by polynomial factors* in n, m, and d [BS23, GKK+23a, AL24]. Note that while the $mn^{3/2}$ gradient complexity bound of [BS23] may *appear* to be better than $\beta^{1/4}(nm)^{9/8}$ in certain parameter regimes (e.g. $m > n^3$ or $\beta \gg nm$), this is not the case: the result of [BS23] requires $m < n$ and $\beta < \sqrt{n/m}$.

Remark 10.3.4 (Mild assumptions). Note that theorems 10.3.1 and 10.3.2 do not require any bound on the smoothness parameter β, and only require the number of users to grow logarithmically: $n^{1-o(1)} \geq \widetilde{\Omega}(1/\varepsilon)$. Contrast this with the results of previous works (e.g. [BS23]).

A challenge in proving theorem 10.3.2 is getting a tight bound on the variance of the the noisy minibatch stochastic gradients $\widehat{g}_i^t$ that are used in Algorithm 32 (lines 12-14). Conditional on $\mathcal{S}_i^t = D_i^t$, it is easy to obtain a variance bound of the form $\mathbb{E}\|\widehat{g}_i^t - \nabla \widehat{F}_i(x_i^t)\|^2 \lesssim d\sigma_i^2 + \frac{L^2}{K_i}$, since we are sampling K_i users uniformly at random. However, this bound is too weak to obtain theorem 10.3.2, since it does not scale with m. To prove theorem 10.3.2, we need the following stronger result:

Lemma 10.3.5 (Variance Bound for Algorithm 32)**.** *Let $\delta \leq 1/(nm), \varepsilon \lesssim 1$. Denote $\widetilde{F}_i(x) := \frac{1}{n_i}\sum_{Z_{i,j} \in D_i} \widehat{F}(x, Z_{i,j})$. Then, conditional on $\mathcal{S}_i^t = D_i^t$ for all $i \in [l], t \in [T_i]$, we have*

$$\mathbb{E}\|g_i^t - \nabla \widetilde{F}_i(x_{i-1}^t)\|^2 \lesssim \frac{L^2 \log(ndm)}{Km}$$

for all $i \in [l], t \in [T_i]$, where the expectation is over both the random i.i.d. draw of $\mathcal{D} = (Z_1, \ldots, Z_n) \sim P^{nm}$ and the randomness in Algorithm 33.

The difficulty in proving Lemma 10.3.5 comes from the fact that the iterates x_i^t and the data $\mathcal{D}$ are not independent. To overcome this difficulty, we use the *stability of user-level DP* [BS23] to argue that for all $Z \in D_i$, $\nabla \widehat{f}(x_{i-1}^t, Z)$ is $\approx L/\sqrt{m}$-close to $\nabla F(x_{i-1}^t)$ with high probability, since x_{i-1}^t is user-level DP. A detailed proof is given in Appendix 10.8.

Remark 10.3.6 (Strongly convex losses: Optimal excess risk with state-of-the-art runtime). If $f(\cdot, z)$ is μ-strongly convex, then Algorithm 33 can be combined with the meta-algorithm

of [FKT20, Section 5.1] to obtain optimal excess risk

$$\frac{L^2}{\mu} \cdot \widetilde{O}\left(\frac{1}{nm} + \frac{d\ln(1/\delta)}{\varepsilon^2 n^2 m}\right)$$

with the same gradient complexity stated in theorem 10.3.2. This improves over the previous state-of-the-art gradient complexity $\approx \beta(mn)^{3/2}$ of [AL24].

10.4 An optimal algorithm with subquadratic gradient complexity for non-smooth losses

In this section, we extend our accelerated algorithm from the previous section to non-smooth loss functions. To accomplish this with minimal computational cost, we apply *randomized (convolution) smoothing* [YNS12, DBW12] to approximate non-smooth f by a β-smooth $\tilde{f}$. We can then apply Algorithm 33 to $\tilde{f}$. Since convolution smoothing is by now a standard optimization technique, we defer the details and proof to Appendix 10.9.

Theorem 10.4.1 (Privacy and utility of smoothed Algorithm 33 for non-smooth loss - informal)**.** *Let $\varepsilon \leq 10$, $\delta < 1/(mn)$, and $q > 0$ such that $n^{1-q} > \frac{100\log(20nmde^{\varepsilon}/\delta)}{\varepsilon(1-(1/2)^q)}$. Then, applying Algorithm 33 to the smooth approximation of f yields optimal excess risk:*

$$\mathbb{E}F(x_l) - F^* \leq LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{mn}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon n\sqrt{m}}\right).$$

The gradient complexity of this algorithm is upper bounded by

$$mn\left(1 + n^{3/8}m^{1/4}\varepsilon^{1/4}\right).$$

Remark 10.4.2 (State-of-the-art gradient complexity). The only previous polynomial-time algorithm that can achieve optimal excess risk for non-smooth loss functions is due to [AL24]. The algorithm of [AL24] required $(nm)^3 + (mn)^2\sqrt{d}$ gradient evaluations. Thus, the gradient complexity of the smoothed version of Algorithm 33 offers a *significant improvement over the previous state-of-the-art.* For example, if $\varepsilon = 1$, then our algorithm is faster than the previous state-of-the-art by a multiplicative factor of at least $n^{13/8}m^{7/4}$.

10.5 Conclusion

In this paper, we developed new user-level DP algorithms with improved runtime and excess risk guarantees for stochastic convex optimization without the restrictive assumptions made in prior works. Our accelerated Algorithm 33 achieves optimal excess risk for both smooth and non-smooth loss functions, with significantly smaller computational cost than the previous state-of-the-art. Our linear-time Algorithm 31 achieves state-of-the-art excess risk under much milder, more practical assumptions than existing linear-time approaches.

Our work paves the way for several intriguing future research directions. First, the question of whether there exists a linear-time algorithm that can attain the user-level DP lower bound for smooth losses remains open. In light of our improved gradient complexity bound ($\approx (nm)^{9/8}$), we are now optimistic that the answer to this question is "yes." We believe that our novel techniques will be key to the development of an optimal linear-time algorithm. Specifically, utilizing Lemma 10.2.3 to apply outlier removal to the iterates instead of the gradients appears to be pivotal. Second, the study of user-level DP SCO has been largely limited to approximate (ε, δ)-DP. What rates are achievable under the stronger notion of pure ε-user-level DP? Third, it would be useful to develop fast and optimal algorithms that are tailored to federated learning environments [MRTZ18, GLZW24], where only a small number of users may be available to communicate with the server in each iteration. We hope our work inspires and guides further research in this exciting and practically important area.

10.6 More Preliminaries

10.6.1 Tools from Differential Privacy

Additive Noise Mechanisms Additive noise mechanisms privatize a query by adding noise to its output, with the scale of the noise calibrated to the *sensitivity* of the query.

Definition 10.6.1 (Sensitivity)**.** Given a function $q : \mathcal{Z}^N \to \mathbb{R}^k$ and a norm $\|\cdot\|_p$ on $\mathbb{R}^k$,

the ℓ_p-*sensitivity* of q is defined as

$$\sup_{\mathcal{D}\sim\mathcal{D}'} \|q(\mathcal{D}) - q(\mathcal{D}')\|_p,$$

where the supremum is taken over all pairs of datasets that differ in one user's data.

Definition 10.6.2 (Laplace Distribution)**.** We say $X \sim \text{Lap}(b)$ if the density of X is $f(X = x) = \frac{1}{2b}\exp(-\frac{|x|}{b})$.

Definition 10.6.3 (Laplace Mechanism)**.** Let $\varepsilon > 0$. Given a function $q : \mathcal{Z}^N \to \mathbb{R}^k$ on $\mathbb{R}^k$ with ℓ_1-sensitivity Δ, the *Laplace Mechanism* $\mathcal{M}$ is defined by

$$\mathcal{M}(\mathcal{D}) := q(\mathcal{D}) + (Y_1, \ldots, Y_k),$$

where $\{Y_i\}_{i=1}^k$ are i.i.d., $Y_i \sim \text{Lap}\left(\frac{\Delta}{\varepsilon}\right)$.

Lemma 10.6.4 (Privacy of Laplace Mechanism [DR14])**.** *The Laplace Mechanism is ε-DP.*

Definition 10.6.5 (Gaussian Mechanism)**.** Let $\varepsilon > 0$, $\delta \in (0, 1)$. Given a function $q : \mathcal{Z}^N \to \mathbb{R}^k$ with ℓ_2-sensitivity Δ, the *Gaussian Mechanism* $\mathcal{M}$ is defined by

$$\mathcal{M}(\mathcal{D}) := q(\mathcal{D}) + G$$

where $G \sim \mathcal{N}_k\left(0, \sigma^2\mathbf{I}_k\right)$ and $\sigma^2 = \frac{2\Delta^2 \log(2/\delta)}{\varepsilon^2}$.

Lemma 10.6.6 (Privavcy of Gaussian Mechanism [DR14])**.** *The Laplace Mechanism is (ε, δ)-DP.*

Advanced Composition If we adaptively query a data set T times, then the privacy guarantees of the T-th query is still DP and the privacy parameters degrade gracefully:

Lemma 10.6.7 (Advanced Composition Theorem [DR14])**.** *Let $\varepsilon \geq 0, \delta, \delta' \in [0, 1)$. Assume $\mathcal{A}_1, \cdots, \mathcal{A}_T$, with $\mathcal{A}_t : \mathcal{Z}^n \times \mathcal{X} \to \mathcal{X}$, are each (ε, δ)-DP $\forall t = 1, \cdots, T$. Then, the adaptive composition $\mathcal{A}(\mathcal{D}) := \mathcal{A}_T(\mathcal{D}, \mathcal{A}_{T-1}(\mathcal{D}, \mathcal{A}_{T-2}(X, \cdots)))$ is $(\varepsilon', T\delta + \delta')$-DP for*

$$\varepsilon' = \sqrt{2T\ln(1/\delta')}\varepsilon + T\varepsilon(e^\varepsilon - 1).$$

Privacy Amplification by Subsampling

Lemma 10.6.8 ([Ull17]). *Let $\mathcal{M} : \mathcal{Z}^M \to \mathcal{X}$ be (ε, δ)-DP. Let $\mathcal{M}' : \mathcal{Z}^N \to \mathcal{X}$ that first selects a random subsample $\mathcal{D}'$ of size M from the data set $\mathcal{D} \in \mathcal{Z}^N$ and then outputs $\mathcal{M}(\mathcal{D}')$. Then, $\mathcal{M}'$ is (ε', δ')-DP, where $\varepsilon' = \frac{(e^\varepsilon - 1)M}{N}$ and $\delta' = \frac{\delta M}{N}$.*

AboveThreshold: AboveThreshold algorithm [DR14] which is a key tool in differential privacy to identify whether there is a query $q_i : \mathcal{Z} \to \mathbb{R}$ in a stream of queries $q_1, \ldots, q_T$ that is above a certain threshold Δ. The AboveThreshold Algorithm 34 has the following guarantees:

Algorithm 34: AboveThreshold

Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, threshold $\Delta \in \mathbb{R}$, privacy parameter ε, sequence of T queries $q_1, \cdots, q_T : \mathcal{Z}^n \to \mathbb{R}$, each with ℓ_1-sensitivity 1;
Let $\widehat{\Delta} := \Delta + \mathrm{Lap}(\frac{2}{\varepsilon})$;
for $t = 1$ *to* T **do**
  Receive a new query $q_t : \mathcal{Z}^n \to \mathbb{R}$;
  Sample $\nu_i \sim \mathrm{Lap}(\frac{4}{\varepsilon})$;
  if $q_t(\mathcal{D}) + \nu_i \geq \widehat{\Delta}$ **then**
    Output: $a_t = \top$;
    Halt;
  end
  else
    Output: $a_t = \perp$;
  end
end

Lemma 10.6.9 ([DR14], Theorem 3.24). *Let $\gamma > 0$ and $\alpha = \frac{8 \log(2T/\gamma)}{\varepsilon}$, $k \in [T + 1]$. AboveThreshold is $(\varepsilon, 0)$-DP. Moreover, with probability at least $1 - \gamma$, for all $t \leq k$, we have:*

- *if $a_t = \top$, then $q_t(\mathcal{D}) \geq \Delta - \alpha$; and*
- *if $a_t = \perp$, then $q_t(\mathcal{D}) \leq \Delta + \alpha$.*

10.6.2 SubGaussian and Norm-SubGuassian Random Vectors

Definition 10.6.10. Let $\zeta > 0$. We say a random vector X is *SubGaussian* (SG(ζ)) with parameter ζ if $\mathbb{E}[e^{\langle v, X - \mathbb{E}X\rangle}] \le e^{\|v\|^2\zeta^2/2}$ for any $v \in \mathbb{R}^d$. Random vector $X \in \mathbb{R}^d$ is *Norm-SubGaussian* with parameter ζ (nSG(ζ)) if $\mathbb{P}[\|X - \mathbb{E}X\| \ge t] \le 2e^{-\frac{t^2}{2\zeta^2}}$ for all $t > 0$.

Lemma 10.6.11 ([JNG+19])**.** *There exists an absolute constant c, such that if X is* nSG(ζ)*, then for any fixed unit vector $v \in \mathbb{R}^d$, $\langle v, X\rangle$ is $c\zeta$ norm-SubGaussian.*

10.7 Proof of theorem 10.2.1

Theorem 10.7.1 (Formal statement of theorem 10.2.1)**.** *Suppose $n^{1-q} \ge (100/(1-1/2^q))\log(n/\delta)/\varepsilon$ for some small $q > 0$, and $m \le n^J$ for some large $J > 0$. Choose $p = J + 3/2$ and $\eta = R/(L\sqrt{dmn\varepsilon})$. in Algorithm 31. Then, Algorithm 31 is (ε, δ)-user-level DP and achieves excess risk*

$$\mathbb{E}F(x_l) - F^* \le LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{nm\varepsilon}} + \frac{\sqrt{d\log(1/\delta)}}{\sqrt{n\varepsilon}\sqrt{m}}\right),$$

using nm gradient evaluations, provided $\beta \le (L/R)\sqrt{dmn\varepsilon}$.

The gradient complexity is clear by inspection of the algorithm: The number of stochastic gradients computed during the algorithm is

$$\sum_{i=1}^{l} T_i C = \sum_{i=1}^{l} N_i m C = \sum_{i=1}^{l} n_i m \le nm.$$

Next, we will prove the privacy statement in theorem 10.7.1. The following lemma ensures that if the Laplace noise added in Algorithm 31 is sufficiently small and outlier detection succeeds, then the sensitivity of $\tilde{x}_i$ is $\widetilde{O}(\tau_i/C)$ with high probability.

Lemma 10.7.2. *[AL24, Slight modification of Lemma 3.5] Let $i \in [l]$ and $\zeta > 0$. Suppose $\mathcal{D}_i$ and $\mathcal{D}_i'$ differ in the data of one user and we are in phase i of Algorithm 31. Let E_i be the event that the Laplace noise added to the concentration score $s_i(\tau_i)$ for $\mathcal{D}_i$ has absolute value less than $2C/15$ and define E_i' similarly for data $\mathcal{D}_i'$. Denote $a_i := \mathcal{I}(\widehat{s}_i(\tau_i) \ge 4C/5)$ and $a_i' := \mathcal{I}(\widehat{s}_i'(\tau_i) \ge 4C/5)$, where $\widehat{s}_i(\tau_i)$ and $\widehat{s}_i'(\tau_i)$ are the noisy concentration scores that*

we get when running phase i of Algorithm 31 on neighboring $\mathcal{D}_i$ and $\mathcal{D}'_i$, respectively. Then, conditional on $a_i = a'_i$ and $E_i \bigcap E'_i$, there is a coupling Γ_i over $\tilde{x}_i$ and $\tilde{x}'_i$ such that for (y_i, y'_i) drawn from Γ_i, we have

$$\|y_i - y'_i\| \lesssim \frac{\tau_i \log(1/\zeta)}{C}$$

with probability at least $1 - \zeta$.

With Lemma 10.7.2 in hand, we proceed to prove that Algorithm 31 is (ε, δ)-user-level DP:

Proof of theorem 10.7.1 - Privacy. **Privacy:** Since the $\{D_i\}_{i=1}^{l}$ are disjoint, parallel composition of DP [McS09] implies that it suffices to prove that phase i is (ε, δ)-user-level-DP for any fixed i and fixed x_{i-1}. To that end, let $\mathcal{D}$ and $\mathcal{D}'$ be adjacent datasets differing in the data of one user, say $Z_{i,1} \neq Z'_{i,1}$ without loss of generality. We will show that the outputs of phase i when run on $\mathcal{D}$ and $\mathcal{D}'$, $x_i := x_i(\mathcal{D})$ and $x'_i := x_i(\mathcal{D}')$ respectively, are (ε, δ)-indistinguishable.

Let E_i be the event that the Laplace noise added in phase i (for data set $\mathcal{D}$) has absolute value less than $2C/15$ and define E'_i analogously for data set $\mathcal{D}'$. Note that E_i and E'_i are independent and $\mathbb{P}(E_i, E'_i) \geq 1 - \delta/10e^{\varepsilon}$. Denote $\zeta := \delta/10e^{\varepsilon}$. Let $a_i := \mathcal{I}(\widehat{s}_i(\tau_i) \geq 4C/5)$ and $a'_i := \mathcal{I}(\widehat{s}'_i(\tau_i) \geq 4C/5)$, where $\widehat{s}_i(\tau_i)$ and $\widehat{s}'_i(\tau_i)$ are the noisy concentration scores that we get when running phase i of Algorithm 31 on neighboring $\mathcal{D}_i$ and $\mathcal{D}'_i$, respectively. By Lemma 10.7.2 and our choice of C, we know that, conditional on $E_i \bigcap E'_i$ and on $a_i = a'_i$, there exists a coupling Γ over $(\tilde{x}_i, \tilde{x}'_i)$ such that for (y_i, y'_i) drawn from Γ, we have

$$\|y_i - y'_i\| \lesssim \frac{\tau_i \log(1/\zeta)}{C} \tag{10.1}$$

with probability at least $1 - \zeta$.

Note that the sensitivity of s_i is less than or equal to 2. Thus, by the privacy guarantees of the Laplace mechanism (Lemma 10.6.4), we have

$$\mathbb{P}(a_i = b) \leq e^{\varepsilon/4}\mathbb{P}(a'_i = b) \tag{10.2}$$

for any $b \in \{0,1\}$. Further, this implies

$$\mathbb{P}(a_i = b, E_i) \leq e^{\varepsilon/4}\left[\mathbb{P}(a_i' = b, E_i') + \zeta\right]. \tag{10.3}$$

By the bound (10.1), the privacy guarantee of the Gaussian mechanism (Lemma 10.6.6), our choice of σ_i, and independence of the Laplace and Gaussian noises that we add in Algorithm 31, we have

$$\mathbb{P}(x_i \in \mathcal{O} \mid E_i, a_i = 1) \leq e^{\varepsilon/4}\mathbb{P}(x_i' \in \mathcal{O} \mid E_i', a_i' = 1) + \frac{\delta}{n} + \zeta, \tag{10.4}$$

for any event $\mathcal{O} \subset \mathcal{X}$.

Moreover, since the algorithm halts and returns $x_i = 0$ if $a_i = 0$, we know that

$$\mathbb{P}(x_i \in \mathcal{O} \mid E_i, a_i = 0) = \mathbb{P}(x_i' \in \mathcal{O} \mid E_i', a_i' = 0) \tag{10.5}$$

for any event $\mathcal{O} \subset \mathcal{X}$.

Therefore,

$$\begin{aligned}
\mathbb{P}(x_i \in \mathcal{O}) &= \mathbb{P}(x_i \in \mathcal{O} \mid E_i)\mathbb{P}(E_i) + \mathbb{P}(x_i \in \mathcal{O} \mid E_i^c)\mathbb{P}(E_i^c) \\
&\leq \mathbb{P}(x_i \in \mathcal{O} \mid E_i, a_i = 1)\mathbb{P}(E_i, a_i = 1) + \mathbb{P}(x_i \in \mathcal{O} \mid E_i, a_i = 0)\mathbb{P}(E_i, a_i = 0) + \zeta \\
&\overset{(i)}{\leq} e^{\varepsilon/4}\mathbb{P}(x_i' \in \mathcal{O} \mid E_i', a_i' = 1)e^{\varepsilon/4}\left[\mathbb{P}(E_i', a_i' = 1) + \zeta\right] \\
&\quad + \mathbb{P}(x_i' \in \mathcal{O} \mid E_i', a_i' = 0)e^{\varepsilon/4}\left[\mathbb{P}(E_i', a_i' = 0) + \zeta\right] + \zeta \\
&\leq e^{\varepsilon/2}\mathbb{P}(x_i' \in \mathcal{O}, E_i') + \zeta\left(2e^{\varepsilon/2} + 1\right) \\
&\leq e^{\varepsilon}\mathbb{P}(x_i' \in \mathcal{O}) + \delta,
\end{aligned}$$

where (i) follows from inequalities (10.3), (10.4), and (10.5). Thus, x_i is (ε, δ)-user-level-DP. This completes the privacy proof.

□

Next, we turn to the excess risk proof. The following lemma is immediate from [FKT20, Lemma 4.5]:

Lemma 10.7.3. *Let $\eta_i \le 1/\beta$. Then, for any $y \in \mathcal{X}$ and all i, j, we have*

$$\mathbb{E}[F(\tilde{x}_{i,j}) - F(y)] \le \frac{\mathbb{E}\|y - x_{i-1}\|^2}{\eta_i T_i} + \eta_i L^2.$$

The next novel lemma is crucial in our analysis:

Lemma 10.7.4 (Re-statement of Lemma 10.2.3)**.** *Assume $f(\cdot, z)$ is convex, L-Lipschitz, and β-smooth on $\mathcal{X}$ with $\eta \le 1/\beta$. Let $\tilde{x} \leftarrow SGD(D, \eta, T, x_0)$ and $\tilde{y} \leftarrow SGD(D', \eta, T, x_0)$ be two independent runs of projected SGD, where $D, D' \sim P^N$ are i.i.d. Then, with probability at least $1 - \zeta$, we have*

$$\|\tilde{x} - \tilde{y}\| \lesssim \eta L \sqrt{T \log(dT/\zeta)}.$$

Proof. Let $g_t := \nabla f(x_t, z_t)$ for z_t drawn uniformly from D without replacement and $g_t' := \nabla f(y_t, z_t')$ for z_t' drawn uniformly from D' without replacement. Let $F(x) := \mathbb{E}_{z \sim P}[f(x, z)]$.

We will prove that $\|x_t - y_t\| \lesssim \eta L \sqrt{T \log(dT/\zeta)}$ with probability at least $1 - \zeta/t$ for all $t \in [T]$. Note that this implies the lemma. We proceed by induction. The base case, when $t = 0$, is trivially true since $x_0 = y_0$. For the inductive hypothesis, suppose there is an absolute constant $c > 0$ such that with probability at least $1 - t\zeta/T$, we have

$$\|x_i - y_i\| \le c\eta L \sqrt{i \cdot \log(dT/\zeta)} + 2\eta L,$$

$\forall i \le t$. Then, for the inductive step, we have by non-expansiveness of projection onto convex sets, that

$$\begin{aligned}
\|x_{t+1} - y_{t+1}\|^2 &\le \|x_t - \eta g_t - (y_t - \eta g_t')\|^2 \\
&= \|x_t - \eta \nabla F(x_t) - (y_t - \eta \nabla F(y_t)) - \eta(g_t - \nabla F(x_t) - g_t' + \nabla F(y_t))\|^2 \\
&= \|x_t - \eta \nabla F(x_t) - (y_t - \eta \nabla F(y_t))\|^2 \\
&\quad - 2\eta \langle x_t - \eta \nabla F(x_t) - (y_t - \eta \nabla F(y_t)), g_t - \nabla F(x_t) - g_t' + \nabla F(y_t) \rangle \\
&\quad + \eta^2 \|g_t - \nabla F(x_t) - g_t' + \nabla F(y_t)\|^2 \\
&\overset{(i)}{\le} \|x_t - y_t\|^2 - 2\eta \langle x_t - \eta \nabla F(x_t) - (y_t - \eta \nabla F(y_t)), g_t - \nabla F(x_t) - g_t' + \nabla F(y_t) \rangle \\
&\quad + 4\eta^2 L^2, \qquad (10.6)
\end{aligned}$$

where (i) follows from the non-expansive property of gradient descent on smooth convex function for $\eta \leq 1/\beta$ [HRS16].

Define $a_t := -2\eta\langle x_t - \eta\nabla F(x_t) - (y_t - \eta\nabla F(y_t)), g_t - \nabla F(x_t) - g_t' + \nabla F(y_t)\rangle$. By Inequality (10.6) and the inductive hypothesis, we obtain

$$\|x_{t+1} - y_{t+1}\|^2 \leq 4\eta^2 L^2 t + \sum_{i=1}^{t} a_t.$$

It remains to bound $\sum_{i=1}^{t} a_i$. Note that $\mathbb{E}[a_i \mid a_1, \cdots, a_{i-1}] = 0$, and by Lemma 10.6.11 we know there is a constant $c > 0$ such that a_i is $\mathrm{nSG}(c\eta L\|x_i - y_i\|)$ for all i. Hence by Theorem 8.11.2, we know

$$\mathbb{P}\left[\left|\sum_{i=1}^{t} a_i\right| \geq c\eta L\sqrt{\log(dT/\gamma)\sum_{i\leq t}\|x_i - y_i\|^2}\right] \leq 1 - \zeta/T.$$

Conditional on the event that $\|x_i - y_i\| \leq c\sqrt{\log(dT/\zeta)}\eta L\sqrt{i}$ for all $i \leq t$ (which happens with probability $1 - t\zeta/T$ by the inductive hypothesis), we know

$$\mathbb{P}\left[\left|\sum_{i=1}^{t} a_i\right| \geq c^2(t+1)L^2\eta^2\log(dT/\zeta)\middle|\|x_i - y_i\| \leq c\log(dT/\zeta)\eta L\sqrt{i}, \forall i \leq t\right] \leq 1 - \zeta/T.$$

Hence we know

$$\mathbb{P}\left[\|x_{t+1} - y_{t+1}\|^2 \geq c^2\log(dT/\zeta)\eta^2 L^2(t+1)\middle|\|x_i - y_i\| \leq c\log(dT/\zeta)\eta L\sqrt{i}, \forall i \leq t\right] \leq 1 - \zeta/T.$$

Combining the above pieces completes the inductive step, showing that $\|x_{t+1} - y_{t+1}\| \leq c\sqrt{(t+1)\log(dT/\zeta)}\eta L + 2\eta L$ with probability at least $1 - (t+1)\zeta/T$. This completes the proof.

□

By combining Lemmas 10.2.3 and 10.7.3 with the localization proof technique of [FKT20], we can now prove the excess risk guarantee of theorem 10.7.1:

Proof of theorem 10.2.1 - Excess risk. **Excess Risk:** First, we will argue that $\tilde{x}_i = \frac{1}{C}\sum_{j=1}^{C}\tilde{x}_{i,j}$

for all i with high probability $\geq 1 - 3/nm$. Lemma 10.2.3 implies that

$$\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \leq \tau_i$$

for all $i \in [l]$, $j, j' \in [C]$ with probability at least $1 - 1/nm$. Thus, $s_i(\tau_i) = C$ with probability at least $1 - 1/nm$. Now, conditional on $s_i(\tau_i) = C$, we have $\widehat{s}_i(\tau_i) \geq 4C/5$ for all i with probability at least $1 - 1/nm$ by Laplace concentration and a union bound. Moroever, if $\|\tilde{x}_{i,j} - \tilde{x}_{i,j'}\| \leq \tau_i$ for all j, j', then $p_{i,j} = 1$ for all j and hence there are no outliers: $\mathcal{S}_i = \{\tilde{x}_{i,j}\}_{j\in[C]}$. By a union bound, we conclude that $\mathcal{S}_i = \{\tilde{x}_{i,j}\}_{j\in[C]}$ and hence $\tilde{x}_i = \frac{1}{\mathcal{S}_i}\sum_{D_{i,j}\in\mathcal{S}_i} \tilde{x}_{i,j}$ for all i with probability at least $\geq 1 - 3/nm$. By the law of total expectation and Lipschitz continuity, it suffices to condition on this high probability good event that $\tilde{x}_i = \frac{1}{C}\sum_{j=1}^{C} \tilde{x}_{i,j}$ for all i: the total expected excess risk can only be larger than the conditional excess risk by an additive factor of at most $3LR/nm$.

Now, conditional on $\tilde{x}_i = \frac{1}{\mathcal{S}_i}\sum_{D_{i,j}\in\mathcal{S}_i} \tilde{x}_{i,j}$, Lemma 10.7.3 and Jensen's inequality implies

$$\mathbb{E}[F(\tilde{x}_i) - F(\tilde{x}_{i-1})] \lesssim \frac{\mathbb{E}\|\tilde{x}_{i-1} - x_{i-1}\|^2}{\eta_i T_i} + \eta_i L^2 = \frac{d\sigma_{i-1}^2}{\eta_i T_i} + \eta_i L^2. \tag{10.7}$$

Next, let $x_0^* := x^* = \operatorname{argmin}_{x\in\mathcal{X}} F(x)$, and write

$$\begin{aligned}
\mathbb{E}[F(x_l) - F^*] &= \sum_{i=1}^{l} \mathbb{E}[F(x_i^*) - F(x_{i-1}^*)] + \mathbb{E}[F(x_l) - F(x_l^*)] \\
&\lesssim \frac{R^2}{\eta T_1} + \eta L^2 + \sum_{i=2}^{l} \left[\eta_{i-1} L^2 d + \eta_i L^2\right] + L^2\sqrt{d}\eta_l\sqrt{T_l} \\
&\lesssim \frac{R^2}{\eta T_1} + d\eta L^2 + L^2\sqrt{d}\eta_l\sqrt{T_l}.
\end{aligned}$$

Plugging in the prescribed algorithmic parameters completes the excess risk proof. □

10.8 Proofs of Results in Section 10.3

Theorem 10.8.1 (Re-statement of theorem 10.3.1)**.** *Let $\varepsilon \leq 10$, $q > 0$ such that $n^{1-q} > \frac{100\log(20nmde^{\varepsilon}/\delta)}{\varepsilon(1-(1/2)^q)}$. Then, Algorithm 33 is (ε, δ)-DP.*

We require the following lemma, which is a direct consequence of [AL24, Lemma 3.5]:

Lemma 10.8.2. *Consider Algorithm 32. Let $\mathcal{D}_i'$ and $\mathcal{D}_i'$ be two data sets that differ in the data of one user. Let $E_i = \{|v_i^t| \le K_i/20 \ \forall t \in [T_i] \cap |\xi_i| \le K_i/20\}$. Define E_i' similarly for independent draws of random Laplace noise: $E_i' = \left\{|(v_i^t)'| \le K_i/20 \ \forall t \in [T_i] \cap |\xi_i'| \le K_i/20\right\}$. Let $a_i^t = \mathcal{I}(\widehat{s}_i^t(\mathcal{D}_i) \ge 4K_i/5)$ and $b_i^t = \mathcal{I}(\widehat{s}_i^t(\mathcal{D}_i') \ge 4K_i/5)$ denote the concentration scores in iteration t. Then, conditional on $E_i \bigcap E_i'$ and conditional on $a_i^t = b_i^t$, there exists a coupling Γ_i^t over $g_i^t(\mathcal{D}_i)$ and $g_i^t(\mathcal{D}_i')$ such that for (h, h') drawn from Γ_i, we have*

$$\|h - h'\| \lesssim \frac{\tau \log(1/\zeta)}{K_i}$$

with probability at least $1 - \zeta$.

Proof of theorem 10.8.1. Note that our assumption on n^{1-q} being sufficiently large implies that $n_i \gtrsim \frac{\log(nmd/\delta)}{\varepsilon}$ for all $i \in [l]$. By parallel composition [McS09] and post-processing, it suffices to show that $\{\widehat{g}_i^t\}_{t=1}^{T_i}$ satisfies (ε, δ)-user-level DP for any $i \in [l]$. To that end, fix any $i \in [l]$ and let $\mathcal{D}$ and $\mathcal{D}'$ be adjacent datasets that differ in the data of one user such that $\mathcal{D}_i \ne \mathcal{D}_i'$. We will show that $\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i}$ and $\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i}$ are (ε, δ)-indistinguishable, which will imply that Algorithm 33 is (ε, δ)-user-level DP.

Let $E_i = \{|v_i^t| \le K_i/20 \ \forall t \in [T_i] \cap |\xi_i| \le K_i/20\}$. Define E_i' similarly for independent draws of random Laplace noise: $E_i' = \left\{|(v_i^t)'| \le K_i/20 \ \forall t \in [T_i] \cap |\xi_i'| \le K_i/20\right\}$. Our choice of $K_i \ge \frac{500 \log(nmde^\varepsilon/\delta)}{\varepsilon}$ ensures that

$$\mathbb{P}\left(E_i \bigcap E_i'\right) \ge 1 - \delta/(10e^\varepsilon),$$

by Laplace concentration and a union bound. Let $\zeta := \delta/(10T_i e^\varepsilon)$.

Let $a_i^t = \mathcal{I}(\widehat{s}_i^t(\mathcal{D}) \ge 4K_i/5)$ and $b_i^t = \mathcal{I}(\widehat{s}_i^t(\mathcal{D}') \ge 4K_i/5)$. Note that if $a_i^t = b_i^t = 0$, then $\widehat{g}_i^t(\mathcal{D}) = 0 = \widehat{g}_i^t(\mathcal{D}')$.

Conditional on the good event that $a_i^t = b_i^t$ for all t and conditional on $E_i \bigcap E_i'$, we can bound the ℓ_2-sensitivity of g_i^t with high probability, via Lemma 10.8.2 and a union bound:

$$\|g_i^t(\mathcal{D}) - g_i^t(\mathcal{D}')\| \lesssim \frac{\tau \log(1/\zeta)}{K_i} \lesssim \frac{\tau \log(nme^\varepsilon/\delta)}{K_i} \tag{10.8}$$

for all $t \in [T_i]$ with probability at least $1 - T_i\zeta = 1 - \delta/(10e^{\varepsilon})$.

Note that $\{\widehat{s}_i^t(\mathcal{D})\}_{t=1}^{T_i}$ and $\{\widehat{s}_i^t(\mathcal{D}')\}$ are $\varepsilon/2$-indistinguishable by the DP guarantees of AboveThreshold in Lemma 10.6.9, since the sensitivity of s_i^t is upper bounded by 2. Therefore,

$$\mathbb{P}(\{a_i^t\}_{t=1}^{T_i} = v, E_i) \le e^{\varepsilon/2}\left[\mathbb{P}(\{b_i^t\}_{t=1}^{T_i} = v, E_i') + \zeta\right] \tag{10.9}$$

for any $v \in \{0,1\}^{T_i}$.

Now, by the sensitivity bound (10.8), the privacy guarantee of the Gaussian mechanism (Lemma 10.6.6) and our choice of σ_i, the advanced composition theorem (Lemma 10.6.7), and privacy amplification by subsampling (Lemma 10.6.8), we have

$$\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i\{a_i^t\}_{t=1}^{T_i} = v) \le e^{\varepsilon/2}\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i', \{b_i^t\}_{t=1}^{T_i} = v) + (T_i+1)\zeta, \tag{10.10}$$

for any event $\mathcal{O} \subset \mathcal{X}$. Here we also used the fact that $K_i \ge \frac{n_i\varepsilon}{\sqrt{T_i}}$.

For short-hand, write $\{a_i^t\}_{t=1}^{T_i} = 1$ if $a_i^t = 1$ for all $t \in [T_i]$ and $\{a_i^t\}_{t=1}^{T_i} = 0$ if $a_i^t = 0$ for some $t \in [T_i]$; similarly for b_i^t. Then since the algorithm halts and returns $\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} = 0$ if $\{a_i^t\}_{t=1}^{T_i} = 0$, we know that

$$\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i, \{a_i^t\}_{t=1}^{T_i} = 0) = \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i', \{b_i^t\}_{t=1}^{T_i} = 0), \tag{10.11}$$

for any event $\mathcal{O} \subset \mathcal{X}$.

Combining the above pieces, we have

$$\begin{aligned}
\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O}) &= \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i)\mathbb{P}(E_i) + \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i^c)\mathbb{P}(E_i^c) \\
&\le \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i, \{a_i^t\}_{t=1}^{T_i} = 1)\mathbb{P}(E_i, \{a_i^t\}_{t=1}^{T_i} = 1) \\
&\quad + \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i, \{a_i^t\}_{t=1}^{T_i} = 0)\mathbb{P}(E_i, \{a_i^t\}_{t=1}^{T_i} = 0) + 2T\zeta \\
&\overset{(i)}{\le} e^{\varepsilon/2}\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i', \{b_i^t\}_{t=1}^{T_i} = 1)e^{\varepsilon/4}\left[\mathbb{P}(E_i', \{b_i^t\}_{t=1}^{T_i} = 1) + T\zeta\right] \\
&\quad + \mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O} \mid E_i', \{b_i^t\}_{t=1}^{T_i} = 0)e^{\varepsilon/2}\left[\mathbb{P}(E_i', \{b_i^t\}_{t=1}^{T_i} = 0) + T\zeta\right] + T\zeta \\
&\le e^{\varepsilon/2}\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O}, E_i') + 4T\zeta\left(2e^{\varepsilon/2} + 1\right)
\end{aligned}$$

$$\leq e^{\varepsilon}\mathbb{P}(\{\widehat{g}_i^t(\mathcal{D}')\}_{t=1}^{T_i} \in \mathcal{O}) + \delta,$$

where (i) follows from inequalities (10.9), (10.10), and (10.11). Thus, $\{\widehat{g}_i^t(\mathcal{D})\}_{t=1}^{T_i}$ is (ε, δ)-user-level-DP, which implies the result.

□

Theorem 10.8.3 (Formal statement of theorem 10.3.2)**.** *Let $\varepsilon \leq 10$ and $\delta < 1/(mn)$. Then, choosing $\lambda = \frac{L}{R}\left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{\varepsilon n\sqrt{m}}\right)$ and $p \geq 3q + 2.5 + \log_n(\sqrt{m})$ in Algorithm 33 yields optimal excess risk:*

$$\mathbb{E}F(x_l) - F^* \leq LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{mn}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon n\sqrt{m}}\right).$$

The gradient complexity of this algorithm is upper bounded by

$$mn\left(1 + \varepsilon\left(\frac{\beta R}{L}\right)^{1/4}\left((mn)^{1/8} \wedge \left(\frac{\varepsilon^2 n^2 m}{d}\right)^{1/8}\right)\right) + \sqrt{\frac{\beta R}{L}}\left(\frac{n^{1/4}m^{5/4}}{\varepsilon} + \left(\frac{n^{1/2}m^{5/4}}{d^{1/4}\varepsilon^{1/2}}\right)\right).$$

We will need the following bound on the excess empirical risk of accelerated (noisy) SGD for the proof of theorem 10.8.3:

Lemma 10.8.4. *[GL13a, Proposition 7] Let x^T be computed by T steps of (multi-stage) Accelerated Minibatch SGD on λ-strongly convex, β-smooth $\widehat{F}$ with unbiased stochastic gradient estimator g^t such that $\mathbb{E}\|g^t - \nabla\widehat{f}(x^t)\|^2 \leq V^2$ for all $t \in [T]$. Then,*

$$\mathbb{E}[\widehat{f}(x^T) - \min_{x\in\mathcal{X}}\widehat{f}(x)] \lesssim [\widehat{f}(x^0) - \min_{x\in\mathcal{X}}\widehat{f}(x)]\exp\left(-T\sqrt{\frac{\lambda}{\beta}}\right) + \frac{V^2}{\lambda T}.$$

Remark 10.8.5. As noted in Lemma 10.8.4, we technically need to call a *multi-stage implementation* of Algorithm 32 in line 7 of Algorithm 33 (as in [GL13a]) to get the desired excess risk bound for minimizing the regularized ERM problem in each iteration. For improved readability, we omitted these details in the main body.

Next, we obtain a bound on the variance of the noisy stochastic minibatch gradient estimator $\widehat{g}_i^t$ in Algorithm 32, which can then be plugged in for V^2 in Lemma 10.8.4.

Lemma 10.8.6 (Re-statement of Lemma 10.3.5)**.** *Let $\delta \le 1/(nm), \varepsilon \lesssim 1$. Denote $\widetilde{F}_i(x) := \frac{1}{n_i}\sum_{Z_{i,j}\in D_i}\widehat{F}(x, Z_{i,j})$. Then, conditional on $\mathcal{S}_i^t = D_i^t$ for all $i \in [l], t \in [T_i]$, we have*

$$\mathbb{E}\|g_i^t - \nabla\widetilde{F}_i(x_{i-1}^t)\|^2 \lesssim \frac{L^2\log(ndm)}{Km}$$

for all $i \in [l], t \in [T_i]$, where the expectation is over both the random i.i.d. draw of $\mathcal{D} = (Z_1, \ldots, Z_n) \sim P^{nm}$ and the randomness in Algorithm 33.

Proof. By [LJCJ17, Lemma A.1], we know that, conditional on the draw of the data D_i and for fixed x_{i-1}^t, the variance of the minibatch estimator of the gradient of the empirical loss is

$$\begin{aligned}\mathbb{E}\left[\left\|g_i^t - \nabla\widetilde{F}_i(x_{i-1}^t)\right\|^2 \middle| D_i, x_{i-1}^t\right] &= \mathbb{E}_{\{i_l\}_{l=1}^K\sim\mathrm{Unif}([n])}\left[\left\|\frac{1}{Km}\sum_{l=1}^K\sum_{j=1}^m \nabla f(x_{i-1}^t, z_{i_l,j}^t)) - \nabla\widetilde{F}_i(x_{i-1}^t)\right\|^2 \middle| D_i, x_{i-1}^t\right] \\ &\le \frac{\mathcal{I}(K=n)}{K}\frac{1}{n}\sum_{i=1}^n\left\|\frac{1}{m}\sum_{j=1}^m[\nabla f(x_{i-1}^t, z_{i,j}^t) - \nabla\widetilde{F}_i(x_{i-1}^t)]\right\|^2.\end{aligned} \tag{10.12}$$

Recall $\widetilde{F}_i(x) := \frac{1}{n_i m}\sum_{z\in D_i} f(x,z)$ is the empirical loss of D_i.

Now, for any fixed x and any $Z \in D_i$, Hoeffding's inequality implies that

$$\|\nabla\widehat{f}(x,Z) - \nabla F(x)\| \le \tau = O\left(\frac{L\sqrt{\log(nd/\gamma)}}{\sqrt{m}}\right)$$

with probability at least $1-\gamma$, where $\widehat{f}(x,Z) := \frac{1}{m}\sum_{z\in Z} f(x,z)$ is user Z's empirical loss. Thus, by the stability of user-level DP (see [BS23, Theorem 3.4]), for any $i \in [l], t \in [t]$, we have that

$$\|\nabla\widehat{f}(x_{i-1}^t, Z) - \nabla F(x_{i-1}^t)\| \le \tau \tag{10.13}$$

for all $Z \in \mathcal{D}$ with probability at least $1-\gamma' = n(e^{2\varepsilon}\gamma + \delta)$, since x_{i-1}^t is (ε,δ)-user-level DP. To make $\gamma' \lesssim 1/m$, we choose $\gamma = 1/mn$ and use the assumptions that $\varepsilon \lesssim 1$ and

$\delta \leq 1/mn$. Thus, for any fixed i, t we have

$$\|\nabla \widehat{f}(x_{i-1}^t, Z) - \nabla F(x_{i-1}^t)\| \lesssim \frac{L\sqrt{\log(n^2md)}}{\sqrt{m}}$$

for all $Z \in \mathcal{D}$ with probability at least $1 - 1/m$, which implies

$$\mathbb{E}\|\nabla \widehat{f}(x_{i-1}^t, Z) - \nabla F(x_{i-1}^t)\|^2 \lesssim \frac{L^2 \log(nmd)}{m}.$$

This also implies

$$\mathbb{E}\|\nabla \widehat{f}_{\mathcal{D}}(x_{i-1}^t) - \nabla F(x_{i-1}^t)\|^2 \lesssim \frac{L^2 \log(nmd)}{m},$$

by Jensen's inequality, where $\widehat{f}_{\mathcal{D}}(x)$ is the empirical loss over the entire data set $\mathcal{D}$.

Plugging the above bounds into (10.12) then yields

$$\begin{aligned}\mathbb{E}\|g_i^t - \nabla \widetilde{F}_i(x_{i-1}^t)\|^2 \lesssim \frac{L^2 \log(ndm)}{Km} &= \mathbb{E}_{\mathcal{D}\sim P^{nm}, \{i_l\}_{l=1}^K \sim \mathrm{Unif}([n])} \left[\left\| \frac{1}{Km} \sum_{l=1}^{K} \sum_{j=1}^{m} \nabla f(x_{i-1}^t, z_{i_l,j}^t) - \nabla \widehat{F}_{\mathcal{D}}(x_{i-1}^t) \right\|^2 \right. \\ &\lesssim \frac{\mathcal{I}(K=n)}{K} \cdot \frac{L^2 \log^2(nmd)}{m},\end{aligned}$$

completing the proof.

□

We are now ready to prove theorem 10.8.3:

Proof of theorem 10.8.3. **Excess risk:** Note that the assumption in the theorem ensures that $K_i \leq n_i$ for all i. By similar arguments to those used in [AL24, Proposition 3.7], we will show that with high probability $\geq 1 - 2/(nm)$, for all $i \in [l], t \in [T_i]$, $\mathcal{S}_i^t = D_i^t$ and hence g_i^t is an unbiased estimator of $\nabla \widehat{f}_{D_i^t}(x_i^t)$. To show this, first note that for any $\gamma > 0$ and any fixed x,

$$\|\nabla \widehat{f}(x, Z_j) - \nabla F(x)\| \leq \frac{L \log(nd/\gamma)}{\sqrt{m}}$$

with probability at least $1 - \gamma/K_i$ by Hoeffding's inequality (see [AL24, Lemma 4.3]). Next,

we invoke the stability of differential privacy to show that for all $t \in [T_i]$, $(q_t(Z_{i,1}^t), \ldots, q_t(Z_{i,K_i}^t))$ is τ-*concentrated* (i.e. there exists $q^* \in \mathbb{R}^d$ such that $\|q_t(Z_{i,j}^t - q^*\| \le \tau)$ with probability at least $1 - T_i(e^{2\varepsilon}\gamma + \delta)$ (see [BS23, Theorem 4.3]). By a union bound and the choice of $\gamma = 1/[(nm)^{5/4}\log(ndm)e^{2\varepsilon}]$, we have that $(q_t(Z_{i,1}^t), \ldots, q_t(Z_{i,K_i}^t))$ is τ-concentrated for all $i \in [l], t \in [T_i]$ with probability at least $1 - 1/nm$. Now, τ-concentration of $(q_t(Z_{i,1}^t), \ldots, q_t(Z_{i,K_i}^t))$ implies $s_i^t(\tau) = K_i$. Further, $s_i^t(\tau) = K_i$ implies $\widehat{s}_i^t(\tau) \ge 4K_i/5$ with probability at least $1-\zeta$ if $K_i \ge 500\log(nm/\zeta)$, by Laplace concentration and a union bound. Next, note that τ-concentration of $(q_t(Z_{i,1}^t), \ldots, q_t(Z_{i,K_i}^t))$ implies $p_i^t(Z_{i,j}^t) = 1$ for all $j \in [K_i]$ and $\mathcal{S}_i^t = D_i^t$. Thus, $\mathcal{S}_i^t = D_i^t$ for all i, t with probability at least $1 - 1/nm$. Setting $\zeta = 1/nm$ and using a union bound shows that with probability at least $1 - 2/nm$, we have $\mathcal{S}_i^t = D_i^t$ and $g_i^t = \nabla \widehat{f}_{D_i^t}(x_i^t)$ for all i, t.

Now, Lemma 10.8.4 implies that, *if the outlier-removal procedure in Algorithm 32 leads to an unbiased gradient estimator* g_i^t (line 12) for all $t \in [T_i = \widetilde{\Theta}(1 + \sqrt{\beta/\lambda_i})]$, then

$$\mathbb{E}[\widehat{f}_i(x_i^{T_i}) - \min_{x \in \mathcal{X}} \widehat{f}_i(x)] \lesssim \frac{V_i^2}{\lambda_i T_i}, \tag{10.14}$$

where $V_i^2 = \max_{t \in [t_i]} \mathbb{E}\|\widehat{g}_i^t - \nabla \widehat{f}_i(x_i^t)\|^2 \lesssim d\sigma_i^2 + \frac{\log(ndm)L^2}{K_i m}$ (unconditionally, after taking expectation over the random draw of $\mathcal{D} \sim P^{nm}$, by Lemma 10.3.5). We have shown that the event GOOD $:= \{g_i^t = \nabla \widehat{f}_{D_i^t}(x_i^t)$ for all $i \in [l], t \in [T_i]\}$ occurs with probability at least $1 - 2/nm$. We will condition on GOOD for the rest of the proof: note that the Lipschitz assumption implies that the total (unconditional) excess risk will only by larger than the conditional (on GOOD) excess risk by an additive factor of at most $2LR/\sqrt{nm}$.

By stability of regularized ERM (see [SSSSS09]), we have

$$\mathbb{E}[F(x_i^*) - F(y)] \lesssim \frac{L^2}{\lambda_i n_i m} + \lambda_i \mathbb{E}[\|x_{i-1} - y\|^2] \tag{10.15}$$

for all i, where $x_i^* := \operatorname{argmin}_x \widehat{f}_i(x)$. By strong convexity and (10.14), we have

$$(\lambda_i/2)\mathbb{E}\|x_i - x_i^*\|^2 \le \mathbb{E}\widehat{f}_i(x_i) - \widehat{f}_i^* \lesssim \frac{d\sigma_i^2}{\lambda_i T_i} + \frac{L^2\log(ndm)}{\lambda_i T_i K_i m}. \tag{10.16}$$

Thus,

$$\mathbb{E}\|x_i - x_i^*\|^2 \lesssim \frac{d\sigma_i^2}{\lambda_i T_i} + \frac{L^2 \log(ndm)}{\lambda_i T_i K_i m} \lesssim \frac{d\tau^2 \log(1/\delta)}{\lambda_i^2 \varepsilon^2 n_i^2} + \frac{L^2 \log(ndm)}{\lambda_i^2 T_i K_i m} \tag{10.17}$$

Now, letting $x_0^* := x^* = \operatorname{argmin}_{x\in\mathcal{X}} F(x)$ and hiding logarithmic factors, we have:

$$\begin{aligned}
\mathbb{E}[F(x_l) - F^*] &= \sum_{i=1}^{l} \mathbb{E}[F(x_i^*) - F(x_{i-1}^*)] + \mathbb{E}[F(x_l) - F(x_l^*)] \\
&\lesssim \frac{L^2}{\lambda_1 n_1 m} + \lambda_1 R^2 + \sum_{i=2}^{l} \mathbb{E}\left[\frac{L^2}{\lambda_i n_i m} + \lambda_i \|x_{i-1} - x_{i-1}^*\|^2\right] + L\mathbb{E}\|x_l - x_l^*\| \\
&\lesssim \frac{L^2}{\lambda n m} + \lambda R^2 + \sum_{i=2}^{l}\left[\frac{L^2}{\lambda_i n_i m} + \lambda_i\left(\frac{d\tau^2\log(1/\delta)}{\lambda_{i-1}^2\varepsilon^2 n_{i-1}^2} + \frac{L^2}{\lambda_{i-1}^2 T_{i-1} K_{i-1} m}\right)\right] \\
&\quad + L\frac{\sqrt{d}\tau\sqrt{T_l \log(1/\delta)}}{\lambda_l \varepsilon n_l},
\end{aligned}$$

where the first inequality used (10.15) and Lipschitz continuity, the second inequality used (10.17).

Note that $K_i T_i \geq n_i$. Further, our choice of sufficiently large p makes λ_l large enough that $L\frac{\sqrt{d}\tau\sqrt{T_l\log(1/\delta)}}{\lambda_l \varepsilon n_l} \leq \frac{LR\sqrt{d}}{\varepsilon n\sqrt{m}}$. Therefore, upper bounding the sum by it's corresponding geometric series gives us

$$\mathbb{E}[F(x_l) - F^*] \lesssim \frac{LR\sqrt{d}}{\varepsilon n\sqrt{m}} + \frac{L^2}{\lambda}\left(\frac{1}{nm} + \frac{d\tau^2\log(1/\delta)}{\varepsilon^2 n^2}\right) + \lambda R^2. \tag{10.18}$$

Plugging in λ completes the excess risk proof.

Gradient Complexity: The gradient complexity is $\sum_{i=1}^{l} T_i K_i m$. Plugging in the prescribed choices of T_i and K_i completes the proof. □

10.9 Details on the non-smooth algorithm and the proof of Theorem 10.4.1

For any loss function $f(\cdot, z)$, we define the convolution function $f_r(\cdot, z) := f(\cdot, z) * n_r$ where n_r is the uniform density in the ℓ_2 ball of radius r centered at the origin in $\mathbb{R}^d$. Specifically, $n_r(y) = \frac{\Gamma(\frac{d}{2}+1)}{\pi^{\frac{d}{2}} r^d}$ for $\|y\| \leq r$, and $n_r(y) = 0$ otherwise. For simplicity, we omit the dependence on z in the following Lemma:

Lemma 10.9.1 (Randomized Smoothing, [YNS12, DBW12]). *For any $r > 0$, let $\mathcal{X}_r := \mathcal{X} + \{x \in \mathbb{R}^d : \|x\| \le r\}$. If f is convex and L-Lipschitz over $\mathcal{X}_r$, then the convolution function f_r has the following properties:*

- *$f_r(x) \le f(x) \le f_r(x) + Lr$, for all $x \in \mathcal{X}$.*
- *f_r is L-Lipschitz and convex.*
- *f_r is $\frac{L\sqrt{d}}{r}$-smooth.*
- *For random variables $y \sim n_r$, we have $\mathbb{E}_y[\nabla f(x+y)] = \nabla f_r(x)$.*

The following lemma can be easily seen from the proofs of theorems 10.3.1 and 10.3.2:

Lemma 10.9.2 (Privacy and utility of Algorithm 33 for general K_i, T_i). *Let $\varepsilon \le 10$, $q > 0$ such that $n^{1-q} > \frac{100\log(20nmde^{\varepsilon}/\delta)}{\varepsilon(1-(1/2)^q)}$.*

- *If $K_i \gtrsim \frac{n_i\varepsilon}{\sqrt{T_i}} + \frac{\log(nmde^{\varepsilon}/\delta)}{\varepsilon}$, then Algorithm 33 is (ε,δ)-user-level DP.*
- *If $T_iK_i \ge n_i$ and $T_i \gtrsim (1+\sqrt{\beta/\lambda_i})\log(ndm)$ for all i, then Algorithm 33 achieves optimal excess risk.*

Theorem 10.9.3 (Formal statement of theorem 10.4.1). *Let $\varepsilon \le 10$, $\delta < 1/(mn)$, and $q > 0$ such that $n^{1-q} > \frac{100\log(20nmde^{\varepsilon}/\delta)}{\varepsilon(1-(1/2)^q)}$. Suppose that for any z, $f(,z)$ is convex and L-Lipschitz over $\mathcal{X}_r$ for $\mathcal{X}_r := \mathcal{X} + \{x \in \mathbb{R}^d : \|x\| \le r\}$ where $r = \frac{\sqrt{d}}{\varepsilon n\sqrt{m}}R$. Then, running Algorithm 33 with functions $\{f_r(x;z)\}_{z\in\mathcal{D}}$ yields optimal excess risk:*

$$\mathbb{E}F(x_l) - F^* \le LR \cdot \widetilde{O}\left(\frac{1}{\sqrt{mn}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon n\sqrt{m}}\right).$$

The gradient complexity of this algorithm is upper bounded by

$$mn\left(1 + n^{3/8}m^{1/4}\varepsilon^{1/4}\right).$$

Proof. By Lemma 10.9.1 and our choice of r, we have $|f_r(x,z)-f(x,z)| \le Lr = O(LR\frac{\sqrt{d}}{\varepsilon n\sqrt{m}})$. Set $\lambda = \frac{1}{\sqrt{mn}}$. Then we know that

$$\mathbb{E}\, F(x_l) - F^* \le \mathbb{E}\,[F_r(x_l) - F_r^*] + O(LR\frac{\sqrt{d}}{\varepsilon n\sqrt{m}}).$$

Further, F_r is β-smooth for $\beta \le \frac{L}{R}\varepsilon n\sqrt{m}$. Set $T_i = (1+\sqrt{\beta/\lambda_i})\log(ndm) = 1 + n_i^{3/4}m^{1/2}\varepsilon^{1/2}\log(ndm)$ and $K_i = \frac{n_i\varepsilon}{\sqrt{T_i}} + \frac{\log(nmde^{\varepsilon}/\delta)}{\varepsilon}$. Then Lemma 10.9.2 implies that Algorithm 33 is (ε,δ) user-level DP, and yields the excess risk bound

$$\mathbb{E}\, F_r(x_l) - F_r^* \le LR\cdot\tilde{O}\left(\frac{1}{\sqrt{mn}} + \frac{\sqrt{d\log(1/\delta)}}{\varepsilon n\sqrt{m}}\right),$$

as desired. The number of gradient evaluations is

$$\sum_{i=1}^{l} T_i K_i m \lesssim mn\left(1 + n^{3/8}m^{1/4}\varepsilon^{1/4}\right).$$

This completes the proof. □

10.10 Limitations

Our work weakens the assumptions on the smoothness parameter and the number of users that are needed for user-level DP SCO. Nevertheless, our results still require certain assumptions that may not always hold in practice. For example, we assume convexity of the loss function. In deep learning scenarios, this assumption does not hold and our algorithms should not be used. Thus, user-level DP *non-convex* optimization is an important direction for future research []. Furthermore, the assumption that the loss function is convex and uniformly Lipschitz continuous may not hold in certain applications, motivating the future study of user-level DP stochastic optimization with heavy tails [LR23, ALT24].

Our algorithms are also faster than the previous state-of-the-art, including a linear-time Algorithm 31 with state-of-the-art excess risk. However, our error-optimal accelerated Algorithm 33 runs in super-linear time. Thus, in certain applications where a linear-time algorithm is needed due to strict computational constraints, Algorithm 31 should be used

instead.

Part V

OTHER SETTINGS

Chapter 11

WHEN DOES DIFFERENTIALLY PRIVATE LEARNING NOT SUFFER IN HIGH DIMENSIONS?

11.1 Introduction

Recent works have shown that large publicly pretrained models can be differentially privately fine-tuned on small downstream datasets with performance approaching those attained by non-private models. In particular, past works showed that pretrained BERT [DCLT18] and GPT-2 [RNSS18, RWC+19] models can be fine-tuned to perform well for text classification and generation under a privacy budget of $\varepsilon \in [2, 6]$ [LTLH21, YNB+21]. More recently, it was shown that pretrained ResNets [HZRS16] and vision-Transformers [DBK+20] can be fine-tuned to perform well for ImageNet classification under single digit privacy budgets [DBH+22, MTKC22].

One key ingredient in these successes has been the use of large pretrained models with millions to billions of parameters. These works generally highlighted the importance of two phenomena: (i) large pretrained models tend to experience good privacy-utility trade-offs when fine-tuned, and (ii) the trade-off improves with the improvement of the quality of the pretrained model (correlated with increase in size). While the power of scale and pretraining have been demonstrated numerous times in non-private deep learning [KMH+20], one common wisdom in private learning had been that large models tend to perform worse. This intuition was based on (a) results in differentially private convex optimization, most of which predicted that errors would scale proportionally with the dimension of the learning problem in the worst case, and (b) empirical observations that the noise injected to ensure privacy tends to greatly exceed the gradient in magnitude for large models [YZCL21, Kam20].

For instance, consider the problem of differentially private convex *empirical risk minimization* (ERM). Here, we are given a dataset of n examples $\mathcal{D} = \{s_j\}_{j=1}^n \in \mathcal{S}^n$, a convex

set $\mathcal{K} \subseteq \mathbb{R}^d$ (not necessarily bounded), and the goal is to perform the optimization

$$\text{minimize}_{x \in \mathcal{K}} F(x; \mathcal{D}) = \frac{1}{n} \sum_{j=1}^{n} f(x; s_j)$$

subject to differential privacy, where $f(\cdot; s)$ is convex over $\mathcal{K}$ for all $s \in \mathcal{S}$. For bounded $\mathcal{K}$, past works presented matching upper and lower bounds that are dimension-dependent under the usual Lipschitz assumption on the objective [BST14, CMS11]. These results seem to suggest that the performance of differentially private ERM algorithms inevitably degrades with increasing problem size in the worst case, and present a seeming discrepancy between recent empirical results on large-scale fine-tuning.[1]

To better understand the relation between problem size and the performance of differentially private learning, we study the following question both theoretically and empirically:

When does the performance of differentially private stochastic gradient descent (DP-SGD) not degrade with increasing problem dimension?

On the theoretical front, we show that DP-SGD can result in dimension-independent error bounds even when gradients span the entire ambient space for unconstrained optimization problems. We identify that the standard dependence on the dimension of the ambient space can be replaced by the magnitudes of gradients projected onto subspaces of varying dimensions. We formalize this in a condition that we call *restricted Lipschitz continuity* and derive refined bounds for the excess empirical and population risks for DP-SGD when loss functions obey this condition. We show that when the restricted Lipschitz coefficients decay rapidly, both the excess empirical and population risks become dimension-independent. This extends a previous work which derived rank-dependent bounds for learning generalized linear models in an unconstrained space [SSTT21].

Our theoretical results shed light on the recent success of large-scale differentially private fine-tuning. We empirically show that gradients of language models during fine-tuning are mostly controlled by a few principal components — a behavior that is similar to conditions

[1] We judiciously choose to describe the discrepancy as seeming, since the refined analysis presented in the current work suggests that the discrepancy is likely non-existent.

under which we obtain dimension-independent bounds for private convex ERM. This provides a possible explanation for the observation that densely fine-tuning with DP-SGD need not necessarily experience much worse performance than sparsely fine-tuning [LTLH21]. Moreover, it suggests that DP-SGD can be adaptive to problems that are effectively low-dimensional (as characterized by restricted Lipschitz continuity) without further algorithmic intervention.

We summarize our contributions below.

(1) We introduce a condition on the objective function that we term restricted Lipschitz continuity. This condition generalizes the usual Lipschitz continuity notion and gives rise to refined analyses when magnitudes of gradients projected onto diminishing subspaces decay rapidly.

(2) Under restricted Lipschitz continuity, we present refined bounds on the excess empirical and population risks for DP-SGD when optimizing convex objectives. These bounds generalize previous dimension-independent results [SSTT21] and are broadly applicable to cases where gradients are full rank but most coordinates only marginally influence the objective.

(3) Our theory sheds light on recent successes of large-scale differentially private fine-tuning of language models. We show that gradients obtained through fine-tuning mostly lie in a subspace spanned by a few principal components — a behavior similar to when optimizing a restricted Lipschitz continuous loss with decaying coefficients. These empirical results provide a possible explanation for the recent success of large-scale private fine-tuning.

11.2 Preliminaries

We define the notation used throughout this work and state the problems of differentially private empirical risk minimization and differentially private stochastic convex optimization. Finally, we give a brief recap of differentially private stochastic gradient descent, and existing dimension-dependent and dimension-independent results in the literature.

Notation & Terminology. For a positive integer $n \in \mathbb{N}_+$, define the shorthand $[n] = \{1, \ldots, n\}$. For a vector $x \in \mathbb{R}^d$, denote its ℓ_2-norm by $\|x\|_2$. Given a symmetric $M \in \mathbb{R}^{d \times d}$, let $\lambda_1(M) \geq \lambda_2(M) \geq \cdots \geq \lambda_d(M)$ denote its eigenvalues. Given a positive semidefinite matrix A, let $\|x\|_A = (x^\top A x)^{1/2}$ denote the induced Mahalanobis norm. For scalar functions f and g, we write $f \lesssim g$ if there exists a positive constant C such that $f(x) \leq Cg(x)$ for all input x in the domain.

11.2.1 Differentially Private Empirical Risk Minimization and Stochastic Convex Optimization

Before stating the theoretical problem of interest, we recall the basic concepts of Lipschitz continuity, convexity, and approximate differential privacy.

Definition 11.2.1 (Lipschitz Continuity)**.** The loss function $h : \mathcal{K} \to \mathbb{R}$ is G-Lipschitz with respect to the ℓ_2 norm if for all $x, x' \in \mathcal{K}$, $|f(x) - f(x')| \leq G\|x - x'\|_2$.

Definition 11.2.2 (Convexity)**.** The loss function $h : \mathcal{K} \to \mathbb{R}$ is convex if $h(\alpha x + (1-\alpha)y) \leq \alpha h(x) + (1-\alpha)h(y)$, for all $\alpha \in [0, 1]$, and x, y in a convex domain $\mathcal{K}$.

Definition 11.2.3 (Approximate Differential Privacy [DR14])**.** A randomized algorithm $\mathcal{M}$ is (ε, δ)-differentially private if for all neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ by a single record and all sets $\mathcal{O} \subset \text{range}(\mathcal{M})$, the following expression holds

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta.$$

In this work, we study both *differentially private empirical risk minimization* (DP-ERM) for convex objectives and *differentially private stochastic convex optimization* (DP-SCO).

In DP-ERM for convex objectives, we are given a dataset $\mathcal{D} = \{s_j\}_{j \in [n]} \in \mathcal{S}^n$ of n examples. Each per-example loss $f(\cdot; s_j)$ is convex over the convex body $\mathcal{K} \subseteq \mathbb{R}^d$ and G-Lipschitz. We aim to design an (ε, δ)-DP algorithm that returns a solution x^{priv} which approximately minimizes the empirical risk $F(x; \mathcal{D}) := \frac{1}{n} \sum_{s_j \in \mathcal{D}} f(x; s_j)$. The term $\mathbb{E}_{x^{\text{priv}}} \left[F(x^{\text{priv}}; \mathcal{D}) - \min_{x \in \mathcal{K}} F(x; \mathcal{D}) \right]$ is referred to as the *excess empirical risk*.

In DP-SCO, we assume the per-example loss $f(\cdot; s)$ is convex and G-Lipschitz for all $s \in \mathcal{S}$, and we are given n examples drawn i.i.d. from some (unknown) distribution $\mathcal{P}$. The goal is to design an (ε, δ)-DP algorithm which approximately minimizes the population risk $F(x; \mathcal{P}) := \mathbb{E}_{s\sim\mathcal{P}}[f(x; s)]$. The term $\mathbb{E}_{x^{\text{priv}}}\left[F(x^{\text{priv}}; \mathcal{P}) - \min_{x\in\mathcal{K}} F(x; \mathcal{P})\right]$ is referred to as the *excess population risk*.

11.2.2 Differentially Private Stochastic Gradient Descent

Differentially Private Stochastic Gradient Descent (DP-SGD) [ACG+16, SCS13] is a popular algorithm for DP convex optimization. For the theoretical analysis, we study DP-SGD for *unconstrained* optimization. To facilitate analysis, we work with the ℓ_2 regularized objective expressed as

$$F_\alpha(x; \mathcal{D}) = \frac{1}{n}\sum_{j=1}^{n} f(x; s_j) + \frac{\alpha}{2}\|x - x^{(0)}\|_2^2.$$

To optimize this objective, DP-SGD independently samples an example in each iteration and updates the solution by combining the gradient with an isotropic Gaussian whose scale is proportional to G, the Lipschitz constant of f. Algorithm 35 presents the pseudocode.

Algorithm 35: DP-SGD for optimizing regularized finite-sum objectives

Input: Initial iterate $x^{(0)}$, dataset $\mathcal{D} = \{s_j\}_{j\in[n]}$, per-step noise magnitude σ, number of updates T, learning rate η, Lipschitz constant G of f.
for $t = 1, \ldots, T$ **do**
  $j_t \sim \text{Uniform}([n])$
  $x^{(t)} = x^{(t-1)} - \eta\Big(\nabla f(x^{(t-1)}; s_{j_t}) + \alpha(x^{(t-1)} - x^{(0)}) + G \cdot \zeta_t\Big), \quad \zeta_t \sim \mathcal{N}(0, \sigma^2 I_d)$
end
Return: $\overline{x} := \frac{1}{T}\sum_{t=1}^{T} x^{(t)}$.

It is straightforward to show that Algorithm 35 satisfies differential privacy. The following lemma quantifies the overall privacy spending and builds on a long line of work on accounting the privacy loss of compositions [ACG+16, BBG18].

Lemma 11.2.4 ([KLL21])**.** *There exists constants c_1 and c_2 such that for $n \geq 10$, $\varepsilon < c_1 T/n^2$ and $\delta \in (0, \frac{1}{2}]$, DP-SGD (Algorithm 35) is (ε, δ)-DP whenever $\sigma \geq \frac{c_2\sqrt{T\log(1/\delta)}}{\varepsilon n}$.*

11.2.3 On the Dimension Dependence of Private Learning

Early works on bounding the excess empirical and population risks for privately optimizing convex objectives focused on a constrained optimization setup where algorithms typically project iterates back onto a fixed bounded domain after each noisy gradient update. Results in this setting suggested that risks are inevitably dimension-dependent in the worst case. For instance, it was shown that the excess empirical risk bound $\Theta(G\left\|\mathcal{K}\right\|_2 \sqrt{d\log(1/\delta)}n^{-1}\varepsilon^{-1})$ and excess population risk bound $\Theta(G\left\|\mathcal{K}\right\|_2 (n^{-1/2} + \sqrt{d\log(1/\delta)}n^{-1}\varepsilon^{-1}))$ are tight when privately optimizing convex G-Lipschitz objectives in a convex domain of diameter $\left\|\mathcal{K}\right\|_2$ [BST14]. Moreover, the lower bound instances in these works imply that such dimension-dependent lower bounds also apply when one considers the class of loss functions whose gradients are low-rank.

The body of work on unconstrained convex optimization is arguably less abundant, with the notable result that differentially private gradient descent need not suffer from a dimension-dependent penalty when learning generalized linear models with low-rank data (equivalently stated, when gradients are low-rank) [SSTT21]. Our main theoretical results (Theorems 11.3.3 and 11.3.5) extend this line of work and show that dimension-independence is achievable under weaker conditions.

11.3 Dimension-Independence via Restricted Lipschitz Continuity

In this section, we introduce the restricted Lipschitz continuity condition and derive improved bounds for the excess empirical and population risks for DP-SGD when optimizing convex objectives.

Definition 11.3.1 (Restricted Lipschitz Continuity)**.** We say that the loss function $h : \mathcal{K} \to \mathbb{R}$ is restricted Lipschitz continuous with coefficients $\{G_k\}_{k=0}^d$, if for all $k \in \{0, \ldots, d\}$,

there exists an orthogonal projection matrix P_k with rank k such that

$$\|(I - P_k)\nabla h(x)\|_2 \leq G_k,$$

for all $x \in \mathcal{K}$ and all subgradients $\nabla h(x) \in \partial h(x)$.

Note that any G-Lipschitz function is also trivially restricted Lipschitz continuous with coefficients $G = G_0 = G_1 = \cdots = G_d$, since orthogonal projections never increase the ℓ_2-norm of a vector (generally, it is easy to see that $G = G_0 \geq G_1 \geq G_2 \geq \cdots \geq G_d = 0$). On the other hand, we expect that a function which exhibits little growth in some subspace of dimension k to have a restricted Lipschitz coefficient G_{d-k} of almost 0. Our bounds on DP convex optimization characterize errors through the use of restricted Lipschitz coefficients. We now summarize the main assumption.

Assumption 11.3.2. *The per-example loss function $f(\cdot; s)$ is convex and G-Lipschitz continuous for all $s \in \mathcal{S}$. The empirical loss $F(\cdot; \mathcal{D})$ is restricted Lipschitz continuous with coefficients $\{G_k\}_{k=0}^d$.*

11.3.1 Bounds for Excess Empirical Loss

We present the main theoretical result on DP-ERM for convex objectives. The result consists of two components: Equation (11.1) is a general bound that is applicable to any sequence of restricted Lipschitz coefficients; Equation (11.2) specializes the previous bound when the sequence of coefficients decays rapidly and demonstrates dimension-independent error scaling.

Theorem 11.3.3 (Excess Empirical Loss)**.** *Let $\delta \in (0, \frac{1}{2}]$ and $\varepsilon \in (0, 10]$. Under Assumption 11.3.2, for all $k \in [d]$, setting $T = \Theta(n^2 + d\log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T\log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^S s^2 2^s G^2_{2^{s-1}k}}$, where $S = \lfloor \log(d/k) \rfloor + 1$, DP-SGD (Algorithm 35) is (ε, δ)-DP, and*

$$\mathbb{E}\left[F(\overline{x}; \mathcal{D}) - \min_x F(x; \mathcal{D})\right] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^S s^2 2^s G^2_{2^{s-1}k}}, \tag{11.1}$$

where $\|x^{(0)} - \operatorname{argmin}_x F(x;\mathcal{D})\|_2 \le D$, $\overline{x}$ *is the (random) output of DP-SGD (Algorithm 35), and the expectation is over the randomness of* $\overline{x}$. *Moreover, if for some* $c > 1/2$, *we have* $G_k \le G_0 k^{-c}$ *for all* $k \in [d]$, *and in addition* $n \ge \varepsilon^{-1}\sqrt{\log(1/\delta)}$, *minimizing the right hand side of (11.1) with respect to* k *yields*

$$\mathbb{E}\left[F(\overline{x};\mathcal{D}) - \min_x F(x;\mathcal{D})\right] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}. \tag{11.2}$$

We include a sketch of the proof techniques in Subsection 11.3.3 and defer the full proof to Subsection 11.3.4.

Remark 11.3.4. Consider DP-ERM for learning generalized linear models with convex and Lipschitz losses. When the (empirical) data covariance is of rank $r < d$, the span of gradients $\operatorname{span}(\{\nabla_x F(x)\})$ is also of rank r. Thus, the average loss is restricted Lipschitz continuous with coefficients where $G_{r'} = 0$ for all $r' > r$. Setting $k = r$ in (11.1) yields an excess empirical risk bound of order $O\left(G_0 D\sqrt{r \cdot \log(1/\delta)}\varepsilon^{-1}n^{-1}\right)$. This recovers the previous dimension-independent result in [SSTT21].

The restricted Lipschitz continuity condition can be viewed as a generalized notion of rank. The result captured in (11.2) suggests that the empirical loss achieved by DP-SGD does not depend on the problem dimension if the sequence of restricted Lipschitz coefficients decays rapidly. We leverage these insights to build intuition for understanding privately fine-tuning language models in Section 11.4.

11.3.2 Bounds for Excess Population Loss

For DP-SCO, we make use of the *stability* of DP-SGD to bound its generalization error [BE02], following previous works [BFTGT19, BFGT20, SSTT21]. The bound on the excess population loss follows from combining the bounds on the excess empirical risk and the generalization error.

Theorem 11.3.5 (Excess Population Loss)**.** *Let* $\delta \in (0, \frac{1}{2}]$ *and* $\varepsilon \in (0, 10]$. *Under Assumption 11.3.2, for all* $k \in [d]$, *by setting* $T = \Theta(n^2 + d\log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T\log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2(T/n + k\sigma^2)}}$ *and* $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$, *where* $S = \lfloor \log(d/k) \rfloor + 1$, *DP-SGD*

(Algorithm 35) is (ε, δ)-DP, and

$$\mathbb{E}\left[F(\overline{x}; \mathcal{P}) - \min_x F(x; \mathcal{P})\right] \lesssim \frac{G_0 D}{\sqrt{n}} + \frac{G_0 D \sqrt{k \log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2},$$

where $\|x^{(0)} - \operatorname{argmin}_x F(x; \mathcal{P})\|_2 \le D$, $\overline{x}$ is the (random) output of DP-SGD (Algorithm 35), and the expectation is over the randomness of $\overline{x}$.

Moreover, if for some $c > 1/2$, we have $G_k \le G_0 k^{-c}$ for all $k \in [d]$, and in addition $n > \varepsilon^{-1}\sqrt{\log(1/\delta)}$, minimizing the above bound with respect to k yields

$$\mathbb{E}\left[F(\overline{x}; \mathcal{P}) - \min_x F(x; \mathcal{P})\right] \lesssim \frac{G_0 D}{\sqrt{n}} + G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

Remark 11.3.6. Our result on DP-SCO also recovers the DP-SCO rank-dependent result for learning generalized linear models with convex and Lipschitz losses [SSTT21].

Remark 11.3.7. When $c > 1/2$, $\varepsilon = \Theta(1)$ and $\delta = 1/\mathrm{poly}(n)$, the population loss matches the (non-private) informational-theoretical lower bound $\Omega(G_0 D/\sqrt{n})$ [ABRW12].

Remark 11.3.8. Our results on DP-ERM and DP-SCO naturally generalize to (full-batch) DP-GD.

11.3.3 Overview of Proof Techniques

The privacy guarantees in Theorems 11.3.3 and 11.3.5 follow from Lemma 8.9.2. It suffices to prove the utility guarantees. We give an outline of the main proof techniques first and present full proofs afterwards. The following is a sketch of the core technique for deriving (11.2) in Theorem 11.3.3. For simplicity, we write $f_j(\cdot)$ for $f(\cdot; s_j)$ and $F(\cdot)$ for $F(\cdot; \mathcal{D})$ when there is no ambiguity.

By convexity, the error term of SGD is upper bounded as follows

$$f_j(x^{(t)}) - f_j(x^*) \le \nabla f_j(x^{(t)})^\top (x^{(t)} - x^*), \tag{11.3}$$

where $j \in [n]$ is the random index sampled at iteration t.

By definition of G_k, we know that there is a k dimensional subspace U such that the gradient component orthogonal to U is small when G_k is small. Naively, one decomposes the gradient $\nabla f_j(x^{(t)}) = g_1 + g_2$, where $g_1 \in U$ and $g_2 \in U^\perp$, and separately bounds the two terms $g_1^\top (x^{(t)} - x^*)$ and $g_2^\top (x^{(t)} - x^*)$. Since g_1 lies in a k dimensional subspace, one can follow existing arguments on DP-SGD to bound $g_1^\top (x^{(t)} - x^*)$. Unfortunately, this argument does not give a dimension-independent bound. Although $\|g_2\|_2 \le G_k$ (which can be small for large k), the term $\|x^{(t)} - x^*\|_2$ is as large as $\Omega(\sqrt{d})$ with high probability due to the isotropic Gaussian noise injected in DP-SGD.

Our key idea is to partition the whole space $\mathbb{R}^d$ into $\lfloor \log(d/k) \rfloor + 2$ orthogonal subspaces, expressing the error term $\nabla f_j(x^{(t)})^\top (x^{(t)} - x^*)$ as the sum of individual terms, each of which corresponds to a projection to a particular subspace. Fix a $k \in [d]$, and consider the following subspaces: Let $U_0 = \text{range}(P_k)$, U_s be the subspace orthogonal to all previous subspaces such that $\bigoplus_{i=0}^{s} U_i = \text{range}(P_{2^s k})$ for $s = 1, 2, \cdots, \lfloor \log(d/k) \rfloor$, and U_S be the subspace such that the orthogonal direct sum of all subspaces $\{U_i\}_{i=0}^{S}$ is $\mathbb{R}^d$, where $S = \lfloor \log(d/k) \rfloor + 1$. Here, P_i is the orthogonal projection matrix with rank i promised by G_i in Assumption 11.3.2. Let Q_s be the orthogonal projection to the subspace U_s, and observe that $\text{rank}(Q_s) \le 2^s k$ and $\|Q_s \nabla F(x)\|_2 \le G_{2^{s-1}k}$ for all x and all $s \ge 1$. Rewriting the right hand side of (11.3) with this decomposition yields

$$f_j(x^{(t)}) - f_j(x^*) \le \left(Q_0 \nabla f_j(x^{(t)}) + \sum_{s=1}^{S} Q_s \nabla f_j(x^{(t)}) \right)^\top (x^{(t)} - x^*).$$

On the one hand, if G_k decays quickly, $\|\mathbb{E}_j[Q_s \nabla f_j]\|_2$ can be small for large s. On the other hand, we expect $\|Q_s(x^{(t)} - x^*)\|_2$ to be small for small s where Q_s is an orthogonal projection onto a small subspace. Thus, for each s, $\nabla f_j(x^{(t)})^\top Q_s (x^{(t)} - x^*)$ is small either due to a small gradient (small $Q_s \nabla f_j$ in expectation over the random index) or small noise (small $Q_s(x^{(t)} - x^*)$), since noise injected in DP-SGD is isotropic. More formally, in Lemma 11.3.9, we show that for any projection matrix Q with rank r, $\|Q(x^{(t)} - x^{(0)})\|_2$ can be upper bounded by a term that depends only on r (rather than d).

11.3.4 Proof of Theorem 11.3.3

Before bounding the utility of DP-SGD, we first bound $x^{(t)} - x^{(0)}$ in expectation.

Lemma 11.3.9. *Suppose Assumption 11.3.2 holds. Let Q be an orthogonal projection matrix with rank r and suppose that $\|Q\nabla f(x;s)\|_2 \le G_Q$ for all $x \in \mathbb{R}^d$ and $s \in \mathcal{S}$. If we set $\eta \le \frac{1}{2\alpha}$ in DP-SGD, then for all $t > 0$, we have*

$$\mathbb{E}\,\|Q(x^{(t)} - x^{(0)})\|_2^2 \le \frac{4G_Q^2}{\alpha^2} + \frac{2\eta G_0^2}{\alpha}(1 + r\sigma^2).$$

Proof of Lemma 11.3.9. By the assumption, we know $\|Q\nabla F(x)\|_2 \le G_Q$ and $\|\nabla F(x)\|_2 \le G_0$. Let $z^{(t)} = x^{(t)} - x^{(0)}$. Note that

$$\begin{aligned} z^{(t+1)} &= x^{(t+1)} - x^{(0)} \\ &= x^{(t)} - \eta\Big(\nabla f_j(x^{(t)}) + \alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\Big) - x^{(0)} \\ &= (1 - \alpha\eta)z^{(t)} - \eta(\nabla f_j(x^{(t)}) + G_0 \cdot \zeta), \end{aligned}$$

where $\zeta \sim \mathcal{N}(0, \sigma^2 I_d)$ is the isotropic Gaussian noise drawn in $(t+1)$th step. For simplicity, we use $\tilde{\nabla} f_j(x^{(t)})$ to denote the noisy subgradient $\nabla f_j(x^{(t)}) + G_0 \cdot \zeta$. Hence, we have

$$\|Qz^{(t+1)}\|_2^2 = (1 - \alpha\eta)^2\|Qz^{(t)}\|_2^2 - 2\eta(1 - \alpha\eta)(Qz^{(t)})^\top Q\tilde{\nabla} f_j(x^{(t)}) + \eta^2\|Q\tilde{\nabla} f_j(x^{(t)})\|_2^2.$$

Taking expectation over the random sample f_j and random Gaussian noise ζ, we have

$$\begin{aligned} \mathbb{E}\,\|Qz^{(t+1)}\|_2^2 =& (1 - \alpha\eta)^2\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 - 2\eta(1 - \alpha\eta) \cdot \mathbb{E}\left((Qz^{(t)})^\top(Q\nabla F(x^{(t)}))\right) \\ &+ \eta^2\,\mathbb{E}\,\|Q\tilde{\nabla} f_j(x^{(t)})\|_2^2 \\ \le& (1 - \alpha\eta)\,\mathbb{E}[\|Qz^{(t)}\|_2^2] + 2\eta G_Q \cdot \mathbb{E}[\|Qz^{(t)}\|_2] + \eta^2 G_0^2(1 + r\sigma^2), \end{aligned}$$

where we used the fact that ζ has zero mean, $\|\nabla f_j(x^{(t)})\|_2 \le G_0$, $\|Q\nabla F(x^{(t)})\|_2 \le G_Q$ and $\eta \le \frac{1}{2\alpha}$. Further simplifying and taking expectation over all iterations, we have

$$\mathbb{E}\,\|Qz^{(t+1)}\|_2^2 \le (1 - \alpha\eta)\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 + 2\eta(\frac{\alpha}{4}\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 + \frac{1}{\alpha}G_Q^2) + \eta^2 G_0^2(1 + r\sigma^2)$$

$$\leq (1 - \frac{\alpha\eta}{2}) \mathbb{E} \|Qz^{(t)}\|_2^2 + \frac{2\eta}{\alpha} G_Q^2 + \eta^2 G_0^2 (1 + r\sigma^2). \tag{11.4}$$

Using that $z^{(0)} = 0$, we know $\mathbb{E} \|Qz^{(0)}\|_2^2 = 0$. Solving the recursion (Equation (11.4)) gives

$$\mathbb{E} \|Qz^{(t)}\|_2^2 \leq \frac{2}{\alpha\eta} (\frac{2\eta}{\alpha} G_Q^2 + \eta^2 G_0^2 (1 + r\sigma^2))$$

for all t. This concludes the proof. □

Now, we are ready to bound the utility. The proof builds upon the standard mirror descent proof.

Lemma 11.3.10. *Let $\delta \in (0, \frac{1}{2}]$, and $\varepsilon \in (0, 10]$. Under Assumption 11.3.2, let $x^{(0)}$ be the initial iterate and $x^* \in \mathbb{R}^d$ be such that $\|x^{(0)} - x^*\|_2 \leq D$. For all $k \in [d]$, setting $T = \Theta(n^2 + d \log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T \log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$, we have*

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D \sqrt{k \log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2},$$

where $S = \lfloor \log(d/k) \rfloor + 1$, $\overline{x}$ is the output of DP-SGD, and the expectation is under the randomness of DP-SGD.

Moreover, if $G_k \leq G_0 k^{-c}$ for each k for some $c > 1/2$, and in addition $n > \varepsilon^{-1}\sqrt{\log(1/\delta)}$, picking the best $k \in [d]$ for the bound above gives

$$\mathbb{E}[F(\overline{x}; \mathcal{D}) - F(x^*; \mathcal{D})] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

Proof of Lemma 11.3.10. The above statement is true for $k = d$ by standard arguments in past work [BST14, SSTT21]. Now fix a $k \in \{1, \ldots, d-1\}$. Our key idea is to split the whole space $\mathbb{R}^d$ into different subspaces. We define the following set of subspaces:

- $U_0 = \text{range}(P_k)$.
- For $s = 1, 2, \ldots, \lfloor \log(d/k) \rfloor$, let $U_s \subseteq \text{range}(P_{2^s k})$ be a subspace with maximal dimen-

sion such that $U_s \perp U_i$ for all $i = 0, \ldots, s-1$.

- For $S = \lfloor \log(d/k) \rfloor + 1$, let $U_S \subseteq \mathbb{R}^d$ be the subspace such that $\bigoplus_{i=0}^{S} U_i = \mathbb{R}^d$, and $U_S \perp U_i$ for all $i = 0, \ldots, S-1$.

Recall P_i is the orthogonal projection matrix with rank i that gives rise to G_i in Assumption 11.3.2. In the above, we have assumed that the base of log is 2. Let Q_s be the orthogonal projection matrix that projects vectors onto the subspace U_s. Note that $\text{rank}(Q_s) \leq 2^s k$ since $U_s \subseteq \text{range}(P_{2^s k})$. Moreover, it's clear that $U_s \perp \text{range}(P_{2^{s-1}k})$ for all $s \in \{1, \ldots, S\}$. This is true by construction $\bigoplus_{i=0}^{s-1} U_i \supseteq \text{range}(P_{2^{s-1}k})$ and that $U_s \perp \bigoplus_{i=0}^{s-1} U_i$. Thus,

$$\|Q_s \nabla F(x)\|_2 = \|Q_s (I - P_{2^{s-1}k}) \nabla F(x)\|_2 \leq \|Q_s\|_{\text{op}} \|(I - P_{2^{s-1}k}) \nabla F(x)\|_2 \leq G_{2^{s-1}k} \quad (11.5)$$

for all $x \in \mathbb{R}^d$ and all $s \in \{1, \ldots, S\}$.

Let $j \in [n]$ be the (uniformly random) index sampled in iteration t of DP-SGD. By convexity of the individual loss f_j,

$$f_j(x^{(t)}) - f_j(x^*) \leq \nabla f_j(x^{(t)})^\top (x^{(t)} - x^*).$$

By construction, $\mathbb{R}^d$ is the orthogonal direct sum of the subspaces $\{U_j\}_{j=0}^{S}$, and thus any vector $v \in \mathbb{R}^d$ can be rewritten as the sum $\sum_{i=0}^{S} Q_i v$. We thus split the right hand side of the above as follows

$$f_j(x^{(t)}) - f_j(x^*) \leq \left(Q_0 \nabla f_j(x^{(t)}) + \sum_{s=1}^{S} Q_s \nabla f_j(x^{(t)}) \right)^\top (x^{(t)} - x^*). \quad (11.6)$$

We use different approaches to bound $(Q_0 \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$ and $(Q_s \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$ when $s \geq 1$, and we discuss them separately in the following.

Bounding $(Q_0 \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$: Recall that

$$x^{(t+1)} = x^{(t)} - \eta \Big(\nabla f_j(x^{(t)}) + \alpha (x^{(t)} - x^{(0)}) + G_0 \cdot \zeta \Big)$$

for some Gaussian $\zeta \sim \mathcal{N}(0, \sigma^2 I_d)$. Hence, we have

$$\begin{aligned}(\nabla f_j(x^{(t)}))^\top Q_0(x^{(t)} - x^*) = & \left(\frac{1}{\eta}(x^{(t)} - x^{(t+1)}) - \alpha(x^{(t)} - x^{(0)}) - G_0 \cdot \zeta\right)^\top Q_0(x^{(t)} - x^*) \\ = & \left(\frac{1}{\eta} Q_0(x^{(t)} - x^{(t+1)})\right)^\top Q_0(x^{(t)} - x^*) - \left(\alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\right)^\top Q_0(x^{(t)} - x^* \\ = & \frac{1}{2\eta}\left(\|Q_0(x^{(t)} - x^*)\|_2^2 - \|Q_0(x^{(t+1)} - x^*)\|_2^2 + \|Q_0(x^{(t)} - x^{(t+1)})\|_2^2\right) \\ & - \left(\alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\right)^\top Q_0(x^{(t)} - x^*), \end{aligned} \tag{11.7}$$

where we used the fact that $Q_0^2 v = Q_0 v$ for any $v \in \mathbb{R}^d$ (since Q_0 is a projection matrix), and the last equality follows from

$$\begin{aligned}& 2(Q_0(x^{(t)} - x^{(t+1)}))^\top Q_0(x^{(t)} - x^*) \\ & \qquad = \|Q_0(x^{(t)} - x^*)\|_2^2 - \|Q_0(x^{(t+1)} - x^*)\|_2^2 + \|Q_0(x^{(t)} - x^{(t+1)})\|_2^2.\end{aligned}$$

Taking expectation on ζ over both sides of Equation (11.7) and making use of the fact that ζ has mean 0, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\zeta}(Q_0 \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*) = & \frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)} - x^*)\|_2^2 - \mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)} - x^*)\|_2^2 + \mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)} - x^{(t+1)})\|_2^2\right) \\ & - \alpha \mathop{\mathbb{E}}_{\zeta}\left((x^{(t)} - x^{(0)})^\top Q_0(x^{(t)} - x^*)\right).\end{aligned}$$

Recalling the definition of Q_0 and that Q_0 has rank at most k, one has

$$\begin{aligned}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)} - x^{(t+1)})\|_2^2 &= \eta^2 \mathop{\mathbb{E}}_{\zeta}\|Q_0\big(\nabla f_j(x^{(t)}) + \alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\big)\|_2^2 \\ &= \eta^2 \mathop{\mathbb{E}}_{\zeta}\|Q_0\big(\nabla f_j(x^{(t)}) + \alpha(x^{(t)} - x^{(0)})\big)\|_2^2 + \eta^2 G_0^2 k\sigma^2 \\ &\le 2\eta^2 G_0^2(1 + k\sigma^2) + 2\eta^2\alpha^2 \mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)} - x^{(0)})\|_2^2.\end{aligned}$$

Moreover, one has

$$-\alpha(x^{(t)} - x^{(0)})^\top Q_0(x^{(t)} - x^*) = -\alpha(x^{(t)} - x^{(0)})^\top Q_0(x^{(t)} - x^{(0)}) - \alpha(x^{(t)} - x^{(0)})^\top Q_0(x^{(0)} - x^*)$$

$$\leq -\frac{\alpha}{2}\|Q_0(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha}{2}\|Q_0(x^{(0)}-x^*)\|_2^2.$$

Therefore, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\zeta}(Q_0\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*) &\leq \frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)}-x^*)\|_2^2\right)+\eta G_0^2(1+k\sigma^2)\\ &\quad+\eta\alpha^2\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(0)})\|_2^2-\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(0)}-x^*)\|_2^2\\ &\leq \frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)}-x^*)\|_2^2\right) \qquad (11.8)\\ &\quad+\eta G_0^2(1+k\sigma^2)+\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(0)}-x^*)\|_2^2, \qquad (11.9)\end{aligned}$$

where we used $\eta\leq\frac{1}{2\alpha}$ at the end.

Bounding $(Q_s\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*)$ **:** We bound the objective above for each s separately. By taking expectation over the random f_j, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{f_j}(Q_s\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*) &= (Q_s\nabla F(x^{(t)}))^\top(x^{(t)}-x^*)\\ &\leq \|Q_s\nabla F(x^{(t)})\|_2\cdot\|Q_s(x^{(t)}-x^*)\|_2\\ &\leq \frac{1}{\alpha_s}\|Q_s\nabla F(x^{(t)})\|_2^2+\frac{\alpha_s}{4}\|Q_s(x^{(t)}-x^*)\|_2^2\\ &\leq \frac{G_{2^{s-1}k}^2}{\alpha_s}+\frac{\alpha_s}{2}\|Q_s(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha_s}{2}\|Q_s(x^{(0)}-x^*)\|_2^2,\end{aligned} \qquad (11.10)$$

where we chose $\alpha_s=\alpha s^{-2}2^{-s}$ and used the bound (11.5) and Young's inequality at the end.

Bounding Equation (11.6)**:** Combining both the terms (11.9) and (11.10) and taking expectation over all randomness, we have

$$\begin{aligned}\mathbb{E}[F(x^{(t)})-F(x^*)] &\leq \frac{1}{2\eta}(\mathbb{E}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathbb{E}\|Q_0(x^{(t+1)}-x^*)\|_2^2)+\eta G_0^2(1+k\sigma^2)+\frac{\alpha}{2}\mathbb{E}\|Q_0(x^{(0)}-x^*)\|_2^2\\ &\quad+\sum_{s=1}^{S}\frac{G_{2^{s-1}k}^2}{\alpha_s}+\frac{1}{2}\sum_{s=1}^{S}\alpha_s\mathbb{E}\|Q_s(x^{(t)}-x^{(0)})\|_2^2+\frac{1}{2}\sum_{s=1}^{S}\alpha_s\mathbb{E}\|Q_s(x^{(0)}-x^*)\|_2^2\\ &\leq \frac{1}{2\eta}(\mathbb{E}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathbb{E}\|Q_0(x^{(t+1)}-x^*)\|_2^2)+\eta G_0^2(1+k\sigma^2)+\frac{\alpha}{2}\|x^{(0)}-x^*\|_2^2\end{aligned}$$

$$+\sum_{s=1}^{S}\frac{G_{2^{s-1}k}^2}{\alpha_s}+\frac{1}{2}\sum_{s=1}^{S}\alpha_s\,\mathbb{E}\,\|Q_s(x^{(t)}-x^{(0)})\|_2^2.$$

Recall $\alpha\cdot\eta\le 1/2$. Under the other assumptions, by Lemma 11.3.9, one can show

$$\begin{aligned}\mathbb{E}\,\|Q_s(x^{(t)}-x^{(0)})\|^2 &\le \frac{4G_{2^{s-1}k}^2}{\alpha^2}+\frac{2\eta G_0^2}{\alpha}(1+2^s k\sigma^2)\\ &\le \frac{4G_{2^{s-1}k}^2}{\alpha_s^2}+\frac{2\eta G_0^2}{\alpha_s s^2}(1+k\sigma^2).\end{aligned}$$

Using $\sum_{s=1}^{\infty}s^{-2}\le 2$, we have

$$\begin{aligned}\mathbb{E}\,F(x^{(t)})-\mathbb{E}\,F(x^*) &\le \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)}-x^*)\|_2^2-\mathbb{E}\,\|Q_0(x^{(t+1)}-x^*)\|_2^2)+\frac{\alpha}{2}\|x^{(0)}-x^*\|^2\\ &\quad+\eta G_0^2(1+k\sigma^2)+3\sum_{s=1}^{S}\frac{G_{2^{s-1}k}^2}{\alpha_s}+2\eta G_0^2(1+k\sigma^2)\\ &\le \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)}-x^*)\|_2^2-\mathbb{E}\,\|Q_0(x^{(t+1)}-x^*)\|_2^2)+\frac{\alpha}{2}\|x^{(0)}-x^*\|_2^2\\ &\quad+3\eta G_0^2(1+k\sigma^2)+\frac{3}{\alpha}\sum_{s=1}^{S}s^2 2^s G_{2^{s-1}k}^2.\end{aligned}$$

Summing up over $t=1,2,\cdots,T$, by the assumption that $\|x^{(0)}-x^*\|_2\le D$ and convexity of the function, we have

$$\mathbb{E}[F(\overline{x})-F(x^*)]\le\frac{D^2}{2\eta T}+3\eta G_0^2(1+k\sigma^2)+\frac{\alpha}{2}D^2+\frac{3}{\alpha}\sum_{s=1}^{S}s^2 2^s G_{2^{s-1}k}^2. \tag{11.11}$$

Set the parameters $T=c_1(n^2+d\log^2 d)$, $\sigma=\frac{c_2\sqrt{T\log(1/\delta)}}{n\varepsilon}$, $\eta=\sqrt{\frac{D^2}{T\cdot G_0^2\cdot k\sigma^2}}$ and $\alpha=\frac{1}{D}\sqrt{\sum_{s=1}^{S}s^2 2^s G_{2^{s-1}k}^2}$ for some large constants c_1,c_2. Note that this choice of parameters satisfies

$$\begin{aligned}\eta\cdot\alpha &= \sqrt{\frac{D^2}{T\cdot G_0^2\cdot k\sigma^2}}\cdot\frac{\sqrt{\sum_{s=1}^{S}s^2 2^s G_{2^{s-1}k}^2}}{D}\\ &\le \sqrt{\frac{G_0^2(2d)\log^3(2d)}{T\cdot G_0^2\cdot k\sigma^2}}=\frac{n\varepsilon}{c_2 T}\sqrt{\frac{(2d)\log^3(2d)}{k\cdot\log(1/\delta)}}\end{aligned}$$

$$\leq \frac{n\varepsilon\sqrt{(2d)\log^3(2d)}}{c_2 T} \leq \frac{1}{2},$$

where we used the fact that $G_k \leq G_0$, $s \leq S \leq \log(2d)$, $T \geq n^2 + d\log^2 d$, and c_2 is large enough.

Using the parameters we pick, we have

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$$

Moreover, assuming $G_k \leq G_0 k^{-c}$ for some $c > 1/2$, we have $\sqrt{\sum_s s^2 2^s G_{2^{s-1}k}^2} \lesssim G_0/k^c$. Hence,

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + \frac{G_0 D}{k^c}.$$

Since the above bound holds for all $k \in \{1, \ldots, d\}$, we may optimize it with respect to k. Recall by assumption that $n \geq \varepsilon^{-1}\sqrt{\log(1/\delta)}$. Letting

$$k = \min\left\{d, \left\lceil \left(\frac{\varepsilon n}{\sqrt{\log(1/\delta)}}\right)^{\frac{2}{1+2c}} \right\rceil\right\}$$

yields the bound

$$\mathbb{E}[F(\overline{x}; \mathcal{D}) - F(x^*; \mathcal{D})] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

□

Combining the privacy guarantee in Lemma 8.9.2 and Lemma 11.3.10 directly results in Theorem 11.3.3.

11.3.5 Proof of Theorem 11.3.5

We study the generalization error of DP-SGD and make use of its stability. The bound on the excess population loss follows from combining bounds on the excess empirical loss and the generalization error. Before stating the proof, we first recall two results in the literature.

Lemma 11.3.11 ([BE02, Lemma 7]). *Given a learning algorithm $\mathcal{A}$, a dataset $\mathcal{D} = \{s_1, \cdots, s_n\}$ formed by n i.i.d. samples drawn from the underlying distribution $\mathcal{P}$, and we replace one random sample in $\mathcal{D}$ with a freshly sampled $s' \sim \mathcal{P}$ to obtain a new neighboring dataset $\mathcal{D}'$. One has*

$$\mathop{\mathbb{E}}_{\mathcal{D},\mathcal{A}}[F(\mathcal{A}(\mathcal{D}); \mathcal{P}) - F(\mathcal{A}(\mathcal{D}); \mathcal{D})] = \mathop{\mathbb{E}}_{\mathcal{D},s'\sim\mathcal{P},\mathcal{A}}[f(\mathcal{A}(\mathcal{D}); s') - f(\mathcal{A}(\mathcal{D}'); s')],$$

where $\mathcal{A}(\mathcal{D})$ is the output of $\mathcal{A}$ with input $\mathcal{D}$.

Lemma 11.3.12 ([BFGT20, Theorem 3.3]). *Suppose Assumption 11.3.2 holds, running DP-SGD with step size η on any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ for T steps yields the following bound*

$$\mathbb{E}\left[\|\overline{x} - \overline{x}'\|_2\right] \leq 4G_0\eta\left(\frac{T}{n} + \sqrt{T}\right),$$

where $\overline{x}$ and $\overline{x}'$ are the outputs of DP-SGD with datasets $\mathcal{D}$ and $\mathcal{D}'$, respectively.

Proof of Theorem 11.3.5. Let $\overline{x}$ and $\overline{x}'$ be the outputs of DP-SGD when applied to the datasets $\mathcal{D}$ and $\mathcal{D}'$, respectively. $\mathcal{D}'$ is a neighbor of $\mathcal{D}$ with one example replaced by $s' \sim \mathcal{P}$ that is independently sampled. Combining Lemma 11.3.11 and Lemma 11.3.12 yields

$$\begin{aligned}
\mathbb{E}[F(\overline{x}; \mathcal{P}) - F(\overline{x}; \mathcal{D})] =& \ \mathbb{E}[f(\overline{x}; s') - f(\overline{x}'; s')] \\
\leq& \ \mathbb{E}[G_0\|\overline{x} - \overline{x}'\|_2] \\
\leq& \ 4G_0^2\eta\left(\frac{T}{n} + \sqrt{T}\right).
\end{aligned}$$

Similar to the DP-ERM case, by setting $T = c_1(n^2 + d\log^2 d)$, $\sigma = \frac{c_2\sqrt{T\log(1/\delta)}}{n\varepsilon}$, $\eta = \sqrt{\frac{D^2}{T\cdot G_0^2(T/n + k\sigma^2)}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$ for some large positive constants c_1 and c_2,

we conclude that $\eta \cdot \alpha \leq 1/2$. Hence, Equation (11.11) shows that, for any fixed dataset $\mathcal{D}$ and any x^* such that $\left\|x^{(0)} - x^*\right\|_2 \leq D$, we have

$$\mathbb{E}[F(\overline{x}; \mathcal{D}) - F(x^*; \mathcal{D})] \leq \frac{D^2}{2\eta T} + 3\eta G_0^2(1 + k\sigma^2) + \frac{\alpha}{2}D^2 + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2.$$

We can rewrite the population loss as follows

$$\begin{aligned}\mathbb{E}[F(\overline{x}; \mathcal{P}) - F(x^*; \mathcal{P})] = \; & \mathbb{E}[F(\overline{x}; \mathcal{P}) - F(\overline{x}; \mathcal{D})] + \mathop{\mathbb{E}}_{\mathcal{D}}[F(\overline{x}; \mathcal{D}) - F(x^*; \mathcal{D})] \\ & \leq 4G_0^2\eta\left(\frac{T}{n} + \sqrt{T}\right) + \frac{D^2}{2\eta T} + 3\eta G_0^2(1 + k\sigma^2) + \frac{\alpha}{2}D^2 + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2.\end{aligned}$$

Substituting in the values for parameters T, σ, η, and α yields

$$\mathbb{E}[F(\overline{x}; \mathcal{P}) - F(x^*; \mathcal{P})] \lesssim \frac{G_0 D}{\sqrt{n}} + \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$$

for all $k \in [d]$.

Similarly, if we have $G_k \leq G_0 k^{-c}$ for some $c > 1/2$, and in addition $n > \varepsilon^{-1}\log(1/\delta)$, it immediately follows that

$$\mathbb{E}[F(\overline{x}; \mathcal{P}) - \min_x F(x; \mathcal{P})] \lesssim \frac{G_0 D}{\sqrt{n}} + G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

This completes the proof. □

11.4 Numerical Experiments

The aim of this section is twofold. In Section 11.4.1, we study a synthetic example that matches our theoretical assumptions and show that DP-SGD attains dimension-independent empirical and population loss when the sequence of restricted Lipschitz coefficients decays rapidly—even when gradients span the entire ambient space. In Section 11.4.2, we study a stylized example of privately fine-tuning large language models. Building on the previous theory, we provide insights as to why dense fine-tuning can yield good performance.

11.4.1 Synthetic Example: Estimating the Generalized Geometric Median

We privately estimate the geometric median which minimizes the average Mahalanobis distance. Specifically, let $x_i \in \mathbb{R}^d$ for $i \in [n]$ be feature vectors drawn i.i.d. from some distribution P_x, each of which is treated as an individual record. Denote the entire dataset as $\mathcal{D} = \{x_i\}_{i=1}^n$. Subject to differential privacy, we perform the following optimization

$$\min_{x \in \mathbb{R}^d} F_\alpha(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) + \frac{\alpha}{2} \left\| x - x^{(0)} \right\|_2^2 = \frac{1}{n} \sum_{i=1}^n \|x - x_i\|_A + \frac{\alpha}{2} \left\| x - x^{(0)} \right\|_2^2, \quad (11.12)$$

where we adopt the shorthand $f_i(x) = f(x; x_i) = \|x - x_i\|_A$. When $A = I_d$ and $\alpha = 0$ (without the regularization term), the problem reduces to estimating the usual geometric median (commonly known as center of mass).

For this example, individual gradients are bounded since $\|\nabla f_i(x)\|_2 = \|A(x - x_i)/\|x - x_i\|_A\|_2 \leq \lambda_1(A^{1/2}) = G_0$. More generally, the restricted Lipschitz coefficients of $F(x)$ are the eigenvalues of $A^{1/2}$, since

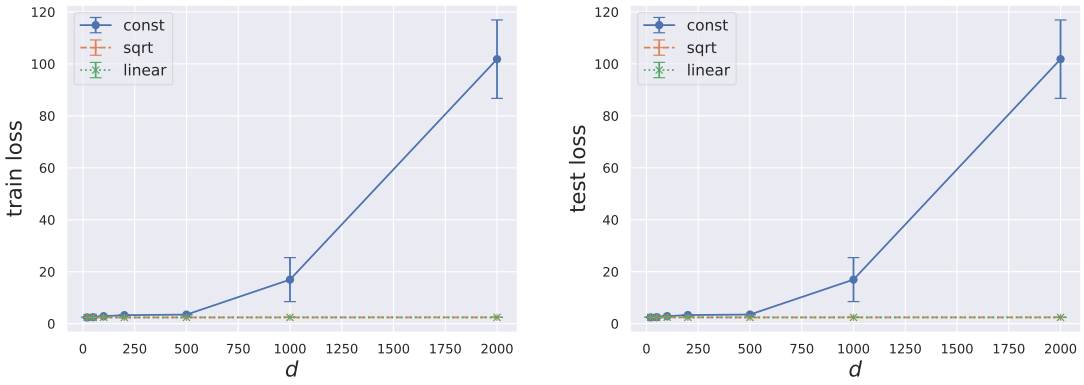
$$\|Q_k \nabla F(x)\|_2 = \left\| Q_k A^{1/2} \frac{1}{n} \sum_{i=1}^n \frac{A^{1/2}(x - x_i)}{\|x - x_i\|_A} \right\|_2 \leq \|Q_k A^{1/2}\|_{\text{op}} = \lambda_{k+1}(A^{1/2}) = G_k,$$

where $Q_k = I - P_k$ is chosen to be the rank $(d - k)$ orthogonal projection matrix that projects onto the subspace spanned by the bottom $(d - k)$ eigenvectors of $A^{1/2}$.

To verify our theory, we study the optimization and generalization performance of DP-SGD for minimizing (11.12) under Mahalanobis distances induced by different A as the problem dimension grows. The optimization performance is measured by the final training error, and the generalization performance is measured by the population quantity $\mathbb{E}_{x \sim P_x, \overline{x}}[\|\overline{x} - x\|_A]$, where $\overline{x}$ denotes the random output of DP-SGD. We study the dimension scaling behavior for A being one of

$$A_{\text{const}} = \text{diag}(1, \ldots, 1), \quad A_{\text{sqrt}} = \text{diag}(1, 1/\sqrt{2}, \ldots, 1/\sqrt{d}), \quad A_{\text{linear}} = \text{diag}(1, 1/2, \ldots, 1/d),$$

where $\text{diag} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ maps vectors onto square matrices with inputs on the diagonal. In all cases, the span of gradients $\text{span}(\{\nabla F(x)\})$ is the ambient space $\mathbb{R}^d$, since A is of



(a) empirical loss

(b) (estimated) population loss

Figure 11.1: The empirical and population losses grow with increasing problem dimension when the sequence of restricted Lipschitz coefficients remain constant. On the other hand, these losses remain almost constant when the sequence of restricted Lipschitz coefficients decays rapidly. Error bars represent one standard deviation over five runs of DP-SGD with the same hyperparameters which were tuned on separate validation data. For the same A, the optimal training error $\min_{x \in \mathbb{R}^d} F(x)$ is the same for problem instances with different dimensions (thus errors do not scale if learning was non-private). Each training run was performed with $\varepsilon = 2$, $\delta = 10^{-6}$, and $n = 10000$.

full rank. To ensure the distance from the initial iterate $\beta^{(0)} = 0$ to the optimum is the same for problem instances of different dimensions, we let feature vectors $\{x_i\}_{i=1}^n$ take zero values in any dimension $k > d_{\text{min}}$, where d_{min} is the dimension of the smallest problem in our experiments. Our theoretical bounds suggest that when the sequence of restricted Lipschitz coefficients is constant (when $A = A_{\text{const}}$), the excess empirical loss grows with the problem dimension, whereas when the sequence of kth-Lipschitz constants rapidly decays with k (when $A = A_{\text{sqrt}}$ or $A = A_{\text{linear}}$), the excess empirical loss does not grow beyond a certain problem dimension. Figure 11.1 empirically captures this phenomenon.

11.4.2 Why Does Dense Fine-Tuning Work Well for Pretrained Language Models?

Stated informally, our bounds in Theorem 11.3.5 imply that DP-SGD obtains dimension-independent errors if gradients approximately reside in a subspace much smaller than the ambient space. Inspired by these results for the convex case, we now turn to study dense

language model fine-tuning [LTLH21] and provide a possible explanation for their recent intriguing success — fine-tuning gigantic parameter vectors frequently results in moderate performance drops compared to non-private learning.

In the following, we present evidence that gradients obtained through fine-tuning mostly lie in a small subspace. We design subsequent experiments to work under a simplified setup. Specifically, we fine-tune DistilRoBERTa [SDCW19, LOG+19] under $\varepsilon = 8$ and $\delta = 1/n^{1.1}$ for sentiment classification on the SST-2 dataset [SPW+13]. We reformulate the label prediction problem as templated text prediction [LTLH21], and fine-tune only the query and value matrices in attention layers.

We focus on fine-tuning these specific parameter matrices due to the success of LoRA for non-private learning [HSW+21] which focuses on adapting the attention layers. Unlike LoRA, we fine-tune all parameters in these matrices rather than focusing on low-rank updates. This gives a setup that is lightweight enough to run spectral analyses computationally tractably but retains enough parameters (≈ 7 million) such that a problem of similar scale outside of fine-tuning results in substantial losses in utility.[2] For our setup, DP-SGD obtains a dev set accuracy approximately of 90% and 92%, privately and non-privately, respectively. These numbers are similar to previous results obtained with the same pretrained model [YNB+21, LTLH21].

To provide evidence for the small subspace hypothesis, we sample gradients during fine-tuning and study their principal components. Specifically, we "over-train" by privately fine-tuning for $r = 2 \times 10^3$ updates and collect all the non-privatized average clipped gradients along the optimization trajectory. While fine-tuning for 200 and 2k updates have similar final dev set performance under our hyperparameters, the increased number of steps allows us to collect more gradients around the converged solution. This yields a gradient matrix $H \in \mathbb{R}^{r \times p}$, where $p \approx 7 \times 10^6$ is the size of the parameter vector. We perform PCA for H with the orthogonal iteration algorithm [Dem97] and visualize the set of estimated singular values $\sigma_i(H) = \lambda_i(H^\top H)^{1/2}$ in terms of both (i) the density estimate, and (ii) their

[2] For instance, an off-the-shelf ResNet image classifier has 10 to 20+ million parameters. A plethora of works report large performance drops when training these models from scratch [YZCL21, LWAFF21, DBH+22].

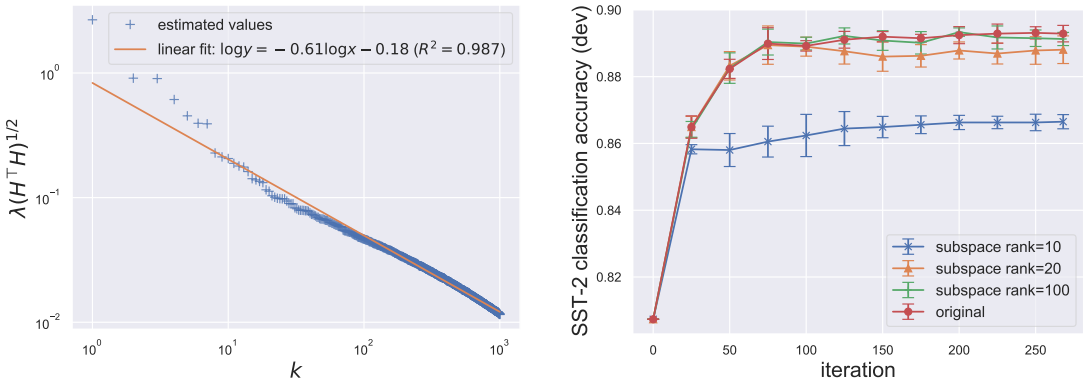
relation with the rank. Figure 11.2 (a) shows the top 1000 singular values sorted and plotted against their rank k and the least squares fit on log-transformed inputs and outputs. The plot displays few large singular values which suggests that gradients are controlled through only a few principal directions. The linear fit suggests that singular values decay rapidly (at a rate of approximately $k^{-0.6}$).

To study the effects that different principal components have on fine-tuning performance, we further perform the following re-training experiment. Given the principal components, we privately re-fine-tune with gradients projected onto the top $k \in \{10, 20, 100\}$ components. Note that this projection applies only to the (non-privatized) average clipped gradients and the isotropic DP noise is still applied to all dimensions. Figure 11.2 (b) shows that the original performance can be attained by optimizing within a subspace of only dimension $k = 100$, suggesting that most of the dimensions of the 7 million parameter vector encode a limited learning signal.

While these empirical results present encouraging insights for the dimension-independent performance of fine-tuning, we acknowledge that this is not a complete validation of the restricted Lipschitz continuity condition and fast decay of coefficients (even locally near the optimum). We leave a more thorough analysis with additional model classes and fine-tuning tasks to future work.

11.5 Related Work

DP-ERM and DP-SCO are arguably the most well-studied areas of differential privacy [CMS11, KST12, BST14, SCS13, WYX17, FTS17, BFTGT19, MRTZ17, ZZMW17, WLK+17, FKT20, INS+19, BFGT20, STT20, LL21, AFKT21, BGN21, GTU22, GLL22]. Tight dependence on the number of model parameters and the number of samples is known for both DP-ERM [BST14] and DP-SCO [BFTGT19]. In particular, for the error on general convex losses, an explicit polynomial dependence on the number of optimization variables is necessary. However, it is shown that if gradients lie in a fixed low-rank subspace M, the dependence on dimension d can be replaced by $\mathsf{rank}(M)$ which can be significantly smaller [JT14, STT20]. We extend this line of work to show that under a weaker assumption (restricted Lipschitz continuity with decaying coefficients) one can obtain analogous



(a) singular values decay with rank

(b) retrain in fixed subspace

Figure 11.2: Gradients obtained through fine-tuning are controlled by a few principal components. *Left:* Singular values decay rapidly with their rank. *Right:* Retraining with gradients projected onto a subspace (but noise is not projected!) is sufficient to recover original performance.

error guarantees that are independent of d, but do not require the gradients of the loss to strictly lie in any fixed low-rank subspace M. As a consequence, our results provide a plausible explanation for the empirical observation that dense fine-tuning can be effective and that fine-tuning a larger model under DP can generally be more advantageous in terms of utility than fine-tuning a smaller model [LTLH21, YNB+21]. A concurrent work shows that the standard dimension dependence of DP-SGD can be replaced by a dependence on the trace of the Hessian assuming the latter quantity is uniformly bounded [MMZ22].

A complementary line of work designed variants of DP-SGD that either explicitly or implicitly control the subspace in which gradients are allowed to reside [AGM+21, LVS+21, AFKT21, KDRT21, YZCL21]. They demonstrated improved dependence of the error on the dimension if the true gradients lie in a "near" low-rank subspace. Our results are incomparable to this line of work because of two reasons: (i) Our algorithm is vanilla DP-SGD and does not track the gradient subspace either explicitly or implicitly, and hence does not change the optimization landscape. Our improved dependence on dimensions is an artifact of the analysis. (ii) Our analytical results do not need the existence of any public data to obtain tighter dependence on dimensions. All prior works mentioned above need

the existence of public data to demonstrate any improvement.

On the empirical front, past works have observed that for image classification tasks, gradients of ConvNets converge to a small subspace spanned by the top directions of the Hessian. In addition, this span remains stable for long periods of time during training [GARD18]. While insightful, this line of work does not look at language model fine-tuning. Another line of work measures for language model fine-tuning the *intrinsic dimension*—the minimum dimension such that optimizing in a randomly sampled subspace of such dimension approximately recovers the original performance [LFLY18, AZG20]. We note that a small intrinsic dimension likely suggests that gradients are approximately low rank. Yet, this statement should not be interpreted as a strict implication, since the notion of intrinsic dimension is at best vaguely defined (e.g., there's no explicit failure probability threshold over the randomly sampled subspace in the original statement), and the definition involves not a fixed subspace but rather a randomly sampled one.

11.6 Conclusion

We made an attempt to reconcile two seemingly conflicting results: (i) in private convex optimization, errors are predicted to scale proportionally with the dimension of the learning problem; while (ii) in empirical works on large-scale private fine-tuning through DP-SGD, privacy-utility trade-offs become better with increasing model size. We introduced the notion of restricted Lipschitz continuity, with which we gave refined analyses of DP-SGD for DP-ERM and DP-SCO. When the magnitudes of gradients projected onto diminishing subspaces decay rapidly, our analysis showed that excess empirical and population losses of DP-SGD are independent of the model dimension. Through preliminary experiments, we gave empirical evidence that gradients of large pretrained language models obtained through fine-tuning mostly lie in the subspace spanned by a few principal components. Our theoretical and empirical results together give a possible explanation for recent successes in large-scale differentially private fine-tuning.

Given our improved upper bounds on the excess empirical and population risks for differentially private convex learning, it is instructive to ask if such bounds are tight in the mini-max sense. We leave answering this inquiry to future work. In addition, while

we have presented encouraging empirical evidence that fine-tuning gradients mostly lie in a small subspace, more work is required to study the robustness of this phenomenon with respect to the model class and fine-tuning problem. Overall, we hope that our work leads to more research on understanding conditions under which DP learning does not degrade with increasing problem size, and more generally, how theory can inform and explain the practical successes of differentially private deep learning.

Chapter 12

PRIVATE STOCHASTIC CONVEX OPTIMIZATION WITH HEAVY TAILS: NEAR-OPTIMALITY FROM SIMPLE REDUCTIONS

12.1 Introduction

Differentially private stochastic convex optimization (DP-SCO), where an algorithm aims to minimize a population loss given samples from a distribution, is a fundamental problem in statistics and machine learning. In this problem, given n samples from a distribution $\mathcal{P}$ over a sample space $\mathcal{S}$, our goal is to privately find an approximate minimizer $\widehat{x} \in \mathcal{X} \subset \mathbb{R}^d$ for the population loss

$$F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s \sim \mathcal{P}} [f(x; s)] ,$$

where $f(\cdot; s)$ is a convex function for all $s \in \mathcal{S}$. The quality of an algorithm is measured by the excess population loss of its output $\widehat{x}$, that is $F_{\mathcal{P}}(\widehat{x}) - \min_{x\mathrm{OPT} \in \mathcal{X}} F_{\mathcal{P}}(x\mathrm{OPT})$.

Extensive research efforts have been devoted to DP-SCO, resulting in important progress over the past few years [BFTGT19, FKT20, AFKT21, BGN21, ALD21, KLL21]. In an important milestone, [BFTGT19] developed optimal algorithms (in terms of the excess population loss) for DP-SCO under a uniform Lipschitz assumption (i.e., where every $f(\cdot; s)$ is assumed to have the same Lipschitz bound), and [FKT20] followed this result with efficient and optimal algorithms that run in linear time for smooth functions. DP-SCO has also been explored in other notable settings, including developing faster algorithms for non-smooth settings [AFKT21, KLL21, CJJ+23], different geometries imposed on the solution space [AFKT21, BGN21, GLL+23], and different notions of privacy [ALD21].

Most existing results in DP-SCO are based on the assumption that the function $f(\cdot; s)$ is uniformly G-Lipschitz for all $s \in \mathcal{S}$. This assumption is convenient for private algorithm design, because it allows us to straightforwardly bound the *sensitivity* of iterates of private algorithms, i.e., how far a pair of iterates defined via algorithms induced by neighboring datasets drift apart. Under the uniform Lipschitz assumption, the DP-SCO problem is

relatively well-understood, as optimal and efficient algorithms exist (sometimes requiring additional regularity assumptions) [BFTGT19, FKT20].[1] State-of-the-art SCO algorithms satisfying (ε, δ)-differential privacy (Definition 12.2.1) in the uniform Lipschitz setting result in excess population loss

$$GD\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d\log(\frac{1}{\delta})}}{\varepsilon n}\right), \tag{12.1}$$

where D is the diameter of $\mathcal{X}$. However, the assumption of uniformly G-Lipschitz gradients is strong, and may be violated in real-life applications where the distribution in question has heavy tails (see e.g. discussion in [ACG+16]). As a simple motivating example, consider mean estimation, where each $f(\cdot; s) = \frac{1}{2}\|\cdot - s\|^2$, so the minimizer of $F_{\mathcal{P}}$ is the population mean. The uniform Lipschitz requirement amounts to $\mathcal{P}$ having a bounded support over $\mathcal{X}$, whereas an algorithm that can handle heavy tails only posits the weaker assumption that $\mathcal{P}$ has bounded k-th moments. However, as pointed out by [WXDX20], many real-world datasets [MM97, BDFS07, IIW15], especially those from biomedicine and finance, are usually unbounded or even heavy-tailed. As a result, existing algorithms for DP-SCO may have overly pessimistic performance bounds when G is large or even unbounded, necessitating the search for new private algorithms handling heavy-tailed gradients.

Motivated by this weakness of existing DP-SCO analyses, several papers studied the problem of DP-SCO with heavy-tailed gradients [WXDX20, ADF+21, KLZ22, LR23], formally defined in Definition 12.2.6. Rather than assuming uniformly Lipschitz gradients, this line of work builds on the more realistic assumption that the norm of the gradients has bounded k^{th}-moments. In particular, [ADF+21] studied heavy-tailed private optimization for the related empirical loss, while [WXDX20] initiated an analogous study for the population loss. More recently, [KLZ22, LR23] also proposed algorithms to solve the heavy-tailed DP-SCO problem based on clipped stochastic gradient methods.

Despite the significant progress made in addressing heavy-tailed DP-SCO, it remains notably less understood compared to the uniformly Lipschitz setting. As a benchmark, under a notion called ρ-concentrated differential privacy (CDP, see Definition 12.2.3), which trans-

[1] One notable exception is the lack of linear-time algorithms in the non-smooth setting.

lates to (ε, δ)-DP for $\rho \approx \varepsilon^2 \log^{-1}(\frac{1}{\delta})$, [LR23] established that the best excess population loss achievable scales as

$$G_2 D \cdot \frac{1}{\sqrt{n}} + G_k D \cdot \left(\frac{\sqrt{d}}{n\sqrt{\rho}} \right)^{1-\frac{1}{k}}, \tag{12.2}$$

where G_j^j is the j^{th} moment bound on the Lipschitz constant of sampled functions, see Definition 12.2.6. Note that as $k \to \infty$, the rate in (12.2) recovers the uniform Lipschitz rate in (12.1).

Unfortunately, existing works on heavy-tailed DP-SCO assume stringent conditions on problem parameters and are suboptimal in the general case. For example, [KLZ22] requires the loss functions to be uniformly smooth with various parameter bounds in order to guarantee optimal rates, while the recent work [LR23] obtains a suboptimal rate scaling as[2] $G_2 D \cdot \frac{1}{\sqrt{n}} + G_k D \cdot (\frac{\sqrt{d}}{n\sqrt{\rho}})^{1-\frac{2}{k}}$, which is worse than (12.2) by polynomial factors in the dimension for any constant k.

12.1.1 Our contributions

Motivated by the suboptimality of existing results for heavy-tailed DP-SCO, we develop the first algorithm for this problem, which achieves the optimal rate (12.2) up to logarithmic factors with no additional assumptions. Along the way, we give several simple reduction-based tools for overcoming technical barriers encountered by prior works. To state our results (deferring a formal problem statement to Definition 12.2.1), we assume that for some $k \geq 2$ and all $j \in [k]$, we have

$$\mathop{\mathbb{E}}_{s \sim \mathcal{P}} \left[\max_{x \in \mathcal{X}} \|\nabla f(x; s)\|^j \right] \leq G_j^j.$$

Our results hold in several settings and are based on different reductions which allow us to apply strategies for DP-SCO from the uniform Lipschitz setting.

---

[2]The rate in [LR23] is stated slightly differently (see their Theorem 6), as they parameterize their error bound via G_{2k} despite assuming only k bounded moments. However, under the assumption that G_{2k} is finite (so the [LR23] result is usable), the optimal rate scales as in (12.2) where k is replaced with $2k$, leaving a polynomial gap.

Near-optimal rates for heavy-tailed DP-SCO (Section 12.3). We design an algorithm for the k-heavy-tailed DP-SCO problem, which satisfies ρ-CDP[3] and attains near-optimal excess loss

$$G_2 D \cdot \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{n}} + G_k D \cdot \left(\frac{\sqrt{d}\log\left(\frac{1}{\delta}\right)}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}. \tag{12.3}$$

This matches the lower bounds recently proved by [KLZ22, LR23] for ρ-concentrated DP algorithms up to logarithmic factors, stated in (12.2). Standard conversions from CDP to (ε, δ)-DP imply that our algorithm also obtains loss $\approx G_2 D \cdot \sqrt{\frac{1}{n}} + G_k D \cdot (\frac{\sqrt{d\log^3(1/\delta)}}{n\varepsilon})^{1-\frac{1}{k}}$ under this parameterization. We note that our bound (12.3) holds with high probability $\geq 1-\delta$, whereas the lower bound (12.2) is for an error which holds only in expectation (see Theorem 13, [LR23]). Our lossiness in (12.3) is due to a natural sample-splitting strategy used to boost our failure probability, and we conjecture that (12.3) may be optimal in the high-probability error bound regime.

As in [LR23], to establish our result we begin by deriving utility guarantees for a clipped stochastic gradient descent subroutine on an empirical loss, where clipping ensures privacy but induces bias, parameterized by a dataset-dependent quantity $b_{\mathcal{D}}^2$ defined in (12.25). We give a standard analysis of this subroutine in Proposition 12.3.1, a variant of which (with slightly different parameterizations) also appeared as Lemma 27, [LR23]. However, the key technical barrier encountered by the [LR23] analysis, when converting to population risk, was bounding $\mathbb{E}\, b_{\mathcal{D}}^2$ over the sampled dataset, which naively depends on the $2k^{\text{th}}$ moment of gradients. This either incurs an overhead depending on G_{2k}, or in the absence of such a bound (which is not given under the problem statement), leads to the aforementioned suboptimal rate in [LR23] losing a factor of $(\frac{\sqrt{d}}{n\sqrt{\rho}})^{\frac{1}{k}}$ in the utility. We give a further discussion of natural strategies and barriers towards directly bounding $\mathbb{E}\, b_{\mathcal{D}}^2$ in Appendix 12.11.

Where we depart from the strategy of [LR23] is in the use of a new *population-level localization* framework we design (see Algorithm 37), inspired by similar localization techniques

[3] We state the privacy guarantee of most of our results, save our algorithm in Appendix 12.7 which employs the sparse vector technique of [DNR+09, DR14], in terms of CDP, for simpler comparison to the lower bound (12.2).

in prior work [FKT20] (discussed in more detail in Section 12.1.2). This strategy allows us to use constant-success probability bounds on the quantity $b_{\mathcal{D}}$ (which also bound $b_{\mathcal{D}}^2$), which are easy to achieve depending only on G_k rather than G_{2k} via Markov's inequality. This bypasses the need in [LR23] for bounding $\mathbb{E}\, b_{\mathcal{D}}^2$. The motivation for population-level localization is that we wish to aggregate empirical solutions to multiple datasets, some of which have small $b_{\mathcal{D}}$, and others which do not. However, each dataset has a different empirical minimizer, so it is unclear how to argue about convergence if we apply the empirical localization. Instead, we aggregate solutions close to the population-level minimizer and share them across datasets via a simple geometric aggregation technique, showing that it suffices for a constant fraction of datasets to have this desirable property for us to carry out our population-level localization argument. We formally state our main result achieving the rate (12.3) as Theorem 12.3.8.

Interestingly, as a straightforward corollary of our new localization framework, we achieve a tight rate for high-probability stochastic convex optimization under a bounded-variance gradient estimator parameterization, perhaps the most well-studied formulation of SCO. To our knowledge, this result was only first achieved very recently by [CH24].[4] However, we find it a promising proof-of-concept that our new framework directly yields the same result. For completeness, we include a derivation in Appendix 12.9 (see Theorem 12.9.4) as a demonstration of the utility of our framework.

Optimal rates with known Lipschitz constants (Section 12.6). We next consider the *known Lipschitz* setting, where each sample function $f(\cdot; s)$ arrives with a value $\overline{L}_s$ which is an overestimate of its Lipschitz constant, such that $\mathbb{E}\, \overline{L}_s^j$ is bounded for all $j \in [k]$ (see Assumption 12.6.1). As motivation, consider the problem of learning a generalized linear model (GLM), where $f(\cdot; s) = \sigma(\langle \cdot, s \rangle)$ for a known convex activation function σ. Typically, the Lipschitz constant for $f(\cdot; s)$ is simply the Lipschitz constant of σ times $\|s\|$, which

[4] We mention that an alternative route to obtaining a near-optimal high-probability rate was given slightly earlier in [SZ23], but lost a polylogarithmic factor in the failure probability. We also wish to acknowledge that in an independent and concurrent work [JST24] involving the third author, the authors slightly sharpened and generalized the result of [SZ23], which inspired us to consider this application of our population-level localization framework.

can be straightforwardly calculated. Thus, for GLMs, our known Lipschitz heavy-tailed assumption amounts to moment bounds on the distribution $\mathcal{P}$.

Our second result, Theorem 12.6.7, shows a natural strategy obtains optimal rates in this known Lipschitz setting, eliminating logarithmic factors from Theorem 12.3.8. As mentioned previously, this result holds for the important family of GLMs. Our algorithm is based on a straightforward reduction to the uniformly Lipschitz setting: after simply iterating over the input samples, and replacing samples whose Lipschitz constant exceeds a given threshold with a new dummy sample, we show existing Lipschitz DP-SCO algorithms then obtain the optimal heavy-tailed excess population loss (12.2). Despite the simplicity of this result, to the best of our knowledge, it was not previously known.

Efficient algorithms for smooth functions (Appendices 12.7 and 12.8). Finally, we propose algorithms with improved query efficiency for general smooth functions or smooth GLMs, with moderate smoothness bounds. Our strategy is to analyze the stability of clipped-DP-SGD in the smooth heavy-tailed setting, and use localization-based reductions to transform a stable algorithm into a private one [FKT20]. This results in linear-time algorithms for the smooth case with near-optimal rates. In order to prove the privacy of our smooth, heavy-tailed algorithm, we analyze a careful interplay of our clipped stochastic gradient method with the sparse vector technique (SVT) [DNR+09, DR14]. At a high level, our use of SVT comes from the fact that under clipping, smooth gradient steps no longer enjoy the type of contraction guarantees applicable in the uniform Lipschitz setting (see Fact 12.7.1), so we must take care to not clip too often. The SVT is then used to ensure privacy of our count of how many clipping operations were used. In Appendix 12.10, we provide a simple counterexample showing that the noncontractiveness of contractive steps after applying clipping is inherent. Our general smooth heavy-tailed DP-SCO result is stated as Theorem 12.7.6.

We believe the use of SVT within an optimization algorithm to ensure privacy may be of independent interest, as it is one of few such instances that have appeared in the private optimization literature to our knowledge; it is inspired by a simpler application of this technique carried out in [AL24].

On the other hand, we make the simple observation that for GLMs, clipping cannot make a contractive gradient step noncontractive, by taking advantage of the fact that the derivative of $f(x; s) = \sigma(\langle x, s\rangle)$ is a multiple of s for any $x \in \mathcal{X}$ (see Lemma 12.8.1). We use this observation to give a straightforward adaptation of the smooth algorithm in [FKT20] to the heavy-tailed setting, proving Theorem 12.8.2, which attains both a linear gradient query complexity and the optimal rate (12.2).

12.1.2 Prior work

The best-known rates for heavy-tailed DP-SCO were recently achieved by [KLZ22, LR23]. As discussed previously, their results do not provide the same optimality guarantees as our Theorem 12.3.8. The rate achieved by [LR23] is polynomially worse than the optimal loss (12.2) for any constant k. On the other hand, the work of [KLZ22] uses a different assumption on the gradients than Assumption 12.2.5, which is arguably more nonstandard: in particular, they require that the k^{th}-order central moments of each coordinate $\nabla_j f(x; s)$ is bounded. Moreover, their algorithms require each sample function $f(\cdot; s)$ to be β-smooth, and the final rates have a strong dependence on the condition number $\kappa = \frac{\beta}{\lambda}$ where λ is the strong convexity parameter (see Appendix C in [LR23] for additional discussion).

Our result in the heavy-tailed setting assuming β-smoothness of sample functions, Theorem 12.7.6, is most directly related to Theorem 15 of [LR23]. These two results respectively require

$$\beta = O\left(\frac{G_k}{D} \cdot \varepsilon^{1.5}\sqrt{\frac{n}{d}}\right) \text{ and } \beta = O\left(\frac{G_k}{D} \cdot \left(\frac{d^5}{\varepsilon n}\right)^{\frac{1}{18}}\right),$$

omitting logarithmic factors in our bound for simplicity, to obtain near-optimal rates. These regimes are different and not generally comparable. However, we find it potentially useful that our upper bound on β grows as more samples are taken, whereas the [LR23] bound degrades with larger n. It is worth mentioning that [LR23]'s Theorem 15 shaves roughly one logarithmic factor in the error bound from our Theorem 12.7.6. On the other hand, Theorem 12.7.6 actually requires a looser condition than mentioned above (see (12.19)), which can improve its guarantees in a wider range of parameters.

Finally, we briefly contextualize our population-level localization framework in regard

to previous localization schemes proposed by [FKT20]. The two localization schemes in [FKT20] (see Sections 4.1 and 5.1 of that work) both follow the same strategy of gradually improving distance bounds to a minimizer in phases. However, their implementation is qualitatively different than our Algorithm 37, preventing their direct application in our algorithm. For instance, Section 4.1 of [FKT20] does not use strong convexity and, therefore cannot take advantage of generalization bounds afforded to strongly convex losses (see discussion in [SSSSS09]). On the other hand, the scheme in Section 5.1 of [FKT20] serves a different purpose than Algorithm 37, aiming to solve strongly convex optimization by reducing it to non-strongly convex optimization; our Algorithm 37, on the other hand, directly targets non-strongly convex optimization as its goal. We view our approach as complementary to these prior frameworks and are optimistic it will find further utility in applications.

12.2 Preliminaries

General notation. We use $[d]$ to denote the set $\{i \in \mathbb{N} \mid i \le d\}$. We use $\mathrm{sign}(x) \in \{\pm 1\}$ to denote the sign for $x \in \mathbb{R}$, with $\mathrm{sign}(0) = 1$. We use $\mathcal{N}(\mu, \mathbf{\Sigma})$ to denote the multivariate normal distribution of specified mean and covariance. We denote the all-ones and all-zeroes vectors of dimension d by $\mathbb{1}_d$ and $\mathbb{0}_d$. We use $\|\cdot\|$ to denote the Euclidean (ℓ_2) norm. We use $\mathbf{I}_d$ to denote the identity matrix on $\mathbb{R}^d$. We use $\mathbb{B}(C)$ to denote the ℓ_2 ball of radius C, and for $x \in \mathbb{R}^d$, $\mathbb{B}(x, C)$ is used to denote $\{x' \in \mathbb{R}^d \mid \|x' - x\| \le C\}$. For a set $\mathcal{X} \subseteq \mathbb{R}^d$, we let $\mathrm{diam}(\mathcal{X}) := \sup_{x,x' \in \mathcal{X}} \|x - x'\|$, and we let $\Pi_{\mathcal{X}}(x)$ denote the Euclidean projection of x to $\mathcal{X}$, i.e. $\mathrm{argmin}_{x' \in \mathcal{X}} \|x' - x\|$, which exists and is unique when $\mathcal{X}$ is compact. We use $f_{\mathcal{X}}$ to denote the restriction of a function f to $\mathcal{X}$, i.e.

$$f_{\mathcal{X}}(x) = \begin{cases} f(x) & x \in \mathcal{X} \\ \infty & x \notin \mathcal{X} \end{cases}. \tag{12.4}$$

For $x \in \mathbb{R}^d$, we use $\Pi_C(x)$ as shorthand for $\Pi_{\mathbb{B}(C)}(x)$, i.e. $\Pi_C(X)$ denotes the clipped vector $x \cdot \min(\frac{C}{\|x\|}, 1)$. We say two datasets $\mathcal{D}$, $\mathcal{D}'$ are *neighboring* if they differ in one entry, and $|\mathcal{D}| = |\mathcal{D}'|$. We say $x \in \mathcal{X}$ is an ε-approximate minimizer to $f : \mathcal{X} \to \mathbb{R}$ if

$f(x) - \inf_{x^\star \in \mathcal{X}} f(x^\star) \le \varepsilon$. For two densities μ, ν on the same probability space, and $\alpha > 1$, we define the α-Rényi divergence

$$D_\alpha(\mu \| \nu) := \frac{1}{\alpha - 1} \log \left(\int \left(\frac{\mu(\omega)}{\nu(\omega)} \right)^\alpha \mathrm{d}\nu(\omega) \right).$$

For an event $\mathcal{E}$ on a probability space clear from context, we let $\mathcal{I}_\mathcal{E}$ denote the 0-1 indicator of $\mathcal{E}$. We say $f : \mathcal{X} \to \mathbb{R}$ is L-Lipschitz if $|f(x) - f(x')| \le L \|x - x'\|$ for all $x, x' \in \mathcal{X}$; if f is differentiable and convex, an equivalent characterization is $\|\nabla f(x)\| \le L$ for all $x \in \mathcal{X}$. We say $f : \mathcal{X} \to \mathbb{R}$ is μ-strongly convex if $f(\lambda x' + (1-\lambda)x) \le \lambda f(x') + (1-\lambda) f(x) - \frac{\mu\lambda(1-\lambda)}{2} \|x - x'\|^2$ for all $x, x' \in \mathcal{X}$. We say differentiable $f : \mathcal{X} \to \mathbb{R}$ is β-smooth if for all $x, x' \in \mathcal{X}$, $\|\nabla f(x) - \nabla f(x')\| \le \beta \|x - x'\|$.

Differential privacy. We begin with a definition of standard differential privacy.

Definition 12.2.1 (Differential privacy)**.** Let $\varepsilon \ge 0$, $\delta \in [0, 1]$. We say a mechanism (randomized algorithm) $\mathcal{M} : \mathcal{S}^n \to \Omega$ satisfies (ε, δ)-differential privacy (alternatively, $\mathcal{M}$ is (ε, δ)-DP) if for any neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$, and any $S \subseteq \Omega$, $\Pr[\mathcal{M}(\mathcal{D}) \in S] \le \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta$.

More generally, for random variables $X, Y \in \Omega$ satisfying $\Pr[X \in S] \le \exp(\varepsilon) \Pr[Y \in S] + \delta$ for all $S \subseteq \Omega$, we say that X, Y are (ε, δ)-indistinguishable.

Throughout the paper, other notions of differential privacy will frequently be useful for our accounting of privacy loss in our algorithms. For example, we define the following variants of DP.

Definition 12.2.2 (Rényi DP)**.** Let $\alpha > 1$, $\varepsilon \ge 0$. We say a mechanism $\mathcal{M} : \mathcal{S}^n \to \Omega$ satisfies (α, ε)-Rényi differential privacy (RDP) if for any neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$, $D_\alpha(\mathcal{M}(\mathcal{D}) \| \mathcal{M}(\mathcal{D}')) \le \varepsilon$.

Definition 12.2.3 (CDP)**.** Let $\rho \ge 0$. We say a mechanism $\mathcal{M} : \mathcal{S}^n \to \Omega$ satisfies ρ-concentrated differential privacy (alternatively, $\mathcal{M}$ satisfies ρ-CDP) if for any neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$, and any $\alpha \ge 1$, $D_\alpha(\mathcal{M}(\mathcal{D}) \| \mathcal{M}(\mathcal{D}')) \le \alpha\rho$.

For an extended discussion of RDP and CDP and their properties, we refer the reader to [BS16, Mir17, BDRS18]. We summarize the main facts about these notions we use here.

Lemma 12.2.4 ([Mir17])**.** *RDP has the following properties.*

1. *(Composition): Let $\mathcal{M}_1 : \mathcal{S}^n \to \Omega$ satisfy (α, ε_1)-RDP and $\mathcal{M}_2 : \mathcal{S}^n \times \Omega \to \Omega'$ satisfy (α, ε_2)-RDP for any input in Ω. Then the composition of $\mathcal{M}_2$ and $\mathcal{M}_1$, i.e. the randomized algorithm which takes $\mathcal{D}$ to $\mathcal{M}_2(\mathcal{D}, \mathcal{M}_1(\mathcal{D}))$, satisfies $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.*

2. *(RDP to DP): If $\mathcal{M}$ satisfies (α, ε)-RDP, it satisfies $(\varepsilon + \frac{1}{\alpha - 1} \log \frac{1}{\delta}, \delta)$-DP for all $\delta \in (0, 1)$.*

3. *(Gaussian mechanism): Let $f : \mathcal{S}^n \to \mathbb{R}^d$ be an L-sensitive randomized function for $L \geq 0$, i.e. for any neighboring $\mathcal{D}$, $\mathcal{D}'$, we have $\|f(\mathcal{D}) - f(\mathcal{D}')\| \leq L$. Then for any $\sigma > 0$, the mechanism which outputs $f(\mathcal{D}) + \xi$ for $\xi \sim \mathcal{N}(\not{\vdash}_d, \sigma^2 \mathbf{I}_d)$ satisfies $\frac{L^2}{2\sigma^2}$-CDP.*

Private SCO. Throughout the paper, we study the problem of private stochastic convex optimization (SCO) with heavy-tailed gradients. We first define the assumptions used in our algorithms.

Assumption 12.2.5 (k-heavy-tailed distributions)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact, convex set. Let $\mathcal{P}$ be a distribution over a sample space $\mathcal{S}$, such that each $s \in \mathcal{S}$ induces a continuously-differentiable, convex, L_s-Lipschitz loss function $f(\cdot; s) : \mathcal{X} \to \mathbb{R}$,[5] where $L_s := \max_{x \in \mathcal{X}} \|\nabla f(x; s)\|$ is unknown. For $k \in \mathbb{N}$ satisfying $k \geq 2$, we say $\mathcal{P}$ satisfies the* k-heavy tailed assumption *if, for a sequence of monotonically nondecreasing $\{G_j\}_{j \in [k]}$,[6] we have $\mathbb{E}_{s \sim \mathcal{P}}[L_s^j] \leq G_j^j < \infty$ for all $j \in [k]$.*

In Appendix 12.6, we consider a variant of Assumption 12.2.5 where we have explicit access to upper bounds on the Lipschitz constants L_s, formalized in Assumption 12.6.1.

[5] The assumed moment bounds shows that $f(\cdot; s)$ has a finite Lipschitz constant, except for a probability-zero set of s. Moreover, convex functions are differentiable almost everywhere. Therefore, if $f(\cdot; s)$ is Lipschitz, perturbing its first argument by an infinitesimal Gaussian makes it differentiable there with probability 1, and negligibly affects the function value. We thus assume for simplicity that $f(\cdot; s)$ is differentiable everywhere.

[6] This assumption is without loss of generality by Jensen's inequality.

Our goal is to approximately optimize a population loss over sample functions satisfying Assumptions 12.2.5 or 12.6.1, formalized in the following.

Definition 12.2.6 (k-heavy-tailed private SCO)**.** In the *k-heavy-tailed private SCO* problem, $\mathcal{X} \subset \mathbb{R}^d$ is a compact, convex set with $\mathrm{diam}(\mathcal{X}) = D$. Further, $\mathcal{P}$ is a distribution over a sample space $\mathcal{S}$ satisfying Assumption 12.2.5. Our goal is to design an algorithm which provides an approximate minimizer in expectation to the population loss, $F_{\mathcal{P}}(x) := \mathbb{E}_{s\sim\mathcal{P}}\left[f(x;s)\right]$, subject to satisfying differential privacy. We say such an algorithm queries N *sample gradients* if it queries $\nabla f(x;s)$ for N different pairs $(x,s) \in \mathcal{X}\times\mathcal{S}$. If $\mathcal{P}$ further satisfies Assumption 12.6.1, we call the corresponding problem the *known Lipschitz k-heavy-tailed private SCO* problem.

We first observe the following consequence of Assumption 12.2.5, deferring a proof to Appendix 12.5.

Lemma 12.2.7. *Let $\mathcal{P}$ be a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5. Then $F_{\mathcal{P}}$ is G_1-Lipschitz.*

We require the following claim which bounds the bias of clipped heavy-tailed distributions.

Fact 12.2.8 ([BD14], Lemma 3)**.** *Let $k > 1$ and $X \in \mathbb{R}^d$ be a random vector with $\mathbb{E}[\|X\|^k] \le G^k$. Then,*

$$\mathbb{E}\left\|\Pi_C(X) - X\right\| \le \mathbb{E}[\|X\|\mathcal{I}_{\|X\|\ge C}] \le \frac{G^k}{(k-1)C^{k-1}}.$$

We also use the following standard claim on geometric aggregation.

Fact 12.2.9 ([KLL+23], Claim 1)**.** *Let $S := \{x_i\}_{i\in[k]} \subset \mathbb{R}^d$ have the property that for (unknown) $z \in \mathbb{R}^d$, $|\{i \in [k] \mid \|x_i - z\| \le R\}| \ge 0.51k$ for some $R \ge 0$. There is an algorithm* Aggregate *which runs in time $O(dk^2)$ and outputs $x \in S$ such that $\|x - z\| \le 3R$.*

Finally, given a dataset $\mathcal{D} \in \mathcal{S}^*$ of arbitrary size, and $\lambda \ge 0$, we use the following shorthand to denote the regularized empirical risk minimization (ERM) objective corresponding to the dataset:

$$F_{\mathcal{D},\lambda}(x) := \frac{1}{|\mathcal{D}|}\sum_{s\in\mathcal{D}} f(x;s) + \frac{\lambda}{2}\|x\|^2. \tag{12.5}$$

When $\lambda = 0$, we simply denote the function above by $F_{\mathcal{D}}(x) := \frac{1}{|\mathcal{D}|}\sum_{s\in\mathcal{D}} f(x;s)$.

12.3 Heavy-Tailed Private SCO

In this section, we obtain near-optimal algorithms for the problem in Definition 12.2.6 using a new *population-level localization* framework, combined with geometric aggregation for boosting weak subproblem solvers to succeed with high probability (Fact 12.2.9). Our algorithm's main ingredient, in Section 12.3.1, is a clipped DP-SGD subroutine for privately minimizing a regularized ERM subproblem, under a condition on a randomly sampled dataset holding with constant probability. Next, in Section 12.3.2 we show that our algorithm from Section 12.3.1 returns points near the minimizer of a regularized loss function over the population, using generalization arguments. Finally, we develop our population-level localization scheme in Section 12.3.3, and combine it with our subproblem solver to give our overall method for heavy-tailed private SCO. Several proofs and a generalization to strongly convex functions (Corollary 12.5.1) are deferred to Appendix 12.5.

12.3.1 Strongly convex DP-ERM solver

We give a parameterized subroutine for minimizing a DP-ERM objective $F_{\mathcal{D},\lambda}(x)$ associated with a dataset $\mathcal{D}$ and a regularization parameter $\lambda \geq 0$ (recalling the definition (12.5)). In this section only, for notational convenience we identify elements of $\mathcal{D}$ with $[n]$ where $n := |\mathcal{D}|$, so we will also write

$$F_{\mathcal{D},\lambda}(x) := \frac{1}{n}\sum_{i\in[n]} f_i(x) + \frac{\lambda}{2}\left\|x\right\|^2,$$

i.e. we let $f_i(\cdot) := f(\cdot; s)$ where $s \in \mathcal{D}$ is the element identified with $i \in [n]$. Our subroutine is a clipped DP-SGD algorithm (Algorithm 36), which only clips the heavy-tailed portion of $\nabla F_{\mathcal{D},\lambda}$ (i.e. the sample gradients), and leaves both the regularization and additive noise unchanged. The utility of Algorithm 36 is parameterized by the following function of the dataset:

$$b_{\mathcal{D}} := \max_{x\in\mathcal{X}} \left\| \frac{1}{n}\sum_{i\in[n]} \nabla f_i(x) - \frac{1}{n}\sum_{i\in[n]} \Pi_C(\nabla f_i(x)) \right\|.$$

In other words, $b_{\mathcal{D}}$ denotes the maximum bias incurred by the clipped gradient of $F_{\mathcal{D}}$ when compared to the true gradient, over points in $\mathcal{X}$; note the maximum is achieved as $\mathcal{X}$ is compact.

We are now ready to state our algorithm, Clipped-DP-SGD, as Algorithm 36.

Algorithm 36: Clipped-DP-SGD$(\mathcal{D}, C, \lambda, \{\eta_t\}_{t\in[T]}, \sigma^2, T, r, \mathcal{X})$

Input: Dataset $\mathcal{D} \in \mathcal{S}^n$, clip threshold $C \in \mathbb{R}_{\geq 0}$, regularization $\lambda \in \mathbb{R}_{\geq 0}$, step sizes $\{\eta_t\}_{t\in[T]} \subset \mathbb{R}_{\geq 0}$, noise $\sigma^2 \in \mathbb{R}_{\geq 0}$, iteration count $T \in \mathbb{N}$, radius $r \in \mathbb{R}_{\geq 0}$, domain $\mathcal{X} \subset \mathbb{B}(r)$ with $\mathcal{X} \ni \curlywedge_d$

$x_0 \leftarrow \curlywedge_d$

for $0 \leq t < T$ **do**

  $\xi_t \sim \mathcal{N}(\curlywedge_d, \sigma^2 \mathbf{I}_d)$

  $\widehat{g}_t \leftarrow \frac{1}{n}\sum_{i\in[n]} \Pi_C(\nabla f_i(x_t))$

  $x_{t+1} \leftarrow \operatorname{argmin}_{x\in\mathcal{X}_r} \{\eta_t \langle \widehat{g}_t + \xi_t, x\rangle + \frac{\eta_t\lambda}{2}\|x\|^2 + \frac{1}{2}\|x - x_t\|^2\}$

end

Return: $\widehat{x} \leftarrow \frac{\sum_{0\leq t<T}(t+4)x_t}{\sum_{0\leq t<T}(t+4)}$

We provide the following guarantee on Clipped-DP-SGD, by modifying an analysis of [LSB12].

Proposition 12.3.1. *Let $\rho \geq 0$, and $\widehat{x}$ be the output of* Clipped-DP-SGD *with $\eta_t \leftarrow \frac{4}{\lambda(t+1)}$ for all $0 \leq t < T$, $\sigma^2 \leftarrow \frac{2C^2T}{n^2\rho}$, and $T \geq \max(n, \frac{n^2\rho}{d})$.* Clipped-DP-SGD *satisfies ρ-CDP, and*

$$\mathbb{E}[F_{\mathcal{D},\lambda}(\widehat{x}) - F_{\mathcal{D},\lambda}(x\mathrm{OPT})] \leq \frac{32C^2d}{\lambda n^2\rho} + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{7\lambda r^2}{n}, \text{ where } x^\star := \operatorname*{argmin}_{x\in\mathcal{X}} F_{\mathcal{D},\lambda}(x).$$

For ease of use of Proposition 12.3.1, we now provide a simple bound on $b_{\mathcal{D}}$ which holds with constant probability from a dataset drawn from a distribution satisfying Assumption 12.2.5.

Lemma 12.3.2. *Let $\mathcal{D} \sim \mathcal{P}^n$, where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5. With probability at least $\frac{4}{5}$, denoting $b_{\mathcal{D}}$ as in (12.25), we have*

$$b_{\mathcal{D}} \leq \frac{5G_k^k}{(k-1)C^{k-1}}.$$

We therefore have the following corollary of Proposition 12.3.1 and Lemma 12.3.2.

Corollary 12.3.3. *Let $\mathcal{D} \sim \mathcal{P}^n$, where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, and let $x^\star_{\mathcal{D},\lambda} := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{D},\lambda}(x)$, following (12.5). If we run* Clipped-DP-SGD *with parameters in Proposition 12.3.1 and $C \leftarrow G_k \cdot (\frac{25n^2\rho}{32d})^{\frac{1}{2k}}$,* Clipped-DP-SGD *is ρ-CDP, and there is a universal constant C_{erm} such that with probability $\geq \frac{3}{5}$ over the randomness of $\mathcal{D}$ and* Clipped-DP-SGD*, $\widehat{x}$, the output of* Clipped-DP-SGD*, satisfies*

$$\left\|\widehat{x} - x^\star_{\mathcal{D},\lambda}\right\| \leq C_{\mathrm{erm}}\left(\frac{G_k}{\lambda}\left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}} + \frac{r}{\sqrt{n}}\right).$$

Clipped-DP-SGD *queries at most $\max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$).*

Proof. Condition on the conclusion of Lemma 12.3.2, which holds with probability $\frac{4}{5}$. Therefore, Markov's inequality shows that with probability at least $\frac{3}{5}$, after a union bound with Proposition 12.3.1,

$$\begin{aligned}\frac{\lambda}{2}\left\|\widehat{x} - x^\star_{\mathcal{D},\lambda}\right\|^2 &\leq F_{\mathcal{D},\lambda}(\widehat{x}) - F_{\mathcal{D},\lambda}(x^\star_{\mathcal{D},\lambda}) \\ &\leq \frac{160C^2d}{\lambda n^2\rho} + \frac{125G_k^{2k}}{\lambda C^{2(k-1)}} + \frac{7\lambda r^2}{n} \leq \frac{320G_k^2}{\lambda}\left(\frac{d}{n^2\rho}\right)^{1-\frac{1}{k}} + \frac{7\lambda r^2}{n},\end{aligned}$$

where we used strong convexity in the first inequality, and plugged in our choice of C in the last. The conclusion follows by rearranging the above display, and using $\sqrt{a^2+b^2} \leq a+b$ for $a, b \in \mathbb{R}_{\geq 0}$. □

12.3.2 Localizing regularized population loss minimizers

Here, we use generalization arguments from the SCO literature to show how that our algorithm Clipped-DP-SGD from Section 12.3.1 acts as an oracle which, with constant probability, returns a point near the minimizer of a regularized population loss. We begin with a standard helper statement.

Lemma 12.3.4. *Let $\lambda \geq 0$, let $\mathcal{P}$ be a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, let*

$\overline{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex, and let

$$x^\star_{\lambda,\overline{x}} := \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ F_{\mathcal{P}}(x) + \frac{\lambda}{2}\|x-\overline{x}\|^2 \right\}, \text{ where } F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s\sim\mathcal{P}}[f(x;s)]. \tag{12.6}$$

Then $\|\overline{x} - x^\star_{\lambda,\overline{x}}\| \le \frac{2G_1}{\lambda}$.

Next, we apply a result on generalization due to [LR23] to bound the expected distance between a restricted empirical regularized minimizer and the minimizer of the population variant in (12.6).

Lemma 12.3.5. *Let $\lambda \ge 0$, let $\mathcal{D} \sim \mathcal{P}^n$ where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, and let $\overline{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex. Following notation (12.4), (12.5), let*

$$y := \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ [F_{\mathcal{D}}]_{\mathbb{B}(\overline{x},r)}(x) + \frac{\lambda}{2}\|x-\overline{x}\|^2 \right\}, \quad \text{for } r := \frac{2G_1}{\lambda}$$

and let $x^\star_{\lambda,\overline{x}}$ be defined as in (12.6). Then with probability ≥ 0.95 over the randomness of $\mathcal{D} \sim \mathcal{P}^n$,

$$\left\|y - x^\star_{\lambda,\overline{x}}\right\|_2 \le \frac{90G_2}{\lambda\sqrt{n}}.$$

Corollary 12.3.6. *Let $\mathcal{D} \sim \mathcal{P}^n$, where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, and let $\overline{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex. Let $\lambda \ge 0$ and define $x^\star_{\lambda,\overline{x}}$ as in (12.6). There is a ρ-CDP algorithm $\mathcal{A}$ which queries $\max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$). With probability 0.55 over the randomness of $\mathcal{A}$ and $\mathcal{D}$, $\mathcal{A}$ returns $\widehat{x}$ satisfying, for a universal constant $C_{\text{reg-pop}}$,*

$$\left\|\widehat{x} - x^\star_{\lambda,\overline{x}}\right\| \le C_{\text{reg-pop}} \left(\frac{G_k}{\lambda} \left(\frac{\sqrt{d}}{n\sqrt{\rho}} \right)^{1-\frac{1}{k}} + \frac{G_2}{\lambda\sqrt{n}} \right).$$

Proof. Condition on the conclusion of Lemma 12.3.5 holding for our dataset, which loses 0.05 in the failure probability. Next, consider the guarantee of Corollary 12.3.3, when applied to the truncated and shifted functions, $\tilde{f}(x;s) \leftarrow f_{\mathbb{B}(\overline{x},r)}(x-\overline{x};s)$, where r is set as in Lemma 12.3.5. It shows that with probability $\frac{3}{5}$, $\|\widehat{x} + \overline{x} - y\| = O(\frac{G_k}{\lambda}(\frac{\sqrt{d}}{n\sqrt{\rho}})^{1-\frac{1}{k}} + \frac{\sqrt{\lambda}r}{\sqrt{n}})$,

for the point $\widehat{x}$ returned by the algorithm, and y the exact minimizer of the empirical loss restricted to $\mathbb{B}(\overline{x}, r)$. Therefore, the conclusion follows by overloading $\widehat{x} \leftarrow \widehat{x} + \overline{x}$, applying the triangle inequality with the conclusions of Corollary 12.3.3 and 12.3.6, and taking a union bound over their failure probabilities. □

12.3.3 Population-level localization

In this section, we provide a generic population-level localization scheme for stochastic convex optimization, which may be of broader interest. Our localization scheme is largely patterned off of the analogous localization methods developed by [FKT20], but directly argues about contraction to population-level regularized minimizers (as opposed to empirical minimizers), which makes it compatible with our framework in Section 12.3.1 and 12.3.2, specificially the guarantees of Corollary 12.3.6.

Algorithm 37: Population-Localize$(x_0, \mathcal{P}, \lambda, I)$

Input: Initial point $x_0 \in \mathcal{X}$, distribution $\mathcal{P}$ over samples in $\mathcal{S}$, for $\mathcal{X}, \mathcal{S}$ inducing a k-heavy-tailed DP-SCO problem as in Definition 12.2.6, with a population loss $F_{\mathcal{P}} := \mathbb{E}_{s\sim\mathcal{S}}[f(\cdot; s)]$, $\lambda \geq 0$, $I \in \mathbb{N}$

for $i \in [I]$ **do**

$\lambda_i \leftarrow \lambda \cdot 32^i$

$x_i \leftarrow$ any point satisfying

$$\|x_i - x_i^\star\| \leq \frac{\Delta 4^i}{\lambda_i}, \text{ where } x_i^\star := \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ F_{\mathcal{P}}(x) + \frac{\lambda_i}{2} \|x - x_{i-1}\|^2 \right\} \tag{12.7}$$

end

Return: x_I

We briefly discuss the role of the hyperparameters λ, Δ in Algorithm 37 for clarity. The parameter Δ scales with the error guarantee of our regularized ERM solver; in particular, it will be determined by the bound in Corollary 12.3.6. The parameter λ specifies an initial regularization amount that will later be tuned to trade off the terms in the following Proposition 12.3.7.

Proposition 12.3.7. *Following notation of Algorithm 37, let* $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$. *Then,*

$$F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star) \le \frac{G_1\Delta}{\lambda 8^I} + \frac{\Delta^2}{4\lambda} + \frac{\lambda D^2}{2}.$$

In particular, choosing λ *to optimize this bound, we have*

$$F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star) \le 2D\sqrt{\frac{G_1\Delta}{8^I}} + D\Delta.$$

Proof. We denote $x_0^\star := x^\star$ throughout the proof. First, we expand

$$\begin{aligned} F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x_0^\star) &= F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x_I^\star) + F_{\mathcal{P}}(x_I^\star) - F_{\mathcal{P}}(x_0^\star) \\ &= F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x_I^\star) + \sum_{i\in[I]} F_{\mathcal{P}}(x_i^\star) - F_{\mathcal{P}}(x_{i-1}^\star). \end{aligned}$$

Moreover, for each $i \in [I]$, since $x_i^\star$ minimizes $F_{\mathcal{P}}(x) + \frac{\lambda_i}{2}\|x - x_{i-1}\|^2$,

$$F_{\mathcal{P}}(x_i^\star) \le F_{\mathcal{P}}(x_i^\star) + \frac{\lambda_i}{2}\|x_i^\star - x_{i-1}\|^2 \le F_{\mathcal{P}}(x_{i-1}^\star) + \frac{\lambda_i}{2}\left\|x_{i-1}^\star - x_{i-1}\right\|^2.$$

Combining the above two displays, and using that $F_{\mathcal{P}}$ is G_1-Lipschitz (Lemma 12.2.7), we have

$$\begin{aligned} F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star) &\le G_1\|x_I - x_I^\star\| + \sum_{i\in[I]} \frac{\lambda_i}{2}\left\|x_{i-1}^\star - x_{i-1}\right\|^2 \\ &\le \frac{G_1\Delta}{\lambda 8^I} + \sum_{i\in[I-1]} \frac{\Delta^2 16^i}{2\lambda_i} + \frac{\lambda D^2}{2} \le \frac{G_1\Delta}{\lambda 8^I} + \frac{\Delta^2}{4\lambda} + \frac{\lambda D^2}{2}, \end{aligned}$$

where we used the diameter bound assumption $\operatorname{diam}(\mathcal{X}) = D$, as in Definition 12.2.6. □

In particular, note that Corollary 12.3.6 shows that by using n samples from $\mathcal{P}$ and a CDP budget of ρ, with constant probability, we can satisfy the requirement (12.7) with $\Delta 4^i = O(G_k(\frac{\sqrt{d}}{n\sqrt{\rho}})^{1-\frac{1}{k}} + \frac{G_2}{\sqrt{n}})$. By plugging this guarantee into the aggregation subroutine in Fact 12.2.9, we have our SCO algorithm.

Theorem 12.3.8. *Consider an instance of* k*-heavy-tailed private SCO, following nota-*

Algorithm 38: Aggregate-ERM$(\overline{x}, \lambda, J, \rho, \{s_\ell\}_{\ell\in[nJ]}, R)$

Input: Regularization center $\overline{x} \in \mathcal{X}$, regularization $\lambda \in \mathbb{R}_{\geq 0}$, sample split parameter $J \in \mathbb{N}$, privacy parameter $\rho \in \mathbb{R}_{\geq 0}$, samples $\{s_\ell\}_{\ell\in[nJ]} \subset \mathcal{S}$, distance bound $R \in \mathbb{R}_{\geq 0}$

for $j \in [J]$ **do**

  $\mathcal{D}^j \leftarrow \{s_\ell\}_{(j-1)n<\ell\leq jn}$ for all $j \in [J]$

  $x^j \leftarrow$ result of Corollary 12.3.6 using $\mathcal{D}^j$, on loss defined by $\overline{x}, \lambda$ with privacy parameter ρ, i.e., x^j is a point satisfying, with probability 0.55, for a universal constant $C_{\text{reg-pop}}$,

$$\left\|x^j - x^\star_{\lambda,\overline{x}}\right\| \leq C_{\text{reg-pop}} \left(\frac{G_k}{\lambda}\left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}} + \frac{G_2}{\lambda\sqrt{n}}\right)$$

end

$x \leftarrow$ Aggregate$(\{x^j\}_{j\in[J]}, R)$ (see Fact 12.2.9)

Return: x

tion in Definition 12.2.6, let $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$*, and let* $\rho \geq 0$*,* $\delta \in (0,1)$*. Algorithm 37 using Algorithm 38 in Line 5 is a* ρ*-CDP algorithm which draws* $\mathcal{D} \sim \mathcal{P}^n$*, queries* $C_{\text{sco}} \max(n^2, \frac{n^3\rho}{d})$ *sample gradients (using samples in* $\mathcal{D}$*) for a universal constant* C_{sco}*, and outputs* $x \in \mathcal{X}$ *satisfying, with probability* $\geq 1-\delta$*,*

$$F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star) \leq C_{\text{sco}} \left(G_k D \cdot \left(\frac{\sqrt{d}\log\left(\frac{1}{\delta}\right)}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}} + G_2 D \cdot \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{n}}\right).$$

12.4 Conclusion

In this work, we consider the DP-SCO with heavy-tailed gradients. When the k-th moments of gradients are bounded, we propose the population-level localization framework and attain near-optimal excess loss $G_2 D \cdot \sqrt{\frac{\log(\frac{1}{\delta})}{n}} + G_k D \cdot (\frac{\sqrt{d}\log(\frac{1}{\delta})}{n\sqrt{\rho}})^{1-\frac{1}{k}}$ with probability at least $1-\delta$ and satisfy ρ-CDP. We can achieve a tight rate for high-probability SCO under a bounded-variance gradient estimator parameterization by applying the population-level localization framework. Moreover, we improve this basic result under additional assumptions, including an optimal algorithm under a known-Lipschitz constant assumption, a near-linear time

algorithm for smooth functions, and an optimal linear time algorithm for smooth generalized linear models, with interesting techniques adapted to each setting.

It leaves many intriguing open problems in this direction. For example, can we design near-linear time algorithms for non-smooth functions? Can the population-level localization framework be applied to solve other problems? Can we establish a high-probability lower bound or eliminate the additional logarithmic term if we are only concerned with the excess bound in expectation? Can we evaluate the algorithm's performance through numerical simulations or real-world datasets? We leave these questions for future research.

12.5 Deferred proofs from the main body

12.5.1 Deferred proofs from Section 1.2

Lemma 12.2.7. *Let $\mathcal{P}$ be a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5. Then $F_{\mathcal{P}}$ is G_1-Lipschitz.*

Proof. This follows from the derivation

$$\max_{x\in\mathcal{X}} \left\| \mathop{\mathbb{E}}_{s\sim\mathcal{P}} [\nabla f(x;s)] \right\| \le \max_{x\in\mathcal{X}} \mathop{\mathbb{E}}_{s\sim\mathcal{P}} \|\nabla f(x;s)\| \le \mathop{\mathbb{E}}_{s\sim\mathcal{P}} \max_{x\in\mathcal{X}} \|\nabla f(x;s)\| \le G_1.$$

□

12.5.2 Deferred proofs from Section 12.3

Proposition 12.3.1. *Let $\rho \ge 0$, and $\widehat{x}$ be the output of* Clipped-DP-SGD *with $\eta_t \leftarrow \frac{4}{\lambda(t+1)}$ for all $0 \le t < T$, $\sigma^2 \leftarrow \frac{2C^2T}{n^2\rho}$, and $T \ge \max(n, \frac{n^2\rho}{d})$.* Clipped-DP-SGD *satisfies ρ-CDP, and*

$$\mathbb{E}[F_{\mathcal{D},\lambda}(\widehat{x}) - F_{\mathcal{D},\lambda}(x\mathrm{OPT})] \le \frac{32C^2d}{\lambda n^2\rho} + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{7\lambda r^2}{n}, \text{ where } x^\star := \operatorname*{argmin}_{x\in\mathcal{X}} F_{\mathcal{D},\lambda}(x).$$

Proof. For the privacy claim, note that each call to Line 6 is a postprocessing of a $\frac{2C}{n}$-sensitive statistic of the dataset $\mathcal{D}$, since neighboring databases can only change $\frac{1}{n}\sum_{i\in[n]} \Pi_C(\nabla f_i(x_t))$ by $\frac{2C}{n}$ in the ℓ_2 norm, via the triangle inequality. Therefore, applying the first and third parts of Lemma 12.2.4 shows that after T iterations, the CDP of the mechanism is at most $T \cdot \frac{2C^2}{n^2\sigma^2} \le \rho$.

We next prove the utility claim. For each $0 \le t \le T$, denote

$$\Delta_t := \mathbb{E}\left[F_{\mathcal{D},\lambda}(x_t) - F_{\mathcal{D},\lambda}(x\mathrm{OPT})\right],\ \Phi_t := \mathbb{E}\left[\frac{1}{2}\left\|x_t - x\mathrm{OPT}\right\|^2\right],\ g_t := \nabla F_{\mathcal{D}}(x_t),$$

where all expectations are only over randomness used by the algorithm, and not the randomness in sampling $\mathcal{D}$. First-order optimality applied to the definition of x_{t+1} implies, for all $0 \le t < T$,

$$\langle \widehat{g}_t + \xi_t, x_t - x^\star\rangle + \langle \lambda x_{t+1}, x_{t+1} - x^\star\rangle \le \frac{1}{2\eta_t}\left(\|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\right) + \frac{\eta_t}{2}\|\widehat{g}_t + \xi_t\|^2.$$

Adding $\langle g_t - \widehat{g}_t - \xi_t, x_t - x^\star\rangle$ to both sides and rearranging shows

$$\begin{aligned}
&F_{\mathcal{D}}(x_t) + \frac{\lambda}{2}\|x_{t+1}\|^2 - F_{\mathcal{D},\lambda}(x^\star) + \frac{\lambda}{2}\|x_{t+1} - x^\star\|^2 \\
\le&\langle g_t, x_t - x^\star\rangle + \langle \lambda x_{t+1}, x_{t+1} - x^\star\rangle \\
\le&\frac{1}{2\eta_t}\left(\|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\right) + \frac{\eta_t}{2}\|\widehat{g}_t + \xi_t\|^2 + \langle g_t - \widehat{g}_t - \xi_t, x_t - x^\star\rangle \\
\le&\frac{1}{2\eta_t}\left(\|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\right) + \eta_t C^2 + \eta_t\|\xi_t\|^2 + b_{\mathcal{D}}\|x_t - x^\star\| - \langle \xi_t, x_t - x^\star\rangle.
\end{aligned}$$

In the first line, we used strong convexity of the function $\frac{\lambda}{2}\|x\|^2$, and in the last line, we used $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ and the definitions of C and $b_{\mathcal{D}}$. Next, adding $\frac{\lambda}{2}(\|x_t\|^2 - \|x_{t+1}\|^2)$ to both sides above and taking expectations over the first t iterations yields

$$\Delta_t + \lambda\Phi_{t+1} \le \frac{1}{\eta_t}(\Phi_t - \Phi_{t+1}) + \eta_t(C^2 + \sigma^2 d) + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{\lambda}{2}\Phi_t + \frac{\lambda}{2}\left(\mathbb{E}\|x_t\|^2 - \mathbb{E}\|x_{t+1}\|^2\right),$$

where we used the Fenchel-Young inequality to bound $b_{\mathcal{D}}\|x_t - x^\star\| \le \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{\lambda}{4}\|x_t - x^\star\|^2$. Now, plugging in our step size schedule $\eta_t = \frac{4}{\lambda(t+1)}$, multiplying by $t+4$, and rearranging shows

$$\begin{aligned}
(t+4)\Delta_t \le& \frac{\lambda(t+3)(t+4)}{4}\Phi_t - \frac{\lambda(t+5)(t+4)}{4}\Phi_{t+1} \\
&+ \frac{4(t+4)}{\lambda(t+1)}\left(\frac{3C^2Td}{n^2\rho}\right) + \frac{(t+4)b_{\mathcal{D}}^2}{\lambda} + \frac{\lambda(t+4)}{2}\left(\mathbb{E}\|x_t\|^2 - \mathbb{E}\|x_{t+1}\|^2\right),
\end{aligned}$$

where we plugged in the choice of σ^2 and $T \geq \frac{n^2\rho}{d}$, so $C^2 \leq \frac{\sigma^2 d}{2}$. Summing the above for $0 \leq t < T$, using that all iterates and $x^\star$ lie in $\mathbb{B}(r)$, and dividing by $Z := \sum_{0\leq t<T}(t+4) \geq \frac{T^2}{2}$, shows

$$\begin{aligned}\frac{1}{Z}\sum_{0\leq t<T}(t+4)\Delta_t &\leq \frac{3\lambda\Phi_0}{Z} + \frac{16C^2T^2d}{\lambda Z n^2\rho} + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{\lambda}{2Z}\sum_{t\in[T]}\mathbb{E}\left\|x_t\right\|^2 \\ &\leq \frac{6\lambda r^2}{T^2} + \frac{32C^2d}{\lambda n^2\rho} + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{\lambda r^2}{T} \leq \frac{32C^2d}{\lambda n^2\rho} + \frac{b_{\mathcal{D}}^2}{\lambda} + \frac{7\lambda r^2}{T}.\end{aligned}$$

The conclusion follows from convexity of $F_{\mathcal{D},\lambda}$, the definition of $\widehat{x}$, and $T \geq n$. □

Lemma 12.3.2. *Let $\mathcal{D} \sim \mathcal{P}^n$, where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5. With probability at least $\frac{4}{5}$, denoting $b_{\mathcal{D}}$ as in (12.25), we have*

$$b_{\mathcal{D}} \leq \frac{5G_k^k}{(k-1)C^{k-1}}.$$

Proof. For every $s \in \mathcal{S}$ let $x^\star(s) := \operatorname{argmax}_{x\in\mathcal{X}} \left\|\nabla f(x;s) - \Pi_C(\nabla f(x;s))\right\|_2$. Then, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}[b_{\mathcal{D}}] &= \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\max_{x\in\mathcal{X}}\left\|\frac{1}{n}\sum_{i\in[n]}\nabla f_i(x) - \frac{1}{n}\sum_{i\in[n]}\Pi_C(\nabla f_i(x))\right\|\right] \\ &\leq \frac{1}{n}\sum_{i\in[n]}\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\max_{x\in\mathcal{X}}\left\|\nabla f_i(x) - \Pi_C(\nabla f_i(x))\right\|\right] \\ &= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\left\|\nabla f(x^\star(s);s) - \Pi_C(\nabla f(x^\star(s);s))\right\|\right] \leq \frac{\mathbb{E}\left[\left\|\nabla f(x^\star(s);s)\right\|^k\right]}{(k-1)C^{k-1}} \\ &\leq \frac{G_k^k}{(k-1)C^{k-1}}.\end{aligned}$$

The last line used independence of samples, used Fact 12.2.8 on the random vector $\nabla f(x^\star(s);s)$, and applied Assumption 12.2.5 with the definition of $x^\star(s)$. The conclusion uses Markov's inequality. □

Lemma 12.3.4. *Let $\lambda \geq 0$, let $\mathcal{P}$ be a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, let*

$\overline{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex, and let

$$x^\star_{\lambda,\overline{x}} := \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ F_{\mathcal{P}}(x) + \frac{\lambda}{2}\left\|x-\overline{x}\right\|^2 \right\}, \text{ where } F_{\mathcal{P}}(x) := \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[f(x;s)\right]. \tag{12.6}$$

Then $\left\|\overline{x} - x^\star_{\lambda,\overline{x}}\right\| \le \frac{2G_1}{\lambda}$.

Proof. Let $r := \left\|\overline{x} - x^\star\right\|$. By strong convexity and the definition of $x^\star_{\lambda,\overline{x}}$,

$$\frac{\lambda r^2}{2} \le F_{\mathcal{P}}(\overline{x}) - F_{\mathcal{P}}(x^\star) - \frac{\lambda}{2}\left\|x^\star - \overline{x}\right\|^2 \le F_{\mathcal{P}}(\overline{x}) - F_{\mathcal{P}}(x^\star) \le G_1 r.$$

Here, we used that $F_{\mathcal{P}}$ is G_1-Lipschitz (Lemma 12.2.7), and rearranging yields the conclusion. □

Lemma 12.3.5. *Let $\lambda \ge 0$, let $\mathcal{D} \sim \mathcal{P}^n$ where $\mathcal{P}$ is a distribution over $\mathcal{S}$ satisfying Assumption 12.2.5, and let $\overline{x} \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex. Following notation (12.4), (12.5), let*

$$y := \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \left[F_{\mathcal{D}}\right]_{\mathbb{B}(\overline{x},r)}(x) + \frac{\lambda}{2}\left\|x-\overline{x}\right\|^2 \right\}, \quad \text{for } r := \frac{2G_1}{\lambda}$$

and let $x^\star_{\lambda,\overline{x}}$ be defined as in (12.6). Then with probability ≥ 0.95 over the randomness of $\mathcal{D} \sim \mathcal{P}^n$,

$$\left\|y - x^\star_{\lambda,\overline{x}}\right\|_2 \le \frac{90G_2}{\lambda\sqrt{n}}.$$

Proof. For each $f(x;s)$, define a restricted variant $\tilde{f}(x;s) := f_{\mathbb{B}(\overline{x},r)}(x;s)$, and let $\widetilde{F}_{\mathcal{P}} := \mathbb{E}_{s\sim\mathcal{S}}\,\tilde{f}(\cdot;s)$. Similarly, define $\widetilde{F}_{\mathcal{D}}$ to be the restricted variant of the empirical loss $F_{\mathcal{D}}$. Because $\widetilde{F}_{\mathcal{P}}$ is pointwise larger than $F_{\mathcal{P}}$ and $x^\star_{\lambda,\overline{x}} \in \mathbb{B}(\overline{x},r)$ by Lemma 12.3.4, it is clear that

$$x^\star_{\lambda,\overline{x}} = \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \widetilde{F}_{\mathcal{P}}(x) + \frac{\lambda}{2}\left\|x-\overline{x}\right\|^2 \right\},$$

and y is the minimizer of the empirical (restricted) variant of the above display. Moreover, each of the regularized functions $\tilde{f}(x;s) + \frac{\lambda}{2}\left\|x-\overline{x}\right\|^2$ has a Lipschitz constant at most $\lambda r = 2G_1$ larger than its unregularized counterpart in $\mathcal{X} \cap \mathbb{B}(\overline{x},r)$, so these functions satisfy the moment bound in Assumption 12.2.5 for $j = 2$ with a bound of $2G_2^2 + 8G_1^2$. Now,

applying Proposition 29, [LR23] yields

$$\begin{aligned}
&\mathbb{E}\left[\left(\widetilde{F}_{\mathcal{P}}(y)+\frac{\lambda}{2}\left\|y-\overline{x}\right\|^2\right)-\left(\widetilde{F}_{\mathcal{P}}(x^\star_{\lambda,\overline{x}})+\frac{\lambda}{2}\left\|x^\star_{\lambda,\overline{x}}-\overline{x}\right\|^2\right)\right]\\
&=\mathbb{E}\left[\left(\widetilde{F}_{\mathcal{D}}(y)+\frac{\lambda}{2}\left\|y-\overline{x}\right\|^2\right)-\left(\widetilde{F}_{\mathcal{D}}(x^\star_{\lambda,\overline{x}})+\frac{\lambda}{2}\left\|x^\star_{\lambda,\overline{x}}-\overline{x}\right\|^2\right)\right]\\
&+\mathbb{E}\left[\left(\widetilde{F}_{\mathcal{P}}(y)+\frac{\lambda}{2}\left\|y-\overline{x}\right\|^2\right)-\left(\widetilde{F}_{\mathcal{D}}(y)+\frac{\lambda}{2}\left\|y-\overline{x}\right\|^2\right)\right]\\
&\le 0+\frac{4G_2^2+16G_1^2}{\lambda n}=\frac{4G_2^2+16G_1^2}{\lambda n}\le\frac{20G_2^2}{\lambda n}.
\end{aligned}$$

The first equality used that $x^\star_{\lambda,\overline{x}}$ is independent of sampling $\mathcal{D}$, and the second used $\widehat{x}$ is the empirical risk minimizer. The conclusion follows from Markov's inequality and strong convexity. □

Theorem 12.3.8. *Consider an instance of k-heavy-tailed private SCO, following notation in Definition 12.2.6, let $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$, and let $\rho \ge 0$, $\delta \in (0,1)$. Algorithm 37 using Algorithm 38 in Line 5 is a ρ-CDP algorithm which draws $\mathcal{D}\sim\mathcal{P}^n$, queries $C_{\mathrm{sco}}\max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$) for a universal constant C_{sco}, and outputs $x\in\mathcal{X}$ satisfying, with probability $\ge 1-\delta$,*

$$F_{\mathcal{P}}(x)-F_{\mathcal{P}}(x^\star)\le C_{\mathrm{sco}}\left(G_kD\cdot\left(\frac{\sqrt{d}\log\left(\frac{1}{\delta}\right)}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}+G_2D\cdot\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{n}}\right).$$

Proof. Throughout, we assume that $\frac{1}{\delta}$ is at least a large enough constant (where lossiness can be absorbed into C_{sco}), and that n is at least a sufficiently large constant multiple of $\log\frac{1}{\delta}$ (because the entire range of $F_{\mathcal{P}}$ is $\le G_2D$). We first handle the case where $\frac{1}{\delta}$ is larger than $\mathrm{polylog}(n)$, deferring the case of small $\frac{1}{\delta}$ to the end of the proof. Let $I, J\in\mathbb{N}$ be chosen such that

$$I:=\left\lfloor\log_2\left(\frac{n}{J}\right)\right\rfloor,\ J\in\left[400\log\left(\frac{I}{\delta}\right),500\log\left(\frac{I}{\delta}\right)\right],$$

which is achievable with $I=O(\log n)$ and $J=O(\log\frac{\log n}{\delta})=O(\log\frac{1}{\delta})$. Let $m:=\frac{n}{J}$, and assume without loss that m is a power of 2, which we can guarantee by discarding $\le\frac{1}{2}$ our

samples, losing a constant factor in the claim. For each $i \in [I]$, let $m_i := \frac{m}{2^i}$. We subdivide $\mathcal{D}$ into J portions, each with m samples, and subdivide each portion into I parts each with m_i samples. For $j \in [J]$ and $i \in [I]$, we denote the samples corresponding to the i^{th} part of the j^{th} portion by $\mathcal{D}_i^j$, so

$$\bigcup_{i \in [I]} \bigcup_{j \in [J]} \mathcal{D}_i^j \subseteq \mathcal{D}, \ |\mathcal{D}_i^j| = m_i \text{ for all } j \in [J], \ \mathcal{D}_i^j \cap \mathcal{D}_{i'}^{j'} = \emptyset \text{ for all } (i,j) \neq (i',j').$$

Next, we show how to implement Line 5 in Algorithm 37, for an iteration $i \in [I]$, by calling Algorithm 38 with appropriate parameters. Let $n \leftarrow m_i$, $\rho \leftarrow \rho$, and initialize Algorithm 38 with the dataset $\cup_{j \in [J]} \mathcal{D}_i^j$ and $R := \frac{\Delta 4^i}{\lambda_i}$, where

$$\Delta := 3C_{\text{reg-pop}} \left(G_k \cdot \left(\frac{\sqrt{d}}{m\sqrt{\rho}} \right)^{1-\frac{1}{k}} + \frac{G_2}{\sqrt{m}} \right).$$

By Corollary 12.3.6, each independent run outputs $x_i^j \in \mathcal{X}$ satisfying, with probability 0.55,

$$\left\| x_i^j - x_i^\star \right\| \leq \frac{C_{\text{reg-pop}}}{\lambda_i} \left(G_k \cdot \left(\frac{\sqrt{d}}{m_i\sqrt{\rho}} \right)^{1-\frac{1}{k}} + \frac{G_2}{\sqrt{m_i}} \right) \leq \frac{\Delta 4^i}{3\lambda_i} = \frac{R}{3}. \tag{12.8}$$

Therefore, by a Chernoff bound, with probability $\geq 1 - \frac{\delta}{I}$, at least $0.51J$ of the copies satisfy the above bound, so Fact 12.2.9 yields x_i satisfying $\|x_i - x_i^\star\| \leq R = \frac{\Delta 4^i}{\lambda_i}$ with the same probability. Union bounding over all I iterations of Algorithm 37 yields the failure probability, and so we obtain the claim from Proposition 12.3.7, after plugging in $n = O(m \log(\frac{1}{\delta}))$, since the dominant term is $D\Delta$. The privacy proof follows from the first part of Lemma 12.2.4 since for each pair of neighboring databases, exactly one of the datasets $\mathcal{D}_i^j$ are neighboring, and Corollary 12.3.6 guarantees privacy of the empirical risk minimization algorithm using that dataset; privacy for all other datasets used is immediate from postprocessing properties of privacy. The gradient complexity comes from aggregating all of the IJ calls to Corollary 12.3.6, where we recall the sample sizes decay geometrically.

Finally, if $\frac{1}{\delta}$ is smaller than $\text{polylog}(n)$, for the i^{th} iteration of Algorithm 37 we instead set $J_i \in [400 \log(\frac{I}{\delta_i}), 500 \log(\frac{I}{\delta_i})]$ where $\delta_i := \frac{\delta}{2^i}$. Then we subdivide a consecutive batch

of $\frac{n}{2^i}$ samples into J_i portions, and follow the above proof. It is straightforward to check that (12.8) still holds with the new value of $m_i = \lfloor \frac{n}{2^i J_i} \rfloor$ because the 4^i factor growth on the right-hand side continues to outweigh the change in m_i. The error bound follows from Proposition 12.3.7, and the privacy proof is identical. □

12.5.3 Strongly convex heavy-tailed private SCO via localization

Finally, by following the template of standard localization reductions in the literature (see e.g. Theorem 5.1, [FKT20] or Lemma 5.5, [KLL21]), Theorem 12.3.8 obtains an improved rate when all sample functions are strongly convex. For completeness, we state this result below.

Corollary 12.5.1. *In the setting of Theorem 12.3.8, suppose $f(x; s)$ is μ-strongly convex for all $s \in \mathcal{S}$. There is an algorithm which draws $\mathcal{D} \sim \mathcal{P}^n$, queries $C_{\mathrm{sco}} \max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$) for a universal constant C_{sco}, and outputs $x \in \mathcal{X}$ satisfying, with probability $\geq 1 - \delta$,*

$$F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star) \leq C_{\mathrm{sco}} \left(\frac{G_k^2}{\mu} \cdot \left(\frac{d \log^3\left(\frac{1}{\delta}\right)}{n^2 \rho} \right)^{1 - \frac{1}{k}} + \frac{G_2^2}{\mu} \cdot \frac{\log\left(\frac{1}{\delta}\right)}{n} \right).$$

Proof. This is immediate from the development in Section 5.1 (and the proof of Theorem 5.1) of [FKT20], but we mention one slight difference here. Our guarantees in Theorem 12.3.8 do not scale with the initial distance bound to the function minimizer, and instead scale with the domain size, which makes it less directly compatible with the standard localization framework in [FKT20]. However, because Theorem 12.3.8 holds with high probability, we also have explicit bounds on the domain size via function error, as seen in the proof of Theorem 5.1 in [FKT20], so we can explicitly truncate our domain to have smaller domain without removing the minimizer. With this modification, the claim follows directly from Theorem 5.1 in [FKT20]. □

12.6 Optimal Algorithms in the Known Lipschitz Setting

Compared to the standard Lipschitz setting (i.e. the ∞-heavy-tailed private SCO problem), our algorithm in Section 12.3 has two downsides: it pays a polylogarithmic overhead in the utility, and it requires an extra aggregation step. In this section, assuming we are in the *known Lipschitz k-heavy-tailed* setting (see the following Assumption 12.6.1, and Definition 12.2.6), we provide a simple reduction to the standard Lipschitz setting, resulting in optimal rates.

Assumption 12.6.1 (Known Lipschitz k-heavy-tailed distributions)**.** *In the setting of Assumption 12.2.5, suppose that for each $s \in \mathcal{S}$ we know a value $\overline{L}_s \geq L_s$. For $k \in \mathbb{N}$ satisfying $k \geq 2$, we say $\mathcal{P}$ satisfies the* known Lipschitz k-heavy tailed assumption *if, for a sequence of monotonically nondecreasing $\{G_j\}_{j\in[k]}$, we have $\mathbb{E}_{s\sim\mathcal{P}}[\overline{L}_s^j] \leq G_j^j < \infty$ for all $j \in [k]$.*

Note that Assumption 12.6.1 clearly implies Assumption 12.2.5, but gives us additional access to Lipschitz overestimates with bounded moments. We require some additional definitions used throughout the section. First, we augment $\mathcal{S}$ with a designated element $s_0 \notin \mathcal{S}$, and define

$$f(x; s_0) = 0 \text{ for all } x \in \mathcal{X}. \tag{12.9}$$

We also define a truncated distribution parameterized by $C \geq 0$, where we use $f(\cdot; s_0)$ in place of sample functions with large Lipschitz overestimates, following notation of Assumption 12.6.1:

$$f^C(x; s) := \begin{cases} f(x; s) & \overline{L}_s \leq C \\ f(x; s_0) & \overline{L}_s > C \end{cases}, \; f^C(x; s_0) := f(x; s_0), \; F_{\mathcal{P}}^C(x; s) := \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[f^C(x; s)\right]. \tag{12.10}$$

We denote $\mathcal{S}_0 := \mathcal{S} \cup \{s_0\}$, and for $\mathcal{D} \in \mathcal{S}^n$, the dataset $\mathcal{D}^C \in \mathcal{S}_0^n$ replaces all $s \in \mathcal{D}$ satisfying $\overline{L}_s > C$ with s_0. We additionally provide a second reduction in the known Lipschitz heavy-tailed setting, when all sample functions are assumed to be μ-strongly convex. Because our treatments of these cases are slightly different, we use different notation when $\mu = 0$ and $\mu > 0$, for convenience of exposition. Fixing an arbitrary point $\overline{x} \in \mathcal{X}$,

for $\mu > 0$, instead of using the constant 0 function as in (12.9), we define a strongly convex alternative $f(\cdot; s_\mu)$, for a designated element s_μ:

$$f(x; s_\mu) = \frac{\mu}{2}\|x - \overline{x}\|^2, \text{ for all } x \in \mathcal{X}. \tag{12.11}$$

The truncated distribution parameterized by $C \geq \mu D$, is defined in a similar way:

$$f_\mu^C(x; s) := \begin{cases} f(x; s) & \overline{L}_s \leq C \\ f(x; s_\mu) & \overline{L}_s > C \end{cases}, \; f_\mu^C(x; s_\mu) := f(x; s_\mu), \; F_{\mathcal{P}}^{C,\mu}(x; s) := \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[f_\mu^C(x; s)\right]. \tag{12.12}$$

We denote $\mathcal{S}_\mu := \mathcal{S} \cup \{s_\mu\}$, and for $\mathcal{D} \in \mathcal{S}^n$, the dataset $\mathcal{D}_\mu^C \in \mathcal{S}_\mu^n$ replaces every $s \in \mathcal{D}$ such that $\overline{L}_s > C$ with s_μ. Our focus on the regime $C \geq \frac{\mu D}{4}$ is motivated by the following well-known claim.

Lemma 12.6.2. *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be compact and convex satisfying $\operatorname{diam}(\mathcal{X}) = D$, and suppose $f : \mathcal{X} \to \mathbb{R}$ is L-Lipschitz and μ-strongly convex. Then, $L \geq \frac{\mu D}{4}$.*

Proof. Let $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} f(x)$. By strong convexity, for all $x \in \mathcal{X}$,

$$\frac{\mu}{2}\|x - x^\star\|^2 \leq f(x) - f(x^\star) \leq L\|x - x^\star\|.$$

Now, choose x such that $\|x - x^\star\| \geq \frac{D}{2}$. To see this is always possible, let $x, x' \in \mathcal{X}$ realize $\|x - x'\| = D$; then at least one of x, x' must have distance $\geq \frac{D}{2}$ from $x^\star$ by the triangle inequality. The conclusion follows by rearranging after using our choice of x. □

In other words, if $C < \frac{\mu D}{4}$ then no sample function will survive the truncation in (12.12). Finally, we parameterize the performance of algorithms in the standard Lipschitz setting.

Definition 12.6.3 (Lipschitz private SCO algorithm)**.** We say $\mathcal{A}$ is an L-Lipschitz private SCO algorithm if it takes input $(\mathcal{D}, \rho, \mathcal{X})$, where $\mathcal{D} \in \mathcal{S}^n$ is drawn i.i.d. from $\mathcal{P}$, a distribution over $\mathcal{S}$ where every $s \in \mathcal{S}$ induces L-Lipschitz $f(\cdot; s)$ over $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{A}(\mathcal{D}, \rho, \mathcal{X}) \in \mathcal{X}$, and $\mathcal{A}$ satisfies ρ-CDP. We denote

$$\operatorname{Err}(\mathcal{A}) := \mathop{\mathbb{E}}_{\mathcal{A}}\left[F_{\mathcal{P}}\left(\mathcal{A}(\mathcal{D}, \rho, \mathcal{X})\right)\right] - \min_{x\in\mathcal{X}} F_{\mathcal{P}}(x),$$

where $F_{\mathcal{P}}(x) := \mathbb{E}_{s\sim\mathcal{P}} f(x;s)$, and denote the number of sample gradients queried by $\mathcal{A}$ by $\mathsf{N}(\mathcal{A})$. Moreover, if each Lipschitz function $f(;s)$ is μ-strongly convex over the convex domain $\mathcal{X}$, we say $\mathcal{A}$ is an L-Lipschitz, μ-strongly convex private SCO algorithm, and define $\mathrm{Err}(\mathcal{A})$, $\mathsf{N}(\mathcal{A})$ as before.

With this notation in place, we state our reduction.

Algorithm 39: $\mathsf{KnownLipReduction}(\mathcal{D}, C, \mu, \rho, \mathcal{X}, \mathcal{A})$

Input: Dataset $\mathcal{D} \in \mathcal{S}^n$, clip threshold $C \in \mathbb{R}_{\geq 0}$, strong convexity parameter $\mu \in \mathbb{R}_{\geq 0}$, privacy parameter $\rho \in \mathbb{R}_{>0}$, domain $\mathcal{X} \in \mathbb{R}^d$, C-Lipschitz private SCO algorithm $\mathcal{A}$ (if $\mu = 0$), or C-Lipschitz μ-strongly convex private SCO algorithm $\mathcal{A}$ (if $\mu > 0$)
if $\mu = 0$ **then**
  Return: $\mathcal{A}(\mathcal{D}^C, \rho, \mathcal{X})$
end
else
  Return: $\mathcal{A}(\mathcal{D}^C_\mu, \rho, \mathcal{X})$
end

We begin with a simple bound relating $F^C_{\mathcal{P}}, F^{C,\mu}_{\mathcal{P}}$ and $F_{\mathcal{P}}$.

Lemma 12.6.4. *Let $F_{\mathcal{P}}$ be defined as in Definition 12.2.6, where $\mathcal{P}$ satisfies Assumption 12.6.1, and define $F^C_{\mathcal{P}}$ as in (12.10). Then, $F_{\mathcal{P}} - F^C_{\mathcal{P}}$ is $\frac{G_k^k}{(k-1)C^{k-1}}$-Lipschitz, and $F_{\mathcal{P}} - F^{C,\mu}_{\mathcal{P}}$ is $\frac{G_k^k}{(k-1)C^{k-1}} + \frac{4G_k^{k+1}}{C^k}$-Lipschitz.*

Proof. For $s \in \mathcal{S}$, let $\pi(s) := s_0$ if $\overline{L}_s > C$, and otherwise let $\pi(s) := s$. For any $x, x' \in \mathcal{X}$, we have

$$\begin{aligned}
\left(F_{\mathcal{P}}(x) - F^C_{\mathcal{P}}(x)\right) - \left(F_{\mathcal{P}}(x') - F^C_{\mathcal{P}}(x')\right) &= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[f(x;s) - f(x;\pi(s)) - f(x';s) + f(x';\pi(s))\right] \\
&= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\left(f(x;s) - f(x';s)\right)\mathcal{I}_{\overline{L}_s > C}\right] \\
&\leq \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\overline{L}_s \left\|x - x'\right\| \mathcal{I}_{\overline{L}_s > C}\right] \leq \frac{G_k^k}{(k-1)C^{k-1}}\left\|x - x'\right\|.
\end{aligned}$$

In the second line, we used that $\pi(s) = s$ unless $\overline{L}_s > C$, in which case $f(\cdot;\pi(s)) = 0$ uniformly. The last line used the definition of $\overline{L}_s$ and Fact 12.2.8 with $X \leftarrow \overline{L}_s$, recalling Assumption 12.6.1.

Next, we analyze $F_{\mathcal{P}}^{C,\mu}$. Overloading $\pi(s) := s_\mu$ if $\overline{L}_s > C$, and letting $\pi(s) := s$ otherwise,

$$\begin{aligned}
&\left(F_{\mathcal{P}}(x) - F_{\mathcal{P}}^{C,\mu}(x)\right) - \left(F_{\mathcal{P}}(x') - F_{\mathcal{P}}^{C,\mu}(x')\right) \\
&= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[f(x;s) - f(x;\pi(s)) - f(x';s) + f(x';\pi(s))\right] \\
&= \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\left(f(x;s) - f(x';s) + f(x';s_\mu) - f(x;s_\mu)\right)\mathcal{I}_{\overline{L}_s>C}\right] \\
&\le \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[\overline{L}_s\left\|x-x'\right\|\mathcal{I}_{\overline{L}_s>C}\right] + \mathop{\mathbb{E}}_{s\sim\mathcal{P}}\left[4G_k\left\|x-x'\right\|\mathcal{I}_{\overline{L}_s>C}\right] \\
&\le \left(\frac{G_k^k}{(k-1)C^{k-1}} + \frac{4G_k^{k+1}}{C^k}\right)\left\|x-x'\right\|.
\end{aligned}$$

In the third line, we used that $\mu D \le 4G_1 \le 4G_k$ by Lemma 12.6.2 and Lemma 12.2.7 to show that $f(\cdot; s_\mu)$ is $4G_k$-Lipschitz over $\mathcal{X}$. Finally, the last line used Markov's inequality to bound $\mathbb{E}[\mathcal{I}_{\overline{L}_s>C}]$. □

Using Lemma 12.6.4, we provide a straightforward analysis of Algorithm 39.

Proposition 12.6.5. *Consider an instance of known-Lipschitz k-heavy-tailed private SCO (Definition 12.2.6), and let $\rho \ge 0$. If $\mathcal{A}$ is a C-Lipschitz private SCO algorithm (Definition 12.6.3) and $\mu = 0$, Algorithm 39 using $\mathcal{A}$ is a ρ-CDP algorithm which outputs $x \in \mathcal{X}$ satisfying*

$$\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star)\right] \le \mathrm{Err}(\mathcal{A}) + \frac{G_k^k D}{(k-1)C^{k-1}}, \text{ where } x^\star := \mathop{\mathrm{argmin}}_{x\in\mathcal{X}} F_{\mathcal{P}}(x).$$

Further, if $f(\cdot; s)$ is μ-strongly convex for all $s \in \mathcal{S}$ and $\mathcal{A}$ is a C-Lipschitz, μ-strongly convex private SCO algorithm for $\mu > 0$, Algorithm 39 using $\mathcal{A}$ is a ρ-CDP algorithm which outputs $x \in \mathcal{X}$ satisfying

$$\begin{aligned}
\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star)\right] &\le \mathrm{Err}(\mathcal{A}) \\
&+ \left(\frac{G_k^k}{(k-1)C^{k-1}} + \frac{4G_k^{k+1}}{C^k}\right)\left(\frac{2G_k^k}{\mu(k-1)C^{k-1}} + \frac{8G_k^{k+1}}{\mu C^k} + \sqrt{\frac{2}{\mu}\cdot\mathrm{Err}(\mathcal{A})}\right),
\end{aligned}$$

where $x^\star := \mathop{\mathrm{argmin}}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$.

In either case, Algorithm 39 queries $\mathsf{N}(\mathcal{A})$ *sample gradients (using samples in* $\mathcal{D}$*).*

Proof. For the first utility claim, letting $x^{\star,C} := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}^{C}(x)$, we have

$$\begin{aligned}\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star)\right] &= \mathbb{E}\left[F_{\mathcal{P}}^{C}(x) - F_{\mathcal{P}}^{C}(x^\star)\right] \\ &\quad + \mathbb{E}\left[\left(F_{\mathcal{P}}(x) - F_{\mathcal{P}}^{C}(x)\right) - \left(F_{\mathcal{P}}(x^\star) - F_{\mathcal{P}}^{C}(x^\star)\right)\right] \\ &\le \mathbb{E}\left[F_{\mathcal{P}}^{C}(x) - F_{\mathcal{P}}^{C}(x^{\star,C})\right] + \frac{G_k^k}{(k-1)C^{k-1}}\,\mathbb{E}\left[\|x - x^\star\|\right] \\ &\le \mathrm{Err}(\mathcal{A}) + \frac{G_k^k D}{(k-1)C^{k-1}},\end{aligned} \tag{12.13}$$

where the first inequality used the definition of $x^{\star,C}$ and Lemma 12.6.4, and the second used the definition of Err and $\operatorname{diam}(\mathcal{X}) = D$. For the second claim, we first have

$$\mathbb{E}\left[\frac{\mu}{2}\left\|x - x^{\star,C}\right\|^2\right] \le \mathrm{Err}(\mathcal{A})$$

by the definition of $\mathrm{Err}(\mathcal{A})$ and μ-strong convexity of $F_{\mathcal{P}}^{C,\mu}$, so that $\mathbb{E}[\left\|x - x^{\star,C}\right\|] \le (\frac{2}{\mu}\,\mathrm{Err}(\mathcal{A}))^{1/2}$ by Jensen's inequality. Moreover, we also have

$$\begin{aligned}\frac{\mu}{2}\left\|x^{\star,C} - x^\star\right\|^2 &\le F_{\mathcal{P}}(x^{\star,C}) - F_{\mathcal{P}}(x^\star) \\ &\le \left(F_{\mathcal{P}}(x^{\star,C}) - F_{\mathcal{P}}^{C}(x^{\star,C})\right) - \left(F_{\mathcal{P}}(x^\star) - F_{\mathcal{P}}^{C}(x^\star)\right) \\ &\le \left(\frac{G_k^k}{(k-1)C^{k-1}} + \frac{4G_k^{k+1}}{C^k}\right)\left\|x^{\star,C} - x^\star\right\|,\end{aligned}$$

where we use optimality of $x^{\star,C}$ in the second inequality, and Lemma 12.6.4 in the third. Combining,

$$\mathbb{E}\left\|x - x^\star\right\| \le \frac{2}{\mu}\cdot\left(\frac{G_k^k}{(k-1)C^{k-1}} + \frac{4G_k^{k+1}}{C^k}\right) + \sqrt{\frac{2}{\mu}\cdot\mathrm{Err}(\mathcal{A})},$$

and then the claim follows by substituting this bound into (12.13). □

We can use any existing optimal algorithms for DP-SCO to instantiate our reduction. In particular, we can use the algorithm of [FKT20], denoted by $\mathcal{A}_{\mathsf{Lip}}$, which has the following

guarantees. For simplicity of exposition, we focus on the case where our functions do not possess additional regularity properties e.g. smoothness, and we also focus on the simplest $\mathcal{A}_{\mathsf{Lip}}$ which attains the optimal utility bound. Because of the generality of our reduction, however, improvements can be made by using more structured or faster subroutines as $\mathcal{A}_{\mathsf{Lip}}$, such as the smooth DP-SCO algorithms of [FKT20] or the Lipschitz DP-SCO algorithms of e.g. [AFKT21, KLL21, CJJ+23], which are more query-efficient, sometimes at the cost of logarithmic factors in the utility (in the case of [CJJ+23]).

Proposition 12.6.6. *Let $\mathcal{P}$ be a distribution over $\mathcal{S}$ such that $f(\cdot; s)$ is L-Lipschitz and convex for all $s \in \mathcal{S}$. There exists a constant C_{Lip} such that given $\mathcal{D} \sim \mathcal{S}^n$, the algorithm $\mathcal{A}_{\mathsf{Lip}}$ is ρ-CDP and outputs x_{priv} such that, for a universal constant C_{Lip}, letting $x^\star := \mathrm{argmin}_{x \in \mathcal{X}} F_{\mathcal{P}}(x)$,*

$$\mathbb{E}[F_{\mathcal{P}}(x_{\mathrm{priv}}) - F_{\mathcal{P}}(x^\star)] \le C_{\mathsf{Lip}} \cdot \left(\frac{G_2 D}{\sqrt{n}} + \frac{LD\sqrt{d}}{n\sqrt{\rho}} \right),$$

and $\mathcal{A}_{\mathsf{Lip}}$ queries $\le C_{\mathsf{Lip}} \max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$), where G_2 is defined as in Assumption 12.2.5. Moreover, if $f(\cdot; s)$ is μ-strongly convex for all $s \in \mathcal{S}$, then

$$\mathbb{E}[F_{\mathcal{P}}(x_{\mathrm{priv}}) - F_{\mathcal{P}}(x^\star)] \le C_{\mathsf{Lip}} \cdot \left(\frac{G_2^2}{\mu n} + \frac{L^2 d}{\mu n^2 \rho} \right),$$

and $\mathcal{A}_{\mathsf{Lip}}$ queries $\le C_{\mathsf{Lip}} \max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$).

Proof. This follows from developments in [FKT20], but we briefly explain any discrepancies. The $\mu = 0$ case applies Theorem 4.8 in [FKT20], where for simplicity we consider the full-batch variant which does not subsample.[7] Moreover, Theorem 4.8 in [FKT20] is stated with a dependence on L rather than G_2 on the $n^{-1/2}$ term, but inspecting the proof shows it only uses a second moment bound. The $\mu > 0$ case follows from Theorem 5.1 of [FKT20], using Theorem 4.8 as a subroutine. □

We are now ready to present our main result in this section, using our reduction with

[7] The subsampled variant only satisfies a weaker variant of CDP called truncated CDP, with an upside of using n times fewer sample gradient queries, but this is less comparable to the lower bounds in [LR23].

$\mathcal{A}_{\mathsf{Lip}}$.

Theorem 12.6.7. *Consider an instance of known-Lipschitz k-heavy-tailed private SCO (Definition 12.2.6), let $\rho \geq 0$, and let $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$. Algorithm 39 with $C \leftarrow G_k(\frac{n\sqrt{\rho}}{\sqrt{d}})^{\frac{1}{k}}$ using $\mathcal{A}_{\mathsf{Lip}}$ in Proposition 12.6.6 is a ρ-CDP algorithm which outputs $x \in \mathcal{X}$ satisfying, for a universal constant C_{HT},*

$$\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star)\right] \leq C_{\mathrm{HT}}\left(\frac{G_2 D}{\sqrt{n}} + G_k D \cdot \left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}\right),$$

querying $\leq C_{\mathrm{HT}} \max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$). Further, if $f(\cdot; s)$ is μ-strongly convex for all $s \in \mathcal{S}$, Algorithm 39 with $C \leftarrow G_k(\frac{n^2\rho}{d})^{\frac{1}{2k}}$ using $\mathcal{A}_{\mathsf{Lip}}$ in Proposition 12.6.6 is a ρ-CDP algorithm which outputs $x \in \mathcal{X}$ satisfying

$$\mathbb{E}\left[F_{\mathcal{P}}(x) - F_{\mathcal{P}}(x^\star)\right] \leq C_{\mathrm{HT}}\left(\frac{G_2^2}{\mu n} + \frac{G_k^2}{\mu} \cdot \left(\frac{d}{n^2\rho}\right)^{1-\frac{1}{k}}\right),$$

querying $\leq C_{\mathrm{HT}} \max(n^2, \frac{n^3\rho}{d})$ sample gradients (using samples in $\mathcal{D}$).

Proof. Throughout the proof, assume without loss of generality that $d \leq n^2\rho$, as otherwise all stated bounds are vacuous since the additive function value range over $\mathcal{X}$ is at most $G_1 D \leq \frac{4G_1^2}{\mu}$ by Lemma 12.6.2 and Lemma 12.2.7. This also implies that $C \geq G_k$ in either case.

In the $\mu = 0$ case, Proposition 12.6.5 and the guarantees of $\mathcal{A}_{\mathsf{Lip}}$ in Proposition 12.6.6 imply that

$$\begin{aligned}
\mathbb{E}\left[F_{\mathcal{P}}(\widehat{x}) - F_{\mathcal{P}}(x\mathrm{OPT})\right] &\leq \mathrm{Err}(\mathcal{A}_{\mathsf{Lip}}) + \frac{G_k^k D}{C^{k-1}} \\
&\leq C_{\mathsf{Lip}} \cdot \left(\frac{G_2 D}{\sqrt{n}} + \frac{CD\sqrt{d}}{n\sqrt{\rho}}\right) + \frac{G_k^k D}{C^{k-1}} \\
&\leq (C_{\mathsf{Lip}} + 2)\left(\frac{G_2 D}{\sqrt{n}} + G_k D \cdot \left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}\right),
\end{aligned}$$

where the last inequality follows from our choice of C. Next, we consider $\mu > 0$. Propo-

sition 12.6.5 and the guarantees of $\mathcal{A}_{\mathsf{Lip}}$ in Proposition 12.6.6 for this case imply that, assuming $C_{\mathsf{Lip}} \geq 2$ without loss,

$$\begin{aligned}\mathbb{E}\left[F_{\mathcal{P}}(\widehat{x}) - F_{\mathcal{P}}(x\mathrm{OPT})\right] &\leq \mathrm{Err}(\mathcal{A}_{\mathsf{Lip}}) + \frac{5G_k^k}{C^{k-1}}\left(\sqrt{\frac{2\,\mathrm{Err}(n,d,\rho,C,D)}{\mu}} + \frac{10G_k^k}{\mu C^{k-1}}\right) \\ &\leq C_{\mathsf{Lip}} \cdot \left(\frac{G_2^2}{\mu n} + \frac{C^2 d}{\mu n^2 \rho} + \frac{5G_k^k}{C^{k-1}}\left(\frac{G_2}{\mu\sqrt{n}} + \frac{C\sqrt{d}}{\mu n\sqrt{\rho}} + \frac{10G_k^k}{\mu C^{k-1}}\right)\right) \\ &\leq (C_{\mathsf{Lip}} + 61) \cdot \left(\frac{G_2^2}{\mu n} + \frac{G_k^2}{\mu} \cdot \left(\frac{d}{n^2\rho}\right)^{1-\frac{1}{k}}\right),\end{aligned}$$

where we used $C \geq G_k$ to simplify bounds, and applied our choice of C. □

12.7 Fast Algorithms for Smooth Functions

In this section, we develop a linear-time algorithm for the smooth setting where we additionally assume $f(\cdot; s)$ is β-smooth for all $s \in \mathcal{S}$. Our algorithm attains nearly-optimal rates for a sufficiently small value of β, and is based on the localization framework of [FKT20]. To apply this framework, we show that a variant of clipped DP-SGD (see Algorithm 40) is stable in the heavy-tailed setting with high probability. We then ensure that stability holds for any input dataset (not necessarily sampled from a distribution P), by using the sparse vector technique [DR14] to verify that the number of clipped gradients is not too large. In Appendix 12.7.1, we provide some standard preliminary results from the literature. We use these results in Appendix 12.7.2, where we state our algorithm in full as Algorithm 42 and analyze it in Theorem 12.7.6, the main result of this section.

12.7.1 Helper tools

First, we state a standard bound on the contractivity of smooth gradient descent iterations.

Fact 12.7.1 (Lemma 3.7, [HRS16]). *Let $f : \mathcal{X} \to \mathbb{R}$ be β-smooth, and let $\eta \leq \frac{2}{\beta}$. Then for any $x, x' \in \mathcal{X}$,*

$$\left\|(x - x') - \eta(\nabla f(x) - \nabla f(x'))\right\| \leq \left\|x - x'\right\|.$$

Next, we provide a standard utility bound on a one-pass SGD algorithm using clipped

gradients.

Algorithm 40: OnePass-Clipped-SGD$(\mathcal{D}, C, \eta, T, \mathcal{X}, x_0)$

Input: Dataset $\mathcal{D} = \{s_t\}_{t\in[T]} \in \mathcal{S}^T$, clip threshold $C \in \mathbb{R}_{\geq 0}$, step size $\eta \in \mathbb{R}_{\geq 0}$, iteration count $T \in \mathbb{N}$, domain $\mathcal{X} \subset \mathbb{B}(x_0, D)$ for $x_0 \in \mathcal{X}$
for $0 \leq t < T$ **do**
  $x_{t+1} \leftarrow \operatorname{argmin}_{x\in\mathcal{X}}\{\eta \langle \Pi_C(\nabla f(x_t; s_{t+1})), x\rangle + \frac{1}{2}\|x - x_t\|^2\}$
end
Return: $\widehat{x} \leftarrow \frac{1}{T}\sum_{0\leq t<T} x_t$

Lemma 12.7.2. *Consider an instance of k-heavy-tailed private SCO, following notation in Definition 12.2.6, and let $u \in \mathcal{X}$ be independent of $\mathcal{D}$. Assuming $\mathcal{D} \sim \mathcal{P}^T$ i.i.d., Algorithm 40 outputs $\widehat{x} \in \mathcal{X}$ satisfying*

$$\mathbb{E}\left[F_{\mathcal{P}}(\widehat{x}) - F_{\mathcal{P}}(u)\right] \leq \frac{\|x_0 - u\|^2}{2\eta T} + \frac{\eta G_2^2}{2} + \frac{G_k^k D}{(k-1)C^{k-1}}.$$

Proof. To simplify notation, let $g_t := \nabla f(x_t; s_{t+1})$ for all $0 \leq t < T$, and let $\widehat{g}_t := \mathcal{T}_C(g_t)$. Because $s_{t+1} \sim \mathcal{P}$ is independent of x_t, we have that $\mathbb{E}\, g_t = \nabla F_{\mathcal{P}}(x_t)$. Therefore, in iteration t,

$$\begin{aligned} F_{\mathcal{P}}(x_t) - F_{\mathcal{P}}(u) &= \mathbb{E}\left[\langle g_t, x_t - u\rangle\right] \\ &\leq \mathbb{E}\left[\langle \widehat{g}_t, x_t - u\rangle + \|g_t - \widehat{g}_t\|\, D\right] \\ &\leq \mathbb{E}\left[\frac{1}{2}\|x_t - u\|^2 - \frac{1}{2}\|x_{t+1} - u\|^2 + \frac{\eta G_2^2}{2}\right] + \frac{G_k^k D}{(k-1)C^{k-1}}, \end{aligned} \tag{12.14}$$

where all expectations are conditional on the first t iterations of the algorithm, and taken over the randomness of s_{t+1}. In the third line, we used the first-order optimality condition on x_{t+1}, applied Fact 12.2.8 to bound $\mathbb{E}\,\|g_t - \widehat{g}_t\|$, and used

$$\mathbb{E}\,\|\widehat{g}_t\|^2 \leq \mathbb{E}\,\|g_t\|^2 \leq G_2^2. \tag{12.15}$$

Summing across all iterations and dividing by T yields the result upon iterating expectations. □

We also note the following straightforward generalization of Lemma 12.7.2 to the case of randomized clipping thresholds, which is used in our later development.

Corollary 12.7.3. *For $C, \widehat{C} \geq 0$ and $g \in \mathbb{R}^d$, define the operation*

$$\Pi_{C,\widehat{C}}(g) := \begin{cases} \Pi_C(g) & \|g\| \geq \widehat{C} \\ g & \text{else} \end{cases}.$$

If Algorithm 40 is run with $\Pi_C(\nabla f(x_t; s_{t+1}))$ replaced by $\Pi_{C,\widehat{C}_t}(\nabla f(x_t; s_{t+1}))$ where $\widehat{C}_t$ is independent of s_{t+1} and satisfies $\widehat{C}_t \geq \frac{C}{2}$ for all $0 \leq t < T$, then following notation in Lemma 12.7.2,

$$\mathbb{E}\left[F_{\mathcal{P}}(\widehat{x}) - F_{\mathcal{P}}(u)\right] \leq \frac{\|x_0 - u\|^2}{2\eta T} + 2\eta G_2^2 + \frac{G_k^k D}{(k-1)(\frac{C}{2})^{k-1}}.$$

Proof. For a fixed iteration $0 \leq t < T$, the calculation (12.15) changes in two ways. First, in place of the variance bound (12.15) (which used $\|\widehat{g}_t\| \leq \|g_t\|$ deterministically), when using the modified clipping operators we require the modified deterministic bound

$$\|\widehat{g}_t\| \leq 2\|g_t\|,$$

which follows because $\|\widehat{g}_t\| \neq \|g_t\|$ (which implies $C = \|\widehat{g}_t\|$) only if $\|g_t\| \geq \frac{C}{2}$. Moreover, in place of the bias bound $\mathbb{E}\|g_t - \widehat{g}_t\| \leq \frac{G_k^k}{(k-1)C^{k-1}}$ which followed from Fact 12.2.8, we instead have

$$\mathbb{E}\left\|\Pi_{C,\widehat{C}_t}(g_t) - g_t\right\| = \mathbb{E}\left[\left|\frac{C}{\|g_t\|} - 1\right| \|g_t\| \mathcal{I}_{\|g_t\| \geq \max(\widehat{C}_t, C)}\right] \leq \mathbb{E}\left[\|g_t\| \mathcal{I}_{\|g_t\| \geq \frac{C}{2}}\right] \leq \frac{G_k^k}{(k-1)(\frac{C}{2})^{k-1}}.$$

The conclusion follows by adjusting these constants appropriately in Lemma 12.7.2. □

Next, for $R, \tau \geq 0$, we let $\mathrm{BLap}(R, \tau)$ denote the bounded Laplace distribution with scale parameter R and truncation threshold τ be defined as the conditional distribution of $\xi \sim \mathrm{Lap}(R)$ on the event $|\xi| \leq \tau$ (recall that $\mathrm{Lap}(R)$ has a density function $\propto \exp(-\frac{1}{R}|\xi|)$).

It is a standard calculation that

$$\Pr_{\xi\sim\mathrm{Lap}(R)}\left[|\xi| \le R\log\left(\frac{1}{\delta}\right)\right] = 1-\delta, \tag{12.16}$$

so that the total variation distance between $\mathrm{Lap}(R)$ and $\mathrm{BLap}(R, R\log(\frac{1}{\delta}))$ is δ. We hence have the following bounded generalization of the privacy given by the Laplace mechanism.

Lemma 12.7.4. *Let $\varepsilon, \delta \in (0,1)$. If $S(\mathcal{D}) \in \mathbb{R}$ is a Δ-sensitive statistic of the dataset $\mathcal{D}$, i.e. for neighboring datasets $\mathcal{D}, \mathcal{D}'$ we have that $|S(\mathcal{D}) - S(\mathcal{D}')| \le \Delta$, then the* bounded Laplace *mechanism which outputs $S(\mathcal{D}) + \xi$ where $\xi \sim \mathrm{BLap}(\frac{\Delta}{\varepsilon}, \tau)$ for any $\tau \ge \frac{\Delta}{\varepsilon}\log(\frac{4}{\delta})$ satisfies (ε, δ)-DP.*

Proof. For notational simplicity, let $\mathcal{A}$ denote the Laplace mechanism (which samples $\xi \sim \mathrm{Lap}(\frac{\Delta}{\varepsilon})$ instead of $\mathrm{BLap}(\frac{\Delta}{\varepsilon}, \tau)$), let $\overline{\mathcal{A}}$ denote the bounded Laplace mechanism, and let $\mathcal{E} \subseteq \mathbb{R}$ be an event in the outcome space. By standard guarantees on $(\varepsilon, 0)$-DP of $\mathcal{A}$ (e.g. Theorem 3.6, [DR14]),

$$\begin{aligned}\Pr\left[\overline{\mathcal{A}}(\mathcal{D}) \in \mathcal{E}\right] &\le \Pr\left[\mathcal{A}(\mathcal{D}) \in \mathcal{E}\right] + \frac{\delta}{4} \\ &\le \exp(\varepsilon)\Pr\left[\mathcal{A}(\mathcal{D}') \in \mathcal{E}\right] + \frac{\delta}{4} \le \exp(\varepsilon)\Pr\left[\overline{\mathcal{A}}(\mathcal{D}') \in \mathcal{E}\right] + \delta,\end{aligned} \tag{12.17}$$

for any neighboring datasets, where we used $\exp(\varepsilon) \le 3$ and that the total variation distance between $(\mathcal{A}(\mathcal{D}), \overline{\mathcal{A}}(\mathcal{D}))$ and $(\mathcal{A}(\mathcal{D}'), \overline{\mathcal{A}}(\mathcal{D}'))$ are bounded by $\frac{\delta}{4}$ by (12.16). □

We also use the sparse vector technique (SVT) [DR14], which has been used recently in private optimization in the user-level setting [AL24]. Given an input dataset $\mathcal{D} = \{s_i\}_{i\in[n]} \in \mathcal{S}^n$, SVT takes a stream of queries $q_1, q_2, \ldots, q_T : \mathcal{D} \to \mathbb{R}$ in an online manner. We assume each q_i is Δ-sensitive, i.e. $|q_i(\mathcal{D}) - q_i(\mathcal{D}')| \le \Delta$ for neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{S}^n$. One notable difference is that our SVT algorithm will use the bounded Laplace mechanism, rather than the Laplace mechanism, but this distinction is handled similarly to Lemma 12.7.4. We provide a guarantee on this variant of SVT in Lemma 12.7.5, and pseudocode is provided as Algorithm 41.

Algorithm 41: $\mathsf{SVT}(\mathcal{D}, \{q_i\}_{i\in[T]}, c, L, R, \tau)$

Input: Dataset $\mathcal{D} = \{s_t\}_{t\in[n]} \in \mathcal{S}^n$, Δ-sensitive queries $\{q_i : \mathcal{S}^n \to \mathbb{R}\}_{i\in[T]}$, count threshold $c \in \mathbb{N}$, query threshold $L \in \mathbb{R}$, scale parameter $R \in \mathbb{R}_{\geq 0}$, truncation threshold $\tau \in \mathbb{R}_{\geq 0}$
$i \leftarrow 1$, count $\leftarrow 0$
$b \leftarrow L + \xi$ for $\xi \sim \mathrm{BLap}(R, \tau)$
while $i \in [T]$ **and** count $< c$ **do**
  $\xi \sim \mathrm{BLap}(2R, 2\tau)$
  if $q_i(\mathcal{D}) + \xi < b$ **then**
    Output: $a_i \leftarrow \perp$
    $i \leftarrow i + 1$
  end
  else
    Output: $a_i \leftarrow \top$
    $i \leftarrow i + 1$, count $\leftarrow$ count $+ 1$
    $b \leftarrow L + \xi$ for $\xi \sim \mathrm{BLap}(R, \tau)$
  end
end
Halt

Lemma 12.7.5. *Let $\delta, \varepsilon \in (0, 1)$ and suppose*

$$R \geq \frac{6\Delta}{\varepsilon}\sqrt{c \log\left(\frac{5}{\delta}\right)}, \ \tau \geq R \log\left(\frac{10T}{\delta}\right) \tag{12.18}$$

Algorithm 41 outputs a sequence of answers $\{a_i \in \{\perp, \top\}\}_{i\in[k]}$ for some $k \in [T]$, and is (ε, δ)-DP.

Proof. The proof is analogous to Lemma 12.7.4. Let $\mathcal{A}$ denote SVT run with Laplace noise in place of bounded Laplace noise (i.e. $\tau = \infty$), and let $\overline{\mathcal{A}}$ denote SVT run with bounded Laplace noise. We first claim that $\mathcal{A}$ is $(\varepsilon, \frac{\delta}{5})$-DP, which is immediate from Theorem 3.23 and Theorem 3.20 in [DR14].

Next, by a union bound on all of the $\leq 2T$ random variables sampled, the total variation distance between $(\mathcal{A}(\mathcal{D}), \overline{\mathcal{A}}(\mathcal{D}))$ for any dataset $\mathcal{D}$ is bounded by $\frac{\delta}{5}$. Then, for neighboring

datasets $\mathcal{D}, \mathcal{D}'$ and some event $\mathcal{E}$ in the outcome space, repeating the calculation (12.17),

$$\begin{aligned}\Pr\left[\overline{\mathcal{A}}(\mathcal{D}) \in \mathcal{E}\right] &\leq \Pr\left[\mathcal{A}(\mathcal{D}) \in \mathcal{E}\right] + \frac{\delta}{5} \\ &\leq \exp(\varepsilon) \Pr\left[\mathcal{A}(\mathcal{D}') \in \mathcal{E}\right] + \frac{\delta}{5} + \frac{\delta}{5} \leq \exp(\varepsilon) \Pr\left[\overline{\mathcal{A}}(\mathcal{D}') \in \mathcal{E}\right] + \delta.\end{aligned}$$

□

12.7.2 Algorithm statement and analysis

In this section, we present the full details of our algorithm (see Algorithm 42) and prove its corresponding guarantees, separating out the privacy analysis and utility analysis.

Algorithm 42: Localized-Clipped-DP-SGD$(\mathcal{D}, x_0, \eta, c, \varepsilon, \delta)$

Input: Dataset $\mathcal{D} \in \mathcal{S}^n$, initial point $x_0 \in \mathcal{X}$, step size $\eta \in \mathbb{R}_{>0}$, parameters $C, c, \omega \in \mathbb{R}_{>0}$, privacy parameters $(\varepsilon, \delta) \in \mathbb{R}^2_{>0}$

$I \leftarrow \lfloor \log_2 n \rfloor$, $n \leftarrow 2^I$

for $i \in [I]$ **do**

  $n_i \leftarrow \frac{n}{2^i}$, $\eta_i \leftarrow \frac{\eta}{4^i}$, $\omega_i \leftarrow \omega \cdot 6C\eta_i\beta$

  $\widehat{C} \leftarrow C + \mathrm{BLap}(\omega_i, \omega_i \log(\frac{30n_i}{\delta}))$, $\widehat{c}_i \leftarrow c + \mathrm{BLap}(\frac{3}{\varepsilon}, \frac{c}{2})$, count $\leftarrow 0$

  $x_{i,1} \leftarrow x_{i-1}$

  for $j \in [n_i]$ **do**

    $s_{i,j} \leftarrow (\sum_{i' \in [i]} n_{i'} + j)^{\text{th}}$ element of $\mathcal{D}$

    $\nu_{i,j} \sim \mathrm{BLap}(2\omega_i, 2\omega_i \log(\frac{30n_i}{\delta}))$

    if $\|\nabla f(x_{i,j}; s_{i,j})\| + \nu_{i,j} \geq \widehat{C}$ **then**

      count $\leftarrow$ count $+ 1$

      $g_{i,j} \leftarrow \Pi_C(\nabla f(x_{i,j}; s_{i,j}))$

      $\widehat{C} \leftarrow C + \mathrm{BLap}(\omega_i, \omega_i \log(\frac{30n_i}{\delta}))$

    end

    else

      $g_{i,j} \leftarrow \nabla f(x_{i,j}; s_{i,j})$

    end

    if count $\geq \widehat{c}_i$ **then**

      Return: $\perp$

    end

    $x_{i,j+1} \leftarrow \Pi_{\mathcal{X}}(x_{i,j} - \eta_i g_{i,j})$

  end

  $\overline{x}_i \leftarrow \frac{1}{n_i} \sum_{j \in [n_i]} x_{i,j}$

  $x_i \leftarrow \overline{x}_i + \zeta_i$, where $\zeta_i \sim \mathcal{N}(0, \sigma_i^2 \mathbf{I}_d)$ with $\sigma_i = \frac{30C\eta_i\sqrt{\log(3/\delta)}}{\varepsilon}$

end

Return: x_I

The following theorem summarizes the guarantees of Algorithm 42.

Theorem 12.7.6. *Consider an instance of k-heavy-tailed private SCO, following notation in Definition 12.2.6, and let $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} F_{\mathcal{P}}(x)$, and $\varepsilon, \delta \in (0,1)$. Algorithm 42 run with parameters*

$$\eta \leftarrow \min\left(\sqrt{\frac{4}{n}}\cdot\frac{D}{G_2}, \frac{DI}{G_k n}\cdot\left(\frac{n^2\varepsilon^2}{14400 d\log^2(\frac{15n}{\delta})}\right)^{\frac{k-1}{2k}}\right),$$

$$C \leftarrow 2\left(\frac{G_k^k DIn\varepsilon^2}{14400 d\eta\log^2(\frac{15n}{\delta})}\right)^{\frac{1}{k+1}},\ c \leftarrow \frac{240\sqrt{d}\log(\frac{15n}{\delta})}{\varepsilon},\ \omega \leftarrow \frac{18}{\varepsilon}\sqrt{2c\log\left(\frac{15}{\delta}\right)},$$

is (ε,δ)-DP and outputs x_I that satisfies, for a universal constant C_{smooth},

$$\mathbb{E}\left[F_{\mathcal{P}}(x_k) - F_{\mathcal{P}}(x^\star)\right] \le C_{\text{smooth}}\left(\frac{G_2 D}{\sqrt{n}} + G_k D\cdot\left(\frac{\sqrt{d\log^3(\frac{n}{\delta})}}{n\varepsilon}\right)^{1-\frac{1}{k}}\right),$$

assuming $f(\cdot;s)$ is β-smooth for all $s\in\mathcal{S}$, where

$$\begin{aligned}\beta &\le \frac{\varepsilon^{1.5}}{24000\eta\sqrt{d}\log^2(\frac{30n}{\delta})}\\ &= \Theta\left(\max\left(\frac{G_2}{D}\cdot\frac{\sqrt{n}\varepsilon^{1.5}}{\sqrt{d}\log^2(\frac{n}{\delta})}, \frac{G_k}{D}\cdot\frac{\varepsilon^{1.5}n}{\sqrt{d}\log(n)\log^2(\frac{n}{\delta})}\cdot\left(\frac{d\log^2(\frac{n}{\delta})}{n^2\varepsilon^2}\right)^{\frac{k-1}{2k}}\right)\right).\end{aligned} \tag{12.19}$$

We now proceed to prove Theorem 12.7.6.

Privacy proof overview. We first overview the structure of our privacy proof. Consider two neighboring datasets $\mathcal{D}, \mathcal{D}'$ that differ on a single sample $s_{i,j_0} \neq s'_{i,j_0}$. The core argument used to prove privacy is controlling the total number of times when gradients are clipped, so we introduce the variable "count." Note that we have $\|x_{i,j_0+1} - x'_{i,j_0+1}\| = O(C\eta)$ due to the clip operation. If no clip ever happened afterward, then we know $\|x_{i,n_i} - x'_{i,n_i}\| \le \|x_{i,j_0+1} - x'_{i,j_0+1}\| = O(C\eta)$ due to our smoothness assumption (see Fact 12.7.1), which means the algorithm is private. When count is not too large, we can still bound the sensitivity between $\|x_{i,n_i} - x'_{i,n_i}\|$ by $O(C\eta)$. However, when the value of count is larger, there is a risk that the sensitivity of x_{i,n_i} is not bounded as before, and hence we halt the algorithm when count

exceeds some appropriate cutoff point $\widehat{c}_i$.

One subtle difference between our algorithm and standard uses of SVT is that we add Laplace noise to the cutoff point c to obtain a randomized cutoff $\widehat{c}_i$. This is because the sensitivity of the count increment at the j_0^{th} iteration of phase i is bounded by one, even though $\|\nabla f(x_{i,j_0}; s_{i,j_0})\| - \|\nabla f(x'_{i,j_0}; s'_{i,j_0})\|$ can be arbitrarily large. The guarantees of the bounded Laplace mechanism imply that the noise added in $\widehat{c}_i$ hence suffices to privatize count.

In summary, we can control the sensitivity between $\|x_{i,j} - x'_{i,j}\|$ for all j due to the termination condition in Line 18 and our use of bounded Laplace noise, and hence can control the sensitivity of the query for $\|\nabla f(x_{i,j}; s_{i,j})\| - \|\nabla f(x'_{i,j}; s'_{i,j})\|$ for all $j \neq j_0$. By adding Laplace noise on the cutoff c, we handle the issue of the sensitivity of the j_0^{th} query $\|\nabla f(x_{i,j_0}; s_{i,j_0})\|$ being unbounded. If the algorithm succeeds and returns x_k, we know the sensitivity $\|x_{i,n_i} - x'_{i,n_i}\|$ is $O(C\eta_i)$ and the privacy guarantee follows from the Gaussian mechanism. If the algorithm fails and outputs $\perp$, the privacy guarantee follows from the bounded Laplace noise on the cutoff point and the guarantees of SVT.

Privacy proof. We now provide our formal privacy analysis following this overview. To fix notation in the remainder of the privacy proof, we consider running Algorithm 42 on two neighboring datasets $\mathcal{D}, \mathcal{D}'$ that differ on a single sample $s_{i,j_0} \neq s'_{i,j_0}$, for some $i \in [I]$. By standard postprocessing properties of differential privacy, it suffices to argue that the i^{th} phase (i.e. the run of the loop in Lines 3 to 25 corresponding to this value of i) is private, so we fix $i \in [I]$ in the following discussion.

We let $\{x_{i,j}\}_{j\in[n_i]}$ and $\{x'_{i,j}\}_{j\in[n_i]}$ be the iterates of the i^{th} phase of Algorithm 42 using $\mathcal{D}$ and $\mathcal{D}'$, and we let $Y_{i,j}$ and $Y'_{i,j}$ be the respective 0-1 indicator variables that count increases by 1 in iteration j. We also let count_j and count'_j denote the values of count at the end of the j^{th} iteration, and abusing notation we let $\widehat{c}_i$, $\widehat{c}'_i$ be the values of $\widehat{c}_i$ in the i^{th} phase when using $\mathcal{D}$ or $\mathcal{D}'$ respectively. Finally, we denote $\overline{x}_i := \frac{1}{n_i}\sum_{j\in[n_i]} x_{i,j}$ and let $\overline{x}'_i$ denote the average iterate using $\mathcal{D}'$ similarly.

We first bound the sensitivity between the iterates $\{x_{i,j}\}_{j\in[n_i]}$ and $\{x'_{i,j}\}_{j\in[n_i]}$ in the following lemma, assuming count_j and $\text{count}_{j'}$ are bounded. The proof is deferred to Sec-

tion 12.12.

Lemma 12.7.7. *Let $t \in [n_i]$, and suppose that $192\eta_i\beta c \le 1$ and $C \ge 8\omega_i \log(\frac{30n_i}{\delta})$. If $\mathrm{count}_t < \widehat{c}_i$, $\mathrm{count}'_t < \widehat{c}'_i$, and $Y_{i,j} = Y'_{i,j}$ for all $j < t$ with $j \ne j_0$, then*

$$\|x_{i,t} - x'_{i,t}\| \le 6C\eta_i.$$

Using this bound on the sensitivity, we are now ready to prove privacy of the algorithm.

Lemma 12.7.8. *Algorithm 42 is (ε, δ)-DP if it is run with parameters satisfying*

$$C \ge 8\omega_i \log\left(\frac{30n_i}{\delta}\right),\ c \ge \frac{6}{\varepsilon}\log\left(\frac{12}{\delta}\right),\ \omega \ge \frac{18}{\varepsilon}\sqrt{2c\log\left(\frac{15}{\delta}\right)},\ 192\eta_i\beta c \le 1.$$

Proof. Recall our assumption that $\mathcal{D}$ and $\mathcal{D}'$ only differ in s_{i,j_0}, the j_0^{th} sample used in the i^{th} phase of the algorithm. The privacy of all phases of the algorithm other than phase i is immediate from postprocessing properties of DP, so it suffices to argue that phase i is (ε, δ)-DP. Note also that the conditions of Lemma 12.7.7 are met after reparameterizing $\delta \leftarrow \frac{\delta}{4}$. We split our privacy argument into two cases, depending on whether the algorithm terminates on Line 18 or Line 26.

Termination on Line 18. We begin with the case where the algorithm outputs $\perp$. We introduce some simplifying notation. For iterations $S \subseteq [n_i]$, define $W_S := \{Y_{i,j}\}_{j \in S}$ to be the 0-1 indicator variables for whether count incremented on iterations $j \in S$ (when run on $\mathcal{D}$), and define $[W]_S := \sum_{j \in S} Y_{i,j}$ to be their sum. Similarly, define W'_S and $[W']_S$ for when the algorithm is run on $\mathcal{D}'$. Observe that the algorithm outputs $\perp$ iff the following event occurs:

$$Y_{i,j_0} + [W]_{[n_i]\setminus\{j_0\}} \ge \widehat{c}_i \iff (Y_{i,j_0} - \widehat{c}_i) + [W]_{[n_i]\setminus[j_0]} \ge -[W]_{[j_0-1]}.$$

The right-hand side $-[W]_{[j_0-1]}$ is independent of whether the dataset used was $\mathcal{D}$ or $\mathcal{D}'$, so it suffices to argue about the privacy loss of the random variables $Y_{i,j_0} - \widehat{c}_i$ and $W_{[n_i]\setminus[j_0]}$ as a function of the dataset used. First, $Y_{i,j_0} - c$ is clearly a 1-sensitive statistic, so Lemma 12.7.4 implies $Y_{i,j_0} - \widehat{c}_i$ is $(\frac{\varepsilon}{3}, \frac{\delta}{3})$-indistinguishable as a function of the dataset used. Next, conditioning on the value of $Y_{i,j_0} - \widehat{c}_i$, the random variable $W_{[n_i]\setminus[j_0]}$ is an instance of Algorithm 41

run with a fixed threshold $\widehat{c}_i - Y_{i,j_0} - [W]_{[j_0-1]} \le 2c$, where we rename the output variables $\{\perp, \top\}$ to $\{0, 1\}$. Moreover, Lemma 12.7.7 and smoothness of each sample function implies that the sensitivity of each query $\|\nabla f(\cdot; s_{i,j})\|$ is bounded by $\Delta := 6C\eta_i\beta$. Therefore, Lemma 12.7.5 shows that $W_{[n_i]\setminus[j_0]}$ is $(\frac{\varepsilon}{3}, \frac{\delta}{3})$-indistinguishable, where we note that we adjusted constants appropriately in ω and the failure probabilities everywhere. By basic composition of DP, this implies $Y_{i,j_0} - \widehat{c}_i + [W]_{[n_i]\setminus[j_0]}$ (a postprocessing of $Y_{i,j_0} - \widehat{c}_i$ and $W_{[n_i]\setminus[j_0]} \mid Y_{i,j_0} - \widehat{c}_i$) is $(\frac{2\varepsilon}{3}, \frac{2\delta}{3})$-DP, as required.

Termination on Line 26. Finally, we argue about the privacy when the algorithm does not terminate on Line 18. As before, the sensitivity of $\overline{x}_i$ is bounded by $6C\eta_i$ via Lemma 12.7.7 and the triangle inequality, conditioned on a $(\frac{2\varepsilon}{3}, \frac{2\delta}{3})$-indistinguishable event (i.e. the values of $Y_{i,j_0} - \widehat{c}_i$ and $W_{[n_i]\setminus[j_0]} \mid Y_{i,j_0} - \widehat{c}_i$). Then x_i is $(\frac{\varepsilon}{3}, \frac{\delta}{3})$-indistinguishable by standard bounds on the Gaussian mechanism (Theorem A.1, [DR14]), which completes the proof upon applying basic composition. □

Utility proof. The utility proof follows the standard analysis of localized SGD algorithms and a specialized analysis of clipped SGD (Corollary 12.7.3). We first state a utility guarantee in each phase.

Lemma 12.7.9. *Following notation in Algorithm 42, fix $i \in [I]$, and suppose $\mathcal{D} \sim \mathcal{P}^n$ i.i.d. where $\mathcal{P}$ satisfies Assumption 12.2.5. For any $x \in \mathcal{X}$, if $C \ge 8\omega_i \log(\frac{30n_i}{\delta})$ and $\frac{c}{4} \ge \max(n \cdot (\frac{2G_k}{C})^k, 6\log(n))$,*

$$\mathbb{E}[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(x)] \le \frac{\|x - x_{i-1}\|^2}{2\eta_i n_i} + 2\eta_i G_2^2 + \frac{G_k^k D}{(k-1)(\frac{C}{2})^{k-1}} + \frac{G_2 D}{n^2}.$$

Proof. By Markov's inequality, $\mathbb{E}_{s\sim\mathcal{P}}[\mathcal{I}_{L_s > \frac{C}{2}}] \le (\frac{2G_k}{C})^k$, so the total number of expected samples with $L_s > \frac{C}{2}$ is at most $\frac{c}{4}$. Hence by applying a Chernoff bound,

$$\Pr_{\mathcal{D}\sim\mathcal{P}^n}\left[\underbrace{\sum_{s\in\mathcal{D}} \mathcal{I}_{L_s > \frac{C}{2}} \le \frac{c}{2}}_{:=\mathcal{E}}\right] \ge 1 - \frac{1}{n^2}.$$

Conditional on $\mathcal{E}$, the algorithm will not halt (i.e., return $\perp$) and is running one-pass clipped-SGD (Algorithm 40) using the modified clipping operation defined in the precondition in Corollary 12.7.3. Then, the statement follows from Corollary 12.7.3 as follows: letting $\mathcal{E}^c$ denote the complement of $\mathcal{E}$,

$$\begin{aligned}
\mathbb{E}[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(x)] &= \mathbb{E}[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(x) \mid \mathcal{E}] \Pr[\mathcal{E}] + \mathbb{E}[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(x) \mid \mathcal{E}^c] \Pr[\mathcal{E}^c] \\
&\le \frac{\|x - x_{i-1}\|^2}{2\eta_i n_i} + 2\eta_i G_2^2 + \frac{G_k^k D}{(k-1)(\frac{C}{2})^{k-1}} + \mathbb{E}[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(x) \mid \mathcal{E}^c] \Pr[\mathcal{E}^c] \\
&\le \frac{\|x - x_{i-1}\|^2}{2\eta_i n_i} + 2\eta_i G_2^2 + \frac{G_k^k D}{(k-1)(\frac{C}{2})^{k-1}} + G_2 D \Pr[\mathcal{E}^c] \\
&\le \frac{\|x - x_{i-1}\|^2}{2\eta_i n_i} + 2\eta_i G_2^2 + \frac{G_k^k D}{(k-1)(\frac{C}{2})^{k-1}} + \frac{G_2 D}{n^2},
\end{aligned}$$

where we used that $F_{\mathcal{P}}$ is $G_1 \le G_2$-Lipschitz by Lemma 12.2.7. □

Combining our privacy and utility guarantees, we are ready to prove this section's main theorem.

Proof of Theorem 12.7.6. For simplicity, let $\overline{x}_0 := x^\star$ and $\zeta_0 := x_0 - x^\star$, so $\|\zeta_0\| \le D$ by assumption. Also, suppose that n is a power of 2, as the adjustment on Line 2 only affects n (and hence the guarantees) by constant factors. The privacy claim follows immediately from Lemma 12.7.8 assuming its preconditions are met, which we verify at the end of the proof. By applying Lemma 12.7.9 in each phase $i \in [I]$ to $x \leftarrow x_i$, assuming its preconditions are met, we have

$$\begin{aligned}
\mathbb{E}\left[F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star)\right] &\le \sum_{i \in [I]} \left(\frac{\mathbb{E}\left[\|\zeta_{i-1}\|^2\right]}{2\eta_i n_i} + 2\eta_i G_2^2 + \frac{G_k^k D}{(\frac{C}{2})^{k-1}} \right) + \frac{G_2 D I}{n^2} \\
&\quad + \mathbb{E}\left[F_{\mathcal{P}}(x_k) - F_{\mathcal{P}}(\overline{x}_k)\right] \\
&\le \frac{4D^2}{\eta n} + \frac{\eta G_2^2}{2} + \frac{G_k^k D I}{(\frac{C}{2})^{k-1}} + \frac{G_2 D}{\sqrt{n}} + G_2 \sigma_I \sqrt{d} \\
&\quad + \sum_{i \in [I-1]} \left(\frac{3600 C^2 d \eta_i \log(\frac{3}{\delta})}{n_i \varepsilon^2} + \frac{\eta_i G_2^2}{2} \right).
\end{aligned}$$

In the first inequality, we used $G_1 \le G_2$-Lipschitzness of $F_{\mathcal{P}}$ by Lemma 12.2.7, and in the second inequality, we pulled out the $i = 1$ term and adjusted indices, and bounded $I \le n$ and used Jensen's inequality to bound $(\mathbb{E}\,\|\zeta_I\|)^2 \le \mathbb{E}\,\|\zeta_I\|^2 = \sigma_I^2 d$. Now using that $\frac{\eta_i}{n_i}$ and η_i are geometrically decaying sequences, we continue bounding the above display using our choice of C:

$$\begin{aligned}\mathbb{E}\left[F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star)\right] &\le \frac{4D^2}{\eta n} + \eta G_2^2 + \frac{14400(\frac{C}{2})^2 d\eta \log(\frac{3}{\delta})}{n\varepsilon^2} + \frac{G_k^k DI}{(\frac{C}{2})^{k-1}} + \frac{G_2 D}{\sqrt{n}} + G_2\sigma_I\sqrt{d} \\ &\le \frac{4D^2}{\eta n} + \eta G_2^2 + 2(A\eta)^{\frac{k-1}{k+1}}\left(G_k^k DI\right)^{\frac{2}{k+1}} + \frac{G_2 D}{\sqrt{n}} + G_2\sigma_I\sqrt{d}, \\ \text{for } A := \frac{14400 d \log^2(\frac{15n}{\delta})}{n\varepsilon^2},\ C &= 2\left(\frac{G_k^k DI}{A\eta}\right)^{\frac{1}{k+1}}.\end{aligned}$$

Next, plugging in our choice of

$$\eta = \min\left(\underbrace{\sqrt{\frac{4}{n}}\cdot\frac{D}{G_2}}_{:=\eta_1}, \underbrace{\frac{DI}{G_k n}\cdot\left(\frac{n}{A}\right)^{\frac{k-1}{2k}}}_{:=\eta_2}\right), \tag{12.20}$$

we have the claimed utility bound upon simplifying, and using that $G_2\sigma_I\sqrt{d}$ is a low-order term.

We now verify our parameters satisfy the conditions in Lemma 12.7.8 and Lemma 12.7.9, which concludes the proof. First, it is straightforward to check that both sets of conditions are implied by

$$\frac{96\eta\beta c}{\sqrt{\varepsilon}}\log\left(\frac{30n}{\delta}\right) \le 1,\ c \ge 4n\cdot\left(\frac{2G_k}{C}\right)^k,\ \text{and } c \ge \frac{26}{\varepsilon}\log\left(\frac{15n}{\delta}\right), \tag{12.21}$$

given that we chose $\omega = \frac{18}{\varepsilon}\sqrt{2c\log(\frac{15}{\delta})} \le \frac{c}{\sqrt{\varepsilon}}$. Indeed, $C \ge 8\omega_i \log(\frac{30n_i}{\delta}) \impliedby 2\eta\beta\omega\log(\frac{30n}{\delta}) \le 1$ which is subsumed by the first condition in (12.21). Clearly, $c \ge \frac{26}{\varepsilon}\log(\frac{15n}{\delta})$, giving the third condition in (12.21). Next, a direct computation with the definition of η_2 in (12.20)

yields

$$c = 2\sqrt{An} = 4n \cdot \sqrt{\frac{A}{n}} = 4n \cdot \left(G_k \cdot \left(\frac{A\eta_2}{G_k^k DI} \right)^{\frac{1}{k+1}} \right)^k .$$

Now because C depends inversely on $\eta \leq \eta_2$ defined in (12.20), the second condition in (12.21) holds:

$$c = 4n \cdot \left(G_k \cdot \left(\frac{A\eta_2}{G_k^k DI} \right)^{\frac{1}{k+1}} \right)^k \geq 4n \cdot \left(G_k \cdot \left(\frac{A\eta}{G_k^k DI} \right)^{\frac{1}{k+1}} \right)^k = 4n \cdot \left(\frac{2G_k}{C} \right)^k .$$

Finally, the first condition in (12.21) now follows from our upper bound on β. □

12.8 Improved Smoothness Bounds for Generalized Linear Models

In this section, we give an improved algorithm for heavy-tailed private SCO when the sample functions $f(x; s)$ are instances of a smooth generalized linear model (GLM). That is, we assume the sample space $\mathcal{S} \subseteq \mathbb{R}^d$, and that for a convex function $\sigma : \mathbb{R} \to \mathbb{R}$,

$$f(x; s) = \sigma\left(\langle s, x \rangle\right). \tag{12.22}$$

We also assume that all $f(x; s)$ are β-smooth. Observe that

$$\nabla f(x; s) = \sigma'(\langle s, x \rangle) s, \tag{12.23}$$

so that for all $x \in \mathcal{X}$, $\nabla f(x; s)$ are all scalar multiples of the same vector s. We prove that under this assumption, clipped gradient descent steps can only improve contraction, in contrast to Fact 12.10.2.

Lemma 12.8.1. *Let $s, s' \in \mathbb{R}$ and let $x, x', g \in \mathbb{R}^d$. Assume that*

$$\left\| (x - sg) - (x' - s'g) \right\| \leq \left\| x - x' \right\|.$$

Then for any $C \geq 0$, letting $t := \operatorname{sign}(s)\min(|s|, C)$ and $t' := \operatorname{sign}(s')\min(|s'|, C)$, we have

$$\left\|(x - tg) - (x' - t'g)\right\| \leq \left\|x - x'\right\|.$$

Proof. Note that the premise is impossible unless $\operatorname{sign}(s - s') = \operatorname{sign}(\langle x - x', g\rangle)$. Without loss of generality, assume they are both nonnegative, else we can negate s, s', g. In this case,

$$\begin{aligned}\left\|(x - x') - (s - s')g\right\| \leq \left\|x - x'\right\| &\iff (s' - s)^2 \left\|g\right\|^2 \leq 2(s - s')\left\langle x - x', g\right\rangle \\ &\iff s - s' \leq \frac{2\left\langle x - x', g\right\rangle}{\left\|g\right\|^2}.\end{aligned}$$

Now, observe that $t - t' \leq s - s'$ and $\operatorname{sign}(t - t') = \operatorname{sign}(s - s')$, for any value of $C \geq 0$. Therefore, $t - t' \leq \frac{2\langle v, g\rangle}{\|g\|^2}$ as well, and we can reverse the above chain of implications. □

Note that the premise of Lemma 12.8.1 is exactly an instance of Fact 12.7.1 where $\nabla f(x)$ and $\nabla f(x')$ are scalar multiples of the same direction, which is the case for GLMs by (12.23). Hence, Lemma 12.8.1 shows the contraction property in Fact 12.7.1 is preserved after clipping gradients (again, for GLMs).

We can now directly combine Lemma 12.7.2 and our contraction results, used to analyze the stability of Algorithm 40, with the iterative localization framework of [FKT20], Section 4.

Theorem 12.8.2. *Consider an instance of k-heavy-tailed private SCO, following notation in Definition 12.2.6, let $x^\star := \operatorname{argmin}_{x \in \mathcal{X}} F_{\mathcal{P}}(x)$, and let $\rho \geq 0$. Further, assume that for a convex function σ, the sample functions $f(x; s)$ satisfy (12.22) for all $s \in \mathcal{S} \subseteq \mathbb{R}^d$. Finally, assume $f(x; s)$ is β-smooth for all $s \in \mathcal{S}$, where $\beta \leq \max(\sqrt{\frac{n}{2}} \cdot \frac{G_2}{D}, n \cdot (\frac{d}{n^2\rho})^{\frac{k-1}{2k}} \cdot \frac{G_k}{D})$. Algorithm 43 is a ρ-CDP algorithm which draws $\mathcal{D} \sim \mathcal{P}^n$, queries n sample gradients (using samples in $\mathcal{D}$), and outputs $x_I \in \mathcal{X}$ satisfying*

$$\mathbb{E}\left[F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star)\right] \leq 4G_2 D\sqrt{\frac{1}{n}} + 26G_k D\left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}.$$

Proof. We begin with the privacy claim. Consider neighboring datasets $\mathcal{D}$, $\mathcal{D}'$, and suppose

Algorithm 43: OnePass-Clipped-DP-SGD$(\mathcal{D}, n, \mathcal{X}, x_0, \rho)$

Input: Dataset $\mathcal{D} = \{s_i\}_{i\in[n]} \in \mathcal{S}^n$, domain $\mathcal{X} \subset \mathbb{B}(x_0, D)$ for $x_0 \in \mathcal{X}$
$I \leftarrow \lfloor \log_2(n) \rfloor$
$n \leftarrow 2^I$
$\eta \leftarrow \min(\sqrt{\frac{8}{n}} \cdot \frac{D}{G_2}, \frac{1}{n} \cdot (\frac{n^2\rho}{32d})^{\frac{k-1}{2k}} \cdot \frac{2^{\frac{k+1}{2k}} D}{G_k}),\ C \leftarrow (\frac{G_k^k D\rho n}{32\eta d})^{\frac{1}{k+1}}$
for $i \in [I]$ **do**
  $n_i \leftarrow 2^{-i}n,\ \eta_i \leftarrow 16^{-i}\eta,\ C_i \leftarrow 2^i C,\ \sigma_i \leftarrow 2\eta_i C_i \cdot \sqrt{\frac{2}{\rho}}$
  $\mathcal{D}_i \leftarrow$ first n_i elements of $\mathcal{D},\ \mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_i$
  $\overline{x}_i \leftarrow$ OnePass-Clipped-SGD$(\mathcal{D}_i, C_i, \eta_i, n_i, \mathcal{X}, x_{i-1})$
  $\xi_i \sim \mathcal{N}(\not{0}_d, \sigma_i^2 \mathbf{I}_d)$
  $x_i \leftarrow \overline{x}_i + \xi_i$
end
Return: x_I

the datasets differ on the j^{th} entry such that $s_j \in \mathcal{D}_i$ (if the differing entry is not in $\cup_{i\in[I]}\mathcal{D}_i$, Algorithm 43 clearly satisfies 0-CDP). Let $\overline{x}_i$ and $\overline{x}_i'$ be the outputs of Line 8 when run with the same initialization x_{i-1}, and neighboring $\mathcal{D}_i$, $\mathcal{D}_i'$. By the assumption on β, since $\eta_i \le \eta$ for all $i \in [I]$, we can apply Fact 12.7.1 and Lemma 12.8.1 (recalling the characterization (12.23)) to show $\|\overline{x}_i - \overline{x}_i'\| \le 2\eta_i C_i$ with probability 1. Therefore, by our choice of σ_i and the first and third parts of Lemma 12.2.4, the whole algorithm is ρ-CDP regardless of which $\mathcal{D}_i$ contained the differing sample, since all other calls to OnePass-Clipped-SGD are 0-CDP as we can couple all randomness used by the calls.

Next, we prove the utility claim. For simplicity, let $\overline{x}_0 := x^\star$ and $\xi_0 := x_0 - x^\star$, so $\|\xi_0\| \le D$ by assumption. By applying Lemma 12.7.2 for all $i \in [I]$ with $x_0 \leftarrow x_{i-1}$ and $u \leftarrow \overline{x}_{i-1}$, we have

$$\begin{aligned}
\mathbb{E}\left[F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(x^\star)\right] &= \sum_{i\in[I]} \mathbb{E}\left[F_{\mathcal{P}}(\overline{x}_i) - F_{\mathcal{P}}(\overline{x}_{i-1})\right] + \mathbb{E}\left[F_{\mathcal{P}}(x_I) - F_{\mathcal{P}}(\overline{x}_I)\right] \\
&\le \sum_{i\in[I]} \left(\mathbb{E}\left[\frac{\|\xi_{i-1}\|^2}{2\eta_i n_i}\right] + \frac{\eta_i G_2^2}{2} + \frac{G_k^k D}{(k-1)C_i^{k-1}} \right) + G_1 \mathbb{E}\left[\|x_I - \overline{x}_I\|\right] \\
&\le \frac{4D^2}{\eta n} + \sum_{i\in[I-1]} 2^{-i} \left(\frac{32d\eta C^2}{\rho n} + \frac{\eta G_2^2}{2} + \frac{G_k^k D}{C^{k-1}} \right) + \sqrt{\frac{8d}{\rho}} G_1 \eta C \cdot 8^{-I}
\end{aligned}$$

$$\leq \frac{4D^2}{\eta n} + \frac{32d\eta C^2}{\rho n} + \frac{\eta G_2^2}{2} + \frac{G_k^k D}{C^{k-1}} + 24\sqrt{\frac{d}{\rho}} \cdot \frac{G_1 \eta C}{n^3},$$

where the second line applied Lemma 12.2.7, the third used Jensen's inequality to bound $\mathbb{E}[\|x_I - \overline{x}_I\|]^2 \leq \mathbb{E}[\|x_I - \overline{x}_I\|^2]$ and our assumption $k \geq 2$, and the last used the geometric decay of the different parameters. Finally, by plugging in our choices of C, η, we have

$$\frac{4D^2}{\eta n} + \frac{\eta G_2^2}{2} + \frac{32d\eta C^2}{\rho n} + \frac{G_k^k D}{C^{k-1}} = \frac{4D^2}{\eta n} + \frac{\eta G_2^2}{2} + 2\eta^{\frac{k-1}{k+1}} \left(G_k^k D\right)^{\frac{2}{k+1}} \left(\frac{32d}{\rho n}\right)^{\frac{k-1}{k+1}}$$
$$\leq G_2 D\sqrt{\frac{8}{n}} + 8G_k D \left(\frac{\sqrt{d}}{n\sqrt{\rho}}\right)^{1-\frac{1}{k}}.$$

We can also check that the final summand is a low-order term, by using $\eta \leq \frac{1}{n} \cdot (\frac{n^2\rho}{32d})^{\frac{k-1}{2k}} \cdot \frac{2^{\frac{k+1}{2k}} D}{G_k}$:

$$24\sqrt{\frac{d}{\rho}} \cdot \frac{G_1 \eta C}{n^3} \leq \frac{5G_k D}{n^2}.$$

The conclusion follows by adjusting n, since Algorithm 43 is run with a sample count in $[\frac{n}{2}, n]$. □

12.9 High-probability stochastic convex optimization

In this section, to highlight another application of our population-level localization framework, we show that it obtains improved high-probability guarantees for the following standard bounded-variance estimator parameterization of SCO in the non-private setting.

Definition 12.9.1 (Stochastic convex optimization)**.** Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex, with $\mathrm{diam}(\mathcal{X}) = D$. In the *stochastic convex optimization (SCO)* problem, there is a convex function $f : \mathcal{X} \to \mathbb{R}$, and we have query access to a stochastic oracle $g : \mathcal{X} \to \mathbb{R}^d$ satisfying, for all $x \in \mathcal{X}$,

$$\mathbb{E}\left[g(x)\right] \in \partial f(x), \ \mathbb{E}\left[\|g(x)\|^2\right] \leq G^2.$$

For a convex function $\psi : \mathcal{X} \to \mathbb{R}$, our goal in SCO is to optimize the composite function $f + \psi$.

For instance, one can set ψ to the constant zero function to recover the non-composite variant of SCO. We include the composite variant of Definition 12.9.1 as it is a standard extension in the SCO literature, under the assumption that the function ψ is "simple." The specific notion of simplicity we use is that $\psi : \mathcal{X} \to \mathbb{R}$ admits an efficient *proximal oracle* (Definition 12.9.2).

Definition 12.9.2 (Proximal oracle)**.** Let $\mathcal{X} \subset \mathbb{R}^d$ be compact and convex. We say $\mathcal{O}$ is a *proximal oracle* for a convex function $\psi : \mathcal{X} \to \mathbb{R}$ if for any inputs $v \in \mathbb{R}^d$, $\eta \in \mathbb{R}_{\geq 0}$, $\mathcal{O}(v)$ returns

$$\operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \frac{1}{2\eta} \|x - v\|^2 + \psi(x) \right\}.$$

In Theorem 12.9.4, we give an algorithm which uses n queries to each of g and a proximal oracle for ψ, and achieves an error bound for $f + \psi$ of

$$O\left(GD \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}} \right), \tag{12.24}$$

with probability $\geq 1 - \delta$. Similar rates are straightforward to derive using martingale concentration when the estimator g is assumed to satisfy heavier tail bounds, such as a sub-Gaussian norm. To our knowledge, the rate (12.24) was first attained recently by [CH24], who also proved a matching lower bound. Our Theorem 12.9.4 gives an alternative route to achieving this error bound. As was the case in several recent works in the literature [HS16, DDXZ21, Lia24] who studied high-probability variants of stochastic convex optimization, our Theorem 12.9.4 is based on using geometric aggregation techniques within a proximal point method framework (in our case, using Fact 12.2.9 within Algorithm 37). However, these aforementioned prior works all assume additional smoothness bounds on the function f.

We use the following standard result in the literature as a key subroutine.

Lemma 12.9.3 (Lemma 1, [ACJ+21])**.** *In the setting of Definition 12.9.1, assume ψ is λ-strongly convex, let $x^\star := \operatorname{argmin}_{x \in \mathcal{X}} f(x) + \psi(x)$, and let $T \in \mathbb{N}$. There is an algorithm which queries the stochastic oracle g and a proximal oracle for ψ each T times, and produces*

$\overline{x}$ satisfying, with probability $\geq \frac{4}{5}$,

$$\|\overline{x} - x^\star\| \leq \frac{30G}{\lambda\sqrt{T}}.$$

We combine Lemma 12.9.3 with Proposition 12.3.7 to obtain the following high-probability SCO algorithm.

Theorem 12.9.4. *Consider an instance of SCO, following notation in Definition 12.9.1, let $n \in \mathbb{N}$, $x^\star := \operatorname{argmin}_{x\in\mathcal{X}} f(x) + \psi(x)$, and $\delta \in (0, \frac{1}{2})$. There is an algorithm using n queries to g and a proximal oracle for ψ and outputs $x \in \mathcal{X}$ satisfying, for a universal constant C_{sco}, with probability $\geq 1 - \delta$,*

$$f(x) + \psi(x) - f(x^\star) - \psi(x^\star) \leq C_{\text{sco}} \cdot GD \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}}.$$

Proof. Assume without loss of generality that $\frac{1}{\delta}$ is a sufficiently large constant (else we can adjust the constant factor C_{sco}), and that n is sufficiently larger than $\log \frac{1}{\delta}$ (else the result holds because the range of the function is bounded by GD). We instantiate Proposition 12.3.7 with $F_{\mathcal{P}} \leftarrow f + \psi$, $I \leftarrow \frac{1}{2}\log_2 n$, and in each phase $i \in [I]$ of Algorithm 37, we let $n_i := \frac{n}{2^i}$. In the remainder of the proof, we describe how to implement (12.7) in the i^{th} phase, where $F_{\mathcal{P}} \leftarrow f + \psi$, splitting into cases.

If $\frac{1}{\delta}$ is bounded by $\operatorname{polylog}(n)$ and n is sufficiently large, suppose that n is a power of 4, else we can use fewer queries and lose a constant factor in the guarantee. Then we can use a batch of n_i consecutive queries, divided into $48\log(\frac{1}{\delta_i})$ portions, where $\delta_i := \frac{\delta}{2^i}$. We then use Lemma 12.9.3 on each portion of queries, with $f \leftarrow f$ and $\psi \leftarrow \psi + \frac{\lambda_i}{2}\|\cdot - x_{i-1}\|^2$; it is straightforward to see that Definition 12.9.2 generalizes to give a proximal oracle for this new ψ. A Chernoff bound shows that at least $\frac{3}{5}$ of the portions will return a point satisfying the bound in Lemma 12.9.3 except with probability δ_i, so Fact 12.2.9 returns us a point at distance at most $\frac{90G}{\lambda\sqrt{T}}$ from $x_i^\star$, where

$$T = \Omega\left(\frac{n_i}{\log \frac{1}{\delta_i}}\right) = \Omega\left(\frac{n}{2^i\left(\log \frac{1}{\delta} + i\right)}\right),$$

(accounting for rounding error). Therefore, (12.7) holds with

$$\Delta = \frac{C_{\text{sco}}}{2} \cdot G \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}},$$

for sufficiently large C_{sco}. Proposition 12.3.7 then implies that Algorithm 37 outputs x satisfying

$$f(x) + \psi(x) - f(x^\star) - \psi(x^\star) \le 2GD \cdot \sqrt{\frac{\Delta}{n^{1.5}}} + \frac{C_{\text{sco}}}{2} \cdot GD \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}} \le C_{\text{sco}} \cdot GD \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}},$$

where we use that $G_1 \le G$ by Jensen's inequality and our second moment bound in Definition 12.9.1. The failure probability follows from a union bound because we ensured that $\sum_{i \in [I]} \delta_i \le \delta$.

Finally, if $\frac{1}{\delta}$ is larger than polylog(n), then we let $I, J \in \mathbb{N}$ be chosen such that

$$I := \left\lfloor \log_2 \left(\frac{n}{J} \right) \right\rfloor, \; J \ge 48 \log \left(\frac{I}{\delta} \right),$$

which is achievable with $I = O(\log n)$ and $J = O(\log \frac{\log n}{\delta}) = O(\log \frac{1}{\delta})$. Let $m := \frac{n}{J}$, and assume without loss that m is a power of 2, which we can guarantee by discarding $\le \frac{1}{2}$ our queries, losing a constant factor in the error bound. The remainder of the proof follows identically to the first part of this proof, where we union bound over I phases, the i^{th} of which uses J batches of $\frac{m}{2^i}$ unused queries. Again we may apply Lemma 12.9.3 and Fact 12.2.9 with $T = \frac{m}{2^i}$, so (12.7) holds with

$$\Delta = \frac{C_{\text{sco}}}{2} \cdot G \cdot \sqrt{\frac{\log \frac{1}{\delta}}{n}},$$

except with probability $\frac{\delta}{I}$. The conclusion then follows from Proposition 12.3.7. □

12.10 Non-contraction of truncated contractive steps

In this section, we demonstrate that a natural conjecture related to the performance of clipped private gradient algorithms in the smooth setting is false. We state this below

as Conjecture 12.10.1. To motivate it, suppose v is the difference between a current pair of coupled iterates of a private gradient algorithm instantiated on neighboring datasets, and suppose the differing sample function has already been encountered. If we take a coupled gradient step in a sufficiently smooth function, Fact 12.7.1 shows that the step is a contraction. However, to preserve privacy in the heavy-tailed setting, it is natural to ask whether such a contractive step remains contractive after the gradients are clipped, i.e. the statement of Conjecture 12.10.1 (which gives the freedom for C to be lower bounded).

Conjecture 12.10.1. *Let $\|v\|_2 \le C$ for a sufficiently large constant C, and let $\|v - (g - h)\| \le \|v\|$. Let $g' = \Pi_1(g)$ and $h' = \Pi_1(h)$.*[8] *Then, $\|v - (g' - h')\| \le C$.*

We strongly refute Conjecture 12.10.1, by disproving it for any $C \ge 0$. We remark that Lemma 12.10.2 does not necessarily rule out this approach to designing heavy-tailed DP-SCO algorithms in the smooth regime, but demonstrates an obstacle if additional structure of gradients is not exploited.

Lemma 12.10.2. *Conjecture 12.10.1 is false for any choice of $C \ge 0$.*

Proof. We give a 2-dimensional counterexample. Let

$$v = \begin{pmatrix} -C \\ 0 \end{pmatrix}, \; g = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \; h = \begin{pmatrix} \frac{2C+1}{C+1} \\ \frac{C\sqrt{2C+1}}{C+1} \end{pmatrix} = \sqrt{2C+1} \underbrace{\begin{pmatrix} \frac{\sqrt{2C+1}}{C+1} \\ \frac{C}{C+1} \end{pmatrix}}_{:=h'}.$$

Observe that

$$v - (g - h) = \begin{pmatrix} -(C+1) + \frac{2C+1}{C+1} \\ \frac{C\sqrt{2C+1}}{C+1} \end{pmatrix} = \begin{pmatrix} \frac{-C^2}{C+1} \\ \frac{C\sqrt{2C+1}}{C+1} \end{pmatrix} = C \begin{pmatrix} \frac{-C}{C+1} \\ \frac{\sqrt{2C+1}}{C+1} \end{pmatrix}.$$

It is easy to verify $\|v - (g - h)\| = C$ at this point. Moreover,

$$v - (g' - h') = \begin{pmatrix} -(C+1) + \frac{\sqrt{2C+1}}{C+1} \\ \frac{C}{C+1} \end{pmatrix}.$$

[8] By scale-invariance of the claim, the assumption that the truncation threshold is 1 is without loss of generality.

For $C \geq 0$, the first coordinate of this vector is already less than $-C$. □

12.11 Non-decay of empirical squared bias

In this section, we present an obstacle towards a natural approach to improving the logarithmic terms in our algorithm in Section 12.3. We follow the notation of Section 12.3.1, i.e. for samples $\{i \equiv s_i\}_{i\in[n]} \sim \mathcal{P}^n$, we define sample functions $f_i \equiv f(\cdot; s_i)$, and let

$$b_{\mathcal{D}} := \max_{x\in\mathcal{X}} \left\| \frac{1}{n}\sum_{i\in[n]} \nabla f_i(x) - \frac{1}{n}\sum_{i\in[n]} \Pi_C(\nabla f_i(x)) \right\|. \tag{12.25}$$

A basic bottleneck with known approaches following SCO-to-ERM reductions is that they require a strongly convex ERM solver as a primitive, due to known barriers to generalization in SCO without strong convexity (see e.g. discussion in [SSSSS09]). This poses an issue in the heavy-tailed setting, because standard analyses of strongly convex clipped SGD (see e.g. our Proposition 12.3.1) appear to suffer a dependence on $b_{\mathcal{D}}^2$ in the utility bound, which upon taking expectations requires bounding

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n} b_{\mathcal{D}}^2 = \mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n} \left[\max_{x\in\mathcal{X}} \left\| \frac{1}{n}\sum_{i\in[n]} \nabla f_i(x) - \frac{1}{n}\sum_{i\in[n]} \Pi_C(\nabla f_i(x)) \right\|^2 \right]. \tag{12.26}$$

Recall from Lemma 12.3.2 that it is straightforward to bound $\mathbb{E}\, b_{\mathcal{D}} \leq \frac{G_k^k}{C^{k-1}}$, due to Fact 12.2.8. Bounding $\mathbb{E}\, b_{\mathcal{D}}^2$ is more problematic; in [LR23], requiring this bound resulted in a dependence on G_{2k} as opposed to G_k (see the proof of Theorem 31), which we avoid (up to a polylogarithmic overhead) via our population-level localization strategy. We now present an alternative strategy to bound (12.26), avoiding a G_{2k} dependence. Observe that, by using

$(a+b+c)^2 \le 3(a^2+b^2+c^2)$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n} b_{\mathcal{D}}^2 \le 3 \underbrace{\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\max_{x\in\mathcal{X}}\left\|\frac{1}{n}\sum_{i\in[n]}\nabla f_i(x) - \nabla F_{\mathcal{P}}(x)\right\|^2\right]}_{:=T_1} \\ + 3\underbrace{\max_{x\in\mathcal{X}}\left\|\nabla F_{\mathcal{P}}(x) - \mathop{\mathbb{E}}_{s\sim\mathcal{P}}[\Pi_C(\nabla f(x;s))]\right\|^2}_{:=T_2} \\ + 3\underbrace{\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n}\left[\max_{x\in\mathcal{X}}\left\|\frac{1}{n}\sum_{i\in[n]}\Pi_C(\nabla f_i(x)) - \mathop{\mathbb{E}}_{s\sim\mathcal{P}}[\Pi_C(\nabla f(x;s))]\right\|^2\right]}_{:=T_3}.\end{aligned} \tag{12.27}$$

We focus on T_1, as T_3 can be bounded by similar means (as truncation can only improve moment bounds), and $T_2 \le \frac{G_k^{2k}}{C^{2(k-1)}}$ via Fact 12.2.8. Hence, if we can show that $T_1 = O(\frac{G_2^2}{n})$ under the moment bound assumption in Assumption 12.2.5, we can avoid the logarithmic factors lost by our population localization approach. We suggest the following conjecture as an abstraction of this bound.

Conjecture 12.11.1. *Let $\mathcal{P}$ be a distribution over $\mathcal{S}$. For each $x \in \mathcal{X}$, let $g(x;s) \in \mathbb{R}^d$ be a random vector, indexed by $s \sim \mathcal{S}$, satisfying $\mathbb{E}_{s\sim\mathcal{P}}[g(x;s)] = \nvDash_d$ and $\mathbb{E}_{s\sim\mathcal{P}}[\sup_{x\in\mathcal{X}}\|g(x;s)\|^2] \le 1$. Finally for $S \sim \mathcal{P}^n$ and $x \in \mathcal{X}$, let $g(x;S) := \frac{1}{n}\sum_{s\in S} g(x;s)$. Then,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{P}^n}\left[\sup_{x\in\mathcal{X}} g(x;S)^2\right] = O\left(\frac{1}{n}\right).$$

Note that the bound in Conjecture 12.11.1 exactly corresponds to T_1 in (12.27), after rescaling all sample gradients by $\frac{1}{G_2}$, and centering them by subtracting $\nabla F_{\mathcal{P}}(x)$. Hence, if Conjecture 12.11.1 is true, it would yield the following desirable bound in (12.27):

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}^n} b_{\mathcal{D}}^2 = O\left(\frac{G_2^2}{n} + \frac{G_k^k}{(k-1)C^{k-1}}\right).$$

Moreover, it is simple to prove a bound of $O(1)$ on the right-hand side of Conjecture 12.11.1, and as $n \to \infty$ it is reasonable to suppose $g(x;S) \to \nvDash_d$ for all $x \in \mathcal{X}$. Nonetheless, we refute Conjecture 12.11.1 in full generality with a simple 1-dimensional

example.

Lemma 12.11.2. *Conjecture 12.11.1 is false.*

Proof. Let $\mathcal{S} = [0,1]$ and let $\mathcal{P}$ be the uniform distribution over $\mathcal{S}$. Let $\mathcal{X}$ index a set of random $g(x;\cdot) : [0,1] \to [0,1]$ which are nonzero at finitely many points.[9] Then $\mathbb{E}_{s\sim\mathcal{P}}\, g(x;s) = 0$ for all $x \in \mathcal{X}$, and $g(x;s)^2 \le 1$ for all $x \in \mathcal{X}, s \in \mathcal{S}$. However, for any $S \in [0,1]^n$, we have

$$\sup_{x\in\mathcal{X}} g(x;S)^2 = 1.$$

□

While Lemma 12.11.2 does not rule out the approach suggested in (12.27) (or other approaches) to improve the analysis of strongly convex ERM solvers in heavy-tailed settings, it presents an obstacle to applying the natural decomposition strategy in (12.27). To overcome Lemma 12.11.2, one must either use more structure about the index set $\mathcal{X}$ or the iterates encountered by the algorithm, or consider a different decomposition strategy for bounding the squared empirical bias.

12.12 Proof of Lemma 12.7.7

In this section, we prove Lemma 12.7.7. We first require the following standard fact (see e.g. [Sch14]).

Fact 12.12.1. *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex set. Then for any $x, y \in \mathbb{R}^d$, we have*

$$\|\Pi_{\mathcal{K}}(x) - \Pi_{\mathcal{K}}(y)\| \le \|x - y\|.$$

We now set up some notation. Let $\{\psi_j : \mathcal{X} \to \mathcal{X}\}_{j\in[T]}$ and $\{\varphi_j : \mathcal{X} \to \mathcal{X}\}_{j\in[T]}$ be two sequences of operations. We say that an operation pair (ψ, φ) is contractive if for any two

[9] We note there is a bijection between $\mathcal{X}$ and any convex subset $\mathcal{X}'$ of $\mathbb{R}^d$ containing a ball with nonzero radius. To see this, it is well-known that there is a bijection from $[0,1]$ to $\mathbb{R}_{\ge 0}$, and we can simply construct a bijection between $\mathbb{R}_{\ge 0}$ and $\mathcal{X}$ by mapping the interval $[i-1, i]$ to $[0,1]^{2i}$ (where the first i coordinates specify the nonzero points, and the next i coordinates specify their values) for all $i \in \mathbb{N}$. Finally, it is well-known there is a bijection between $[0,1]$ and $\mathbb{R}^d$, and we can construct a bijection between $\mathcal{X}'$ and $\mathbb{R}^d$ by considering each 1-dimensional projection separately.

points $x, y \in \mathcal{X}$,

$$\|\psi(x) - \varphi(y)\| \leq \|x - y\|.$$

We say an operation pair (ψ, φ) is (C, ζ)-contractive if for any x, y where $\|x - y\| \leq C$, we have

$$\|\psi(x) - \varphi(y)\| \leq \|x - y\| + \zeta.$$

Let $\psi^j(x) = \psi_j \circ \psi_{j-1} \circ \ldots \circ \psi_1(x)$, and define φ^j similarly, for all $j \in [T]$.

We prove Lemma 12.7.7 as a consequence of the following more general result.

Lemma 12.12.2. *Let $x_0 = x_0' \in \mathcal{X}$, and consider two sequences of operations $\{\psi_j : \mathcal{X} \to \mathcal{X}\}_{j\in[T]}$ and $\{\psi_j' : \mathcal{X} \to \mathcal{X}\}_{j\in[T]}$ satisfying the following conditions, for $c := \lfloor \frac{C}{\zeta} \rfloor$.*

1. *For at least $T - c - 1$ indices $j \in [T]$, (ψ_j, φ_j) is contractive.*

2. *At most one operation pair, (ψ_k, ψ_k), is (∞, C)-contractive.*

3. *For at most c indices $j \in [T]$, (ψ_j, φ_j) is $(2C, \zeta)$-contractive.*

Then for all $j \in [T]$, we have that $\|\psi^j(x_0) - \varphi^j(y_0)\| \leq 2C$.

Proof. Define $\Delta_j := \|\psi^j(x_0) - \varphi^j(x_0')\|$ for all $j \in [T]$. Let $a_j \leq c$ be the total number of $(2C, \zeta)$-contractive operation pairs (ψ_i, φ_i) where $i \leq j$, and let b_j be the 0-1 indicator variable for $k \leq j$. We use induction to show that $\Delta_j \leq a_j\zeta + b_jC$. When $j = 1$, the claim holds. Now if the claim holds for $j - 1$, then $\Delta_{j-1} \leq a_{j-1}\zeta + b_{j-1}C \leq 2C$. Hence, by definition,

$$\Delta_j \leq \Delta_{j-1} + (a_j - a_{j-1})\zeta + (b_j - b_{j-1})C = a_j\zeta + b_jC,$$

which completes our induction. This also implies $\Delta_T \leq 2C$ as claimed. □

Proof of Lemma 12.7.7. Throughout the following proof, note that $\widehat{c}_i \le 2c$ deterministically (due to our use of $\mathrm{BLap}(\frac{3}{\varepsilon}, c)$ noise), and under the stated parameter bounds,

$$\widehat{C} \in \left[\frac{7C}{8}, \frac{9C}{8}\right] \text{ and } |\nu_{i,j}| \le \frac{C}{4} \text{ for all } j \in [n_i].$$

Let $\{g_{i,j} = \Pi_C(\nabla f(x_{i,j}; s_{i,j}))\}_{j\in[n_i]}$ and $\{g'_{i,j} = \Pi_C(\nabla f(x'_{i,j}; s'_{i,j}))\}$ be the two truncated gradient sequences in the i^{th} phase corresponding to the two datasets, and let $\{x_{i,j}\}_{j\in[n_i]}$ and $\{x'_{i,j}\}_{j\in[n_i]}$ be the corresponding iterate sequences. We set the operation sequences $\psi_j(x) := \Pi_{\mathcal{X}}(x - \eta_i g_{i,j})$ and $\varphi_j(x) := \Pi_{\mathcal{X}}(x - \eta_i g'_{i,j})$. We bound the contractivity of these operation pairs and apply Lemma 12.12.2.

First, note that because count_t, $\mathrm{count}'_t < \widehat{c}_i \le 2c$, the operation pair (ψ_j, φ_j) is an identical untruncated gradient mapping for at least $t - 2c - 1$ indices $j \in [t]$. Because we assume each sample function $f(\cdot; s)$ is β-smooth, it follows that for these indices $j \in [t]$, the operation pair (ψ_j, φ_j) is contractive, by applying Fact 12.7.1, Fact 12.12.1, and $\eta_i\beta \le 1$.

Next, recall the assumption that the datasets $\mathcal{D}$, $\mathcal{D}'$ differ in the j_0^{th} sample only. Because $\|g_{i,j_0}\| \le \frac{9C}{8} + \frac{C}{4} \le \frac{11C}{8}$ by assumption, and similarly $\|g'_{i,j_0}\| \le \frac{11C}{8}$, it follows that the operation pair $(\psi_{j_0}, \varphi_{j_0})$ is $(\infty, 3C\eta_i)$-contractive by applying the triangle inequality and Fact 12.12.1.

For all remaining indices $j \in [t]$, count_t and count'_t both incremented (under the assumption that $Y_{i,j} = Y'_{i,j}$ for these indices). We claim that (ψ_j, φ_j) is $(6\eta_i C, 12\eta_i^2 C\beta)$-contractive for these iterations. To see this, we bound

$$\begin{aligned}
\left\|\psi_j(x_{i,j}) - \varphi_j(x'_{i,j})\right\| &\le \left\|(x_{i,j} - \eta_i g_{i,j}) - (x'_{i,j} - \eta_i g'_{i,j})\right\| \\
&\le \left\|(x_{i,j} - \eta_i \nabla f(x_{i,j}; s_{i,j})) - (x'_{i,j} - \eta_i \nabla f(x'_{i,j}; s_{i,j}))\right\| \\
&\quad + \eta_i \left\|\nabla f(x_{i,j}; s_{i,j}) - \nabla f(x'_{i,j}; s_{i,j})\right\| + \eta_i \left\|g_{i,j} - g'_{i,j}\right\| \\
&\le \left\|x_{i,j} - x'_{i,j}\right\| + 12\eta_i^2 C\beta.
\end{aligned}$$

The first line used Fact 12.12.1, the second used the triangle inequality, and the last used Fact 12.7.1, Fact 12.12.1, and the fact that $\|\nabla f(x_{i,j}; s_{i,j}) - \nabla f(x'_{i,j}; s_{i,j})\| \le 6\eta_i C\beta$ by smoothness, when $\|x_{i,j} - x'_{i,j}\| \le 6C\eta_i$.

Finally, it suffices to apply Lemma 12.12.2 with $C \leftarrow 3C\eta_i$, $\zeta \leftarrow 12\eta_i^2 C\beta$, and $c \leftarrow 2c$, which we can check meets the conditions of Lemma 12.12.2 under the stated parameter bounds. $\square$

Chapter 13

PRIVATE ONLINE LEARNING VIA LAZY ALGORITHMS

13.1 Introduction

Online learning is a fundamental problem in machine learning, where an algorithm interacts with an oblivious adversary for T rounds. First, the oblivious adversary chooses T loss functions $\ell_1, \ldots, \ell_T : \mathcal{X} \to \mathbb{R}$ over a fixed decision set $\mathcal{X}$. Then, at any round t, the algorithm chooses a model $x_t \in \mathcal{X}$, and the adversary reveals the loss function ℓ_t. The algorithm suffers loss $\ell_t(x_t)$, and its goal is to minimize its cumulative loss compared to the best model in hindsight, namely its *regret*:

$$\mathbf{Reg}_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x^\star \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x^\star).$$

In this work, we study two different *differentially private* instances of this problem: differentially private online prediction from experts (DP-OPE) where the model x can be chosen from d experts ($\mathcal{X} = [d]$); and differentially private online convex optimization (DP-OCO) where the model belongs to a convex set $\mathcal{X} \subset \mathbb{R}^d$.

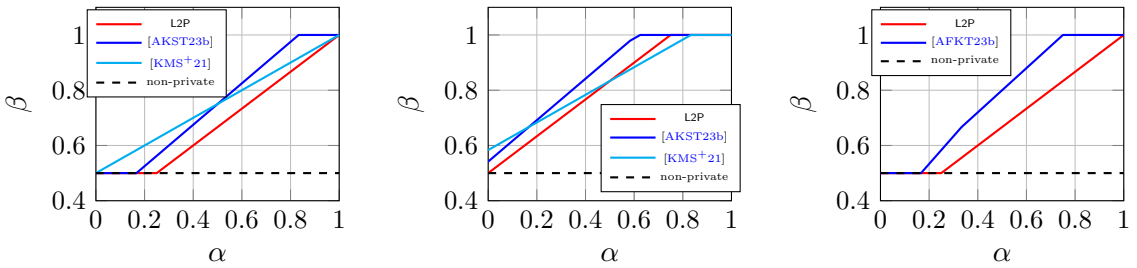
Both problems have been extensively studied recently [JKT12, ST13, JT14, AS17, KMS+21] and an exciting new direction with promising results for this problem is that of designing private algorithms based on low-switching algorithms for online learning [AFKT23b, AFKT23a, AKST23a, AKST23b]. The main idea in these works is that the privacy cost for privatizing a low-switching algorithm can be significantly smaller as these algorithms do not update their models too frequently, allowing them to allocate a larger privacy budget for each update. This has been initiated by [AFKT23b], which used the shrinking dartboard algorithm to design new algorithms for DP-OPE, later revisited by [AKST23a] to design new algorithms for DP-OCO using a regularized follow-the-perturbed-leader approach, and more recently by [AKST23b] which used a lazy and regularized version of the multiplicative

	Prior work	This work
DP-OPE	$\sqrt{T \log d} + \frac{\min\{\sqrt{d}, T^{1/3}\} \log d}{\varepsilon}$ [AS17, AFKT23b]	$\sqrt{T \log d} + \frac{T^{1/3} \log d}{\varepsilon^{2/3}}$
DP-OCO	$\min\left\{\frac{d^{1/4}\sqrt{T}}{\sqrt{\varepsilon}}, \sqrt{T} + \frac{T^{1/3}\sqrt{d}}{\varepsilon} + \frac{T^{3/8}\sqrt{d}}{\varepsilon^{3/4}}\right\}$ [KMS+21, AKST23b]	$\sqrt{T} + \frac{T^{1/3}\sqrt{d}}{\varepsilon^{2/3}}$

Table 13.1: Regret for approximate (ε, δ)-DP algorithms. For readability, we omit logarithmic factors that depend on T and $1/\delta$.

weights algorithm to obtain improved rates for DP-OCO.

While all of these results build on lazy-switching algorithms for designing private online algorithms, each one of them has a different method for achieving privacy and, to a greater extent, a different analysis. Moreover, it is not clear whether these transformations from lazy to private algorithms in prior work have fulfilled the full potential of lazy algorithms for private online learning and whether better algorithms are possible through this approach. Indeed, the regret obtained in prior work [AFKT23b, AKST23b] is $T^{1/3}/\varepsilon$ (omitting dependence on d) for DP-OPE, which implies that the normalized regret is $1/T^{2/3}\varepsilon$: this is different than what exhibited in a majority of scenarios of private optimization, where the normalized error is usually a function of $T\varepsilon$.



(a) DP-OCO ($d = \log T$) (b) DP-OCO ($d = T^{\frac{1}{3}}$) (c) DP-OPE

Figure 13.1: Regret bounds for (a) DP-OCO with $d = \operatorname{poly}\log(T)$, (b) DP-OCO with $d = T^{1/3}$ and (c) DP-OPE with $d = T$. We denote the privacy parameter $\varepsilon = T^{-\alpha}$ and regret T^{β}, and plot β as a function of α (ignoring logarithmic factors).

13.1.1 Our contributions

Our main contribution in this work is a new transformation that converts lazy online learning algorithms into private algorithms with similar regret guarantees, resulting in new state-of-the-art rates for DP-OPE and DP-OCO. We provide a summary in Table 13.1 and Figure 13.1.

L2P: a transformation from lazy to private algorithms (Section 13.3). Our main contribution is a new transformation, we call L2P, that allows converting any lazy algorithm into a private one with only a slight cost in regret. This allows us to use a long line of work on lazy online learning [KV05, GVW10, AT18, CYLK20, SK21, AKST23a] to design new algorithms for the private setting. Our transformation builds on two new techniques: first, we design a new switching rule that only depends on the loss at the current round, so as to minimize the privacy cost of each switching and mitigate the accumulation of privacy loss. Second, we rely on a simple, key observation that by grouping losses in a large batch, we can minimize the effect on the regret of lazy online learning algorithms. We introduce a new analysis for the regret of lazy online algorithms with a large batch size that improves over the existing analysis in [AFKT23b]; this allows us to reduce the total number of "fake switches" needed to guarantees privacy, improving the final regret.

Faster rates for DP-OPE (Section 13.3.1). As a first application, we use our transformation in the DP-OPE problem on the multiplicative weights algorithm [AHK12]. This results in a new algorithm for DP-OPE that has regret $\sqrt{T\log(d)} + T^{1/3}\log(d)/\varepsilon^{2/3}$, improving over the best existing results for the high-dimensional regime in which the regret is $\sqrt{T\log(d)} + T^{1/3}\log(d)/\varepsilon$ [AFKT23b].[1] The improvement is particularly crucial in the high-privacy regime, where $\varepsilon \ll 1$: indeed, our regret shows that (for $d = \mathsf{poly}(T)$) it is sufficient to set $\varepsilon \geq T^{-1/4}$ for matching the optimal non-private regret $\sqrt{T\log d}$, whereas previous results require a much larger $\varepsilon \geq T^{-1/6}$ to get privacy for free. This is also important in practice, when multiple applications of DP-OPE are necessary: using advanced

[1] [AFKT23b] has another algorithm which slightly improves over this regret in the high-privacy regime and obtains regret $T^{2/5}/\varepsilon^{4/5}$. We include this algorithm in Figure 13.1.

composition, our result shows that we can solve $K \approx \sqrt{T}$ instances of DP-OPE with $\varepsilon = 1$ and still obtain the non-private regret of order $\sqrt{T}$; in contrast, prior work only allows to solve $K \approx T^{1/3}$ instances while still attaining the non-private regret.

Faster rates for DP-OCO (Section 13.3.2). As another application, we use our transformation for DP-OCO with the regularized multiplicative weights algorithm of [AKST23b]. We obtain a new algorithm for DP-OCO that has regret $\sqrt{T} + T^{1/3}\sqrt{d}/\varepsilon^{2/3}$, improving over the best existing results that established regret $\sqrt{T} + T^{1/3}\sqrt{d}/\varepsilon + T^{3/8}\sqrt{d}/\varepsilon^{3/4}$ [AKST23b] or $d^{1/4}\sqrt{T}/\sqrt{\varepsilon}$ using DP-FTRL [KMS+21].

Lower bounds for low-switching private algorithms (Section 13.4). To understand the limitations of low-switching private algorithms, we prove a lower bound for the natural family of private algorithms with limited switching, showing that the upper bounds obtained via our reduction are nearly tight for this family of algorithms up to logarithmic factors. This shows that new techniques, beyond limited switching, are required in order to improve upon our upper bounds.

13.1.2 Related work

Lazy online learning. Our transformation and algorithms build on a long line of work in online learning with limited switching [KV05, GVW10, AT18, CYLK20, SK21, AKST23b]. As is evident from prior work in private online learning, the problems of lazy online learning and private online learning are tightly connected [AFKT23b, AFKT23a, AKST23a, AKST23b]. In this problem, the algorithm wishes to minimize its regret while making at most S switches: the algorithm can update the model at most S times throughout the T rounds. Recent work has resolved the lazy OPE problem: [AT18] show a lower bound of $\sqrt{T} + (T/S)\log(d)$ on the regret, which is achieved by several algorithms such as Follow-the-perturbed-leader [KV05] and the shrinking dartboard algorithm [GVW10]. For lazy OCO, however, optimal rates are yet to be known: [AKST23b] recently show that a lazy version of the regularized multiplicative weights algorithm obtains regret $\sqrt{T} + (T/S)\sqrt{d}$, whereas the best lower bound is $\sqrt{T} + T/S$ [SK21].

Differentially Private Stochastic Convex Optimization (DP-SCO). In the non-private setting, it is well known that Online Convex Optimization (OCO) is closely related to the problem of Stochastic Convex Optimization (SCO), where an algorithm is given n i.i.d. samples from some distribution P and aims to minimize a stochastic convex loss function, and the optimal rates for both problems are the same (e.g., [SS12, Haz16]). It is still unclear whether such a result holds in the private setting. While optimal algorithms are known to achieve (normalized) rate $1/\sqrt{n} + \sqrt{d}/(n\varepsilon)$ for DP-SCO [BST14, BFTGT19, FKT20, AFKT21, ALD21], the best algorithms for DP-OCO (Section 13.3.2) achieve a normalized regret $1/\sqrt{T} + \sqrt{d}/(T\varepsilon)^{2/3}$. Recently, for the case of stochastic adversaries (that choose ℓ_t i.i.d. from P), [AFKT23b] gave a reduction from DP-OCO to DP-SCO which shows that (up to logarithmic factors) both problems have the same rates in this case.

13.2 Preliminaries

13.2.1 Problem setup

We consider an interactive T-round game between an algorithm ALG and an oblivious adversary Adv. Before the interaction, the adversary Adv chooses T loss functions $\ell_1, \ldots, \ell_T \in \mathcal{L} = \{\ell \mid \ell : \mathcal{X} \to \mathbb{R}\}$. Then, at each round $t \in [T]$, the algorithm ALG, which observed $\ell_1, \cdots, \ell_{t-1}$ chooses $x_t \in \mathcal{X}$, and then the loss function ℓ_t chosen by Adv is revealed. The regret of the algorithm ALG is defined below:

$$\mathbf{Reg}_T(\mathsf{ALG}) := \sum_{t=1}^{T} \ell_t(x_t) - \min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x^*).$$

We study online optimization under the constraint that the algorithm is differentially private. For an algorithm ALG and a sequence $\mathcal{S} = (\ell_1, \ldots, \ell_T)$ chosen by an oblivious adversary Adv, we let $\mathsf{ALG}(\mathcal{S}) := (x_1, \ldots, x_T)$ denote the output of ALG over the loss sequence $\mathcal{S}$. We have the following definition of privacy against oblivious adversaries.[2]

[2]Our regret bound may be invalid with an adaptive adversary, but our algorithms will satisfy a stronger notion of differential privacy against adaptive adversaries (see [AFKT23b]). However, to keep the notation and analysis simpler, we limit our attention to privacy against oblivious adversaries.

Definition 13.2.1 (Differential privacy)**.** A randomized algorithm ALG is (ε, δ)-differentially private against oblivious adversaries ((ε, δ)-DP) if, for all neighboring sequences $\mathcal{S} = (\ell_1, \ldots, \ell_T) \in \mathcal{L}^T$ and $\mathcal{S}' = (\ell'_1, \ldots, \ell'_T) \in \mathcal{L}^T$ that differ in a single element, and for all events $\mathcal{O}$ in the output space of ALG, we have

$$\Pr[\mathsf{ALG}(\mathcal{S}) \in \mathcal{O}] \leq e^{\varepsilon} \Pr[\mathsf{ALG}(\mathcal{S}') \in \mathcal{O}] + \delta.$$

We focus on two important instances of differentially private online optimization:

(i) DP Online Convex Optimization (DP-OCO). In this problem, the adversary picks loss functions $\ell \in \mathcal{L}_{OCO} := \{\ell \mid \ell : \mathcal{X} \to \mathbb{R} \text{ is convex and } L\text{-Lipschitz}\}$ where $\mathcal{X} \subset \mathbb{R}^d$ is a convex set with diameter $D = \operatorname{diam}(\mathcal{X}) := \sup_{x,y \in \mathcal{X}} \|x - y\|$, and the algorithm chooses $x_t \in \mathcal{X}$. The goal of the algorithm is to minimize regret while being (ε, δ)-differentially private.

(ii) DP Online Prediction from Experts (DP-OPE). In this problem, the adversary picks loss functions $\ell \in \mathcal{L}_{OPE} = \{\ell \mid \ell : [d] \to [0, 1]\}$ where $\mathcal{X} = [d]$ is the set of d experts, and the algorithm chooses $x_t \in [d]$. The goal of the algorithm is to minimize regret while being (ε, δ)-differentially private.

13.2.2 Tools from differential privacy

Our analysis crucially relies on the following divergence between two distributions.

Definition 13.2.2 (δ-Approximate Max Divergence)**.** For two distributions μ and ν, we define

$$D_{\infty}^{\delta}(\mu \| \nu) := \sup_{S \subseteq \operatorname{supp}(\mu), \mu(S) \geq \delta} \ln \frac{\mu(S) - \delta}{\nu(S)}.$$

We let $D_{\infty}^{\delta}(\mu, \nu) := \max\{D_{\infty}^{\delta}(\mu \| \nu), D_{\infty}^{\delta}(\nu \| \mu)\}$.

We also use the notion of indistinguishability between two distributions.

Definition 13.2.3. ((ε, δ)-indistinguishability) Two distributions μ, ν are (ε, δ)-indistinguishable, denoted $\mu \approx_{(\varepsilon,\delta)} \nu$, if $D_{\infty}^{\delta}(\mu, \nu) \leq \varepsilon$.

Note that if an algorithm ALG has $\mathsf{ALG}(\mathcal{S}) \approx_{(\varepsilon,\delta)} \mathsf{ALG}(\mathcal{S}')$ for all neighboring datasets $\mathcal{S}, \mathcal{S}'$ then ALG is (ε, δ)-differentially private.

For our lower bounds, we require the notion of concentrated differential privacy. To this end, we first define the α-Renyi divergence ($\alpha > 1$) between two probability measures:

$$D_\alpha(\mu \| \nu) := \frac{1}{\alpha - 1} \log \left(\int \left(\frac{\mu(\omega)}{\nu(\omega)} \right)^\alpha \mathrm{d}\nu(\omega) \right).$$

Concentrated DP is defined below:

Definition 13.2.4 (concentrated DP)**.** Let $\rho \geq 0$. We say an algorithm ALG satisfies ρ-concentrated differential privacy (ρ-CDP) against oblivious adversaries if for any neighboring sequences $\mathcal{S} = (\ell_1, \ldots, \ell_T) \in \mathcal{L}^T$ and $\mathcal{S}' = (\ell'_1, \ldots, \ell'_T) \in \mathcal{L}^T$ that differ in a single element, and any $\alpha \geq 1$, $D_\alpha(\mathsf{ALG}(\mathcal{D}) \| \mathsf{ALG}(\mathcal{D}')) \leq \alpha\rho$.

Moreover, note that distributions with bounded D_∞^δ satisfy the following property.

Lemma 13.2.5. *Let $\varepsilon \leq 1/10$. If $D_\infty^\delta(\mu, \nu) \leq \varepsilon/2$ then we have*

$$\Pr_{X \sim \mu} \left[e^{-\varepsilon} \leq \frac{\mu(X)}{\nu(X)} \leq e^\varepsilon \right] \geq 1 - 6\delta/\varepsilon \text{ and } \Pr_{X \sim \nu} \left[e^{-\varepsilon} \leq \frac{\mu(X)}{\nu(X)} \leq e^\varepsilon \right] \geq 1 - 6\delta/\varepsilon.$$

Our results use standard results on group privacy and privacy composition.

Lemma 13.2.6. *(Group Privacy) Let ALG be an (ε, δ)-DP algorithm and let $\mathcal{S}, \mathcal{S}' \mathcal{L}^T$ be two datasets that differ in k elements. Then for any measurable set S in the output space of ALG*

$$\Pr[\mathsf{ALG}(\mathcal{S}) \in \mathcal{O}] \leq e^{k\varepsilon} \Pr[\mathsf{ALG}(\mathcal{S}') \in \mathcal{O}] + k e^{(k-1)\varepsilon} \delta.$$

Lemma 13.2.7 (Advanced Composition,[KOV15])**.** *For any $\varepsilon_t > 0, \delta_t \in (0, 1)$ for $t \in [k]$, and $\tilde{\delta} \in (0, 1)$, the class of $(\varepsilon_t, \delta_t)$-DP mechanisms satisfy $(\tilde{\varepsilon}_{\tilde{\delta}}, 1 - (1 - \tilde{\delta})\Pi_{t \in [k]}(1 - \tilde{\delta}_t))$-DP under k-fold adaptive composition, for*

$$\tilde{\varepsilon}_{\tilde{\delta}} = \sum_{t \in [k]} \varepsilon_t + \min \left\{ \sqrt{\sum_{t \in [k]} 2\varepsilon_t^2 \log(e + \frac{\sqrt{\sum_{t \in [k]} \varepsilon_t^2}}{\tilde{\delta}})}, \sqrt{\sum_{t \in [k]} 2\varepsilon_t^2 \log(1/\tilde{\delta})} \right\}. \tag{13.1}$$

We have the following standard conversion from ρ-CDP to (ε, δ)-DP.

Lemma 13.2.8 ([BS16]). *If* ALG *is ρ-CDP with $\rho \leq 1$, then it is $(3\sqrt{\rho \log(1/\delta)}, \delta)$-DP for all $\delta \in (0, 1/4)$.*

13.3 L2P: From Lazy to Private Algorithms for Online Learning

This section presents our L2P transformation, which turns lazy online learning algorithms into private ones. The transformation has an input algorithm $\mathcal{A}$ with measure μ_t at round t and samples x_t from the normalized measure $\overline{\mu}_t$, which satisfies the following condition:

Assumption 13.3.1. *The online algorithm $\mathcal{A}$ has at time t a measure μ_t that is a function of $\ell_1, \ldots, \ell_{t-1}$ (and density function $\overline{\mu}_t$) such that for some $\delta_0 \leq 1$ and $0 < \eta \leq 1/10$ that are data-independent, we have*

- $D_\infty^{\delta_0}(\overline{\mu}_{t+1}, \overline{\mu}_t) \leq \eta$,
- $\mu_{t+1}(x)/\mu_t(x) = \mathsf{func}(\ell_t, x)$ *for all $x \in \mathcal{X}$ where* func *is a data-independent function.*

While algorithms satisfying Assumption 13.3.1 need not be lazy, this assumption is satisfied by most existing lazy online learning algorithms such as the shrinking dartboard (Section 13.3.1) and lazy regularized multiplicative weights (Section 13.3.2). Moreover, any algorithm that satisfies this assumption can be made lazy via our reduction.

Technique Overview: Suppose the neighboring datasets differ from the s_0-th loss function. The high-level intuition behind our framework is that our algorithm only loses the privacy budget when it makes a switch (draws a fresh sample) whenever $t > s_0$. Hence, in the framework, we try to make the algorithm make as few switches as possible. This modification can lead to additional regret compared to lazy online learning algorithms, and we need to balance the privacy-regret trade-off. The family of low-switching algorithms is ideal for privatization because its built-in low-switching property can achieve a better trade-off.

Our starting point is the ideas in [AFKT23b, AKST23b] to privatize low-switching algorithms, which use correlated sampling to argue that a sample from $x_{t-1} \sim \overline{\mu}_{t-1}$ is

likely a good sample from $\overline{\mu}_t$ and therefore switching at round t is often not necessary. In particular, at round t, these algorithms sample a Bernoulli random variable $S_t \sim \mathsf{Ber}(c \cdot \overline{\mu}_t(x_{t-1})/\overline{\mu}_{t-1}(x_{t-1}))$ for some constant c and use the same model $x_t = x_{t-1}$ if $S_t = 1$, and otherwise sample new model $x_t \sim \overline{\mu}_t$ if $S_t = 0$ (which happens with small probability). This guarantees that the marginal probability of the lazy iterates remains the same as the original iterates. Finally, to preserve the privacy of the switching decisions, existing algorithms add a fake switching probability p where the algorithm switches independently of the input. To summarize, *existing* low-switching private algorithms work roughly as follows:

$$\begin{cases} \text{At each round } t\text{:} \\ - \text{ Sample } S_t \sim \mathsf{Ber}(C \cdot \overline{\mu}_t(x_{t-1})/\overline{\mu}_{t-1}(x_{t-1})) \text{ and } S'_t \sim \mathsf{Ber}(1-p) \\ - \text{ Sample new } x_t \sim \overline{\mu}_t \text{ if } S_t = 0 \text{ or } S'_t = 0 \\ - \text{ Otherwise set } x_t = x_{t-1} \end{cases}$$

This sketch is the starting point of our transformation, and we will introduce two new components to improve performance. The first component aims to avoid the accumulation of privacy cost for switching in the current approaches where each user can affect the switching probability for all subsequent rounds: this happens since $\overline{\mu}_t(x_{t-1})/\overline{\mu}_{t-1}(x_{t-1})$ is usually a function of the whole history $\ell_1, \ldots, \ell_t$, and hence the existing low-switching private algorithms lose the privacy budget even it does not make real switches. To address this, we deploy a new correlated sampling strategy in $\mathsf{L2P}$ where the loss ℓ_t at time t affects the switching probability only at time t, hence paying a privacy cost for switching only in a single round. To this end, we construct a parallel sequence of models $\{y_t\}_{t\in[T]}$ (independent of x_t) that is used for normalizing the ratio $\overline{\mu}_t(x_{t-1})/\overline{\mu}_{t-1}(x_{t-1})$ to become independent of the history. In particular, at round t, we switch with probability proportional to

$$\frac{\overline{\mu}_t(x_{t-1})}{\overline{\mu}_{t-1}(x_{t-1})} \cdot \frac{\overline{\mu}_{t-1}(y_{t-1})}{\overline{\mu}_t(y_{t-1})}.$$

The main observation here is that $\frac{\overline{\mu}_t(x_{t-1})}{\overline{\mu}_{t-1}(x_{t-1})} \cdot \frac{\overline{\mu}_{t-1}(y_{t-1})}{\overline{\mu}_t(y_{t-1})} = \frac{\mu_t(x_{t-1})}{\mu_{t-1}(x_{t-1})} \cdot \frac{\mu_{t-1}(y_{t-1})}{\mu_t(y_{t-1})}$ and this ratio

is a function of ℓ_t our input online learning algorithms which satisfies Assumption 13.3.1. This will, therefore, improve the privacy guarantee of the final algorithm.

The second main observation in L2P is that having a large batch size (batching rounds together) does not significantly affect the regret of lazy online algorithms compared to non-lazy algorithms but can further reduce the times to make switches and save the privacy budget. Our main novelty is a new analysis of the effect of batching on the regret of lazy algorithms (Proposition 13.3.3), which states that running a lazy online algorithm with a batch size of B would have an additive error of $TB^2\eta^2$ to the regret where η is a measure of distance between $\overline{\mu}_t$ and $\overline{\mu}_{t-1}$. This significantly improves over existing analysis by [AFKT23b, Theorem 2] which shows that batching can add an additive term of B/η to the regret.

Having reviewed our main techniques, we proceed to present the full details of our L2P transformation in Algorithm 44, denoting $\nu_s = \mu_{(s-1)B+1}$ where B is the batch size.

The regret of our transformation depends on the regret of its input algorithm. For the measure $\{\mu_t\}_{t=1}^T$, we denote its regret

$$\mathbf{Reg}_T(\{\mu_t\}_{t=1}^T) := \sum_{t=1}^T \mathop{\mathbb{E}}_{x_t\sim\overline{\mu}_t}[\ell_t(x_t)] - \min_{x\in\mathcal{X}} \sum_{t=1}^T \ell_t(x).$$

The following theorem summarizes the main guarantees of Algorithm 44.

Theorem 13.3.2. *Let $p \in (0,1)$ and $B \in \mathbb{N}$. Assuming Assumption 13.3.1, $Tp/B \geq 1$, and for any $\delta_1 > 0$ such that $\eta B \log(1/\delta_1)/p \leq 1$, our transformation* L2P *is (ε,δ)-DP with*

$$\varepsilon = \frac{2\eta}{p} + \eta + \frac{3T\eta^2 p\log(1/\delta_1)}{2B} + \sqrt{6T\eta^2 p\log^2(1/\delta_1)/B},$$
$$\delta = 2T(2/\eta + \log(1/\delta_1)/p)eB\delta_0 + 2T\delta_1,$$

and has regret

$$\mathbf{Reg}_T \leq \mathbf{Reg}_T(\{\mu_t\}_{t=1}^T) + O\left(TB^2\eta^2 + \frac{\delta_0 T^2\log(\frac{1}{\delta_1})}{\eta}\right).$$

We begin by proving the utility guarantees of our transformation. It will follow directly

Algorithm 44: L2P

Input: Parameter η, measures $\{\nu_t\}_{t\in[T]}$, batch size B, fake switching parameter p ;
Sample $x_1, y_1 \sim \overline{\nu}_1$;
Observe $\ell_1, \ldots, \ell_B$ and suffer loss $\sum_{i=1}^{B} \ell_i(x_1)$;
for $s = 2, \cdots, T/B$ **do**
    Sample $S_s \sim \mathsf{Ber}\left(\min\left(1, \frac{\nu_s(x_{s-1})}{e^{2B\eta}\nu_{s-1}(x_{s-1})} \cdot \frac{\nu_{s-1}(y_{s-1})}{\nu_s(y_{s-1})}\right)\right)$ and $S'_s \sim \mathsf{Ber}(1-p)$;
    if $S_s = 0$ *or* $S'_s = 0$ **then**
        Sample $x_s \sim \overline{\nu}_s$;
    end
    else
        Set $x_s = x_{s-1}$;
    end
    Sample $A_s \sim \mathsf{Ber}(1-p)$;
    if $A_s = 0$ **then**
        Sample $y_s \sim \overline{\nu}_s$;
    end
    else
        Set $y_s = y_{s-1}$;
    end
    Play x_s;
    Observe $\ell_{(s-1)B+1}, \ldots, \ell_{sB}$ and suffer loss $\sum_{i=(s-1)B+1}^{sB} \ell_i(x_s)$;
end

from the following proposition, which bounds the regret of running L2P over a lazy online learning algorithm.

Proposition 13.3.3 (Regret of Batched Lazy Algorithm)**.** *Let* ALG *be an online learning algorithm that satisfies Assumption 13.3.1. Let* $\eta B \log(1/\delta_1)/p \le 1$*, and* $\delta_1, \eta < 1/2$*. Then running* L2P *with the input algorithm* ALG *has regret*

$$\mathbf{Reg}_T \le \mathbf{Reg}_T(\{\mu_t\}_{t=1}^T) + O\left(TB^2\eta^2 + \frac{\delta_0 T^2 \log(\frac{1}{\delta_1})}{\eta}\right).$$

To prove Proposition 13.3.3, we first show that we can instead analyze the utility of a simpler algorithm that samples from $\overline{\nu}_s$ at each round. This is due to the following lemma, which shows that $\|\widehat{\nu}_s - \overline{\nu}_s\|_{TV}$ is small where $\widehat{\nu}_s$ is the marginal distribution of x_s in Algorithm 44.

Lemma 13.3.4. *Let* $\widehat{\nu}_s$ *be the marginal distribution of* x_s *in Algorithm 44. When* $\eta B \log(1/\delta_1)/p \le 1$*, we have*

$$\|\widehat{\nu}_s - \overline{\nu}_s\|_{TV} \le 3(s-1)(2e + \log(1/\delta_1)/p)B\delta_0.$$

We also require the following lemma which allows to build a coupling over multiple variables, such that the variables are as close as possible. This will be used to construct a coupling between the lazy algorithm and the L2P algorithm that runs it.

Lemma 13.3.5 ([AS19])**.** *Given a collection* S *of random variables, all absolutely continuous w.r.t. a common* σ*-finite measure. Then, there exists a coupling* Γ*, such that for any variables* $X, Y \in S$*, we have*

$$\Pr[X \ne Y] \le \frac{2\|X - Y\|_{TV}}{1 + \|X - Y\|_{TV}}.$$

We are now ready to prove Proposition 13.3.3

Proof. Let $\mathbf{Reg}'_T$ denote the regret when the marginal distribution of x_t is $\overline{\nu}_t$ instead of $\widehat{\nu}_t$

induced in the Algorithm. Since each loss function is bounded,

$$\mathbf{Reg}_T \le \mathbf{Reg}'_T + B \sum_{s\in[T/B]} \|\overline{\nu}_s - \widehat{\nu}_s\|_{TV}.$$

By Lemma 13.3.4, we have

$$\begin{aligned}\mathbf{Reg}_T &\le \mathbf{Reg}'_T + B \sum_{s\in[T/B]} 3(s-1)(2/\eta + \log(1/\delta_1)/p)eB\delta_0 \\ &\le \mathbf{Reg}'_T + 8T^2\delta_0 \log(1/\delta_1)/\eta.\end{aligned}$$

Thus, it now suffices to upper bound $\mathbf{Reg}'_T$.

Due to the preconditions that $D_\infty^{\delta_0}(\overline{\mu}_{i+1}, \overline{\mu}_i) \le \eta$ and $\delta_0 \le \eta$, we know $\|\overline{\mu}_{i+1} - \overline{\mu}_i\|_{TV} \le 2\eta$. Recall that we assume $x_s \sim \overline{\nu}_s$. Suppose z_i is the action taken by the input lazy algorithm $\mathcal{A}$ for $i \in [T]$ and the marginal distribution of z_i is $\overline{\mu}_i$. By Lemma 13.3.5, we can construct a coupling Γ_s between x_s and $\overline{z} := (z_{(s-1)B+1}, \cdots, z_{sB})$, such that

$$\Pr_{(x_s,\overline{z})\sim\Gamma_s}[\exists i \in [(s-1)B+1, sB], z_i \ne x_s] \le B\eta.$$

Letting $I_s = \mathbf{1}(\exists i \in [(s-1)B+1, sB], z_i \ne x_s)$, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{x_s\sim\overline{\nu}_s} \sum_{i=(s-1)B+1}^{sB} \ell_i(x_s) &= \mathop{\mathbb{E}}_{(x_s,\overline{z})\sim\Gamma_s} \sum_{i=(s-1)B+1}^{sB} \ell_i(x_s) \\ &= \mathop{\mathbb{E}}_{x_s,\overline{z}\sim\Gamma_s} (1-I_s) \sum_{i=(s-1)B+1}^{sB} \ell_i(z_i) \\ &\quad + \mathop{\mathbb{E}}_{x_s,\overline{z}\sim\Gamma_s} I_s \sum_{i=(s-1)B+1}^{sB} \ell_i(x_s) \\ &\le \mathop{\mathbb{E}}_{x_s,\overline{z}\sim\Gamma_s} (1-I_s) \sum_{i=(s-1)B+1}^{sB} \ell_i(z_i) \\ &\quad + \mathop{\mathbb{E}}_{x_s,\overline{z}\sim\Gamma_s} I_s \sum_{i=(s-1)B+1}^{sB} (\ell_i(z_i) + O(B\eta))\end{aligned}$$

$$\leq \mathop{\mathbb{E}}_{z_i \sim \overline{\mu}_i} \sum_{i=(s-1)B+1}^{sB} \ell_i(z_i) + O(B\eta \cdot B^2\eta).$$

Hence we get

$$\mathbf{Reg}'_T \leq \mathbf{Reg}_T(\{\mu_t\}_{t=1}^T) + \frac{T}{B} \cdot O(B^3\eta^2),$$

which completes the proof. □

Now we turn to prove the privacy of L2P. We begin with the following lemma, which provides the privacy guarantees of sampling a new model x_t from the distribution μ_t. We defer the proof to Appendix 13.6.

Lemma 13.3.6. *Let $\{\mu_t\}_{t=0}^T$ satisfy Assumption 13.3.1 where $\eta \leq 1/10$. Then for any neighboring sequences $\mathcal{S}$ and $\mathcal{S}'$ with corresponding $\{\mu_t\}_{t=0}^T$ and $\{\mu'_t\}_{t=0}^T$ that differ one loss function, we have*

$$D_\infty^{4\delta_0}(\overline{\mu}_t, \overline{\mu}'_t) \leq 2\eta.$$

We use correlated sampling in the algorithm rather than sampling from x_t directly. To this end, we need the following lemma, which provides upper and lower bounds on the ratio used for correlated sampling.

Lemma 13.3.7. *For any $s \in [T/B]$, if $\eta B \log(1/\delta_1)/p \leq 1$, then with probability at least $1 - (2/\eta + \log(1/\delta_1)/p) \cdot eB\delta_0 - \delta_1$,*

$$\frac{\nu_{s+1}(x_s)}{\nu_s(x_s)} \cdot \frac{\nu_s(y_s)}{\nu_{s+1}(y_s)} \in [e^{-2B\eta}, e^{2B\eta}].$$

One remaining issue is we need to conditional on the high probability events in Lemma 13.3.7 for the privacy guarantee and can not directly apply Advanced Composition (Lemma 13.2.7). Now, we modify the Advanced Composition for our usage. In the classic k-fold adaptive composition experiment, the adversary, after getting the first $i-1$ answers $Y_1, \cdots, Y_{i-1}$ (denoted by $Y_{[i-1]}$ for simplicity), can output two datasets D_i^0 and D_i^1, a query q_i, and receives the answer $Y_i \sim \mathcal{M}_i(D_i^b, q_i)$ for the secret bit $b \in \{0, 1\}$. If each $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-DP, then the joint distributions over the answers $Y_{[k]}$ satisfy the advanced composition theorem.

In our case, however, we know there exists a subset $G_{i-1}(D^b_{[i-1]})$, such that with probability at least $1-\lambda_i$, $Y_{[i-1]} \in G_{i-1}(D^b_{[i-1]})$. Conditional on $Y_{[i-1]} \in \cap_{b\in\{0,1\}} G_{i-1}(D^b_{[i-1]})$,

$$\mathcal{M}_i(D^0_i, q_i \mid Y_{[i-1]} \in \cap_{b\in\{0,1\}} G_{i-1}(D^b_{[i-1]})) \approx_{(\varepsilon_i,\delta_i)} \mathcal{M}_i(D^1_i, q_i \mid Y_{[i-1]} \in \cap_{b\in\{0,1\}} G_{i-1}(D^b_{[i-1]})) \tag{13.2}$$

Then we have the following lemma:

Lemma 13.3.8. *Given the k mechanisms satisfying the Condition (13.2), then the class of mechanisms satisfy $(\tilde{\varepsilon}_{\tilde{\delta}}, 1-(1-\tilde{\delta})\Pi_{t\in[k]}(1-\tilde{\delta}_t)) + 2\sum_{t\in[k]}\lambda_t$-DP under k-fold adaptive composition, with $\tilde{\varepsilon}_{\tilde{\delta}}$ defined in Equation (13.1).*

Proof. Without losing generality, suppose we know the adversary and how they generate the databases and queries. We can construct a series of mechanisms $\mathcal{M}'_i$, such that $\mathcal{M}'_i$ draws Y_i from $\mathcal{M}_i(D^b_i, q_i)$, and outputs Y_i if $Y_i \in \cap_{b\in\{0,1\}} G_{i-1}(D^b_{[i-1]})$, and outputs $\mathbf{0}$ otherwise. Let $(Y'_{1,b}, \cdots, Y'_{k,b})$ be the outputs of $\mathcal{M}'_i$ with secret bit b, and we know the TV distance between $(Y'_{1,b}, \cdots, Y'_{k,b})$ and $(Y_{1,b}, \cdots, Y_{k,b})$ is at most $\sum_{t\in[k]}\lambda_t$ for any $b \in \{0,1\}$. Moreover, we know

$$(Y'_{1,0}, \cdots, Y'_{k,0}) \approx_{\tilde{\varepsilon}_{\tilde{\delta}}, 1-(1-\tilde{\delta})\Pi_{t\in[k]}(1-\tilde{\delta}_t))} (Y'_{1,1}, \cdots, Y'_{k,1})$$

by the advanced composition. The basic composition finishes the proof. □

We are now ready to prove our main theorem.

Proof of Theorem 13.3.2. The regret bound follows directly from Proposition 13.3.3. It suffices to prove the privacy guarantee.

Fix two arbitrary neighboring datasets $\mathcal{S}$ and $\mathcal{S}'$, and suppose the sequences differ at s_0-step, that is $\{\ell_{(s_0-1)B+1}, \cdots, \ell_{s_0B}\}$ differ one loss function from $\{\ell'_{(s_0-1)B+1}, \cdots, \ell'_{s_0B}\}$.

Define ζ_s as the indicator that at least one of A_{s+1}, S_{s+1} and S'_{s+1} is zero. Let $\{(x_s, y_s, \zeta_s)\}_{s\in[T/B]}$ and $\{(x'_s, y'_s, \zeta'_s)\}_{s\in[T/B]}$ be the random variables with neighboring datasets. Let $\Sigma_s = \{(x_\tau, y_\tau, \zeta_\tau)\}_{\tau\in[s]}$ be the random variables for the first s-iterations. We will argue that

$\{(x_s, y_s, \zeta_s)\}_{s\in[T/B]}$ and $\{(x'_s, y'_s, \zeta'_s)\}_{s\in[T/B]}$ are indistinguishable, and privacy will follow immediately.

Let E_s be the event such that $\frac{\nu_{s+1}(x_s)}{\nu_s(x_s)} \cdot \frac{\nu_s(y_s)}{\nu_{s+1}(y_s)} \in [e^{-2B\eta}, e^{2B\eta}]$. Hence we know $\Pr[E_s] \geq 1 - (2 + \log(1/\delta_1)/p)eB\delta_0 - \delta_1$ by Lemma 13.3.7 for any $s \in [T/B]$. Define E'_s in a similar way. Moreover, let E_G be the event that $\sum_{s=2}^{T/B} \mathbf{1}(A_s = 0 \quad \text{or} \quad S'_s = 0) \leq 2Tp\log(1/\delta_1)/B$. By Chernoff bound, we know

$$\Pr(E_G) = \Pr[\sum_{s=2}^{T/B} \mathbf{1}(A_s = 0 \quad \text{or} \quad S'_s = 0) \leq 2Tp\log(1/\delta_1)/B] \geq 1 - \delta_1.$$

Then it suffices to show $\{(x_s, y_s, \zeta_s)\}_{s\in[T/B]}$ and $\{(x'_s, y'_s, \zeta'_s)\}_{s\in[T/B]}$ are (ε, δ_x)-indistinguishable conditional on $E := E_G \cup E'_G \cup_{s\in[T/B]} (E_s \cup E'_s)$. Then this will imply that $\{(x_s, y_s, \zeta_s)\}_{s\in[T/B]}$ and $\{(x'_s, y'_s, \zeta'_s)\}_{s\in[T/B]}$ are $(\varepsilon, \delta_x + (2T+2)\delta_1 + 2T(2/\eta + \log(1/\delta_1)/p)eB\delta_0)$-indistinguishable.

Now we show, conditional on E, any value Σ such that $\Sigma_{s-1} = \Sigma'_{s-1} = \Sigma$, (x_s, y_s, ζ_s) and (x'_s, y'_s, ζ'_s) are $(\varepsilon_s, \delta_0)$-indistinguishable where

$$\varepsilon_s = \begin{cases} 0, & s < s_0 \\ 2\eta/p & s = s_0 \\ \zeta_t \cdot \eta & s > s_0 \end{cases} \tag{13.3}$$

Case 1 $(s < s_0)$**:** It is clear that the claim is correct for $s \leq s_0$ as (x_s, y_s, ζ_s) and (x'_s, y'_s, ζ'_s) have the same distribution then.

Case 2 $(s = s_0)$**:** Now consider the case where $s = s_0$.

Note that (x_{s_0}, y_{s_0}) and (x'_{s_0}, y'_{s_0}) have identical distributions, and it suffices to consider the indistinguishability of ζ_{s_0} and ζ'_{s_0}.

We have

$$\begin{aligned} \frac{\Pr[\zeta_{s_0} = 0 \mid \Sigma_{s_0-1}, x_{s_0}, y_{s_0}]}{\Pr[\zeta'_{s_0} = 0 \mid \Sigma'_{s_0-1}, x'_{s_0}, y'_{s_0}]} &= \frac{(1-p)^2 \frac{\nu_{s_0+1}(x_{s_0})}{e^{2B\eta}\nu_{s_0}(x_{s_0})} \cdot \frac{\nu_{s_0}(y_{s_0})}{\nu_{s_0+1}(y_{s_0})}}{(1-p)^2 \frac{\nu'_{s_0+1}(x'_{s_0})}{e^{2B\eta}\nu'_{s_0}(x'_{s_0})} \cdot \frac{\nu'_{s_0}(y'_{s_0})}{\nu'_{s_0+1}(y'_{s_0})}} \\ &= \frac{\nu_{s_0+1}(x_{s_0})}{\nu'_{s_0+1}(x_{s_0})} \cdot \frac{\nu'_{s_0+1}(y_{s_0})}{\nu_{s_0+1}(y_{s_0})} \end{aligned}$$

$$\leq e^{2\eta}.$$

Similarly, we have

$$\begin{aligned}
\frac{\Pr[\zeta_{s_0}=1 \mid \Sigma_{s_0-1}, x_{s_0}, y_{s_0}]}{\Pr[\zeta'_{s_0}=1 \mid \Sigma'_{s_0-1}, x'_{s_0}, y'_{s_0}]} &= \frac{1-(1-p)^2+(1-p)^2(1-\frac{\nu_{s_0+1}(x_{s_0})}{e^{2B\eta}\nu_{s_0}(x_{s_0})}\frac{\nu_{s_0}(y_{s_0})}{\nu_{s_0+1}(y_{s_0})})}{1-(1-p)^2+(1-p)^2(1-\frac{\nu'_{s_0+1}(x'_{s_0})}{e^{2B\eta}\nu'_{s_0}(x'_{s_0})}\frac{\nu'_{s_0}(y'_{s_0})}{\nu'_{s_0+1}(y'_{s_0})})} \\
&= 1+\frac{(1-p)^2(\frac{\nu'_{s_0+1}(x'_{s_0})}{e^{2B\eta}\nu'_{s_0}(x'_{s_0})}\frac{\nu'_{s_0}(y'_{s_0})}{\nu'_{s_0+1}(y'_{s_0})}-\frac{\nu_{s_0+1}(x_{s_0})}{e^{2B\eta}\nu_{s_0}(x_{s_0})}\frac{\nu_{s_0}(y_{s_0})}{\nu_{s_0+1}(y_{s_0})})}{1-(1-p)^2+(1-p)^2(1-\frac{\nu'_{s_0+1}(x'_{s_0})}{e^{2B\eta}\nu'_{s_0}(x'_{s_0})}\frac{\nu'_{s_0}(y'_{s_0})}{\nu'_{s_0+1}(y'_{s_0})})} \\
&\leq 1+\frac{e^{2\eta}-1}{p} \leq e^{2\eta/p}.
\end{aligned}$$

Case 3 ($s > s_0$)**:** As for the case when $s > s_0$, when $\zeta_s = 0$ ($A_{s+1} = S_{s+1} = S'_{s+1} = 1$), the variables are 0-indistinguishable since $x_s = x_{s-1}$ and $y_s = y_{s-1}$ in this case. Consider the remaining possibility. Given the assumption that μ_{t+1}/μ_t is a function of ℓ_t, for any possible Σ, we have

$$\Pr[\zeta_s = 1 \mid \Sigma_{s-1} = \Sigma] = \Pr[\zeta'_s = 1 \mid \Sigma'_{s-1} = \Sigma].$$

For any set S, by the assumption on $\overline{\mu}_s$, we have

$$\begin{aligned}
&\Pr[\zeta_s = 1, (x_s, y_s) \in S \mid \Sigma_{s-1} = \Sigma] \\
=&\Pr[(x_s, y_s) \in S \mid \Sigma_{s-1} = \Sigma, \zeta_s = 1] \Pr[\zeta_s = 1 \mid \Sigma_{s-1} = \Sigma] \\
=&\Pr[(x_s, y_s) \in S \mid \Sigma_{s-1} = \Sigma, \zeta_s = 1] \Pr[\zeta'_s = 1 \mid \Sigma'_{s-1} = \Sigma] \\
\leq&e^{2\eta} \Pr[(x'_s, y'_s) \in S \mid \Sigma'_{s-1} = \Sigma, \zeta'_s = 1] \Pr[\zeta'_s = 1 \mid \Sigma'_{s-1} = \Sigma] + 4\delta_0 \\
=&e^{2\eta} \Pr[\zeta'_s = 1, (x'_s, y'_s) \in S \mid \Sigma'_{s-1} = \Sigma] + 4\delta_0,
\end{aligned}$$

where the inequality comes from Lemma 13.3.6 by the divergence bound between $\overline{\mu}_t$ and $\overline{\mu}'_t$. This completes the proof of Equation (13.3).

The final privacy guarantee follows from combining Equation (13.3) and the modified Advanced composition (Lemma 13.3.8). □

13.3.1 Application to DP-OPE

This section discusses the first application of our transformation to differentially private online prediction from experts (DP-OPE). Towards this end, we apply our transformation over the multiplicative weights algorithms [AHK12], which can be made lazy as done in the shrinking dartboard algorithm [GVW10]. It has the following measure at round t

$$\mu_t^{\mathsf{mw}}(x) = e^{-\eta \sum_{i=1}^{t-1} \ell_i(x)}. \tag{13.4}$$

The following proposition shows that this measure satisfies the desired properties required by our transformation. We let $\overline{\mu}_t^{\mathsf{mw}}$ denote the density corresponding to μ_t^{mw}.

Lemma 13.3.9. *Assume $\ell_1, \ldots, \ell_T$ where $\ell_t : [d] \to [0,1]$. Then we have that*

1. $D_\infty^{\delta_0}(\overline{\mu}_{t+1}^{\mathsf{mw}}, \overline{\mu}_t^{\mathsf{mw}}) \le \eta$ *with* $\delta_0 = 0$.

2. $\frac{\mu_{t+1}^{\mathsf{mw}}(x)}{\mu_t^{\mathsf{mw}}(x)} = e^{-\eta \ell_t(x)}$ *for all* $x \in [d]$.

Proof. The first item follows from the guarantees of the exponential mechanism as $\ell_t(x) \in [0,1]$ for all $x \in [d]$. The second item follows immediately from the definition of μ^{mw}. □

Having proved our desired properties, our transformation now gives the following theorem.

Theorem 13.3.10 (DP-OPE)**.** *Let $\ell_1, \ldots, \ell_T$ where $\ell_t : [d] \to [0,1]$. Setting $B = 1/\varepsilon$ and $\eta = \min(\varepsilon_0, \varepsilon)^{2/3}/T^{1/3}$ where $\varepsilon_0 = T^{-1/4} \log^{3/4} d$, the* L2P *transformation (Algorithm 44) applied with the measure $\{\mu_t^{\mathsf{mw}}\}_{t=1}^T$ is (ε, δ)-DP and has regret*

$$\mathbf{Reg}_T = O\left(\sqrt{T \log d} + \frac{T^{1/3} \log d}{\varepsilon^{2/3}}\right).$$

Proof. First, based on theorem 13.3.2, note that the setting of $B = 1/\varepsilon$ and $\eta \le \min(\varepsilon_0, \varepsilon)^{2/3}/T^{1/3}$ where $\varepsilon_0 = T^{-1/4} \log^{3/4} d$ guarantee the algorithm is (ε, δ)-DP.

To upper bound the regret, we use existing guarantees of the multiplicative weights algorithm [AHK12], combined with Theorem 13.3.2 to get that the regret is

$$\begin{aligned}\mathbf{Reg}_T &\le O\left(\eta T + \frac{\log(d)}{\eta} + TB^2\eta^2\right)\\ &\le O\left(\eta T + \frac{\log(d)}{\eta} + \frac{T\eta^2}{\varepsilon^2}\right)\\ &\le O\left((T\varepsilon_0)^{2/3} + \frac{T^{1/3}\log(d)}{\varepsilon^{2/3}} + \frac{T^{1/3}}{\varepsilon^{2/3}}\right)\\ &\le O\left(\sqrt{T\log d} + \frac{T^{1/3}\log(d)}{\varepsilon^{2/3}}\right),\end{aligned}$$

where the second inequality follows by setting $B = 1/\varepsilon$, and the third inequality follows by setting $\eta \le \min(\varepsilon_0, \varepsilon)^{2/3}/T^{1/3}$, and the last inequality follows since $\varepsilon_0 = T^{-1/4}\log^{3/4} d$. □

13.3.2 Application to DP-OCO

In this section, we use our transformation for differentially private online convex optimization (DP-OCO) using the regularized multiplicative weights algorithm [AKST23b], which has the following measure

$$\mu_t^{\mathsf{rmw}}(x) = e^{-\beta\left(\sum_{i=1}^{t-1}\ell_i(x) + \lambda\|x\|_2^2\right)}. \tag{13.5}$$

Letting $\overline{\mu}^{\mathsf{rmw}}$ denote the corresponding density function, we have the following properties.

Lemma 13.3.11. *Assume $\ell_1, \dots, \ell_T : \mathcal{X} \to \mathbb{R}$ be convex and L-Lipschitz functions. Then we have that*

1. $D_\infty^{\delta_0}(\overline{\mu}_{t+1}^{\mathsf{rmw}}, \overline{\mu}_t^{\mathsf{rmw}}) \le \eta$ *where* $\eta = \frac{2\beta L^2}{\lambda} + \sqrt{\frac{8\beta L^2\log(2/\delta_0)}{\lambda}}$.
2. $\frac{\mu_{t+1}^{\mathsf{rmw}}(x)}{\mu_t^{\mathsf{rmw}}(x)} = e^{-\beta\ell_t(x)}$ *for all* $x \in \mathcal{X}$.

Proof. The first item follows from Lemma 3.5 in [GLL22, AKST23b]. The second item follows immediately from the definition of μ_t^{rmw}. □

Combining these properties with our transformation, we get the following result.

Theorem 13.3.12 (DP-OCO). *Let $\ell_1, \dots, \ell_T : \mathcal{X} \to \mathbb{R}$ be convex and L-Lipschitz functions. Setting $B = \frac{1}{2\varepsilon \log(1/\delta)}$, $\lambda = \frac{L}{D}\max\{\sqrt{T}, \frac{\sqrt{d \log T}}{\eta}\}$, $\beta = \eta^2\lambda/20L^2$, $\eta = \frac{\varepsilon^{2/3}}{T^{1/3}\log(T/\delta)}$ and $p = \eta/\varepsilon$, the L2P transformation (Algorithm 44) applied with the measure $\{\mu_t^{\mathsf{rmw}}\}_{t=1}^T$ is (ε, δ)-DP and has regret*

$$\mathbf{Reg}_T = LD \cdot O\left(\sqrt{T} + \frac{T^{1/3}\sqrt{d \log T}\log(T/\delta)}{\varepsilon^{2/3}}\right).$$

Proof. First, based on Theorem 13.3.2, note that there are three constraints to make the algorithm private:

$$\eta/p \le \varepsilon/2, \qquad \eta\sqrt{Tp\log(1/\delta)/B} \le \varepsilon/2, \qquad \eta B \log(1/\delta)/p \le 1.$$

Setting of $B = \frac{1}{2\varepsilon \log(1/\delta)}$, $\lambda = \frac{L}{D}\max\{\sqrt{T}, \frac{\sqrt{d \log T}}{\eta}\}$, $\beta = \eta^2\lambda/20L^2$, $\eta = \frac{\varepsilon^{2/3}}{T^{1/3}\log(T/\delta)}$ and $p = \eta/\varepsilon$ guarantees the algorithm is (ε, δ)-DP.

For utility, we use theorem 13.3.2 with the existing regret bounds for the regularized multiplicative weights algorithm (Theorem 4.1 in [AKST23b]) to get that the algorithm has regret

$$\begin{aligned}
\mathbf{Reg}_T &\le O\left(\lambda D^2 + \frac{L^2 T}{\lambda} + \frac{d\log(T)}{\beta} + LDTB^2\eta^2\right) \\
&\le O\left(LD\sqrt{T} + \lambda D^2 + \frac{L^2 d \log T}{\lambda\eta^2} + LDTB^2\eta^2\right) \\
&\le LD \cdot O\left(\sqrt{T} + \frac{T^{1/3}\sqrt{d\log T}\log(T/\delta)}{\varepsilon^{2/3}}\right).
\end{aligned}$$

□

13.4 Lower bound for low-switching private algorithms

In this section, we prove a lower bound for DP-OPE for a natural family of private low-switching algorithms that contains most of the existing low-switching private algorithms such as our algorithms and the ones in [AFKT23b, AKST23b]. Our lower bound matches our upper bounds for DP-OPE and suggests that new techniques beyond limited switching

are required in order to obtain faster rates.

For our lower bounds, we will assume that the algorithm satisfies the following condition:

Condition 13.4.1. *(Limited switching algorithms) The online algorithm* ALG *works as follows: at each round t,* ALG *is allowed to either set $x_{t+1} = x_t$ or sample $x_{t+1} \sim \mu_{t+1}$ where μ_{t+1} is a function of $\ell_1, \ldots, \ell_t$ and is supported over $\mathcal{X}$. The algorithm releases the resampling rounds $\{t_1, \ldots, t_S\}$ and models $\{x_{t_1}, \ldots, x_{t_S}\}$.*

Our lower bound will hold for algorithms that satisfy concentrated differential privacy. We use this notion as it allows to get tight characterization of the composition of private algorithms and in most settings have similar rates to approximate differential privacy. We can also prove a tight lower bound for pure differential privacy using the same techniques.

Theorem 13.4.2. *Let $T \geq 1$ and $\varepsilon \geq 100 \log^{3/2}(dT)/T$. If an algorithm* ALG *satisfies Condition 13.4.1 and is ε^2-CDP, then there exists an oblivious adversary that chooses $\ell_1, \ldots, \ell_T : [d] \to [0, 1]$ such that the regret is lower bounded by*

$$\mathbf{Reg}_T \geq \Omega\left(\sqrt{T} + \frac{T^{1/3}}{\varepsilon^{2/3}}\right).$$

We prove a sequence of lemmas that are needed for the proof. The first lemma shows that the algorithm has to split the privacy budget across all resampling rounds. To this end, let S be a random variable that corresponds to the number of resampling steps in the algorithm, let T_i be the random variable corresponding to the round of the i'th resampling (where we let $T_i = T + 1$ if $i > S$), and let Z_i be the random variable corresponding to the model sampled at time T_i (letting $Z_i = 1$ if $i > S$).

Lemma 13.4.3. *(Composition) Let S, T_i, Z_i and S', T_i', Z_i' denote the random variables for two neighboring datasets. Under the assumptions of Theorem 13.4.2, if* ALG *is ε^2-CDP, then for all $\alpha \geq 1$*

$$\sum_{i=1}^{T} D_\alpha(Z_i || Z_i' \mid T_i) \leq \alpha\varepsilon^2.$$

Proof. As ALG is ε^2-concentrated DP and outputs $T_1, \ldots, T_S$ and $Z_1, \ldots, Z_S$, we have that

$$\begin{aligned}\alpha\varepsilon^2 &\geq D_\alpha(T_1, Z_1, \ldots, T_S, Z_S || T'_1, Z'_1, \ldots, T'_{S'}, Z'_{S'}) \\ &\geq D_\alpha(T_1, Z_1, \ldots, T_T, Z_T || T'_1, Z'_1, \ldots, T'_T, Z'_T) \\ &\geq \sum_{i=1}^{T} D_\alpha(Z_i || Z'_i \mid T_i),\end{aligned}$$

where the second inequality follows as the random variables T_i, Z_i and T'_j, Z'_j are constant for $i > S$ and $j > S'$, and the last inequality follows as Z_i is independent of $(T_1, \ldots, T_i)$ and $(Z_1, \ldots, Z_{i-1})$ given T_i. □

We defer the proof of the following Lemma to the appendix.

Lemma 13.4.4. *Let $T \geq 1$, $\varepsilon \leq 1/T$ and $\delta \leq 1/2$. Assume $\ell : [d] \to \{0, 1\}$ where $\ell[x] \sim \mathsf{Ber}(1/2)$ for each $x \in [d]$. Let $D = (\ell, \ldots, \ell)$ and let ALG be an (ε, δ)-DP algorithm that outputs $(z_1, \ldots, z_T) = \mathsf{ALG}(D)$. Then*

$$\mathbb{E}[\sum_{t=1}^{T} \ell(z_t)] \geq T \cdot \left(\frac{1}{2} - \frac{T\varepsilon}{2}\right) - \frac{T^2 d\delta}{2}.$$

We are now ready to prove our main lower bound.

Proof. (of Theorem 13.4.2) We consider the following construction for the lower bound: the adversary sets $S_{\mathsf{adv}} = (T\varepsilon)^{2/3}$, the sequence of losses will have $E = S_{\mathsf{adv}}^2$ epochs, each of size $B = T/E = T/(T\varepsilon)^{4/3} = \frac{1}{(T\varepsilon)^{1/3}\varepsilon}$. Inside each epoch, the adversary samples $\ell \sim \mathsf{Ber}(1/2)^d$ and plays the same loss function for the whole epoch.

Let S be random variable denoting the number of switches in the algorithm. In this case, we argue that each switch must have a small privacy budget (Lemma 13.4.3), and thus, the price inside each epoch has to be large (Lemma 13.4.4). Let $T_1, \ldots, T_S$ be the rounds where the algorithm resamples ($T_i = T + 1$ for $i > S$) and let $Z_1, Z_2, \ldots, Z_S$ be the resampled models ($Z_i = 1$ for $i > S$). Lemma 13.4.3 implies that

$$\sum_{i=1}^{T} D_\alpha(Z_i || Z'_i \mid T_i) \leq \alpha\varepsilon^2.$$

Now note that inside an epoch e, if the algorithm does not switch, then it will suffer loss $B/2$ in that epoch. Otherwise, if it switches, assume without loss of generality there is at most one switch inside each epoch (see Lemma 13.4.4). Let $j_e \in [T]$ denote the index such that Z_{j_e} was sampled in epoch e. Note that the algorithm in this epoch has $D_\alpha(Z_{j_e} || Z'_{j_e} \mid T_{j_e}) = \alpha\varepsilon_e^2$, hence it is ε_e^2-CDP. Standard conversion from concentrated DP to approximate DP (Lemma 13.2.8) implies that it is $(3\varepsilon_e\sqrt{\log(1/\delta)}, \delta)$-DP where $\delta \leq 1/T^3 d$. Hence Lemma 13.4.4 implies the error for this epoch is $B \cdot \left(\frac{1}{2} - \frac{3B\varepsilon_e\sqrt{\log(1/\delta)}}{2}\right) - 1/T$. Letting $E_{switch} \subset [E]$ denote the epochs where there is a switch, we have that the loss of the algorithm is

$$\begin{aligned}
L(\mathsf{ALG}) &:= \mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t)] \\
&= \mathbb{E}\left[\sum_{e \notin E_{switch}} \frac{B}{2}\right] \\
&\quad + \mathbb{E}\left[\sum_{e \in E_{switch}} B\left(\frac{1}{2} - \frac{3B\varepsilon_e\sqrt{\log(1/\delta)}}{2}\right) - 1/T\right] \\
&= \mathbb{E}\left[(E-S)\frac{B}{2} + S\frac{B}{2} - \sum_{e \in E_{switch}} \frac{3B^2\varepsilon_e\sqrt{\log(1/\delta)}}{2} - 1\right] \\
&= T/2 - 1 - \frac{3B^2\sqrt{\log(1/\delta)}}{2}\,\mathbb{E}\left[\sum_{e \in E_{switch}} \varepsilon_e\right] \\
&\geq T/2 - 1 - \frac{3B^2\sqrt{E\log(1/\delta)}\varepsilon}{2},
\end{aligned}$$

where the last inequality follows since $\sum_{e \in E_{switch}} \varepsilon_e \leq \sqrt{E\sum_{e=1}^{E}\varepsilon_e^2} \leq \sqrt{E}\varepsilon$. Note also that the loss of the best expert is

$$L^\star := \min_{x \in [d]} \sum_{t=1}^{T} \ell_t(x) = T/2 - \sqrt{E}B$$

Overall we get that the regret of the algorithm is

$$L(\mathsf{ALG}) - L^\star \geq \sqrt{E}B - \frac{3B^2\sqrt{E\log(1/\delta)}\varepsilon}{2} - 1$$

$$
\begin{aligned}
&\geq (T\varepsilon)^{2/3}\frac{T}{(T\varepsilon)^{4/3}} - \frac{3\sqrt{\log(1/\delta)}}{2(T\varepsilon)^{2/3}\varepsilon^2}\sqrt{E}\varepsilon - 1 \\
&= \frac{T^{1/3}}{\varepsilon^{2/3}} - \frac{3\sqrt{\log(1/\delta)E}}{2(T\varepsilon)^{2/3}\varepsilon} - 1 \\
&\overset{(i)}{\geq} \frac{T^{1/3}}{\varepsilon^{2/3}} - \frac{3\sqrt{\log(1/\delta)}}{2\varepsilon} - 1 \\
&\overset{(ii)}{=} \Omega\left(\frac{T^{1/3}}{\varepsilon^{2/3}}\right),
\end{aligned}
$$

where (i) follows since $E \leq (T\varepsilon)^{4/3}$, and (ii) holds since $\frac{3\sqrt{\log(1/\delta)}}{2\varepsilon} \leq \frac{T^{1/3}}{2\varepsilon^{2/3}}$ for $\varepsilon \geq 100\log^{3/2}(dT)/T \geq 27\log^{3/2}(1/\delta)/T$. The claim follows. □

Finally, we note that this lower bound only holds for switching-based algorithms: indeed, the binary-tree-based algorithm of [AS17] obtains regret $\sqrt{d}\log(d)/\varepsilon$ which is better in the low-dimensional regime. This motivates the search for new strategies beyond limited switching for the high-dimensional regime.

13.5 Conclusion

In this paper, we proposed a new transformation that allows the conversion of lazy online learning algorithms into private algorithms and demonstrates two applications (DP-OPE and DP-OCO) where this transformation offers significant improvements over prior work. Moreover, for DP-OPE, we show a lower bound for natural low-switching-based private algorithms, which shows that new techniques are required for low-switching algorithms to improve our transformation's regret. This begs the question of whether the same lower bound holds for all algorithms or whether a different strategy that breaks the low-switching lower bound exists. As for DP-OCO, it is interesting to see whether better upper or lower bounds can be obtained. The current normalized regret, omitting logarithmic terms, is proportional to $\sqrt{d}/(\varepsilon T)^{2/3}$. This is different than most applications in private optimization where the normalized error is usually a function of $\sqrt{d}/(\varepsilon T)$. Hence, it is natural to conjecture that the normalized regret can be improved to $d^{1/3}/(\varepsilon T)^{2/3}$.

13.6 Missing Proofs for Section 13.3

13.6.1 Proof of Lemma 13.3.4

We prove this statement by induction. For $t = 1$, the statement is obviously correct. We assume $\|\widehat{\nu}_t - \overline{\nu}_t\|_{TV} \le 3(t-1)(2(e/\eta + \log(1/\delta_1)/p)B\delta_0 + \delta_1)$ prove that $\|\widehat{\nu}_{t+1} - \overline{\nu}_{t+1}\|_{TV} \le 3t(2e + \log(1/\delta_0)/p)B\delta_0$.

Let $X_{good} := \{x : \log \frac{\overline{\nu}_{t+1}(x)}{\overline{\nu}_t(x)} \in [-B\eta, B\eta]\}$ and $Y_{good} := \{y : \log \frac{\overline{\nu}_t(y)}{\overline{\nu}_{t+1}(y)} \le [-B\eta, B\eta]\}$. Let $\widehat{\varphi}_t(y)$ be the distribution of y_t. Note that the distribution of y_t is independent of $\{x_\tau\}_{\tau\in[T/B]}$, while the distribution of x_{t+1} is independent of y_{t+1} but depends on y_t. By the assumption and group privacy, we know $D_\infty^{Be^{B\eta}\delta_0}(\overline{\nu}_{t+1}, \overline{\nu}_t) \le B\eta$, and hence we have

$$\nu_t(Y_{good}^{\complement}) \le e^{B\eta}\delta_0/\eta \le 2e\delta_0/\eta.$$

Let $t_0 \le t$ be largest integer such that $A_{t_0} = 1$, that is, y_t is sampled from $\overline{\nu}_{t_0}$ for some random $t_0 \le t$. We have

$$\nu_{t_0}(Y_{good}^{\complement}) \le e^{B\eta(t-t_0)} \cdot \nu_t(Y_{good}) + (t-t_0)B\delta_0 e^{B\eta(t-t_0)}.$$

With probability at least $1 - \delta_1$, we know $|t - t_0| \le \log(1/\delta_1)/p$. Hence we get

$$\Pr_{y\sim\widehat{\varphi}_t}[y \in Y_{good}] \ge 1 - 2(e/\eta + \log(1/\delta_1)/p)B\delta_0 - \delta_1.$$

We know

$$\begin{aligned}
&\Pr_{x\sim\overline{\nu}_t, y\sim\widehat{\varphi}_t}[x \in X_{good}, y \in Y_{good}] \\
&= \Pr_{y\sim\widehat{\varphi}_t}[y \in Y_{good}] \Pr_{x\sim\overline{\nu}_t}[x \in X_{good} \mid y \in Y_{good}] \\
&\ge 1 - 2(e/\eta + \log(1/\delta_1)/p)B\delta_0 - \delta_1.
\end{aligned}$$

Denote the good set

$$S_{good} = \big\{(x, y) : x \in X_{good}, Y_{good}\big\}.$$

Let $\tilde{\varphi}_t$ be the distribution of y_t conditional on $y_t \in Y_{good}$. Let $\widehat{\Gamma}_t$ be the marginal distribution over (x_t, y_t), that is $x_t \sim \widehat{\nu}_t$ and $y_t \sim \widehat{\varphi}_t$. Let Γ_t be the distribution over (x_t, y_t) where $x_t \sim \overline{\nu}_t, y_t \sim \tilde{\varphi}_t$, and $\overline{\Gamma}_t$ be the distribution of Γ_t conditional on $(x_t, y_t) \in S_{good}$.

We know $\|\widehat{\Gamma}_t - \overline{\Gamma}_t\|_{TV} \leq (2e/\eta + \log(1/\delta_1)/p)B\delta_0(3t-2)$. Let $\overline{q}_{t+1}$ be the distribution of x_{t+1} if (x_t, y_t) is sampled from $\overline{\Gamma}_t$ instead of $\widehat{\Gamma}_t$. By the property that post-processing does not increase the TV distance, we know

$$\|\overline{q}_{t+1} - \widehat{\nu}_{t+1}\|_{TV} \leq \|\widehat{\Gamma}_t - \overline{\Gamma}_t\|_{TV}.$$

Now it suffices to bound the TV distance between $\overline{q}_{t+1}$ and $\overline{\nu}_{t+1}$.

For any set E, we have

$$\begin{aligned}
\overline{q}_{t+1}(E) = & \int (\Pr[S'_t = 0, x_{t+1} \in E \mid x_t = x, y_t = y] \\
& + \Pr[S'_t = 1, S_t = 0, x_{t+1} \in E \mid x_t = x, y_t = y] \\
& + \Pr[S'_t = 1, S_t = 1, x_{t+1} \in E \mid x_t = x, y_t = y])\overline{\Gamma}_t(x, y)\mathrm{d}(x, y) \\
= & \, p\overline{\nu}_{t+1}(E) + (1-p)\overline{\nu}_{t+1}(E) \int (1 - \frac{\nu_{t+1}(x)}{e^{2B\eta}\nu_t(x)} \cdot \frac{\nu_t(y)}{\nu_{t+1}(y)})\overline{\Gamma}_t(x, y)\mathrm{d}(x, y) \\
& + (1-p) \int_{\mathbf{1}(x \in E)} \frac{\nu_{t+1}(x)}{e^{2B\eta}\nu_t(x)} \cdot \frac{\nu_t(y)}{\nu_{t+1}(y)} \overline{\Gamma}_t(x, y)\mathrm{d}(x, y).
\end{aligned}$$

Thus we have

$$\begin{aligned}
|\overline{q}_{t+1}(E) - \overline{\nu}_{t+1}(E)| \leq & \Big| \int_{\mathbf{1}(x \in E)} \frac{\overline{\nu}_{t+1}(x)}{e^{2B\eta}\overline{\nu}_t(x)} \cdot \frac{\overline{\nu}_t(y)}{\overline{\nu}_{t+1}(y)} \overline{\Gamma}_t(x, y)\mathrm{d}(x, y) \\
& - \overline{\nu}_{t+1}(E) \int \frac{\overline{\nu}_{t+1}(x)}{e^{2B\eta}\overline{\nu}_t(x)} \cdot \frac{\overline{\nu}_t(y)}{\overline{\nu}_{t+1}(y)} \overline{\Gamma}_t(x, y)\mathrm{d}(x, y) \Big|.
\end{aligned}$$

Note that for any $(x, y) \in S_{good}$, we have

$$\overline{\Gamma}_t(x, y) = \frac{\overline{\nu}_t(x)\tilde{\varphi}_t(y)}{\Gamma_t(S_{good})}.$$

Fixing any y, we know the above term is bounded by

$$\left|\frac{\overline{\nu}_t(y)}{e^{2B\eta}\overline{\nu}_{t+1}(y)\Gamma_t(S_{good})}(\overline{\nu}_{t+1}(E\cap X_{good})-\overline{\nu}_{t+1}(E)\overline{\nu}_{t+1}(X_{good}))\right| \le 2(e/\eta+B\log(1/\delta_1)/p)\delta_0,$$

where the last inequality follows from $\overline{\nu}_{t+1}(X_{good}) \ge 1-B\delta_0$. Hence, we prove that

$$\|\overline{q}_{t+1}-\overline{\nu}_{t+1}\|_{TV} \le 2(e/\eta+\log(1/\delta_1)/p)B\delta_0.$$

This suggests that

$$\|\widehat{\nu}_{t+1}-\overline{\nu}_{t+1}\|_{TV} \le \|\widehat{\nu}_{t+1}-\overline{q}_{t+1}\|_{TV}+\|\overline{q}_{t+1}-\overline{\nu}_{t+1}\|_{TV} \le 6t(e/\eta+\log(1/\delta_1)/p)B\delta_0.$$

13.6.2 Proof of Lemma 13.3.6

Let $\mathcal{S}=(\ell_1,\dots,\ell_T)$ and $\mathcal{S}'=(\ell'_1,\dots,\ell'_T)$ differ in a single round t_0. We fix t and prove the claim is correct. If $t \le t_0$, then the claim clearly holds as $\overline{\mu}_t=\overline{\mu}'_t$. For $t=t_0+1$, note that Assumption 13.3.1 implies that $D_\infty^{\delta_0}(\overline{\mu}_{t_0+1},\overline{\mu}_{t_0}) \le \eta$ and $D_\infty^{\delta_0}(\overline{\mu}'_{t_0+1},\overline{\mu}_{t_0}) \le \eta$, hence by group privacy we get that $D_\infty^{(e^\eta+1)\delta_0}(\overline{\mu}_t,\overline{\mu}'_t) \le 2\eta$. Finally, for $t>t_0+1$, note that Assumption 13.3.1 implies that $\mu_t=\mu_0\cdot\mathsf{func}(\ell_1)\cdot\mathsf{func}(\ell_2)\cdots\mathsf{func}(\ell_{t-1})$ and $\mu'_t=\mu_0\cdot\mathsf{func}(\ell'_1)\cdot\mathsf{func}(\ell'_2)\cdots\mathsf{func}(\ell'_{t-1})$. Thus, swapping the losses at rounds $t-1$ and t_0 results in the same distributions μ_t and μ'_t, therefore privacy follows from the same arguments as the case when $t=t_0+1$. The final claim follows as $e^\eta+1\le 4$.

13.6.3 Proof of Lemma 13.3.7

To prove lemma 13.3.7, we first prove the same result under a simpler setting where $x_t\sim\nu_t$ and $y_t\sim\nu_t$.

Lemma 13.6.1. *For any $0\le t\le T/B-1$, if $B\eta\le 1/20$, $x_t\sim\nu_t$ and $y_t\sim\nu_t$ independently, then with probability at least $1-6e^{B\eta}\delta_0/\eta$,*

$$\frac{\nu_{t+1}(x_t)}{\nu_t(x_t)}\cdot\frac{\nu_t(y_t)}{\nu_{t+1}(y_t)} \in [e^{-2B\eta},e^{2B\eta}]$$

Proof. Let $Z_t = \int \nu_t(x)\mathrm{d}x$. We know $\overline{\nu}_t = \nu_t/Z_t$ by our notation. Then we have that

$$\begin{aligned}\frac{\nu_{t+1}(x_t)}{\nu_t(x_t)} \cdot \frac{\nu_t(y_t)}{\nu_{t+1}(y_t)} &= \frac{\nu_{t+1}(x_t)Z_t}{\nu_t(x_t)Z_{t+1}} \cdot \frac{\nu_t(y_t)Z_{t+1}}{\nu_{t+1}(y_t)Z_t} \\ &= \frac{\overline{\nu}_{t+1}(x_t)}{\overline{\nu}_t(x_t)} \cdot \frac{\overline{\nu}_t(y_t)}{\overline{\nu}_{t+1}(y_t)}.\end{aligned}$$

The statement follows from the Assumption 13.3.1 and the group privacy

$$D_\infty^{Be^{B\eta}\delta_0}(\overline{\nu}_{t+1}, \overline{\nu}_t) \le B\eta.$$

Then the statement follows from Lemma 13.2.5, the independence between x_t, y_t and Union bound. □

We are now ready to prove Lemma 13.3.7.

Proof. Fix any t. Let $t_0 \le t$ be largest integer such that $A_{t_0} = 1$, that is, y_t is sampled from $\overline{\nu}_{t_0}$ for some random $t_0 \le t$. By the group privacy, we know $D_\infty^{Be^{B\eta(t-t_0)}\delta_0(t-t_0)}(\nu_t, \nu_{t_0}) \le B\eta(t-t_0)$.

Define the bad set

$$S_{bad} = \{y : \frac{\nu_{t+1}(x)}{\nu_t(x)} \cdot \frac{\nu_t(y)}{\nu_{t+1}(y)} \notin [e^{-2B\eta}, e^{2B\eta}], x \sim \nu_t\}.$$

By Lemma 13.6.1, we know

$$\nu_t(y \in S_{bad}) \le 6e^{B\eta} \cdot \delta_0/\eta.$$

Therefore, we have that

$$\begin{aligned}\nu_{t_0}(y \in S_{bad}) &\le e^{B\eta(t-t_0)} \cdot \nu_t(y \in S_{bad}) + (t-t_0)B\delta_0 e^{B\eta(t-t_0)} \\ &\le 2e^{B\eta(t-t_0+1)} \cdot B\delta_0/\eta + B\delta_0(t-t_0)e^{B\eta(t-t_0)}.\end{aligned}$$

By the CDF of the geometric distribution, we know with probability at least $1-\delta_1$, we

get $|t_0 - t| \le \log(1/\delta_1)/p$. Let E be the event that $|t_0 - t| \le \log(1/\delta_1)/p$. Hence we know

$$\begin{aligned}\nu_{t_0}(y \in S_{bad}) &\le \nu_{t_0}(y \in S_{bad} \mid E)\Pr(E) + \Pr(E^c)\\ &\le (2/\eta + \log(1/\delta_1)/p) \cdot eB\delta_0 + \delta_1.\end{aligned}$$

□

13.7 Missing proofs for Section 13.4

13.7.1 Proof of Lemma 13.4.4

Proof. For this lower bound, we assume that the algorithm has full access to D to release $z_1, \dots, z_T$. First, note that if the algorithms picks $z = z_i$ with probability $1/T$ and releases $(z, \dots, z)$, then it has the same error since

$$\mathbb{E}[\sum_{t=1}^{T} \ell(z)] = T\,\mathbb{E}[\ell(z)] = T\,\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T} \ell(z_t)] = \mathbb{E}[\sum_{t=1}^{T} \ell(z_t)].$$

Therefore, we assume that the algorithm releases a single $z = \mathsf{ALG}(D)$ that is (ε, δ)-DP. Denote $D_\ell = (\ell, \dots, \ell)$. Note that as we sample $\ell \sim \mathsf{Ber}(1/2)^d$, the probability $p := \Pr(\ell = \ell_0) = \Pr(\ell = \ell_1)$ for all $\ell_0, \ell_1 \in \{0,1\}^d$. Letting $\overline{\ell} = 1 - \ell$, we have that

$$\begin{aligned}&\mathop{\mathbb{E}}_{\ell \sim \mathsf{Ber}(1/2)^d}\left[\sum_{t=1}^{T} \ell(\mathsf{ALG}(D_\ell))\right]\\ &= T \cdot \mathop{\mathbb{E}}_{\ell \sim \mathsf{Ber}(1/2)^d}[\ell(\mathsf{ALG}(D_\ell))]\\ &= T \cdot \sum_{\ell_0 \in \{0,1\}^d} \Pr_{\ell \sim \mathsf{Ber}(1/2)^d}(\ell = \ell_0) \cdot \mathbb{E}\left[\ell_0(\mathsf{ALG}(D_{\ell_0}))\right]\\ &= \frac{T}{2} \cdot \sum_{\ell_0 \in \{0,1\}^d} p\,\mathbb{E}\left[\ell_0(\mathsf{ALG}(D_{\ell_0})) + \overline{\ell}_0(\mathsf{ALG}(D_{\overline{\ell}_0}))\right]\\ &\ge \frac{T}{2} \cdot \min_{\ell_0 \in \{0,1\}^d} \mathbb{E}\left[\ell_0(\mathsf{ALG}(D_{\ell_0})) + \overline{\ell}_0(\mathsf{ALG}(D_{\overline{\ell}_0}))\right].\end{aligned}$$

Now note that for any ℓ_0 we have

$$\begin{aligned}
&\mathbb{E}\left[\ell_0(\mathsf{ALG}(D_{\ell_0})) + \overline{\ell}_0(\mathsf{ALG}(D_{\overline{\ell}_0}))\right] \\
&= \sum_{z\in[d]} \Pr(\mathsf{ALG}(D_{\ell_0}) = z)\ell_0(z) + \Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)\overline{\ell}_0(z) \\
&= \sum_{z\in[d]} \Pr(\mathsf{ALG}(D_{\ell_0}) = z)\ell_0(z) + \Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)(1-\ell_0(z)) \\
&= 1 + \sum_{z\in[d]} \ell_0(z)\left(\Pr(\mathsf{ALG}(D_{\ell_0}) = z) - \Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)\right) \\
&\geq 1 + \sum_{z\in[d]} \ell_0(z)\left(e^{-T\varepsilon}\Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z) - T\delta - \Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)\right) \\
&\geq 1 - Td\delta + \sum_{z\in[d]} \ell_0(z)\Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)\left(e^{-T\varepsilon} - 1\right) \\
&\geq 1 - Td\delta - \sum_{z\in[d]} \ell_0(z)\Pr(\mathsf{ALG}(D_{\overline{\ell}_0}) = z)T\varepsilon \\
&\geq 1 - Td\delta - T\varepsilon,
\end{aligned}$$

where the first inequality follows since ALG is (ε,δ)-DP and group privacy. The claim follows $\square$

BIBLIOGRAPHY

[AA93] Hédy Attouch and Dominique Aze. Approximation and regularization of arbitrary functions in hilbert spaces by the lasry-lions method. In *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, volume 10, 3, pages 289–312. Elsevier, 1993.

[AAZB+17] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199, 2017.

[ABG+23] Raman Arora, Raef Bassily, Tomás González, Cristóbal A Guzmán, Michael Menart, and Enayat Ullah. Faster rates of convergence to stationary points in differentially private optimization. In *International Conference on Machine Learning*, pages 1060–1092. PMLR, 2023.

[Abo16] John M. Abowd. The challenge of scientific reproducibility and privacy protection for statistical agencies. *Technical report, Census Scientific Advisory Committee*, 2016.

[ABRW12] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Trans. Inf. Theory*, 58(5):3235–3249, 2012.

[AC21] Kwangjun Ahn and Sinho Chewi. Efficient constrained sampling via the mirror-langevin algorithm. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, pages 28405–28418, 2021.

[ACCD12] Ery Arias-Castro, Emmanuel J Candes, and Mark A Davenport. On the funda-

mental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481, 2012.

[ACD+23] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1):165–214, 2023.

[ACG+16] Martin Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016.

[ACJ+21] Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. In *Advances in Neural Information Processing Systems, NeurIPS*, 2021.

[ADF+21] Hilal Asi, John Duchi, Alireza Fallah, Omid Javidbakht, and Kunal Talwar. Private adaptive gradient methods for convex optimization. pages 383–392, 2021.

[AFKT21] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in l1 geometry. In *International Conference on Machine Learning*, pages 393–403. PMLR, 2021.

[AFKT23a] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Near-optimal algorithms for private online optimization in the realizable regime. 2023.

[AFKT23b] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private online prediction from experts: Separations and faster rates. 2023.

[AGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant

theory and polynomial identity testing. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 172–181. ACM, 2018.

[AGM+21] Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M. Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. *CoRR*, abs/2112.00193, 2021.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012.

[AKPS19] Deeksha Adil, Rasmus Kyng, Richard Peng, and Sushant Sachdeva. Iterative refinement for p-norm regression. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1405–1424. SIAM, 2019.

[AKRS19] Jordan Awan, Ana Kenney, Matthew Reimherr, and Aleksandra Slavković. Benefits and pitfalls of the exponential mechanism with applications to hilbert spaces and functional pca. In *International Conference on Machine Learning, ICML*, 2019.

[AKST23a] Naman Agarwal, Satyen Kale, Karan Singh, and Abhradeep Thakurta. Differentially private and lazy online convex optimization. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4599–4632. PMLR, 2023.

[AKST23b] Naman Agarwal, Satyen Kale, Karan Singh, and Abhradeep Thakurta. Improved differentially private and lazy online convex optimization. *arXiv:2312.11534 [cs.CR]*, 2023.

[AL24] Hilal Asi and Daogao Liu. User-level differentially private stochastic convex optimization: Efficient algorithms with optimal rates. In *International Conference on Artificial Intelligence and Statistics, 2024*, volume 238 of *Proceedings of Machine Learning Research*, pages 4240–4248. PMLR, 2024.

[ALD21] Hilal Asi, Daniel Asher Nathan Levy, and John Duchi. Adapting to function difficulty and growth conditions in private optimization. In *Advances in Neural Information Processing Systems*, 2021.

[ALS23] Jayadev Acharya, Yuhan Liu, and Ziteng Sun. Discrete distribution estimation under user-level local differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 8561–8585. PMLR, 2023.

[ALT24] Hilal Asi, Daogao Liu, and Kevin Tian. Private stochastic convex optimization with heavy tails: Near-optimality from simple reductions. *arXiv preprint arXiv:2406.02789*, 2024.

[ANW10] Alekh Agarwal, Sahand N. Negahban, and Martin J. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 37–45. Curran Associates, Inc., 2010.

[App17] Differential Privacy Team Apple. Learning with privacy at scale. *Technical report, Apple*, 2017.

[AS17] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. pages 32–40, 2017.

[AS19] Omer Angel and Yinon Spinka. Pairwise optimal coupling of multiple random variables. *arXiv preprint arXiv:1903.00632*, 2019.

[AT18] Jason Altschuler and Kunal Talwar. Online learning over a finite action set with limited switching. 2018.

[AWBR09] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. *Advances in Neural Information Processing Systems*, 22, 2009.

[AZG20] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[AZH16] Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016.

[Bar20] Alessandro Andrea Barp. *The bracket geometry of statistics*. PhD thesis, Imperial College London, 2020.

[BBG18] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems, NeurIPS*, 2018.

[BC12] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends Mach. Learn.*, 5(1):1–122, 2012.

[BCL94] Keith Ball, Eric A. Carlen, and Elliott H. Lieb. Sharp uniform convexity and smoothness estimates for trace norms. *Inventiones mathematicae*, 115(1):463–482, 1994.

[BD14] Rina Foygel Barber and John C. Duchi. Privacy: A few definitional aspects and consequences for minimax mean-squared error. In *53rd IEEE Conference on Decision and Control, CDC 2014*, pages 1365–1369. IEEE, 2014.

[BDFS07] Atanu Biswas, Sujay Datta, Jason P Fine, and Mark R Segal. Statistical advances in the biomedical science. *(No Title)*, 2007.

[BDKT12] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1269–1284, 2012.

[BDMP17] Nicolas Brosse, Alain Durmus, Éric Moulines, and Marcelo Pereyra. Sampling from a log-concave distribution with compact support with proximal langevin monte carlo. In *Conference on learning theory*, pages 319–342. PMLR, 2017.

[BDRS18] Mark Bun, Cynthia Dwork, Guy N Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–86, 2018.

[BÉ85] Dominique Bakry and Michel Émery. Diffusions hypercontractives. In *Seminaire de probabilités XIX 1983/84*, pages 177–206. Springer, 1985.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[BE19] Sébastien Bubeck and Ronen Eldan. The entropic barrier: Exponential families, log-concave geometry, and self-concordance. *Math. Oper. Res.*, 44(1):264–276, 2019.

[BEL18] Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected Langevin Monte Carlo. *Discrete & Computational Geometry*, 59(4):757–783, 2018.

[BEM+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017.

[Ber18] Espen Bernton. Langevin monte carlo and jko splitting. In *Conference on learning theory*, pages 1777–1798. PMLR, 2018.

[Bes94] Julian Besag. Comments on "representations of knowledge in complex systems" by u. grenander and mi miller. *Journal of the Royal Statistical Society, Series B*, 56:591–592, 1994.

[BFGT20] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33:4381–4391, 2020.

[BFTGT19] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

[BGM21] Raef Bassily, Cristóbal Guzmán, and Michael Menart. Differentially private stochastic optimization: New results in convex and non-convex settings. *Advances in Neural Information Processing Systems*, 34:9317–9329, 2021.

[BGN21] Raef Bassily, Cristóbal Guzmán, and Anupama Nandi. Non-euclidean differentially private stochastic convex optimization. pages 474–499, 2021.

[BJL+19] Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Complexity of highly parallel non-smooth convex optimization. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

[BL76] Herm Jan Brascamp and Elliott H Lieb. On extensions of the brunn-minkowski and prékopa-leindler theorems, including inequalities for log concave functions, and with an application to the diffusion equation. *Journal of Functional Analysis*, 22(4):366–389, 1976.

[BL00] Sergey G Bobkov and Michel Ledoux. From brunn-minkowski to brascamp-lieb and to logarithmic sobolev inequalities. *GAFA, Geometric and Functional Analysis*, 10:1028–1052, 2000.

[BLM20] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. *arXiv preprint arXiv:2003.00563*, 2020.

[BM99] Guy E. Blelloch and Bruce M. Maggs. Parallel algorithms. In Mikhail J. Atallah, editor, *Algorithms and Theory of Computation Handbook*, Chapman & Hall/CRC Applied Algorithms and Data Structures series. CRC Press, 1999.

[BNS13] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, 2013.

[Bor75a] Christer Borell. The brunn-minkowski in gauss space. *Inventiones mathematicae*, 30:207–216, 1975.

[Bor75b] Christer Borell. Convex set functions ind-space. *Periodica Mathematica Hungarica*, 6(2):111–136, 1975.

[Bot12] Léon Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer, 2012.

[BRH12] Nawaf Bou-Rabee and Martin Hairer. Nonasymptotic mixing of the mala algorithm. *IMA Journal of Numerical Analysis*, 33(1):80–110, 2012.

[BRP21] Ghazi Badih, Kumar Ravi, and Manurangsi Pasin. User-level private learning via correlated sampling. *Advances in Neural Information Processing Systems*, 2021.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[BS18] Eric Balkanski and Yaron Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization. *arXiv: 1808.03880*, 2018.

[BS23] Raef Bassily and Ziteng Sun. User-level private stochastic convex optimization with optimal rates. 2023.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS)*, pages 464–473, 2014.

[BT94] Dimitri P Bertsekas and Paul Tseng. Partial proximal minimization algorithms for convex pprogramming. *SIAM Journal on Optimization*, 4(3):551–572, 1994.

[BU17] Mitali Bafna and Jonathan Ullman. The price of selection in differential privacy. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 151–168. PMLR, 07–10 Jul 2017.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

[BV14] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.

[BV19] Victor Balcer and Salil Vadhan. Differential privacy on finite computers. *Journal of Privacy and Confidentiality*, 9:2, 2019.

[BW18] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning, ICML*, 2018.

[CBCG04] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generaliza-

tion ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[CCBJ18] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped langevin MCMC: A non-asymptotic analysis. In *Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 300–323. PMLR, 2018.

[CCSW22] Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. Improved analysis for a proximal algorithm for sampling. In *Conference on Learning Theory*, pages 2984–3014. PMLR, 2022.

[CDHS20] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

[CDJB20] Niladri Chatterji, Jelena Diakonikolas, Michael I Jordan, and Peter Bartlett. Langevin monte carlo without smoothness. In *International Conference on Artificial Intelligence and Statistics, AISTATS*, 2020.

[CDWY20] Yuansi Chen, Raaz Dwivedi, Martin J Wainwright, and Bin Yu. Fast mixing of metropolized hamiltonian monte carlo: Benefits of multi-step gradients. *Journal of Machine Learning Research*, 21:92–1, 2020.

[CE17] Dario Cordero-Erausquin. Transport inequalities for log-concave measures, quantitative forms, and applications. *Canada J. Math*, 69(3):481–501, 2017.

[CE22] Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for markov chains (extended abstract). In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, pages 110–122. IEEE, 2022.

[CEL+22] Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of langevin monte carlo from poincare to log-sobolev. In *Conference on Learning Theory*, pages 1–2. PMLR, 2022.

[CH22] Yair Carmon and Danielle Hausler. Distributionally robust optimization via ball oracle acceleration. *arXiv:2203.13225*, 2022.

[CH24] Yair Carmon and Oliver Hinder. The price of adaptivity in stochastic convex optimization. *CoRR*, abs/2402.10898, 2024.

[Che21a] Yuansi Chen. An almost constant lower bound of the isoperimetric coefficient in the kls conjecture. *GAFA, Geometric and Functional Analysis*, 31:34–61, 2021.

[Che21b] Sinho Chewi. The entropic barrier is n-self-concordant. *CoRR*, abs/2112.10947, 2021.

[Che23] Sinho Chewi. *Log-Concave Sampling.* 2023.

[CHJ+22] Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive monteiro-svaiter acceleration. *arXiv:2205.15371*, 2022.

[Chu10] Laurent Chupin. Fokker-planck equation in bounded domain. In *Annales de l'Institut Fourier*, volume 60, pages 217–255, 2010.

[CJJ+20] Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[CJJ+23] Yair Carmon, Arun Jambulapati, Yujia Jin, Yin Tat Lee, Daogao Liu, Aaron Sidford, and Kevin Tian. Resqueing parallel and private stochastic convex optimization. *arXiv preprint arXiv:2301.00457*, 2023.

[CJJS21] Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Thinking inside the ball: Near-optimal minimization of the maximal loss. In *Conference on Learning Theory, COLT*, 2021.

[CJMP21] Albert Cheu, Matthew Joseph, Jieming Mao, and Binghui Peng. Shuffle private stochastic convex optimization. *arXiv preprint arXiv:2106.09805*, 2021.

[CJST19] Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

[CKS20] Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. 2020.

[Cla90] F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, 1990.

[CLA+21] Sinho Chewi, Chen Lu, Kwangjun Ahn, Xiang Cheng, Thibaut Le Gouic, and Philippe Rigollet. Optimal dimension dependence of the metropolis-adjusted langevin algorithm. In *Conference on Learning Theory, COLT 2021*, volume 134 of *Proceedings of Machine Learning Research*, pages 1260–1300. PMLR, 2021.

[CLL19] Yu Cao, Jianfeng Lu, and Yulong Lu. Exponential decay of rényi divergence under fokker–planck equations. *Journal of Statistical Physics*, 176(5):1172–1184, 2019.

[CM08] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems, NeurIPS*, 2008.

[CMO23] Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion. In *International Conference on Machine Learning*, pages 6643–6670. PMLR, 2023.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[CO19] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

[CP22] Sinho Chewi and Aram-Alexandre Pooladian. An entropic generalization of Caffarelli's contraction theorem via covariance inequalities. *arXiv e-prints*, 2022.

[Cra38] Harald Cramér. Sur un nouveau théorème-limite de la théorie des probabilités. *Act. Sci. et Ind.*, 736, 1938.

[CRT06] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

[CSS13] Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14, 2013.

[CST21] Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 62:1–62:18, 2021.

[CTW+21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, pages 2633–2650, 2021.

[CV19] Zongchen Chen and Santosh S Vempala. Optimal convergence rate of hamiltonian monte carlo for strongly logconcave distributions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques APPROX/RANDOM*, 2019.

[CYLK20] Lin Chen, Qian Yu, Hannah Lawrence, and Amin Karbasi. Minimax regret of switching-constrained online convex optimization: No phase transition. *Advances in Neural Information Processing Systems*, 33:3477–3486, 2020.

[CYS21] Rishav Chourasia, Jiayuan Ye, and Reza Shokri. Differential privacy dynamics of langevin diffusion and noisy gradient descent. *arXiv:2102.05855*, 2021.

[CZ92] Yair Censor and Stavros Andrea Zenios. Proximal minimization algorithm withd-functions. *Journal of Optimization Theory and Applications*, 73(3):451–464, 1992.

[Dal17a] Arnak S. Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*, pages 678–689, 2017.

[Dal17b] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):651–676, 2017.

[DBH+22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DBW12] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DCWY19] Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu. Log-concave sampling: Metropolis-hastings algorithms are fast. *J. Mach. Learn. Res.*, 20:183:1–183:42, 2019.

[DDL+22] Damek Davis, Dmitriy Drusvyatskiy, Yin Tat Lee, Swati Padmanabhan, and Guanghao Ye. A gradient sampling method with complexity guarantees for lipschitz functions in high and low dimensions. *Advances in neural information processing systems*, 35:6692–6703, 2022.

[DDXZ21] Damek Davis, Dmitriy Drusvyatskiy, Lin Xiao, and Junyu Zhang. From low probability to high confidence in stochastic convex optimization. *J. Mach. Learn. Res.*, 22:49:1–49:38, 2021.

[Dem97] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.

[DFK91] Martin E. Dyer, Alan M. Frieze, and Ravi Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[DFO20] Jelena Diakonikolas, Maryam Fazel, and Lorenzo Orecchia. Fair packing and covering on a relative scale. *SIAM J. Optim.*, 30(4):3284–3314, 2020.

[DG19] Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. In *Conference on Learning Theory, COLT*, 2019.

[DG21] Jelena Diakonikolas and Cristóbal Guzmán. Complementary composite minimization, small gradients in general norms, and applications to regression problems. *CoRR*, abs/2101.11041, 2021.

[DGBSX12] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1), 2012.

[DGN14] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.

[DJW13] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[DJW14] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Privacy aware learning. *J. ACM*, 61(6):38:1–38:57, 2014.

[DJWW15] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[DKL18] Etienne De Klerk and Monique Laurent. Comparison of lasserre's measure-based bounds for polynomial optimization to bounds obtained by simulated annealing. *Mathematics of Operations Research*, 43(4):1317–1325, 2018.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2006.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *Advances in Neural Information Processing Systems, NeurIPS*, 2017.

[DM19] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

[DMM19] Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of langevin monte carlo via convex optimization. *Journal of Machine Learning Research*, 20(1):2666–2711, 2019.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of the Third Conf. on Theory of Cryptography (TCC)*, pages 265–284, 2006.

[DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[DNR+09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 381–390. ACM, 2009.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[DRS19] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2019.

[DRS21] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2021.

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS*, 2010.

[DRY18] John Duchi, Feng Ruan, and Chulhee Yun. Minimax bounds on stochastic batched convex optimization. In *Conference On Learning Theory, COLT*, 2018.

[Duc18] John C Duchi. Introductory lectures on stochastic optimization. *The Mathematics of Data*, pages 99–186, 2018.

[DYJG16] Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Big batch sgd: Automated inference using adaptive batch sizes. *arXiv preprint arXiv:1610.05792*, 2016.

[EK09] Stewart N Ethier and Thomas G Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.

[EK11] Ronen Eldan and Bo"az Klartag. Approximately gaussian marginals and the hyperplane conjecture. *Contemporary Math.*, 545:44–68, 2011.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014.

[Fel16] Vitaly Feldman. Generalization of erm in stochastic convex optimization: The dimension strikes back. 2016.

[FG04] Matthieu Fradelizi and Olivier Guédon. The extreme points of subsets of s-concave probabilities and a geometric localization theorem. *Discrete & Computational Geometry*, 31(2):327–335, 2004.

[FKM05] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

[FKT20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

[FLLZ18] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

[FMT22] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 954–964. IEEE, 2022.

[FMTT18] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532. IEEE, 2018.

[FTS17] Kazuto Fukuchi, Quang Khai Tran, and Jun Sakuma. Differentially private empirical risk minimization with input perturbation. In *International Conference on Discovery Science*, 2017.

[GARD18] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

[GC11] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society, Series B*, 73(2):123–214, 2011.

[GDG+19] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, César A Uribe, Bo Jiang, Haoyue Wang, Shuzhong Zhang, Sébastien Bubeck, et al. Near optimal methods for minimizing convex functions with lipschitz p-th derivatives. In *Conference on Learning Theory, COLT*, 2019.

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[GJ23] Benjamin Grimmer and Zhichao Jia. Goldstein stationarity in lipschitz constrained optimization. *arXiv preprint arXiv:2310.03690*, 2023.

[GKK+23a] Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Raghu Meka,

and Chiyuan Zhang. User-level differential privacy with few examples per user. *arXiv preprint arXiv:2309.12500*, 2023.

[GKK+23b] Badih Ghazi, Pritish Kamath, Ravi Kumar, Raghu Meka, Pasin Manurangsi, and Chiyuan Zhang. On user-level private convex optimization. *arXiv preprint arXiv:2305.04912*, 2023.

[GL12] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[GL13a] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[GL13b] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

[GLL22] Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 1948–1989. PMLR, 2022.

[GLL+23] Sivakanth Gopi, Yin Tat Lee, Daogao Liu, Ruoqi Shen, and Kevin Tian. Private convex optimization in general norms. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5068–5089. SIAM, 2023.

[GLOT23] Arun Ganesh, Daogao Liu, Sewoong Oh, and Abhradeep Thakurta. Private (stochastic) non-convex optimization revisited: Second-order stationary points and excess risks. *arXiv:2302.09699 [cs.LG]*, 2023.

[GLZW24] Changyu Gao, Andrew Lowy, Xingyu Zhou, and Stephen J Wright. Private heterogeneous federated learning without a trusted server revisited: Error-optimal and communication-efficient algorithms for convex losses. *arXiv preprint arXiv:2407.09690*, 2024.

[Gol64] A. A. Goldstein. Convex programming in hilbert space. 70(5):709—710, 1964.

[Gol77] Allen A Goldstein. Optimization of lipschitz continuous functions. *Mathematical Programming*, 13:14–22, 1977.

[GT20] Arun Ganesh and Kunal Talwar. Faster differentially private samplers via rényi divergence analysis of discretized Langevin MCMC. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[GTU22] Arun Ganesh, Abhradeep Thakurta, and Jalaj Upadhyay. Langevin diffusion: An almost universal algorithm for private euclidean (convex) optimization. *arXiv:2204.01585*, 2022.

[Gül91] Osman Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

[GV22] Khashayar Gatmiry and Santosh S Vempala. Convergence of the riemannian langevin algorithm. *arXiv preprint arXiv:2204.10818*, 2022.

[GVW10] Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. 2010.

[GW23] Changyu Gao and Stephen J Wright. Differentially private optimization for smooth nonconvex erm. *arXiv preprint arXiv:2302.04972*, 2023.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3-4):157–325, 2016.

[HC57] Harish-Chandra. Differential operators on a semisimple lie groups. *American Journal of Mathematics*, 79:87–120, 1957.

[HK12] Zhiyi Huang and Sampath Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *IEEE 53rd Annual Symposium on Foundations of Computer Science, FOCS*, 2012.

[HK14] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *J. Mach. Learn. Res.*, 15(1):2489–2512, 2014.

[HKRC18] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2883–2892, 2018.

[HLL+22] Yuxuan Han, Zhicong Liang, Zhipeng Liang, Yang Wang, Yuan Yao, and Jiheng Zhang. Private streaming sco in ℓ_p geometry with applications in high dimensional online decision making. In *International Conference on Machine Learning*, pages 8249–8279. PMLR, 2022.

[HRS16] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, 2016.

[HRS20] Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. *arXiv preprint arXiv:2006.10129*, 2020.

[HS16] Daniel J. Hsu and Sivan Sabato. Loss minimization and parameter estimation with heavy tails. *J. Mach. Learn. Res.*, 17:18:1–18:40, 2016.

[HSW+21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714, 2010.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IIW15] Marat Ibragimov, Rustam Ibragimov, and Johan Walden. *Heavy-tailed distributions and robustness in economics and finance*, volume 214. Springer, 2015.

[INS+19] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

[IZ80] C. Itzykson and J.-. Zuber. The planar approximation. ii. *Journal of Mathematical Physics*, 21:411–421, 1980.

[JGN+17] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

[Jia21] Qijia Jiang. Mirror langevin monte carlo: the case under isoperimetry. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, pages 715–725, 2021.

[JKL+23] Michael Jordan, Guy Kornowski, Tianyi Lin, Ohad Shamir, and Manolis Zampetakis. Deterministic nonsmooth nonconvex optimization. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4570–4597. PMLR, 2023.

[JKO98] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker-planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.

[JKT12] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[JLLV21] He Jia, Aditi Laddha, Yin Tat Lee, and Santosh Vempala. Reducing isotropy and volume to kls: an $o(n^3\psi^2)$ volume algorithm. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 961–974, 2021.

[JLSW20] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2020.

[JLT20] Arun Jambulapati, Jerry Li, and Kevin Tian. Robust sub-gaussian principal component analysis and width-independent schatten packing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020.

[JLV22] Arun Jambulapati, Yin Tat Lee, and Santosh S. Vempala. A slightly improved bound for the KLS constant. *CoRR*, abs/2208.11644, 2022.

[JNG+19] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

[JNN19] Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. In *Conference on Learning Theory*, pages 1752–1755. PMLR, 2019.

[JST24] Arun Jambulapati, Aaron Sidford, and Kevin Tian. Closing the computational-query depth gap in parallel stochastic convex optimization. In *The Thirty Seventh Annual Conference on Learning Theory, COLT 2024*, Proceedings of Machine Learning Research. PMLR, 2024.

[JT13] Prateek Jain and Abhradeep Thakurta. Differentially private learning with kernels. In *International conference on machine learning*, pages 118–126. PMLR, 2013.

[JT14] Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484. PMLR, 2014.

[JWW+20] Kaiyi Ji, Zhe Wang, Bowen Weng, Yi Zhou, Wei Zhang, and Yingbin Liang. History-gradient aided batch size adaptation for variance reduced algorithms. In *International Conference on Machine Learning*, pages 4762–4772. PMLR, 2020.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

[Kam20] Gautam Kamath. Lecture 14 — Private ML and Stats: Modern ML. http://www.gautamkamath.com/CS860notes/lec14.pdf, 2020.

[KCK+18] Yu-Hsuan Kuo, Cho-Chun Chiu, Daniel Kifer, Michael Hay, and Ashwin Machanavajjhala. Differentially private hierarchical count-of-counts histograms. *Proceedings of the VLDB Endowment*, 11(11), 2018.

[KD99] Jayesh H Kotecha and Petar M Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, ICASSP*, 1999.

[KDRT21] Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private erm with adagrad rates via publicly estimated subspaces. In *COLT*, 2021.

[KJ16] Shiva Prasad Kasiviswanathan and Hongxia Jin. Efficient private empirical

risk minimization for high-dimensional learning. In *International Conference on Machine Learning*, pages 488–497. PMLR, 2016.

[KL22] Bo'az Klartag and Joseph Lehec. Bourgain's slicing problem and kls isoperimetry up to polylog. *arXiv preprint arXiv:2203.15551*, 2022.

[KL23] Siyu Kong and AS Lewis. The cost of nonconvexity in deterministic nonsmooth optimization. *Mathematics of Operations Research*, 2023.

[Kla06] Bo"az Klartag. On convex perturbations with a bounded isotropic constant. *Geometric and Functional Analysis*, 16(6):1274–1290, 2006.

[KLL21] Janardhan Kulkarni, Yin Tat Lee, and Daogao Liu. Private non-smooth erm and sco in subquadratic steps. In *Advances in Neural Information Processing Systems, NeurIPS*, 2021.

[KLL+23] Jonathan A. Kelner, Jerry Li, Allen X. Liu, Aaron Sidford, and Kevin Tian. Semi-random sparse recovery in nearly-linear time. In *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023*, volume 195 of *Proceedings of Machine Learning Research*, pages 2352–2398. PMLR, 2023.

[KLOS14] Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 217–226. SIAM, 2014.

[KLS95] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13(3):541–559, 1995.

[KLSV22] Yunbum Kook, Yin Tat Lee, Ruoqi Shen, and Santosh S Vempala. Condition-number-independent convergence rate of riemannian hamiltonian monte carlo with numerical integrators. *arXiv preprint arXiv:2210.07219*, 2022.

[KLZ22] Gautam Kamath, Xingtu Liu, and Huanyu Zhang. Improved rates for differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*, pages 10633–10660. PMLR, 2022.

[KM12] Bo"az Klartag and Emanuel Milman. Centroid bodies and the logarithmic laplace transform: a unified approach. *Journal of Functional Analysis*, 262(1):10–34, 2012.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[KMS+21] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. *arXiv:2103.00039 [cs.CR]*, 2021.

[Kol11] Alexander V. Kolesnikov. Mass transportation and contractions. *arXiv preprint arXiv:1103.1479*, 2011.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1376–1385, Lille, France, 07–09 Jul 2015. PMLR.

[KPSW19] Rasmus Kyng, Richard Peng, Sushant Sachdeva, and Di Wang. Flows in almost linear time via adaptive preconditioning. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 902–913. ACM, 2019.

[KPW+24] Guy Kornowski, Swati Padmanabhan, Kai Wang, Zhe Zhang, and Suvrit Sra. First-order methods for linearly constrained bilevel optimization. *arXiv preprint arXiv:2406.12771*, 2024.

[KS21] Guy Kornowski and Ohad Shamir. Oracle complexity in nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 34:324–334, 2021.

[KS22a] Guy Kornowski and Ohad Shamir. On the complexity of finding small subgradients in nonsmooth optimization. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

[KS22b] Guy Kornowski and Ohad Shamir. Oracle complexity in nonsmooth nonconvex optimization. *Journal of Machine Learning Research*, 23(314):1–44, 2022.

[KS24] Guy Kornowski and Ohad Shamir. An algorithm with optimal dimension-dependence for zero-order nonsmooth nonconvex stochastic optimization. *Journal of Machine Learning Research*, 25(122):1–14, 2024.

[KST09] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. *arXiv e-prints*, abs/0910.0610, 2009.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings, 2012.

[KT13] Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, SODA*, 2013.

[KTE88] Leonid G. Khachiyan, Sergei Pavlovich Tarasov, and I. I. Erlikh. The method of inscribed ellipsoids. *Soviet Math. Dokl.*, 37:226–230, 1988.

[KV05] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[KV06] Adam Tauman Kalai and Santosh Vempala. Simulated annealing for convex optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

[LC22] Jiaming Liang and Yongxin Chen. A proximal algorithm for sampling from non-smooth potentials. In *2022 Winter Simulation Conference (WSC)*, pages 3229–3240. IEEE, 2022.

[Led99] Michel Ledoux. Concentration of measure and logarithmic sobolev inequalities. In *Seminaire de probabilites XXXIII*, pages 120–216. Springer, 1999.

[LFLY18] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[LGOGT24] Daogao Liu, Arun Ganesh, Sewoong Oh, and Abhradeep Guha Thakurta. Private (stochastic) non-convex optimization revisited: Second-order stationary points and excess risks. *Advances in Neural Information Processing Systems*, 36, 2024.

[Lia24] Jiaming Liang. Variance reduction and low sample complexity in stochastic optimization via proximal point method. *CoRR*, abs/2402.08992, 2024.

[LJCJ17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2345–2355, 2017.

[LK99] László Lovász and Ravi Kannan. Faster mixing via average conductance. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 282–287, 1999.

[LL86] Jean-Michel Lasry and Pierre-Louis Lions. A remark on regularization in hilbert spaces. *Israel Journal of Mathematics*, 55:257–266, 1986.

[LL21] Daogao Liu and Zhou Lu. Curse of dimensionality in unconstrained private convex ERM. *arXiv:2105.13637*, 2021.

[LL22] Daogao Liu and Zhou Lu. Lower bounds for differentially private erm: Unconstrained and non-euclidean. *arXiv preprint arXiv:2105.13637*, 2022.

[LLH+22] Xuechen Li, Daogao Liu, Tatsunori Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin Tat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? *arXiv:2207.00160*, 2022.

[LLL+24a] Zhuohang Li, Andrew Lowy, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, Bradley Malin, and Ye Wang. Analyzing inference privacy risks through gradients in machine learning. *arXiv preprint arXiv:2408.16913*, 2024.

[LLL+24b] Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. Why does differential privacy with large epsilon defend against practical membership inference attacks? *arXiv preprint arXiv:2402.09540*, 2024.

[LMV21] Jonathan Leake, Colin S. McSwiggen, and Nisheeth K. Vishnoi. Sampling matrices from harish-chandra-itzykson-zuber densities with applications to quantum inference and differential privacy. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1384–1397. ACM, 2021.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LR23] Andrew Lowy and Meisam Razaviyayn. Private stochastic optimization with large worst-case lipschitz parameter: Optimal rates for (non-smooth) convex losses and extension to non-convex losses. In *International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pages 986–1054. PMLR, 2023.

[LRC05] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing statistical hypotheses*. Springer, 2005.

[LRY+20] Songtao Lu, Meisam Razaviyayn, Bo Yang, Kejun Huang, and Mingyi Hong. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. *Advances in Neural Information Processing Systems*, 33:2811–2822, 2020.

[LS93] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993.

[LSA+21] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. Learning with user-level privacy. *Advances in Neural Information Processing Systems*, 34:12466–12479, 2021.

[LSB12] Simon Lacoste-Julien, Mark Schmidt, and Francis R. Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method. *CoRR*, abs/1212.2002, 2012.

[LST20] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Logsmooth gradient concentration and tighter runtimes for metropolized hamiltonian monte carlo. In *Conference on Learning Theory, COLT*, 2020.

[LST21a] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Lower bounds on metropolized sampling methods for well-conditioned distributions. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, pages 18812–18824, 2021.

[LST21b] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured logconcave sampling with a restricted gaussian oracle. In *Conference on Learning Theory*, pages 2993–3050. PMLR, 2021.

[LSV18] Yin Tat Lee, Zhao Song, and Santosh S Vempala. Algorithmic theory of odes

and sampling from well-conditioned logconcave densities. *arXiv:1812.06243*, 2018.

[LSY+20] Yuhan Liu, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: user vs item-level privacy. *Advances in Neural Information Processing Systems*, 33:20965–20976, 2020.

[LT19] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2019.

[LTLH21] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[LTVW22] Ruilin Li, Molei Tao, Santosh S. Vempala, and Andre Wibisono. The mirror langevin algorithm converges with vanishing bias. In *International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 718–742. PMLR, 2022.

[LUW24] Andrew Lowy, Jonathan Ullman, and Stephen Wright. How to make the gradients small privately: Improved rates for differentially private non-convex optimization. In *Forty-first International Conference on Machine Learning*, 2024.

[LV07] László Lovász and Santosh S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

[LV18] Yin Tat Lee and Santosh S Vempala. Convergence rate of riemannian hamiltonian monte carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121, 2018.

[LVS+21] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Zhiwei Steven Wu. Leveraging public data for practical private query release. *arXiv preprint arXiv:2102.08598*, 2021.

[LWAFF21] Zelun Luo, Daniel J Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5059–5068, 2021.

[LZJ22] Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35:26160–26175, 2022.

[Mai07] Francesco Mainardi. Lévy stable distributions in the theory of probability. *Lecture Notes on Mathematical Physics*, 2007.

[MASN16] Kentaro Minami, HItomi Arai, Issei Sato, and Hiroshi Nakagawa. Differential privacy without sensitivity. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016.

[MBM18] Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

[MBST21] Paul Mangold, Aurélien Bellet, Joseph Salmon, and Marc Tommasi. Differentially private coordinate descent for composite empirical risk minimization. *arXiv:2110.11688*, 2021.

[McS09] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[McS21] Colin McSwiggen. The harish-chandra integral: An introduction with examples. *L'Enseignement Mathématique*, 67(3):229–299, 2021.

[MFWB22] Wenlong Mou, Nicolas Flammarion, Martin J Wainwright, and Peter L Bartlett. An efficient sampling algorithm for non-smooth composite potentials. *Journal of Machine Learning Research*, 23(233):1–50, 2022.

[Mir17] Ilya Mironov. Rényi differential privacy. In *IEEE 30th Computer Security Foundations Symposium, CSF*, 2017.

[MM97] Benoit B Mandelbrot and Benoit B Mandelbrot. *The variation of certain speculative prices*. Springer, 1997.

[MMW+21] Wenlong Mou, Yi-An Ma, Martin J Wainwright, Peter L Bartlett, and Michael I Jordan. High-order langevin diffusion yields an accelerated mcmc algorithm. *Journal of Machine Learning Research*, 22:42–1, 2021.

[MMZ22] Yi-An Ma, Teodor Vanislavov Marinov, and Tong Zhang. Dimension independent generalization of dp-sgd for overparameterized smooth convex optimization. *arXiv preprint arXiv:2206.01836*, 2022.

[MRTZ17] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[MRTZ18] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*, 2018.

[MS08] Emanuel Milman and Sasha Sodin. An isoperimetric inequality for uniformly log-concave measures and uniformly convex bodies. *Journal of Functional Analysis*, 254(5):1235–1268, 2008.

[MS13] Renato DC Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science FOCS*, 2007.

[MTKC22] Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2205.02973*, 2022.

[MUA+24] Michael Menart, Enayat Ullah, Raman Arora, Raef Bassily, and Cristóbal Guzmán. Differentially private non-convex optimization under the kl condition with optimal rates. In *International Conference on Algorithmic Learning Theory*, pages 868–906. PMLR, 2024.

[MV21] Oren Mangoubi and Nisheeth K Vishnoi. Sampling from log-concave distributions with infinity-distance guarantees and applications to differentially private optimization. *arXiv:2111.04089*, 2021.

[Nem94] Arkadi Nemirovski. On parallel complexity of nonsmooth convex optimization. *Journal of Complexity*, 10(4):451–463, 1994.

[Nem04] Arkadi Nemirovski. Interior point polynomial time methods in convex programming. *Lecture notes*, 42(16):3215–3224, 2004.

[Nes83] Yu E Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl. Akad. Nauk SSSR,*, 1983.

[Nes98] Yurii Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[Nes05] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[Nes13] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[Nes18] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[Nit14] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. *Advances in Neural Information Processing Systems*, 27:1574–1582, 2014.

[NLST17] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

[NP06] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

[NT02] Yurii E Nesterov and Michael J Todd. On the riemannian geometry defined by self-concordant barriers and interior-point methods. *Foundations of Computational Mathematics*, 2(4):333–361, 2002.

[NY83] Arkadi S. Nemirovski and David B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[OV00] Felix Otto and Cédric Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000.

[PB14] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[Per16] Marcelo Pereyra. Proximal markov chain monte carlo algorithms. *Statistics and Computing*, 26:745–760, 2016.

[Pol64] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[RBHT12] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, 2012.

[Rén61] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, 1961.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Roc76] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[RS16] Sofya Raskhodnikova and Adam Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, 2016.

[RT96] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

[SBB+18] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems, NeurIPS*, 2018.

[Sch14] Rolf Schneider. *Convex bodies: the Brunn–Minkowski theory*. Number 151. Cambridge university press, 2014.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[Sha07] Shai Shalev-Shwartz. Online learning: Theory, algorithms, and applications. PhD thesis, Hebrew University, 2007.

[Sha12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, 2012.

[Sha17] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017.

[SK21] Uri Sherman and Tomer Koren. Lazy oco: Online convex optimization on a switching budget. 2021.

[SL19] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 2098–2109, 2019.

[SLRB17] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[Smi09] Adam Smith. Differential privacy and the secrecy of the sample. https://adamdsmith.wordpress.com/2009/09/02/sample-secrecy/, 2009. Accessed: 2022-11-06.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[SRB11] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *arXiv preprint arXiv:1109.2415*, 2011.

[SS12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[SSSSS09] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Conference on Learning Theory, COLT*, 2009.

[SSTT21] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics, AISTATS*, 2021.

[ST74] Vladimir Sudakov and Boris Tsirelson. Extremal properties of half-spaces for spherically invariant measures. *J. Soviet Math*, 9:9–18, 1974.

[ST13] Adam Smith and Abhradeep Thakurta. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. 2013.

[STT20] Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arXiv preprint arXiv:2006.06783*, 2020.

[SU15] Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *arXiv:1501.06095*, 2015.

[SZ13] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR, 2013.

[SZ23] Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In *Advances in Neural Information Processing Systems 36: Annual Con-*

ference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

[T+15] Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[Tal22] Kunal Talwar. Ppml workshop talk: Open questions in differentially private machine learning. https://machinelearning.apple.com/video/open-questions, 2022. Accessed: 2022-11-06.

[Tan79] Hiroshi Tanaka. Stochastic differential equations with reflecting boundary condition in convex regions. *Hiroshima Mathematical Journal*, 9(1):163–177, 1979.

[Tao13] Terence Tao. The harish-chandra-itzykson-zuber integral formula. https://terrytao.wordpress.com/2013/02/08/the-harish-chandra-itzykson-zuber-integral-formula/, 2013. Accessed: 2023-02-05.

[TC22] Hoang Tran and Ashok Cutkosky. Momentum aggregation for private nonconvex erm. In *Advances in Neural Information Processing Systems*, 2022.

[TCK+22] Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Friendlycore: Practical differentially private aggregation. In *International Conference on Machine Learning*, pages 21828–21863. PMLR, 2022.

[Tea17] Apple Differential Privacy Team. Learning with privacy at scale. Technical report, Apple, 2017.

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[TS13] Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory, COLT*, 2013.

[TS24] Lai Tian and Anthony Man-Cho So. No dimension-free deterministic algorithm computes approximate stationarities of lipschitzians. *Mathematical Programming*, pages 1–24, 2024.

[TTZ14] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

[TTZ15] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly-optimal private lasso. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3025–3033, 2015.

[Ull17] Jonathan Ullman. CS7880: rigorous approaches to data privacy, 2017.

[VEH14] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[Vil08] Cédric Villani. *Optimal transport, old and new*. Springer, 2008.

[VW19] Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

[WA18] Jun-Kun Wang and Jacob D. Abernethy. Acceleration through optimistic no-regret dynamics. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, pages 3828–3838, 2018.

[Wan18] Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. *arXiv:1803.02596*, 2018.

[WB23] Yongqiang Wang and Tamer Başar. Decentralized nonconvex optimization with guaranteed privacy and accuracy. *Automatica*, 150:110858, 2023.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[WBSS21] Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory, COLT*, 2021.

[WCX19] Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535. PMLR, 2019.

[WFW+15] Xi Wu, Matthew Fredrikson, Wentao Wu, Somesh Jha, and Jeffrey F Naughton. Revisiting differentially private regression: Lessons from learning theory and their consequences. *arXiv preprint arXiv:1512.06388*, 2015.

[Wib19] Andre Wibisono. Proximal langevin algorithm: Rapid convergence under isoperimetry. *arXiv preprint arXiv:1911.01469*, 2019.

[WJEG19] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving stochastic nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[WJZ+19] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

[WLK+17] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey F. Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD*, 2017.

[WLYZ21] Puyu Wang, Yunwen Lei, Yiming Ying, and Hai Zhang. Differentially private sgd with non-smooth loss. *arXiv:2101.08925*, 2021.

[WM10] Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. In *Advances in Neural Information Processing Systems, NeurIPS*, 2010.

[WX19] Di Wang and Jinhui Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1182–1189, 2019.

[WX20] Di Wang and Jinhui Xu. Escaping saddle points of empirical risk privately and scalably via dp-trust region method. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 90–106. Springer, 2020.

[WXDX20] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. pages 10081–10091, 2020.

[WYX17] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[WZ10] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[XJY18] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in neural information processing systems*, 31, 2018.

[XZ14] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[XZ24] Zheng Xu and Yanxiang Zhang. Advances in private training for production on-device language models. https://research.google/blog/advances-in-private-training-for-production-on-device-language-models/, 2024. Google Research Blog.

[YNB+21] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[YNS12] Farzad Yousefian, Angelia Nedić, and Uday V Shanbhag. On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica*, 48(1):56–67, 2012.

[YZCL21] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[YZCL22] Xiaodong Yang, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Normalized/clipped sgd with perturbation for differentially private non-convex optimization. *arXiv preprint arXiv:2206.13033*, 2022.

[ZCH+20] Yingxue Zhou, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Arindam Banerjee. Private stochastic non-convex optimization: Adaptive algorithms and tighter generalization bounds. *arXiv preprint arXiv:2006.13501*, 2020.

[ZLJ+20] Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Suvrit Sra, and Ali Jadbabaie. Complexity of finding stationary points of nonconvex nonsmooth functions. In *International Conference on Machine Learning*, pages 11173–11182. PMLR, 2020.

[ZMLX21] Qiuchen Zhang, Jing Ma, Jian Lou, and Li Xiong. Private stochastic non-

convex optimization with improved utility rates. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[ZP19] Tianqing Zhu and S Yu Philip. Applying differential privacy mechanism in artificial intelligence. In *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019.

[ZPFP20] Kelvin Shuangjian Zhang, Gabriel Peyré, Jalal Fadili, and Marcelo Pereyra. Wasserstein control of mirror langevin monte carlo. In *Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3814–3841. PMLR, 2020.

[ZTC24] Qinzi Zhang, Hoang Tran, and Ashok Cutkosky. Private zeroth-order nonsmooth nonconvex optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[ZTOH22] Liang Zhang, Kiran K Thekumparampil, Sewoong Oh, and Niao He. Bring your own algorithm for optimal differentially private stochastic minimax optimization. *Advances in Neural Information Processing Systems*, 35:35174–35187, 2022.

[ZZMW17] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*, 2017.

[ZZX+12] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: regression analysis under differential privacy. *arXiv preprint arXiv:1208.0219*, 2012.