Learning in Structured Multi-agent Systems with Provable Guarantees

Qiwen Cui

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

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Simon S. Du, Chair

Maryam Fazel

Sheng Wang

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Learning in Structured Multi-agent Systems with Provable Guarantees

Qiwen Cui

Chair of the Supervisory Committee:
Simon S. Du
Computer Science & Engineering

Multi-agent systems enable decentralized decision-making and interaction in complex environments, with applications ranging from traffic networks to robotics and economics. This thesis develops algorithms with provable theoretical guarantees, exploiting the structural properties of multi-agent systems to enhance scalability and efficiency.

In offline multi-agent reinforcement learning, we introduce unilateral coverage assumption and design the first efficient algorithms for two-player zero-sum and general-sum Markov games based on the principle of pessimism. We propose a novel strategy-wise concentration technique to reduce sample complexity, overcoming the challenges of joint action spaces.

In online multi-agent reinforcement learning, we propose the independent linear Markov game framework, enabling scalable algorithms that break the curse of multiagents by leveraging individual agent function approximation. We also design the first algorithm that can address non-stationary environments, improving sample complexity guarantees for learning correlated and Nash equilibria.

In congestion games, we design the first algorithm for Nash equilibrium learning and optimal tax learning. By exploiting the game's structure, we achieve scalable performance with sample complexity independent of large action spaces. For tax design, we propose an equilibrium feedback framework and develop an efficient method for approximating socially optimal taxes.

This work advances the theoretical and practical understanding of multi-agent learning,

with implications for diverse real-world applications.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

When I was an undergraduate student, I often heard stories about the joys and challenges of the PhD journey. At that time, becoming a PhD felt like a distant dream. Now, as I reach the end of this journey, I find it hard to believe how far I have come. I am deeply grateful to everyone who has supported me along the way.

First and foremost, I would like to express my deepest gratitude to my advisor, Simon S. Du, for his invaluable guidance and support throughout my PhD. His brilliance in machine learning and related fields has been a constant source of inspiration. I am especially thankful for the countless hours he spent discussing projects with me and for giving me the freedom to explore research areas I am passionate about.

I am also grateful to my dissertation committee, Lillian Ratliff, Sheng Wang, and Maryam Fazel, for their thoughtful suggestions and constructive criticism. I sincerely appreciate their time and effort in reviewing my research.

My journey into machine learning theory began with my undergraduate mentors Zaiwen Wen and Lin F. Yang. Their mentorship introduced me to this fascinating field and sparked my interest in reinforcement learning.

I have been fortunate to collaborate with many wonderful colleagues: Liyu Chen, Yan Dai, Abhishek Gupta, Xiaotian Han, Haozhe Jiang, Sham Kakade, Boyi Liu, Tingkai Liu, Ruoqi Shen, Tao Sun, Yunzhe Tao, Guoyin Wang, Xinqi Wang, Xuwu Wang, Zhihan Xiong, Hongxia Yang, Kaiqing Zhang, Natalia Zhang, Yufeng Zhang, Runlong Zhou, Zhaoyi Zhou, Chuning Zhu. The countless brainstorming sessions, late-night debugging, and conference travels have made this journey both productive and enjoyable.

Beyond research, I am grateful to my friends for the incredible memories we have shared—whether playing basketball, hiking, skiing, rock climbing, or simply gathering for dinner. While I cannot name everyone here, I want to extend my appreciation to all of them

for making my PhD years some of the most unforgettable years of my life.

Finally, I am deeply indebted to my family for their unconditional love and support. To my parents, Yimin Yang and Donghao Cui, thank you for providing the best education I could have. To my wife, Ruoqi Shen, your companion, sacrifices, and belief in me have been my greatest source of strength.

This dissertation is dedicated to all of you.

Chapter 1

INTRODUCTION

The advent of multi-agent systems has profoundly transformed various domains, enabling decentralized decision-making and collaborative/competitive behaviors in complex environments. These systems are ubiquitous, with applications ranging from traffic network optimization to autonomous robotics, distributed sensor networks, and economic systems. However, the inherent complexity and diversity of multi-agent interactions pose significant challenges for algorithm design.

This thesis addresses the problem of designing algorithms for multi-agent systems with provable theoretical guarantees. While generic algorithms have broad applicability, they often fail to fully leverage the unique structures of specific systems, leading to suboptimal performance. Task-specific algorithms, by contrast, can harness these structures to achieve superior efficiency and effectiveness. For example:

- **Markov Games**: These are a foundational framework in multi-agent reinforcement learning, where state transitions exhibit Markovian dynamics. This structure eliminates the dependence on long-term histories, simplifying both analysis and computation.

- **Congestion Games**: Here, agents share facilities, and their utilities depend on facility congestion levels. Exploiting this shared-resource structure can significantly enhance algorithmic design and performance.

The overarching goal of this thesis is to bridge the gap between theoretical rigor and practical applicability by developing algorithms tailored to the unique characteristics of multi-agent systems.

Part I: Provably Efficient Offline Multi-agent Reinforcement Learning

We study what dataset assumption permits solving offline two-player zero-sum Markov games. In stark contrast to the offline single-agent Markov decision process, we show that the single strategy concentration assumption is insufficient for learning the Nash equilibrium (NE) strategy in offline two-player zero-sum Markov games. On the other hand, we propose a new assumption named unilateral concentration and design a pessimism-type algorithm that is provably efficient under this assumption. In addition, we show that the unilateral concentration assumption is necessary for learning an NE strategy. Furthermore, our algorithm can achieve minimax sample complexity without any modification for two widely studied settings: dataset with uniform concentration assumption and turn-based Markov games. Our work serves as an important initial step towards understanding offline multi-agent reinforcement learning.

For the next step, we study offline multi-player general-sum Markov games. We propose the strategy-wise concentration principle which directly builds a confidence interval for the joint strategy, in contrast to the point-wise concentration principle that builds a confidence interval for each point in the joint action space. For two-player zero-sum Markov games, by exploiting the convexity of the strategy-wise bonus, we propose a computationally efficient algorithm whose sample complexity enjoys a better dependency on the number of actions than the prior methods based on the point-wise bonus. Furthermore, for offline multi-agent general-sum Markov games, based on the strategy-wise bonus and a novel surrogate function, we give the first algorithm whose sample complexity only scales $\sum_{i=1}^{m} A_i$ where A_i is the action size of the i-th player and m is the number of players. In sharp contrast, the sample complexity of methods based on the point-wise bonus would scale with the size of the joint action space $\Pi_{i=1}^{m} A_i$ due to the curse of multiagents. Lastly, all of our algorithms can naturally take a pre-specified strategy class Π as input and output a strategy that is close to the best strategy in Π. In this setting, the sample complexity only scales with $\log |\Pi|$ instead of $\sum_{i=1}^{m} A_i$.

Part II: Provably Efficient Online Multi-agent Reinforcement Learning

We propose a new model, *independent linear Markov game*, for multi-agent reinforcement learning with a large state space and a large number of agents. This is a class of Markov games with *independent* linear function approximation, where each agent has its

own function approximation for the state-action value functions that are *marginalized* by other players' policies. We design new algorithms for learning the Markov coarse correlated equilibria (CCE) and Markov correlated equilibria (CE) with sample complexity bounds that only scale polynomially with *each agent's own function class complexity*, thus breaking the curse of multiagents. In contrast, existing works for Markov games with function approximation have sample complexity bounds scale with the size of the *joint action space* when specialized to the canonical tabular Markov game setting, which is exponentially large in the number of agents. Our algorithms rely on two key technical innovations: (1) utilizing policy replay to tackle *non-stationarity* incurred by multiple agents and the use of function approximation; (2) separating learning Markov equilibria and exploration in the Markov games, which allows us to use the full-information no-regret learning oracle instead of the stronger bandit-feedback no-regret learning oracle used in the tabular setting. Furthermore, we propose an iterative-best-response type algorithm that can learn pure Markov Nash equilibria in independent linear Markov potential games, with applications in learning in congestion games. In the tabular case, by adapting the policy replay mechanism for independent linear Markov games, we propose an algorithm with $\widetilde{O}(\epsilon^{-2})$ sample complexity to learn Markov CCE, which improves the state-of-the-art result $\widetilde{O}(\epsilon^{-3})$ in Daskalakis et al. [2022], where ϵ is the desired accuracy, and also significantly improves other problem parameters. Furthermore, we design the first provably efficient algorithm for learning Markov CE that breaks the curse of multiagents.

We also investigate learning the equilibria in non-stationary multi-agent systems and address the challenges that differentiate multi-agent learning from single-agent learning. Specifically, we focus on games with bandit feedback, where testing an equilibrium can result in substantial regret even when the gap to be tested is small, and the existence of multiple optimal solutions (equilibria) in stationary games poses extra challenges. To overcome these obstacles, we propose a versatile black-box approach applicable to a broad spectrum of problems, such as general-sum games, potential games, and Markov games, when equipped with appropriate learning and testing oracles for stationary environments. Our algorithms can achieve $\widetilde{O}\left(\Delta^{1/4}T^{3/4}\right)$ regret when the degree of nonstationarity, as measured by total

variation Δ, is known, and $\widetilde{O}\left(\Delta^{1/5}T^{4/5}\right)$ regret when Δ is unknown, where T is the number of rounds. Meanwhile, our algorithm inherits the favorable dependence on number of agents from the oracles. As a side contribution that may be independent of interest, we show how to test for various types of equilibria by a black-box reduction to single-agent learning, which includes Nash equilibria, correlated equilibria, and coarse correlated equilibria.

Part III: Provably Efficient Learning in Congestion Game

For the first part, we investigate Nash-regret minimization in congestion games, a class of games with benign theoretical structure and broad real-world applications. We first propose a centralized algorithm based on the optimism in the face of uncertainty principle for congestion games with (semi-)bandit feedback, and obtain finite-sample guarantees. Then we propose a decentralized algorithm via a novel combination of the Frank-Wolfe method and G-optimal design. By exploiting the structure of the congestion game, we show the sample complexity of both algorithms depends only polynomially on the number of players and the number of facilities, but not the size of the action set, which can be exponentially large in terms of the number of facilities. We further define a new problem class, Markov congestion games, which allows us to model the non-stationarity in congestion games. We propose a centralized algorithm for Markov congestion games, whose sample complexity again has only polynomial dependence on all relevant problem parameters, but not the size of the action set.

For the second part, we investigate optimal tax learning in congestion games. In multi-player games, self-interested behavior among the players can harm the social welfare. Tax mechanisms are a common method to alleviate this issue and induce socially optimal behavior. In this work, we take the initial step of learning the optimal tax that can maximize social welfare with limited feedback in congestion games. We propose a new type of feedback named *equilibrium feedback*, where the tax designer can only observe the Nash equilibrium after deploying a tax plan. Existing algorithms are not applicable due to the exponentially large tax function space, nonexistence of the gradient, and nonconvexity of the objective. To tackle these challenges, we design a computationally efficient algorithm that leverages several novel components: (1) a piece-wise linear tax to approximate the optimal tax; (2)

extra linear terms to guarantee a strongly convex potential function; (3) an efficient subroutine to find the exploratory tax that can provide critical information about the game. The algorithm can find an ϵ-optimal tax with $O(\beta F^2/\epsilon)$ sample complexity, where β is the smoothness of the cost function and F is the number of facilities.

Part I

PROVABLY EFFICIENT OFFLINE MULTI-AGENT REINFORCEMENT LEARNING

Chapter 2

OFFLINE TWO-PLAYER ZERO-SUM MARKOV GAMES

This chapter is based on Cui and Du [2022a], with Simon S. Du.

2.1 Introduction

Promising empirical advances have been achieved in reinforcement learning (RL), including mastering the game of Go [Silver et al., 2016], Poker [Brown et al., 2017], real-time strategy games [Vinyals et al., 2019a] and robotic control [Kober et al., 2013]. Notably, many of these successes lie in the domain of multi-agent reinforcement learning (MARL). MARL is about multiple agents interacting in a shared environment, and each of them aims to maximize its own long-term reward. During the learning process, each agent not only needs to identify the environment dynamic but also needs to compete/cooperate with other agents. One important subarea of MARL is offline MARL. In many practical scenarios, we only have access to the offline data or it is too expensive to frequently change the policy [Zhang et al., 2021a]. While there are plenty of empirical works on offline MARL [Pan et al., 2021, Jiang and Lu, 2021], the theoretical understanding is still very limited. In this work, we take an initial step towards understanding when offline MARL is provably solvable.

We consider two-player zero-sum Markov games, where two players simultaneously select actions over multiple time steps in a Markovian environment and the first player aims to maximize the total reward while the second player aims to minimize it. In the offline setting, we have access to a fixed dataset collected by a (possibly unknown) exploration policy and the target is to find a (near-)Nash equilibrium (NE) strategy of the underlying two-player zero-sum Markov game.

One of the main difficulties in offline RL is distribution shift, i.e., the dataset distribution is different from the distribution induced by the optimal policy. It is important to understand what is the minimal dataset distribution assumption that permits offline RL.

For single-agent offline RL, it is shown that the pessimism principle allows policy optimization with *single policy concentration*, i.e. the dataset only covers the optimal policy [Jin et al., 2021e, Zanette et al., 2021c, Yin and Wang, 2021b, Rashidinejad et al., 2021b]. This assumption is necessary as it is impossible to learn the optimal policy if it is not covered by the dataset. However, the dataset coverage assumption for MARL is still far from clear. In this work, we want to answer the following question:

What is the minimal dataset coverage assumption that permits learning an NE strategy in offline two-player zero-sum Markov games?

Generally speaking, MARL is much more difficult than single-agent RL due to the following two reasons. First, MARL is known to suffer from the *non-stationary* property, i.e. agents will affect the others during the learning process [Zhang et al., 2021a]. Specifically, the performance may decline if each agent simultaneously tries to improve its own policy depending on others' current policies. In addition, multiple agents incur complicated statistical dependence that makes the theoretical analysis difficult. A line of works study Markov games with online sampling oracle [Bai et al., 2020, Bai and Jin, 2020b, Liu et al., 2021b] or generative model oracle [Sidford et al., 2020, Zhang et al., 2020b, Cui and Yang, 2020], where specialized techniques are developed to tackle the above difficulties. In this paper, we give the first analysis on offline Markov games in the fundamental tabular setting.

2.1.1 Main Contributions

• First, we propose an assumption named *unilateral concentration*, which posits that for all strategies μ, ν, strategy pairs (μ^*, ν) and (μ, ν^*) are covered by the dataset, where μ is the strategy for the first (max) player, ν is the strategy for the second (min) player, and (μ^*, ν^*) is an NE strategy. In Section 2.3, we prove that NE strategy is not learnable even if this assumption is only slightly violated. The intuition behind the hardness result is that to identify an NE strategy, the algorithm has to compare it with strategy pairs that one player uses any other strategies as a reference. This result also implies that the single strategy concentration, which is sufficient for offline single-agent RL, is *not sufficient* for offline MARL.

• Second, we provide positive results showing that NE strategy is PAC learnable under the unilateral concentration assumption. Combined with the hardness results above, we conclude that unilateral concentration assumption is the *necessary and sufficient dataset coverage assumption for solving offline zero-sum* Markov games. Our algorithm is based on the pessimism principle that we maintain pessimistic estimates for both players, respectively. We show that our algorithm achieves $\widetilde{O}(\sqrt{C^*SABH^3/n})$ performance gap under unilateral concentration assumption, where C^* quantifies the coverage of the dataset, S is the number of states, A is the number of the max player's actions, B is the number of the min player's actions, H is the horizon and n is the number of samples.

• Third, we show that our algorithm is *minimax optimal* when the dataset satisfies a stronger assumption, uniform concentration, or the Markov game is turn-based. These are two widely studied settings in the RL community. Uniform concentration assumes that all state-action pairs are covered by the dataset and turn-based Markov game is a variant of zero-sum Markov games where two players select actions in turns instead of simultaneously. Although uniform concentration is about the dataset structure and turn-based Markov games are about the environment structure, our algorithm can adapt to both of them without any modification and achieves minimax sample complexity.

Main Techniques. Our algorithm is motivated by the Bernstein-type bonus and reference advantage function techniques in Xie et al. [2021b] while we make novel adaptations, namely monotonic update and a self-bounding technique, to realize them in Markov games. The Monotonic update allows a sandwich-type argument that bounds the reference function and further bounds the variance term. The self-bounding technique is utilized to bound the performance gap by itself and then solve the inequality to derive the final bound on performance gap.

To summarize, (1) we identify the minimal dataset coverage assumption that allows learning the NE strategy in Markov games; (2) we propose a pessimism-based algorithm that achieves polynomial sample complexity based on novel Markov game techniques; and (3) we further show the algorithm is minimax optimal under the uniform concentration assumption or in turn-based Markov games.

2.1.2 Related Work

Here we focus on the theoretical works on two-player zero-sum Markov games and offline RL.

Two-player zero-sum Markov games. Zero-sum Markov games have been widely studied since the seminal work [Shapley, 1953]. When the transition kernel is unknown, different sampling oracles are utilized to acquire samples, including online sampling [Bai and Jin, 2020b, Xie et al., 2020b, Liu et al., 2021b, Bai et al., 2020, Jin et al., 2021b, Song et al., 2021a], generative model sampling [Sidford et al., 2020, Cui and Yang, 2020, Zhang et al., 2020b, Jia et al., 2019]. For offline sampling oracle, Zhang et al. [2021b] and Abe and Kaneko [2020] consider decentralized algorithm with network communication and offline policy evaluation, both under the uniform concentration assumption. One concurrent work [Zhong et al., 2022] considers zero-sum Markov games with linear function approximation. They also show the single policy coverage is not sufficient and propose a similar unilateral concentration assumption under which they give a provably efficient algorithm. On the other hand, under the unilateral concentration assumption, their sample complexity is worse than ours when specialized to tabular setting because they did not use Bernstein bonus. They show it is impossible to learn in all instances without unilateral concentration. However, they do not show that any assumption weaker than unilateral concentration makes learning impossible, which is a negative result proven in our paper. Lastly, our algorithm is minimax optimal for uniform concentration setting and turn-based Markov games while their algorithms are not.

Offline single-agent RL. Theoretical analysis of offline RL can be traced back to Szepesvári and Munos [2005], under the uniform concentration assumption (analogue to Assumption 2.2.3). This assumption has been extensively investigated [Xie and Jiang, 2021b, Xie et al., 2020c, Yin et al., 2020b, 2021b, Ren et al., 2021b]. Recently, a line of works showed that the pessimism principle allows offline policy optimization under a much weaker assumption, single policy concentration, both in tabular case and with function approximation [Rashidinejad et al., 2021b, Yin and Wang, 2021b, Xie et al., 2021b, Jin et al., 2021e,

Uehara and Sun, 2021b, Uehara et al., 2021, Zanette et al., 2021c, Xie et al., 2021a]. One closely related work is Xie et al. [2021b], which utilizes the reference advantage function technique and Bernstein-type bonus to show a minimax sample complexity $\widetilde{O}(SC^*H^3/n)$ in finite-horizon MDP. We show that the counterpart of single policy concentration in zero-sum Markov games is insufficient for NE strategy learning and use the pessimism principle to design algorithm that works under the unilateral concentration assumption.

2.2 Preliminaries

2.2.1 Two-Player Zero-sum Markov Games

Zero-sum Markov games (MG) generalize single-agent MDP to two-agent case where one agent aims to maximize the total reward while the other one aims to minimize it. A tabular finite-horizon zero-sum Markov game is described by the tuple $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathcal{B}, P, r, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space of the first (max) player, $\mathcal{B}$ is the action space of the second (min) player, $P = (P_1, P_2, \cdots, P_H), P_h \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}||\mathcal{B}|\times|\mathcal{S}|}, \forall h \in [H]$ is the (unknown) transition probability matrix for time step h, $r = (r_1, r_2, \cdots, r_H), r_h \in [0,1]^{|\mathcal{S}||\mathcal{A}||\mathcal{B}|}, \forall h \in [H]$ is the (unknown) deterministic reward vector and H is the horizon length.* This paper focuses on the tabular setting where $|\mathcal{S}|$,$|\mathcal{A}|$,and $|\mathcal{B}|$ are finite. At each timestep h and state s_h, if the max player chooses action a_h and the min player chooses action b_h, then the next state at timestep $h+1$ follows the distribution $s_{h+1} \sim P_h(\cdot|s_h, a_h, b_h)$ and both players receive a reward $r_h(s_h, a_h, b_h)$. Both players sequentially choose H actions and at each timestep, the action is chosen *simultaneously* and then it is revealed to both players. We assume that we have a fixed initial state s_1 and it is straightforward to generalize our results to the case where the initial state is sampled from a fixed distribution.†

Turn-based Markov games are an important subclass of (simultaneous-move) Markov games, where the max player takes action first and the min player can take action after

*It is straightforward to generalize our results to stochastic rewards because the major difficulty is in learning the transitions rather than learning the rewards.

†Stochastic initial state is equivalent to an MDP with deterministic initial state by creating a dummy initial state which transits to the next state following that initial state distribution.

observing the opponent's action. It is a widely studied setting [Sidford et al., 2020, Cui and Yang, 2020, Bai and Jin, 2020b] and we will provide minimax sample complexity result for this setting in Section 2.4.3.

We denote a strategy pair as $\pi = (\mu, \nu)$, where $\mu = (\mu_1, \mu_2, \cdots, \mu_H), \mu_h : \mathcal{S} \to \Delta^{\mathcal{A}}, \forall h \in [H]$ is the strategy of the first player and $\nu = (\nu_1, \nu_2, \cdots, \nu_H), \nu_h : \mathcal{S} \to \Delta^{\mathcal{B}}, \forall h \in [H]$ is the strategy of the second player, where $\Delta^{\mathcal{X}}$ is the probability simplex on the finite set $\mathcal{X}$. A deterministic strategy is a strategy that maps state to a single point distribution. We define the state value function and state-action value function for a strategy pair π similarly as in single-agent MDP:

$$V_h^{\pi}(s_h) := \mathbb{E}\left[\sum_{t=h}^{H} r(s_t, a_t, b_t)|\pi, s_h\right], Q_h^{\pi}(s_h, a_h, b_h) := \mathbb{E}\left[\sum_{t=h}^{H} r(s_t, a_t, b_t)|\pi, s_h, a_h, b_h\right].$$

If the second player's strategy ν is fixed, then the MG degenerates to an MDP and we call the optimal policy in this MDP as the best response strategy $\mathrm{br}_1(\nu)$. Similarly, we can define the $\mathrm{br}_2(\mu)$ as the best response for the second player. We will ignore the subscript in br_1 and br_2 when it is clear in the context. While the best response may not be unique, the best response value is always unique. For all $h \in [H], s_h \in \mathcal{S}$, we define

$$V_h^{*,\nu}(s_h) := V_h^{\mathrm{br}(\nu),\nu}(s_h) = \max_{\mu} V_h^{\mu,\nu}(s_h), V_h^{\mu,*}(s_h) := V_h^{\mu,\mathrm{br}(\mu)}(s_h) = \min_{\nu} V_h^{\mu,\nu}(s_h).$$

It is well known that Nash equilibrium (NE) strategy $\pi^* = (\mu^*, \nu^*)$, i.e., a strategy pair such that no player can benefit from switching its own strategy, exists for zero-sum Markov games with a unique value function [Shapley, 1953]. In other words, μ^* and ν^* are the best responses to each other. We define $V_h^* := V_h^{\mu^*,\nu^*}$ for all $h \in [H]$. The following weak duality property holds for all strategy pairs (μ, ν) in MG:

$$V_h^{\mu,*} \leq V_h^* \leq V_h^{*,\nu}, \forall h \in [H].$$

For a strategy pair $\pi = (\mu, \nu)$, we can then define the corresponding duality gap as

$$\mathrm{Gap}(\pi) = V_1^{*,\nu}(s_1) - V_1^{\mu,*}(s_1).$$

The duality gap is always non-negative and the NE strategy has zero duality gap $\mathrm{Gap}(\pi^*) = 0$. Duality gap measures how well a strategy pair approximates the NE. We say a strategy pair π is an ϵ-approximate NE if $\mathrm{Gap}(\pi) \leq \epsilon$.

2.2.2 *Offline Two-Player Zero-Sum Game*

In offline RL, we are given an offline dataset $D = \{(s_h^\tau, a_h^\tau, b_h^\tau, r_h^\tau, s_{h+1}^\tau)\}_{\tau\in[n]}^{h\in[H]}$ and we cannot do any further sampling [Kakade, 2003]. We assume that the dataset is sampled from some exploration policy $\rho = (\rho_1, \rho_2, \cdots, \rho_H), \rho_h : \mathcal{S} \to \Delta^{\mathcal{A}\times\mathcal{B}}, \forall h \in [H]$.[‡] The target of offline MG is to find an approximate NE with a small duality gap by utilizing the given dataset D. We use $d_h^\pi(s, a, b)$ to denote the probability of s, a, b appears at timestep h in the trajectory generated by strategy π for all $h \in [H]$. The dataset distribution $d_h^\rho(s, a, b)$ is defined similarly. A state-action pair (s, a, b) at timestep h is covered by strategy π if and only if $d_h^\pi(s, a, b) > 0$. Strategy π is covered by dataset generated by exploration strategy ρ if and only if for all (s, a, b) covered by π, it is covered by ρ. In other words, we have

$$\frac{d_h^\pi(s, a, b)}{d_h^\rho(s, a, b)} < \infty, \forall h \in [H], (s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}. \tag{2.1}$$

The sample complexity guarantee will depend on this ratio.

Dataset Coverage Assumptions. Below we list three different dataset coverage assumptions for Markov games.

Assumption 2.2.1. (Single strategy concentration) One NE strategy (μ^*, ν^*) is covered by the dataset.

Assumption 2.2.2. (Unilateral concentration) For all strategies μ and ν, (μ, ν^*) and (μ^*, ν) are covered by the dataset, where (μ^*, ν^*) is one NE strategy.

Assumption 2.2.3. (Uniform concentration) For all $h \in [H]$ and $(s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$, (s, a, b) at timestep h is covered by the dataset.

Assumption 2.2.1 is the weakest assumption and is the most straightforward extension of the single policy concentration in single-agent RL [Rashidinejad et al., 2021b]. Assumption 2.2.3 generalizes the uniform policy concentration in single-agent RL [Yin et al., 2020b]. Assumption 2.2.2 is sandwiched by Assumption 2.2.1 and Assumption 2.2.3 as Assumption

[‡]For simplicity we assume the exploration policy is Markovian. However, our analysis can be directly generalized to arbitrary dataset distribution. See Jin et al. [2021e] for discussions on dataset-dependent bounds.

2.2.2 implies Assumption 2.2.1 and Assumption 2.2.3 implies Assumption 2.2.2. In this work, we will show that Assumption 2.2.2 is the minimal dataset coverage assumption that allows NE learning and we provide sample complexity bounds that depends on the density ratio (2.1).[§]

Notations. We use $\mathrm{Var}_{P(s,a,b)}(V)$ to denote the variance of the random variable $V(s')$ where $s' \sim P(\cdot|s,a,b)$ and $\mathrm{Var}_P(V) \in \mathbb{R}^{SAB}$ to denote a vector whose (s,a,b) component is $\mathrm{Var}_{P(s,a,b)}(V)$. We define $a \vee b := \max\{a,b\}$ and $a \wedge b := \min\{a,b\}$. In addition, if a is a vector and b is a scalar, the operation is taken on each element of a: $[a \vee b]_i = a_i \vee b$. For two vector $a \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, we use $\frac{a}{b} \in \mathbb{R}^n$ to denote the element-wise division: $\left[\frac{a}{b}\right]_i = \frac{a_i}{b_i}$. In addition, if a is scalar, we still use $\frac{a}{b} \in \mathbb{R}^n$ to denote the element-wise division: $\left[\frac{a}{b}\right]_i = \frac{a}{b_i}$. We use S, A, B to denote $|\mathcal{S}|, |\mathcal{A}|, |\mathcal{B}|$.

2.3 Impossibility Results

In this section, we show that no assumption weaker than the unilateral concentration assumption (Assumption 2.2.2), which includes single strategy concentration (Assumption 2.2.1), allows learning the NE strategy. To begin with, we consider the deterministic unilateral concentration assumption.

Assumption 2.3.1. (Deterministic unilateral concentration) For all deterministic strategy μ and ν, (μ, ν^*) and (μ^*, ν) are covered by the dataset, where (μ^*, ν^*) is one NE strategy.

Immediately we can tell that Assumption 2.3.1 is satisfied under Assumption 2.2.2. These two assumptions are equivalent, which is shown by Proposition 2.3.2, because any stochastic strategy can be viewed as a combination of several deterministic strategies.

Proposition 2.3.2. *If for all deterministic strategy μ and ν, (μ, ν^*) and (μ^*, ν) are covered by the dataset, then we have for all (possibly stochastic) strategy μ' and ν', (μ', ν^*) and (μ^*, ν') are covered by the dataset.*

For the hardness examples, we consider bandit games, i.e., Markov games with horizon $H = 1$. The result can be generalized to arbitrary horizon by setting the reward to be 0

[§]Note that there could be different minimal assumptions as the assumption set is a partially ordered set. Here 'minimal' means Assumption 2.2.2 allows NE learning while no weaker assumption allows doing so.

in horizons other than $h = 1$. We consider a class of bandit games and datasets such that Assumption 2.3.1 is almost satisfied while no algorithm can identify the NE strategy for all bandit games and datasets in this class. As Assumption 2.2.2 and Assumption 2.3.1 are equivalent, no assumption weaker than Assumption 2.2.2 allows NE strategy learning. A direct corollary is that single strategy concentration (Assumption 2.2.1) is not sufficient for NE learning.

Theorem 2.3.3. *Define a class $\mathcal{X}$ of bandit game M and exploration strategy ρ that consists of all M and ρ pairs satisfying that there exists at most one deterministic strategy μ or one deterministic strategy ν such that (μ, ν^*) or (μ^*, ν) is not covered and for all other deterministic strategies μ', ν', the density ratio is bounded*

$$\frac{d_h^{\mu^*,\nu'}(s,a,b)}{d_h^{\rho}(s,a,b)} \leq 2A + 2B, \frac{d_h^{\mu',\nu^*}(s,a,b)}{d_h^{\rho}(s,a,b)} \leq 2A + 2B,$$

*for all $h \in [H]$. For any algorithm **ALG**, there exists $(M, \rho) \in \mathcal{X}$ such that the output of the algorithm **ALG** is at most a 0.25-approximate NE strategy no matter how many data are collected.*

Proof. We consider bandit games with two actions for each player here. The action set is $\mathcal{A} = \{a_1, a_2\}$ for the first (max) player and $\mathcal{B} = \{b_1, b_2\}$ for the second (min) player. We construct the following two bandit games with deterministic rewards.

$$\begin{aligned} r(a_1, b_1) &= 0.25 \quad & r(a_1, b_2) &= 0.5 \\ r(a_2, b_1) &= 0 \quad & r(a_2, b_2) &= 0.75 \end{aligned}$$

Bandit Game 1

$$\begin{aligned} r(a_1, b_1) &= 0.25 \quad & r(a_1, b_2) &= 0.5 \\ r(a_2, b_1) &= 1 \quad & r(a_2, b_2) &= 0.75 \end{aligned}$$

Bandit Game 2

Then the (unique) NE of the first bandit game is (a_1, b_1) and the (unique) NE of the second bandit game is (a_2, b_2). Now we set the exploration strategy ρ to be uniform distribu-

tion on $\{(a_1, b_1), (a_1, b_2), (a_2, b_2)\}$. We can verify that both bandit games with exploration strategy ρ is in the class defined in Theorem 2.3.3. Note that the dataset contains data on (a_1, b_1), (a_1, b_2), (a_2, b_2) and no data on (a_2, b_1). It is impossible for an algorithm to distinguish between these two bandit games as they are consistent on the given dataset and they all satisfy the dataset coverage assumption that only one action pair is not covered. With some calculations, we can show that the output of **ALG** is at most a 0.25-approximate NE for one of the instances, which proves the theorem. □

Remark 2.3.4. We can easily extend this instance to arbitrary action space by setting $(a_i, b_j) = 0$ for all $i \notin \{1, 2\}, j \in \{1, 2\}$, and $(a_i, b_j) = 1$ for all $j \notin \{1, 2\}, i \in \{1, 2\}$, and the exploration strategy ρ to be the uniform distribution on (a_i, b_j) such that $(i, j) \in \{(i, j) : i \in \{1, 2\} \text{ or } j \in \{1, 2\}, (i, j) \neq (2, 1)\}$.

Remark 2.3.5. It is straightforward to verify that the hard instance in Theorem 2.3.3 also holds for turn-based Markov games. As a result, no assumption weaker than Assumption 2.2.2 is sufficient for NE learning in turn-based Markov games.

2.4 Provably Efficient Algorithm under Unilateral Concentration

In this section, we show that it is indeed possible to learn the NE with the unilateral concentration assumption. We propose a novel algorithm called Pessimistic Nash Value Iteration (PNVI), which adapts the pessimism principle in single-agent RL to Markov games. Our sample complexity result depends on the following quantity named *unilateral concentrability*:

Definition 2.4.1. (Unilateral concentrability) For Nash equilibrium π^*, we define

$$C^* := \min_{\pi^*=(\mu^*,\nu^*)} \max_{h,(s,a,b),\mu,\nu} \left\{ \frac{d_h^{\mu^*,\nu}(s,a,b)}{d_h^{\rho}(s,a,b)}, \frac{d_h^{\mu,\nu^*}(s,a,b)}{d_h^{\rho}(s,a,b)} \right\}.$$

By definition, C^* is finite if Assumption 2.2.2 is satisfied. For the rest of the paper, π^* denotes the Nash equilibrium that achieves the minimum here. Note that C^* is not provided to the algorithm.

2.4.1 Hoeffding-type Algorithm with Data Splitting

To illustrate our main algorithm design ideas, we first propose an algorithm with Hoeffding-type bonus and random data splitting. Given a dataset $\mathcal{D} = \left\{(s_h^k, a_h^k, b_h^k, r_h^k, s_{h+1}^k)\right\}_{k,h=1}^{n,H}$, we denote $n_h(s,a,b) = \sum_{k=1}^n \mathbf{1}\left((s_h^k, a_h^k, b_h^k) = (s,a,b)\right)$ to be the number of times that (s,a,b) is visited at timestep h. We set the empirical reward and the empirical transition kernel as

$$\widehat{r}_h(s,a,b) = r_h(s,a,b), \widehat{P}_h(s'|s,a,b) = \frac{\sum_{k=1}^n \mathbf{1}\left((s_h^k, a_h^k, b_h^k, s_{h+1}^k) = (s,a,b,s')\right)}{\sum_{k=1}^n \mathbf{1}\left((s_h^k, a_h^k, b_h^k) = (s,a,b)\right)}, \tag{2.2}$$

if $n_h(s,a,b) \geq 1$, and $\widehat{r}_h(s,a,b) = 0$, $\widehat{P}_h(s'|s,a,b) = 1/S$ otherwise. In addition, we use $n_h \in \mathbb{R}^{SAB}$ to denote a vector such that $[n_h]_{s,a,b} = n_h(s,a,b)$.

Now we explain Algorithm 1 in detail. First, we split the dataset $\mathcal{D}$ into H small datasets $\{\mathcal{D}_h\}_{h=1}^H$ with the same size. Then we use $\mathcal{D}_h$ to estimate the reward and the transition matrix at timestep h. The data splitting scheme is to remove the dependence between each timestep. Then the value function is estimated via a value-iteration-type algorithm. At each timestep, we maintain both optimistic and pessimistic estimates by adding/minusing a Hoeffding-type bonus. We use the following Hoeffding-type bonus:

$$\underline{b}_h(s_h, a_h, b_h) = \overline{b}_h(s_h, a_h, b_h) = 4\sqrt{\frac{H^2\iota}{n_h(s,a,b) \vee 1}}, \tag{2.3}$$

where $\iota = \log(HSAB/\delta)$. Then we compute the pessimistic estimate $\overline{Q}$ and $\underline{Q}$:

$$\underline{Q}_h = \left(\widehat{r}_h + (\widehat{P}_h \cdot \underline{V}_{h+1}) - \underline{b}_h\right) \vee 0, \overline{Q}_h = \left(\widehat{r}_h + (\widehat{P}_h \cdot \overline{V}_{h+1}) + \overline{b}_h\right) \wedge (H - h + 1). \tag{2.4}$$

Pessimistic estimate $\underline{Q}_h$ is for the max player, which mimics the pessimism in single-agent RL. $\overline{Q}_h$ using a positive bonus is for the min player, which is also a kind of pessimism as the min player's target is to minimize the reward. We compute the NE strategy of the matrix game $\underline{Q}(s,\cdot,\cdot)$ and $\overline{Q}(s,\cdot,\cdot)$ respectively and use the NE value to be the state value $\underline{V}(s)$ and $\overline{V}(s)$. Note that we only solve a zero-sum matrix game, which is computationally efficient [Chen and Deng, 2006].

Remark 2.4.2. If we compute an ϵ_{NE}/H-approximate NE of the matrix game $\underline{Q}(s,\cdot,\cdot)$ and $\overline{Q}(s,\cdot,\cdot)$ at each timestep, then the performance gap will only be enlarged by $\widetilde{O}(\epsilon_{\mathrm{NE}})$.

Algorithm 1 Pessimistic Nash Value Iteration (PNVI)

Input: Offline dataset $\mathcal{D} = \left\{(s_h^k, a_h^k, b_h^k, r_h^k, s_{h+1}^k)\right\}_{k,h=1}^{n,H}$; Failure Probability δ

Initialization: Set $\underline{v}_{H+1}(\cdot) = \overline{v}_{H+1}(\cdot) = 0$

Randomly split the dataset $\mathcal{D}$ into $\{\mathcal{D}_h\}_{h=1}^H$ with $|\mathcal{D}_h| = n/H$

Set $\widehat{r}_h$, $\widehat{P}_h$, $\underline{b}_h$, and $\overline{b}_h$ as (2.2) and (2.3) using the dataset $\mathcal{D}_h$ for all $h \in [H]$

for $h = H, H-1, \ldots, 1$ **do**

  Set $\underline{q}_h(\cdot,\cdot,\cdot)$ and $\overline{q}_h(\cdot,\cdot,\cdot)$ as (2.4)

  Compute the NE of $\underline{q}_h(\cdot,\cdot,\cdot)$ as $(\underline{m}_h(\cdot), \underline{n}_h(\cdot))$

  Compute $\underline{v}_h(\cdot) = \mathbb{E}_{a\sim\underline{m}_h, b\sim\underline{n}_h}\underline{q}_h(\cdot, a, b)$

  Compute the NE of $\overline{q}_h(\cdot,\cdot,\cdot)$ as $(\overline{m}_h(\cdot), \overline{n}_h(\cdot))$

  Compute $\overline{v}_h(\cdot) = \mathbb{E}_{a\sim\overline{m}_h, b\sim\overline{n}_h}\overline{q}_h(\cdot, a, b)$

end for

Output: $\underline{m} = (\underline{m}_1, \underline{m}_2, \ldots, \underline{m}_H)$, $\overline{n} = (\overline{n}_1, \overline{n}_2, \ldots, \overline{n}_H)$, $\{\underline{v}_h\}_{h=1}^H$, $\{\overline{v}_h\}_{h=1}^H$

Theorem 2.4.3. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$ and strategy μ, ν, with probability $1-\delta$, the pessimistic values $\underline{V}_h$ and $\overline{V}_h$ of Algorithm 1 satisfy*

$$\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \le \widetilde{O}\left(\sqrt{C^* SABH^5/n}\right), \mathbb{E}_{\mu,\nu^*}\left[\overline{V}_h(s_h) - V_h^*(s_h)\right] \le \widetilde{O}\left(\sqrt{C^* SABH^5/n}\right),$$

for all $h \in [H]$, where s_h is sampled from the trajectory following the strategy in the expectation.

Proof Sketch. For simplicity, we only show the guarantee for the strategy $\underline{\mu}$ of the max player. First, we show that under good concentration event, the pessimistic value $\underline{V}_h$ is always smaller than the best response value of $\underline{\mu}$, i.e.

$$\underline{V}_h(s) \le V_h^{\underline{\mu},*}(s), \forall h \in [H], s \in \mathcal{S}.$$

Second, we show that the performance gap of $\underline{\mu}$ is bounded by the expected sum of bonus under the strategy $\mu^*, \underline{\nu}$, i.e.

$$V_h^*(s) - V_h^{\underline{\mu},*}(s) \le V_h^{\mu^*,\underline{\nu}}(s_h) - \underline{V}_h(s_h) \le 2\mathbb{E}_{\mu^*,\underline{\nu}}\left[\sum_{t=h}^H \underline{b}_t(s_t, a_t, b_t) | s_h = s\right].$$

Finally, we define a concatenated strategy $\nu' := (\nu_1, \cdots, \nu_{h-1}, \underline{\nu}_h, \cdots, \underline{\nu}_H)$ and then we have

$$\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \le 2\mathbb{E}_{\mu^*,\nu'}\sum_{t=h}^H \underline{b}_t(s_t, a_t, b_t).$$

As Assumption 2.2.2 suggests that (μ^*, ν') is well covered by the exploration strategy ρ, the expected sum of bonus can be bounded. See Appendix A.2 for details. □

Theorem 2.4.3 provides polynomial bounds on the error of the value estimates in Algorithm 1. It can directly imply the following performance gap bound. In addition, it provides guarantees for the reference function that will be utilized in the next section.

Corollary 2.4.4. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$, with probability $1-\delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 1 satisfies* $\mathrm{Gap}(\pi) \le \widetilde{O}\left(\sqrt{C^* SABH^5/n}\right)$.

Theorem 2.4.4 shows that the output strategy of Algorithm 1 is an $\widetilde{O}\left(\sqrt{C^* SABH^5/n}\right)$-approximate NE. The parameter C^* measures how the exploration strategy ρ covers the unilateral strategies (μ^*, ν) and (μ, ν^*) for all μ and ν.

2.4.2 Bernstein-type Algorithm with Reference Advantage Function Decomposition

In this section, we will derive an improved performance gap bound $\widetilde{O}\left(\sqrt{C^* SABH^3/n}\right)$. The extra H^2 is shaved by using Bernstein-type bonus and reference advantage decomposition technique motivated from Xie et al. [2021b]. However, we want to emphasize that zero-sum Markov games are substantially different from MDP and require novel adaptation, which we will describe later.

Due to the space constraint, we put Algorithm 16 in Appendix A.1. Algorithm 16 is different from Algorithm 1 in two aspects. First, we use the reference advantage decomposition to remove an H factor. The dataset is split into three subset with equal size $\mathcal{D}_{\mathrm{ref}}$, $\mathcal{D}_0$, $\mathcal{D}_1$, and $\mathcal{D}_1$ is further split into H subset with equal size $\{\mathcal{D}_{h,1}\}_{h=1}^H$. We run algorithm 1 on dataset $\mathcal{D}_{\mathrm{ref}}$ and we can obtain pessimistic value estimate $\underline{V}_{\mathrm{ref}}$ and $\overline{V}_{\mathrm{ref}}$ with guarantees by Theorem 2.4.3. Then we use dataset $\mathcal{D}_0$ to estimate $P_h \underline{V}_{h+1}^{\mathrm{ref}}$ and dataset $\mathcal{D}_{h,1}$ to estimate $P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})$. Second, we use a Bernstein-type bonus to remove another H factor. Our updating formulas of $\underline{Q}_h$ and $\overline{Q}_h$ are

$$\underline{Q}_h = \underline{Q}_h^{\mathrm{ref}} \vee [\widehat{r}_{h,0} + (\widehat{P}_{h,0} \cdot \underline{V}_{h+1}^{\mathrm{ref}}) - \underline{b}_{h,0} + (\widehat{P}_{h,1} \cdot (\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})) - \underline{b}_{h,1}], \tag{2.5}$$

$$\overline{Q}_h = \overline{Q}_h^{\mathrm{ref}} \wedge [\widehat{r}_{h,0} + (\widehat{P}_{h,0} \cdot \overline{V}_{h+1}^{\mathrm{ref}}) + \overline{b}_{h,0} + (\widehat{P}_{h,1} \cdot (\overline{V}_{h+1} - \overline{V}_{h+1}^{\mathrm{ref}})) + \overline{b}_{h,1}], \tag{2.6}$$

where we truncate by the reference function to ensure monotonic update so that $\underline{Q}_h$ and $\overline{Q}_h$ are more accurate pessimistic/optimistic estimate compared with the reference function $\underline{Q}_h^{\text{ref}}$ and $\overline{Q}_h^{\text{ref}}$. The bonus functions are defined as

$$\underline{b}_{h,0} = c\left(\sqrt{\frac{\text{Var}_{\widehat{P}_{h,0}}(\underline{V}_{h+1}^{\text{ref}})\iota}{n_{h,0} \vee 1}} + \frac{H\iota}{n_{h,0} \vee 1}\right), \bar{b}_{h,0} = c\left(\sqrt{\frac{\text{Var}_{\widehat{P}_{h,0}}(\overline{V}_{h+1}^{\text{ref}})\iota}{n_{h,0} \vee 1}} + \frac{H\iota}{n_{h,0} \vee 1}\right), \tag{2.7}$$

$$\underline{b}_{h,1} = c\left(\sqrt{\frac{\text{Var}_{\widehat{P}_{h,1}}(\underline{V}_{h+1} - \underline{V}_{h+1}^{\text{ref}})\iota}{n_{h,1} \vee 1}} + \frac{H\iota}{n_{h,1} \vee 1}\right), \bar{b}_{h,1} = c\left(\sqrt{\frac{\text{Var}_{\widehat{P}_{h,1}}(\overline{V}_{h+1} - \overline{V}_{h+1}^{\text{ref}})\iota}{n_{h,1} \vee 1}} + \frac{H\iota}{n_{h,1} \vee 1}\right), \tag{2.8}$$

where c is some universal constant and $\text{Var}_{\widehat{P}_{h,0}}(V)$, $\text{Var}_{\widehat{P}_{h,1}}(V)$, $n_{h,0}$, $n_{h,1}$ are all SAB-dimension vectors and the operations are element-wise.

Theorem 2.4.5. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$ and $n \geq C^* SABH^4$, with probability $1 - \delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 16 satisfies* $\text{Gap}(\pi) \leq \widetilde{O}\left(\sqrt{C^* SABH^3/n}\right)$.

Remark 2.4.6. $n \geq C^* SABH^4$ serves as the burn-in cost, which is standard in the literature. See a more detailed discussion in Li et al. [2021].

Proof of Sketch. For simplicity we only show the guarantee for the strategy $\underline{\mu}$ of the max player. First we show that under good concentration event, the pessimistic value $\underline{V}_h$ is always sandwiched by the reference value $\underline{V}_h^{\text{ref}}$ and the best response value of $\underline{\mu}$, i.e.,

$$\underline{V}_h^{\text{ref}}(s) \leq \underline{V}_h(s) \leq V_h^{\underline{\mu},*}(s), \forall h \in [H], s \in \mathcal{S}.$$

Second, we show that the performance gap of $\underline{\mu}$ is bounded by the expected sum of bonus under the strategy $\mu^*, \underline{\nu}$, i.e.,

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \left[\underline{b}_{h,0}(s_h, a_h, b_h) + \underline{b}_{h,1}(s_h, a_h, b_h)\right].$$

Then we bound the first term by

$$\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,0}(s_h, a_h, b_h) \leq \widetilde{O}\left(\sqrt{C^* SABH^3/n} + \sqrt{C^* SABH^3/n}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right),$$

where $\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}$ is the square root of the term we want to bound. The second term can be bounded similarly. Finally solving the self-bounding inequality for $V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)$ and we have

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq \widetilde{O}\left(\sqrt{C^* SABH^3/n}\right).$$

We utilizes Theorem 2.4.3 to provide guarantee for the error of the reference function and $\underline{V}_h^{\text{ref}}(s) \leq \underline{V}_h(s) \leq V_h^{\underline{\mu},*}(s)$ to bound the variance of the estimation error. See Appendix A.3 for details. □

As MDP are degenerated Markov games with one player having a fixed action, Markov games inherit the lower bounds of MDP. Comparing with the lower bound $\widetilde{\Omega}\left(\sqrt{C^* SH^3/n}\right)$ [Xie et al., 2021b], our bound is already tight in C^*, S, H. The extra AB factor is from the Cauchy-Schwarz inequality and the fact that the NE of zero-sum Markov games can be a mixed strategy while deterministic optimal policy always exists for MDP. It is unknown whether the AB factor is removable and we leave it to future work.

2.4.3 Minimax Optimal Sample Complexity Bounds

In this section, we show that Algorithm 16 directly adapts to two popular settings, i.e. Assumption 2.2.3 (uniform concentration assumption) and turn-based Markov games. In addition, minimax sample complexity can be achieved under both settings. The proof is deferred to Appendix A.4.

Theorem 2.4.7. *Set* $d_m = \min\left\{d_h^\rho(s,a,b) : h \in [H], (s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}\right\}$. *Suppose Assumption 2.2.3 holds. For any* $0 < \delta < 1$ *and* $n \geq H^4/d_m$, *with probability* $1-\delta$, *the output policy* $\pi = (\underline{\mu}, \overline{\nu})$ *of Algorithm 16 satisfies* $\text{Gap}(\pi) \leq \widetilde{O}\left(\sqrt{H^3/(nd_m)}\right)$.

This bound has no explicit dependence on AB because the Cauchy-Schwarz inequality can be applied on $d_h^{\mu^*,\underline{\nu}}$ instead of $\sqrt{d_h^{\mu^*,\underline{\nu}}}$ (See the proof of Theorem A.4.1). As the lower bound $\widetilde{\Omega}\left(\sqrt{H^3/(nd_m)}\right)$ for MDP [Yin and Wang, 2021b] is the lower bound for Markov games, Algorithm 16 achieves minimax sample complexity under assumption 2.2.3.

Theorem 2.4.8. *Suppose Assumption 2.2.2 holds for a turn-based Markov games. For any* $0 < \delta < 1$ *and* $n \geq C^* SH^4$*, with probability* $1 - \delta$*, the output policy* $\pi = (\underline{\mu}, \overline{\nu})$ *of Algorithm 16 satisfies* $\mathrm{Gap}(\pi) \leq \widetilde{O}\left(\sqrt{C^* SH^3/n}\right)$.

As the lower bound is $\widetilde{\Omega}\left(\sqrt{C^* SH^3/n}\right)$ [Xie et al., 2021b], Algorithm 16 can achieve the minimax sample complexity for turn-based Markov games under assumption 2.2.2. The difference is due to turn-based Markov games always have pure NE strategies (See the proof of Theorem A.4.6).

2.5 Conclusion

In this work, we study the minimal dataset coverage assumption for NE learning in two-player zero-sum Markov games. We show that single strategy concentration is not enough for NE learning. Instead, we find a minimal coverage assumption for NE learning and design an algorithm with sample complexity tight in $C^*, \mathcal{S}, H$ under such assumption based on novel techniques. In addition, the algorithm can achieve minimax sample complexity in certain settings. We believe this work can shed new light on offline MARL.

Here we list several open problems for future work. One direction is to find the minimax sample complexity of offline Markov games under the unilateral concentration. Importantly, it is unclear whether AB factor can be reduced [Bai et al., 2020]. Another direction is to design efficient algorithms for offline MARL with a large number of agents without sample complexity scales exponentially with the number of agents.

Chapter 3

OFFLINE MULTI-PLAYER GENERAL-SUM MARKOV GAMES

This chapter is based on Cui and Du [2022b], with Simon S. Du.

3.1 Introduction

Multi-agent reinforcement learning (MARL) is about decision making in a multi-agent system under uncertainty, which has achieved significant success in solving a wide range of tasks such as GO [Silver et al., 2017], Poker [Brown and Sandholm, 2019] and autonomous deriving [Shalev-Shwartz et al., 2016]. One standard setting in MARL is multi-player general-sum Markov games where each player deploys a policy to maximize its own total reward while the evolution of the environment depends on the policies of all the players [Zhang et al., 2021a]. During the learning process, each player needs to identify the environment dynamics as well as compete/cooperate with other agents.

One emerging subarea is offline MARL, where plenty of empirical works have been done while the theoretical understanding is still largely missing [Pan et al., 2021, Jiang and Lu, 2021, Meng et al., 2021]. Offline RL has received tremendous attention because in various practical scenarios, it is expensive to acquire online data while offline log data is accessible.

The offline single-agent RL is well studied in the literature. Researchers have identified the minimal dataset coverage assumption, *single policy coverage* (the dataset only needs to cover an optimal policy), under which one can learn a near-optimal policy efficiently. Furthermore, they have developed algorithms with minimax sample complexity [Xie et al., 2021b, Li et al., 2022b]. For offline MARL, recent works showed that single policy coverage is not sufficient and *unilateral coverage* is necessary for learning a Nash equilibrium (NE) strategy, i.e., the dataset covers all the joint strategies that only differ from an NE at one player [Cui and Du, 2022a, Zhong et al., 2022]. This condition is also sufficient for two-player zero-sum Markov games with sample complexity $\widetilde{O}(AB)$ (ignoring other quantities),

where A, B are the number of actions for each player [Cui and Du, 2022a]. However, it is still unclear if it is sufficient for multi-player general-sum Markov game.

One major challenge in MARL is the *curse of multiagents* [Jin et al., 2021b]. Suppose the number of actions for player j is A_j and there are m players. Then the joint action space is of size $\prod_{j\in[m]} A_j$, which grows exponentially with the number of players m. As a result, any algorithm that depends linearly on the cardinality of the joint action space can hardly be applied to real-world scenarios. In online MARL, Jin et al. [2021b] and Song et al. [2021a] show that finding the coarse correlated equilibrium, which is a weaker equilibrium notion than NE, only requires $\widetilde{O}(\max_{j\in[m]} A_j)$ samples, thus breaking the curse of multiagents. In this paper, we study the following question:

Can we find NE in offline m-player general-sum Markov game with unilateral coverage and without the exponential dependence on the number of players?

In this paper, we answer this question in the affirmative. We highlight our contributions below.

3.1.1 Main Novelties and Contributions

1. Strategy-wise concentration principle. We propose the strategy-wise concentration principle. Point-wise concentration is a standard technique in computing the confidence interval for each state-action pair [Azar et al., 2017, Liu et al., 2021a, Xie et al., 2021b, Cui and Du, 2022a]. However, the straightforward extension to MARL suffers from the curse of multiagents as the NE can be a mixed strategy. Different from the point-wise concentration technique, strategy-wise concentration directly *estimates each strategy, which allows a tighter confidence interval that can avoid the dependence on the joint action space.* We give a technical overview in Section 3.1.2. In addition, we show that the strategy-wise confidence bound is always a convex function so that the empirical *best response strategy can always be a deterministic strategy*, which is critical to the computational efficiency.

2. Improved algorithm for offline two-player zero-sum Markov games. For offline two-player zero-sum Markov games, we utilize its special structure to develop a maximin-optimization-type algorithm. Though the nonlinear strategy-wise bonus breaks

the bilinear structure of the zero-sum game, we show that by solving a maximin optimization problem we can still output a good strategy. In addition, we can solve it efficiently using any black-box algorithms for Lipschitz-continuous convex optimization. Our sample complexity improves the AB factor in Cui and Du [2022a] to $(A + B)$.

3. The first algorithm for offline multi-player general-sum Markov games. For multi-player general-sum Markov games, we develop a *surrogate function* to approximate performance gap and then show that the minimizer of the surrogate function approximates NE well. The surrogate function is constructed by optimistic best response values and pessimistic values. Interestingly, to our knowledge, this is the first time that optimism has been used in offline RL algorithms. Our result validates that unilateral coverage is sufficient for general-sum Markov games and our sample complexity rate scales with $\widetilde{O}(\sum_{j=1}^{m} A_j)$ (ignoring other parameters), thus breaking the curse of multiagents.

4. Incorporating pre-specified strategy class. Lastly, our algorithm allows exploiting the prior knowledge about the NE strategy with an adaptive sample complexity bound. Pre-specified policy class has been widely used in empirical works where the policy class is parameterized by neural networks (e.g., Mnih et al. [2016], Haarnoja et al. [2018], Lowe et al. [2017]), and single-agent RL theory as well (e.g., Auer et al. [2002], Agarwal et al. [2021]), but has not been investigated in MARL theory. In this paper, we take a step to incorporate prior knowledge in the MARL setting. Our performance guarantee only depends on the logarithmic covering number of the pre-specified strategy class, which is always upper bounded by $\sum_{j\in[m]} A_j$, but can be smaller. To the best of our knowledge, this is the first paper that considers a pre-specified strategy class in MARL theory.

3.1.2 Technical Overview of Strategy-wise Concentration

To give some intuition about this technique, let us consider a toy problem. Suppose there are m random variables $\{x^i\}_{i=1}^m$ and we want to obtain a pessimistic estimate of their average $x = \sum_{i\in[m]} x^i/m$. We have n/m observations for each x^i. The point-wise concentration estimate corresponds to estimating each x^i and then aggregating the results. The pessimistic estimate of x^i would be $\widehat{x}^i - \widetilde{O}(\sqrt{m/n})$ where $\widehat{x}^i$ is the empirical mean, and the aggregated

mean of these pessimistic estimates would be $\widehat{x} - \widetilde{O}(\sqrt{m/n})$ where $\widehat{x}$ is the empirical mean of all data. The strategy-wise concentration estimate corresponds to directly using all the samples to estimate the average of $\{x\}_{i=1}^m$ and obtain the pessimistic estimate as $\widehat{x} - \widetilde{O}(1/\sqrt{n})$. This example shows that the point-wise estimate will lead to an extra m factor. In MARL, m is the cardinality of the joint action space, which implies that point-wise concentration can be exponentially worse than strategy-wise concentration. Note that this is not an issue in single-agent MDP as the optimal policy is always deterministic but leads to severe suboptimality in the multi-agent case where NE can be a mixed strategy.

3.1.3 Related Work

Online Multi-agent RL. Markov games can be solved via dynamic programming when the rewards and transition dynamics are given [Hansen et al., 2013, Perolat et al., 2015]. If the environment is unknown, reinforcement learning algorithms are applied with different sampling oracles. One particular line of research is online Markov games, including two-player zero-sum Markov games [Liu et al., 2021a, Dou et al., 2021, Xie et al., 2020a, Bai et al., 2020, Huang et al., 2021] and multi-player general-sum Markov games [Zhong et al., 2021, Mao et al., 2021a, Jin et al., 2021b, Song et al., 2021a]. Rubinstein [2016] proves an exponential (in the number of players) lower bound for learning the NE strategy in m-player general-sum game while others show that the correlated equilibrium and coarse correlated equilibrium admit $\mathrm{poly}(m, \max_{j\in[m]} A_j, H, S)$-sample complexity algorithms [Mao et al., 2021a, Jin et al., 2021b, Song et al., 2021a]. Our upper bounds for m-player general-sum games depend polynomially on all parameters, which do not contradict the hardness result in Rubinstein [2016] because the assumptions on the offline dataset provide additional information about the NE.

Offline Single-agent RL. The simplest dataset assumption for offline RL is uniform coverage, i.e., the dataset covers all the state-action pairs. This assumption dates back to Szepesvári and Munos [2005]. The minimax sample complexity has been well studied for both tabular case and function approximation [Xie and Jiang, 2021a, Yin et al., 2020a, 2021a, Ren et al., 2021a]. Recently it has been shown that only covering the optimal policy

is sufficient for offline RL under different settings [Rashidinejad et al., 2021a, Yin and Wang, 2021a, Xie et al., 2021b, Jin et al., 2021d, Uehara and Sun, 2021a, Zanette et al., 2021b, Xie et al., 2021a]. These works design provably efficient algorithms based on the principle of pessimism.

Offline Multi-agent RL. Offline MARL theory is still at a primary stage. Previous works mostly focused on uniform coverage assumption, i.e. all state-action pairs or all policies are covered [Sidford et al., 2020, Cui and Yang, 2021, Zhang et al., 2020a, 2021b, Abe and Kaneko, 2020, Subramanian et al., 2021]. Recently, Cui and Du [2022a] and Zhong et al. [2022] show that the unilateral coverage assumption is the minimal dataset coverage assumption for learning NE in Markov games. In addition, [Cui and Du, 2022a] proposes a pessimism-type algorithm with $\widetilde{O}(SABH^3C(\pi^*)/\epsilon^2)$ sample complexity for tabular two-player zero-sum Markov game and [Zhong et al., 2022] provides a similar algorithm for linear two-player zero-sum Markov games.

3.2 Preliminaries

Notations. We use $D(\mathcal{X})$ to denote the single point distributions over the finite set $\mathcal{X}$. For example, $D(\mathcal{A})$ to represent the policies that deterministically choose one of the actions in $\mathcal{A}$. We use $\pi^s_{j,h} \in \Delta(\mathcal{A}_j)$ as a concise notation of $\pi_{j,h}(\cdot|s)$ and $P_h(s,\mathbf{a})$ to denote $P_h(\cdot|s,\mathbf{a})$, which will be defined in the following section. We use $-j$ in subscript to denote all the players except player j. We use bold letter to denote vectors, e.g. $\mathbf{a}$ is a vector and a_j is the j-th element of $\mathbf{a}$. We let $O(\cdot)$ hide absolute constants and $\widetilde{O}(\cdot)$ hide polylog terms as well. The L1 norm of a vector in $\mathbb{R}^d$ is $\|\mathbf{a}\|_1 = \sum_{i=1}^d |a_i|$. We denote the projection as $\text{proj}_{[a,b]}(x) := \max\{a, \min\{b, x\}\}$.

Multi-player General-sum Markov Game. A multi-player general-sum Markov game is described by a tuple $\mathcal{G} = (\mathcal{S}, \mathcal{A} = \prod_{j\in[m]} \mathcal{A}_j, P, R, H)$, where $\mathcal{S}$ is the state space with cardinality S, m is the number of players, $\mathcal{A}_j$ is the action space of player j with cardinality A_j, $P = (P_1, P_2, \cdots, P_H)$ with $P_h \in \mathbb{R}^{S\times\prod_{i\in[m]} A_i \times S}$ being the (unknown) transition matrix at timestep $h \in [H]$, $R = \{R_h(\cdot|s_h,\mathbf{a}_h)\}_{h=1}^H$ with $R_h(\cdot|s_h,\mathbf{a}_h)$ being a distribution on $[0,1]^m$ with mean $\mathbf{r}_h(s_h,\mathbf{a}_h) \in [0,1]^m$ as the (unknown) reward distribution at timestep h. At timestep h, all players choose their actions *simultaneously* and a reward vector is

sampled from the reward distribution $\mathbf{r}_h \sim R_h(\cdot|s_h, \mathbf{a}_h)$, where s_h is the current state and $\mathbf{a}_h = (a_{h,1}, a_{h,2}, \cdots, a_{h,m})$ is the joint action. Each player j receives its own reward $r_{h,j}$ with support on $[0, 1]$ and mean $r_{h,j}(s_h, \mathbf{a}_h)$. The state then transits to s_{h+1} following the distribution of $P_h(\cdot \mid s_h, \mathbf{a}_h)$. The game terminates at timestep $H + 1$. We assume that the initial state s_1 is fixed because for a stochastic initial state, one can add s_0 as the initial state instead and it transits to s_1 following the initial distribution.

We denote a joint strategy as $\pi = (\pi_1, \pi_2, \cdots, \pi_m)$, where $\pi_j = (\pi_{1,j}, \pi_{2,j}, \cdots, \pi_{H,j})$ and $\pi_{h,j} : \mathcal{S} \to \Delta(\mathcal{A}_j)$ is the strategy of player j at timestep h where $\Delta(\mathcal{A}_j)$ is the probability simplex over $\mathcal{A}_j$. We use Π^{full} to denote the set of all the possible joint strategies. We define the state value function and state-action value function under strategy π for each player $j \in [m]$:

$$V_{h,j}^{\pi}(s_h) := \mathbb{E}_{\pi}\left[\sum_{t=h}^{H} r_{t,j}(s_t, \mathbf{a}_t) \,\middle|\, s_h\right], Q_{h,j}^{\pi}(s_h, \mathbf{a}_h) := \mathbb{E}_{\pi}\left[\sum_{t=h}^{H} r_{t,j}(s_t, \mathbf{a}_t) \,\middle|\, s_h, \mathbf{a}_h\right],$$

where the expectation is over the randomness of the environment and the joint strategy π. For a fixed player j, if all the other player's strategies are fixed, then player j can play the best response strategy to maximize its own total reward. We define π_{-j} to be the strategy for all players except player j and define the best response value to be $V_{h,j}^{*,\pi_{-j}}(s_h) := \max_{\pi_j} V_{h,j}^{\pi_j,\pi_{-j}}(s_h)$.

It is well-known that Nash equilibrium strategy exists for general-sum Markov games. Note that there could be multiple NE strategies with different value functions. We use the following performance gap to evaluate a strategy π: $\text{Gap}(\pi) := \sum_{j\in[m]} \left[V_{1,j}^{*,\pi_{-j}}(s_1) - V_{1,j}^{\pi}(s_1)\right]$. This metric is always non-negative and we say π is an ϵ-approximate NE if and only if $\text{Gap}(\pi) \le \epsilon$.

Two-player Zero-sum Markov Game. A general-sum Markov game becomes a two-player zero-sum Markov game if there are only two players and the reward $r_h \sim R_h(\cdot|s, a_1, a_2)$ always satisfies $r_{h,1} + r_{h,2} = 0$ for all $h \in [H]$, $s \in \mathcal{S}$, $a_1 \in \mathcal{A}_1$ and $a_2 \in \mathcal{A}_2$. Following the literatures on two-player zero-sum Markov games, we use slightly different notations for this setting. There is only one reward function r shared by both players, which is the reward function $\{r_{h,1}\}_{h=1}^{H}$ for player 1 and the target of player 2 is to minimize the total reward. We denote $\mu = \pi_1$ and $\nu = \pi_2$ to be the strategy for each player, $a = a_1$ and

$b = a_2$ to be the action for each player, $\Pi^{\max} = \Pi_1$ and $\Pi^{\min} = \Pi_2$ to be the strategy class for each player to remove extra subscripts. One can derive the performance gap under the new notations for two-player zero-sum Markov games: $\text{Gap}(\pi) := V_1^{*,\nu}(s_1) - V_1^{\mu,*}(s_1)$.

Offline Markov Game. In offline RL, the dataset is collected beforehand and no further sampling is allowed. Here we consider offline multi-player general-sum Markov game. The framework for offline two-player zero-sum Markov game is similar with the slightly different notations as we mentioned.

We assume that the algorithm has access to an offline dataset $\mathcal{D} = \{(s_h^k, \mathbf{a}_h^k, \mathbf{r}_h^k, s_{h+1}^k)\}_{h,k=1,1}^{H,n}$ that satisfies Assumption 3.2.1. The assumption states that the dataset is independently generated from the underlying Markov game, which is used in [Jin et al., 2021d, Zhong et al., 2022]. The target of offline Markov game is to find a strategy π with as small performance gap as possible by utilizing the dataset $\mathcal{D}$. One closely related assumption is that the dataset is generated from some behavior strategy [Xie et al., 2021b, Cui and Du, 2022a]. Though this kind of dataset does not satisfy Assumption 3.2.1 directly due to the dependence within the trajectory, we can construct a compliant dataset by using the subsampling technique in Li et al. [2022b] while the number of samples is still of the same order.

Assumption 3.2.1. The dataset $\mathcal{D}$ is compliant with the multi-player general-sum markov game, i.e.,

$$\mathbb{P}_{\mathcal{D}}(s_{h+1}^k = s \mid s_h^k, \mathbf{a}_h^k) = P_h(s_{h+1} = s \mid s_h = s_h^k, \mathbf{a}_h = \mathbf{a}_h^k),$$

$$\mathbb{P}_{\mathcal{D}}(\mathbf{r}_h^k = \mathbf{r} | s_h^k, \mathbf{a}_h^k) = R_h(\mathbf{r}_h = \mathbf{r} | s_h = s_h^k, a_h = \mathbf{a}_h^k), \forall j \in [m],$$

for all $h \in [H]$ and $k \in [n]$. In addition, all tuples $(s_h^k, \mathbf{a}_h^k, \mathbf{r}_h^k, s_{h+1}^k)$ are independent.

Pre-specified Policy Class. We also consider the case when we know that the NE is possibly in a given subset of Π^{full}. We denote this subset as Π and our target is to find the best strategy in Π. Note that we do not assume NE is indeed in Π. In addition, by choosing $\Pi = \Pi^{\text{full}}$ we can recover the standard setting. To measure the complexity of Π, we use the covering number.

Definition 3.2.2. (Covering Number) For any error level ϵ_{cover} and strategy class Π, we

define

$$\mathcal{N}(\Pi, \epsilon_{\text{cover}}) := \sum_{s\in\mathcal{S}, h\in[H]} \prod_{j\in[m]} \left|\mathcal{C}(\Pi_{h,j}(s), \epsilon_{\text{cover}})\right|,$$

where $\Pi_{h,j}(s) = \{\pi_h^j(\cdot|s) : \pi \in \Pi\}$ is a subset of $\Delta(\mathcal{A}_i)$ and $\mathcal{C}(\Pi_{h,j}(s), \epsilon_{\text{cover}})$ is an ϵ_{cover}-covering of $\Pi_{h,j}(s)$ with respect to the L1 norm $\|\cdot\|_1$.

Our performance guarantee will only have logarithm dependence on $\mathcal{N}(\Pi, \epsilon_{\text{cover}})$. As $\Pi_{h,j}(s)$ is a subset of $\Delta(\mathcal{A}_j)$, we always have $\log(\mathcal{N}(\Pi, \epsilon_{\text{cover}})) \le \widetilde{O}(\sum_{j\in[m]} A_j \log(1/\epsilon_{\text{cover}}))$ and if Π is a finite set, we have $\log(\mathcal{N}(\Pi, \epsilon_{\text{cover}})) \le \log(SH|\Pi|)$ (see Appendix B.2.1 for the proof). In this paper we will choose $\epsilon_{\text{cover}} = \frac{1}{\sum_{j\in[m]} A_j m H^2 n^2}$, which only leads to logarithm dependence on these quantities. In later sections, we will omit ϵ_{cover} to simplify the notation.

For any joint strategy π, we call (π'_j, π_{-j}) for any strategy π' and $j \in [m]$ as a unilateral strategy of π. Previous works show that only covering an NE is not sufficient, and covering all the unilateral strategies of an NE is necessary for learning the NE in Markov games [Cui and Du, 2022a, Zhong et al., 2022]. We use unilateral coefficient to quantify how the dataset covers all the unilateral strategies of a strategy π. If we assume that the dataset is sampled from some (unknown) distribution, i.e. $(s_h, \mathbf{a}_h) \sim d_h(\cdot,\cdot)$ for all $h \in [H]$, we can define the population unilateral coefficient.

Definition 3.2.3. For any strategy π, the population unilateral coefficient is defined as $C(\pi) := \max_{h,j,\pi',s_h,\mathbf{a}_h} \frac{d_h^{\pi'_j,\pi_{-j}}(s_h,\mathbf{a}_h)}{d_h(s_h,\mathbf{a}_h)}$.

Cui and Du [2022a] provide a sample complexity result for zero-sum Markov games with dependence on $C(\pi^*)$. We can also define the empirical unilateral coefficient using the empirical distribution.

Definition 3.2.4. Define the empirical dataset distribution as $\widehat{d}_h(s,\mathbf{a}) = n_h(s,\mathbf{a})/n$, for all $h \in [H], s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$, where $n_h(s,\mathbf{a})$ is the number of times that $(s,\mathbf{a})$ appears in the dataset for timestep h. For any strategy π, the empirical unilateral coefficient is defined as $\widehat{C}(\pi) := \max_{h,j,\pi',s_h,\mathbf{a}_h} \frac{d_h^{\pi'_j,\pi_{-j}}(s_h,\mathbf{a}_h)}{\widehat{d}_h(s_h,\mathbf{a}_h)}$.

The empirical unilateral coefficient can lead to dataset-dependent bound that has no dependence on the underlying distribution of the dataset. In addition, $\widehat{C}(\pi)$ can be bounded

by $2C(\pi)$ (Proposition 3.2.5) so results based on $\widehat{C}(\pi)$ directly transfer to $C(\pi)$. Note that $\widehat{C}(\pi)$ and $C(\pi)$ are both unknown to the algorithm and only appear in the analysis and theorems.

Proposition 3.2.5. *Suppose* $p_{\min} = \min_{s,\mathbf{a},h}\{d_h(s,\mathbf{a}) : d_h(s,\mathbf{a}) > 0\}$. *If* $n \geq \frac{8\log(S\Pi_{j\in[m]}A_jH/\delta)}{p_{\min}}$, *with probability* $1-\delta$, *for all strategy* π, *we have* $2C(\pi) \geq \widehat{C}(\pi)$.

3.3 An Improved Algorithm for Offline Two-player Zero-sum Markov Game

In this section, we propose a new algorithm for offline zero-sum Markov game based on two novel techniques, i.e., strategy-wise concentration and maximin-optimization-based algorithm. We then show that this algorithm is computationally efficient and can (almost) find the best strategy in strategy class Π with favorable sample complexity.

Let us first define some notations. Given a dataset $\mathcal{D} = \{(s_h^k, a_h^k, b_h^k, r_h^k, s_{h+1}^k)\}_{k,h=1}^{n,H}$, we denote $n_h(s,a,b) = \sum_{k=1}^n \mathbf{1}\left((s_h^k, a_h^k, b_h^k) = (s,a,b)\right)$ and $\mathcal{K}_h(s) = \{(a,b) \in \mathcal{A}\times\mathcal{B} : n_h(s,a,b) \neq 0\}$. If $n_h(s,a,b) \neq 0$, we set

$$\widehat{r}_h(s,a,b) = \frac{\sum_{k=1}^n r_h^k \mathbf{1}\left((s_h^k, a_h^k, b_h^k) = (s,a,b)\right)}{n_h(s,a,b)}, \tag{3.1}$$

$$\widehat{P}_h(s'|s,a,b) = \frac{\sum_{k=1}^n \mathbf{1}\left((s_h^k, a_h^k, b_h^k, s_{h+1}^k) = (s,a,b,s')\right)}{n_h(s,a,b)}, \tag{3.2}$$

otherwise we have

$$\widehat{r}_h(s,a,b) = 0, \widehat{P}_h(s'|s,a,b) = 0. \tag{3.3}$$

Based on this empirical Markov game, we can perform value-iteration-type algorithm. Here we describe our algorithm for player 1. For each timestep h, we first compute the the state-action values based on the estimates at timestep $h+1$:

$$\underline{Q}_h(s,a,b) = \widehat{r}_h(s,a,b) + \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1} \right\rangle, \tag{3.4}$$

Then instead of adding the bonus on state-action estimates directly to ensure pessimism as used in Cui and Du [2022a] and Zhong et al. [2022], we first estimate the state value functions for strategy μ_h^s, ν_h^s and then add the bonus on them instead.

$$\underline{V}_h^{\mu_h^s,\nu_h^s}(s) = \mathbb{E}_{a\sim\mu_h^s, b\sim\nu_h^s}\underline{Q}_h(s,a,b) - b_h(s,\mu_h^s,\nu_h^s), \tag{3.5}$$

Algorithm 2 **S**trategy-wise **B**onus + **M**axi**M**in Optimization (SBMM)

Input: Offline dataset $\mathcal{D}$

Initialization: $\underline{v}_{H+1}(s) = \overline{v}_{H+1}(s) = 0$ for all $s \in \mathcal{S}$

for $h = H, H-1, \ldots, 1$ **do**

Player 1

Approximately solve

$$\underline{m}_h^s = \operatorname*{argmax}_{\mu_h^s \in \Pi_h^{\max}(s)} \min_{\nu_h^s \in D(\mathcal{B})} \underline{v}_h^{\mu_h^s, \nu_h^s}(s)$$

where $\underline{v}_h^{\mu_h^s, \nu_h^s}(s)$ is defined by (3.4) and (3.5) and $\underline{m}_h^s$ satisfies (3.10).

Solve

$$\underline{n}_h^s = \operatorname*{argmin}_{\nu_h^s \in D(\mathcal{B})} \underline{v}_h^{\underline{m}_h^s, \nu_h^s}(s)$$

and set

$$\underline{v}_h(s) = \mathrm{proj}_{[0, H-h+1]}\left(\underline{v}_h^{\underline{m}_h^s, \underline{n}_h^s}(s)\right)$$

Player 2

Approximately solve

$$\overline{n}_h^s = \operatorname*{argmin}_{\nu_h^s \in \Pi_h^{\min}(s)} \max_{\mu_h^s \in D(\mathcal{A})} \overline{v}_h^{\mu_h^s, \nu_h^s}(s)$$

where $\overline{v}_h^{\mu_h^s, \nu_h^s}(s)$ is defined by (3.8) and (3.9) and $\overline{n}_h^s$ satisfies (3.11).

Solve

$$\overline{m}_h^s = \operatorname*{argmax}_{\mu_h^s \in D(\mathcal{A})} \overline{v}_h^{\mu_h^s, \overline{n}_h^s}(s)$$

and set

$$\overline{v}_h(s) = \mathrm{proj}_{[0, H-h+1]}\left(\overline{v}_h^{\overline{m}_h^s, \overline{n}_h^s}(s)\right)$$

end for

Output: $\pi^{\mathrm{output}} = (\underline{m}, \overline{n})$

where

$$b_h(s, \mu_h^s, \nu_h^s) = H\sqrt{\sum_{(a,b) \in \mathcal{K}_h(s)} \frac{\mu_h^s(a)^2 \nu_h^s(b)^2}{n_h(s,a,b)} \log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n, \tag{3.6}$$

with $\iota = 32 \log(2ABSHn/\delta)$. We also present the bonus from point-wise concentration used

in Cui and Du [2022a] to better compare them, $b_h^{\text{point}}(s, \mu_h^s, \nu_h^s) = H \sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h^s(a)\nu_h^s(b)\sqrt{\frac{\iota}{n_h(s,a,b)}}$.

As a concrete example, if μ_h^s and ν_h^s are uniform distribution on $\mathcal{A}$ and $\mathcal{B}$, then $b_h(s, \mu_h^s, \nu_h^s)$ is smaller than $b_h^{\text{point}}(s, \mu_h^s, \nu_h^s)$ for an order of $\sqrt{AB}$. Finally to obtain the pessimistic value estimate, we solve the following optimization problem

$$\underline{V}_h(s) = \max_{\mu_h^s \in \Pi_h^{\max}(s)} \min_{\nu_h^s \in D(\mathcal{B})} \underline{V}_h^{\mu_h^s, \nu_h^s}(s). \tag{3.7}$$

Here recall that $D(\mathcal{B})$ represents all the deterministic strategies in $\mathcal{B}$. Our algorthm is similar for player 2 with the following $\overline{Q}$ and $\overline{V}$ estimation:

$$\overline{Q}_h(s,a,b) = \widehat{r}_h(s,a,b) + \left\langle \widehat{P}_h(s,a,b), \overline{V}_{h+1} \right\rangle + H\mathbf{1}\{(a,b) \notin \mathcal{K}_h(s)\}, \tag{3.8}$$

$$\overline{V}_h^{\mu_h^s, \nu_h^s}(s) = \mathbb{E}_{\mu_h^s, \nu_h^s} \overline{Q}_h(s,a,b) + b_h(s, \mu_h^s, \nu_h^s). \tag{3.9}$$

The additional $H\mathbf{1}\{(a,b) \notin \mathcal{K}_h(s)\}$ term in (3.8) compared with (3.4) is to compensate the underestimate by (3.3).

3.3.1 Computational Efficiency

For computational efficiency, we start with the following characterization about our bonus.

Proposition 3.3.1. *$\underline{V}_h^{\mu_h^s, \nu_h^s}(s)$ is concave and $\overline{V}_h^{\mu_h^s, \nu_h^s}(s)$ is convex w.r.t. μ_h^s and ν_h^s respectively.*

Proposition 3.3.1 explains why the inner minimization in (3.7) is over the deterministic strategy class as the minimum of a concave function over the probability simplex is achieved at the vertexes, i.e. deterministic strategies. The proof of Proposition 3.3.1 is provided in Appendix B.2.2.

Previous works solve the NE (saddle point) of $\underline{V}_h^{\mu_h^s, \nu_h^s}(s)$ as the point-wise bonus maintains the bilinear structure [Cui and Du, 2022a, Zhong et al., 2022]. Though here $\underline{V}_h^{\mu_h^s, \nu_h^s}(s)$ no longer enjoys the strong duality, we will show that solving the maximin problem is enough to obtain a good strategy for player 1. As the inner minimization is only on a feasible set of size B, this problem can be solved efficiently by using projected gradient descent [Bubeck

et al., 2015]. We assume that we solve the maximin and the minimax optimization problem to ϵ_{opt}-optimality, i.e.

$$\min_{\nu_h^s \in D(\mathcal{B})} \underline{V}_h^{\mu_h^s, \nu_h^s}(s) \geq \max_{\mu_h^s \in \Pi_h^{\max}(s)} \min_{\nu_h^s \in D(\mathcal{B})} \underline{V}_h^{\mu_h^s, \nu_h^s}(s) - \epsilon_{\text{opt}}, \tag{3.10}$$

$$\max_{\mu_h^s \in D(\mathcal{A})} \overline{V}_h^{\mu_h^s, \overline{\nu}_h^s}(s) \leq \min_{\nu_h^s \in \Pi_h^{\min}(s)} \max_{\mu_h^s \in D(\mathcal{A})} \overline{V}_h^{\mu_h^s, \nu_h^s}(s) + \epsilon_{\text{opt}}. \tag{3.11}$$

In Appendix B.2.2 we show that projected gradient descent can output an ϵ_{opt}-minimizer with $(H + H\sqrt{\log(\mathcal{N}(\Pi))\iota})/\epsilon_{\text{opt}}^2$ iterations, where each iteration consists of a gradient computation and a projection onto the probability simplex. We note that if we set ϵ_{opt} to $\frac{1}{\sqrt{n}}$, then the optimization error is always of a smaller order term compared to the statistical error.

3.3.2 Sample Complexity Guarantees for SBMM

For the statistical guarantee, we will first provide *assumption-free bounds* in the sense that it holds for arbitrary compliant dataset [Jin et al., 2021d, Yin and Wang, 2021a]. We define the uncertainty at timestep h and state s under strategy μ_h^s and ν_h^s:

$$\widehat{b}_h(s, \mu_h^s, \nu_h^s) := 2b_h(s, \mu_h^s, \nu_h^s) + H \sum_{(a,b) \notin \mathcal{K}_h(s)} \mu_h^s(a)\nu_h^s(b)$$

.

Proposition 3.3.2. *Suppose π^{output} is the output of Algorithm 2. With probability $1-\delta$, we have*

$$\text{Gap}(\pi^{\text{output}}) \leq$$
$$\min_{\pi=(\mu,\nu)\in\Pi} \max_{\pi'=(\mu',\nu')\in\Pi^{\text{det}}} \left[\text{Gap}(\pi) + \mathbb{E}_{\mu,\nu'} \sum_{h=1}^{H} \widehat{b}_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) + \mathbb{E}_{\mu',\nu} \sum_{h=1}^{H} \widehat{b}_h(s_h, \mu_h'^{s_h}, \nu_h) \right] + 2H\epsilon_{\text{opt}}.$$

Proposition 3.3.2 shows that our algorithm can find the best strategy in Π with an additional error of the expected total uncertainty under some unilateral strategies and an extra optimization error term $2H\epsilon_{\text{opt}}$. Then we derive bounds with unilateral coefficients.

Theorem 3.3.3. *Suppose π^{output} is the output of Algorithm 2. With probability $1-\delta$, we have*

$$\text{Gap}(\pi^{\text{output}}) \leq \min_{\pi\in\Pi} \left[\text{Gap}(\pi) + 4H^2\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\pi)\iota/n} \right] + 2H\epsilon_{\text{opt}}.$$

Theorem 3.3.3 directly implies the following corollary.

Corollary 3.3.4. *If* $\Pi = \Pi^{\text{full}}$*, then with probability* $1-\delta$*, we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 S(A+B)\widehat{C}(\pi^*)/n}) + 2H\epsilon_{\text{opt}}$*. If* $\pi^* \in \Pi$*, then with probability* $1-\delta$*, we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 S \log(\mathcal{N}(\Pi))\widehat{C}(\pi^*)/n}) + 2H\epsilon_{\text{opt}}$.

Since $\widehat{C}(\pi)$ can be bounded using $C(\pi)$ (Proposition 3.2.5), we have the following theorem.

Theorem 3.3.5. *Suppose* π^{output} *is the output of Algorithm 2. With probability* $1-\delta$*, we have*

$$\text{Gap}(\pi^{\text{output}}) \le \min_{\pi\in\Pi}\left[\text{Gap}(\pi) + 4H^2\sqrt{S\log(\mathcal{N}(\Pi))C(\pi)\iota^2/n} + HS(A+B)C(\pi)/n\right] + 2H\epsilon_{\text{opt}}.$$

In addition, suppose $p_{\min} = \min_{s,a,b,h}\{d_h^\rho(s,a,b) : d_h^\rho(s,a,b) > 0\}$ *and if* $n \ge \frac{8\log(SABH/\delta)}{p_{\min}}$*, we have* $\text{Gap}(\pi) \le \min_{\pi\in\Pi}\left[\text{Gap}(\pi) + 8H^2\sqrt{S\log(\mathcal{N}(\Pi))C(\pi)\iota^2/n}\right] + 2H\epsilon_{\text{opt}}$.

Theorem 3.3.5 shows that there will be an additional lower order term $S(A+B)C(\pi)/n$, which can be interpreted as the rate of the empirical dataset distribution converges to the population distribution. In addition, for large enough $n \ge \frac{8\log(SABH/\delta)}{p_{\min}}$, there is no lower order term. Here $n \ge \frac{8\log(SABH/\delta)}{p_{\min}}$ serves as a warm-up cost so that the empirical support is the same as the true support of d_h. A similar analysis is used in Yin and Wang [2021a]. With a refined analysis, we can show that there is no lower order term for the standard settings $\Pi = \Pi^{\text{full}}$ in two-player zero-sum Markov games and $\Pi = \Pi^{\text{det}}$ for turn-based Markov games. Note that turn-based Markov games always have a deterministic NE.

Corollary 3.3.6. *If* $\Pi = \Pi^{\text{full}}$*, then with probability* $1-\delta$*, we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 S(A+B)C(\pi^*)/n}) + 2H\epsilon_{\text{opt}}$*. In addition, for turn-based two-player zero-sum Markov games, we can set* $\Pi = \Pi^{\text{det}}$ *and we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 SC(\pi^*)/n}) + 2H\epsilon_{\text{opt}}$.

Corollary 3.3.6 improves the AB dependence in the previous zero-sum Markov games result [Cui and Du, 2022a] and matches the result for turn-based Markov games [Cui and Du, 2022a] up to an extra $\sqrt{H}$ factor. The additional H factor is due to the Hoeffding-type bonus and we believe it can be removed with a more sophisticated Bernstein-type bonus.

3.4 Algorithms and Analyses for Multi-player General-sum Markov Game

In this section, we propose the first provably efficient algorithm for offline multi-player general-sum Markov game. We will use the strategy-wise bonus to achieve a sample complexity that does not scale with $\prod_{j\in[m]} A_j$. However, in general-sum games there is no saddle point structure, so we can no longer use the maximin-optimization-type algorithm. Instead, our algorithm utilizes a novel *surrogate function* to approximately minimize the performance gap.

Given a dataset $\mathcal{D} = \{(s_h^k, \mathbf{a}_h^k, \mathbf{r}_h^k, s_{h+1}^k)\}_{k,h=1}^{n,H}$, we denote $n_h(s,\mathbf{a}) = \sum_{k=1}^n \mathbf{1}\left((s_h^k, \mathbf{a}_h^k) = (s,\mathbf{a})\right)$ and $\mathcal{K}_h(s) = \{\mathbf{a} : n_h(s,\mathbf{a}) \neq 0\}$. If $n_h(s,\mathbf{a}) > 0$, we set

$$\widehat{r}_{h,j}(s,\mathbf{a}) = \frac{\sum_{k=1}^n r_{h,j}^k \mathbf{1}\left((s_h^k, \mathbf{a}_h^k) = (s,\mathbf{a})\right)}{n_h(s,\mathbf{a})}, \widehat{P}_h(s'|s,\mathbf{a}) = \frac{\sum_{k=1}^n \mathbf{1}\left((s_h^k, \mathbf{a}_h^k, s_{h+1}^k) = (s,\mathbf{a},s')\right)}{n_h(s,\mathbf{a})}, \tag{3.12}$$

otherwise we have $\widehat{r}_{h,j}(s,\mathbf{a}) = 0, \widehat{P}_h(s'|s,\mathbf{a}) = 0$.

Based on this empirical multi-player Markov game, we can estimate the value of arbitrary strategy π via policy evaluation (Algorithm 17 in Appendix). We describe Algorithm 17 for the pessimistic estimate. For a player j, strategy π and timestep h, we first compute the state-action value estimates:

$$\underline{Q}_{h,j}^{\pi}(s,\mathbf{a}) = \widehat{r}_{h,j}(s,\mathbf{a}) + \left\langle \widehat{P}_h(s,\mathbf{a}), \underline{V}_{h+1,j}^{\pi} \right\rangle, \tag{3.13}$$

Then we estimate the state value functions and add the strategy-wise bonus to ensure pessimism.

$$\underline{V}_{h,j}^{\pi}(s) = \mathrm{proj}_{[0,H-h+1]}\left\{\mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\underline{Q}_{h,j}^{\pi}(s,\mathbf{a}) - b_h(s,\pi_h^s)\right\}, \tag{3.14}$$

$$\text{where } b_h(s,\pi_h^s) = H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\prod_{j\in[m]}\pi_{h,j}^s(a_j)^2}{n_h(s,\mathbf{a})} S\log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n, \tag{3.15}$$

with $\iota = 32\log(16\prod_{j\in[m]} A_j mSHn/\delta)$. Here the strategy-wise pessimism can remove the $\prod_{j\in[m]} A_j$ dependence as explained in the previous section. By dynamic programming from timestep H to timestep 1 we can obtain the pessimistic estimate $\underline{V}_{1,j}^{\pi}(s_1)$. Compared with the bonus function (3.6) in zero-sum Markov game, there is an extra S factor in (3.15) because here we need to perform concentration on $\left\langle \widehat{P}_h(s,\mathbf{a}), \underline{V}_{h+1,j}^{\pi} \right\rangle$ for all π while in

(3.4) we only need to analyze $\left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1} \right\rangle$ for a single $\underline{V}_{h+1}$. We use an additional ϵ-covering on $\mathbb{R}^S$ which leads to the extra S.

We use Algorithm 18 (in Appendix) to compute the optimistic value of the best response strategy. For a given player j, strategy π_{-j} used by all the other player and timestep h, we first compute the optimistic state-action value estimate:

$$\overline{Q}_{h,j}^{*,\pi_{-j}}(s,\mathbf{a}) = \widehat{r}_{h,j}(s,\mathbf{a}) + \left\langle \widehat{P}_h(s,\mathbf{a}), \overline{V}_{h+1,j}^{*,\pi_{-j}} \right\rangle + H\mathbf{1}\{\mathbf{a} \notin \mathcal{K}_h(s)\}. \tag{3.16}$$

Then we compute the optimistic value for deterministic strategies for player j:

$$\overline{V}_{h,j}(s,a_j) = \mathbb{E}_{\mathbf{a}_{-j}\sim\pi_{h,-j}(\cdot|s)} \overline{Q}_{h,j}^{*,\pi_{-j}}(s,a_j,\mathbf{a}_{-j}) + b_h(s,a_j,\pi_{h,-j}^s). \tag{3.17}$$

Here with a slight abuse of the notation, we use a_j to denote the deterministic strategy of player j that chooses action a_j at state s and timestep h. Finally we use the maximum over all the deterministic strategies to be the best response value function:$\overline{V}_{h,j}^{*,\pi_{-j}}(s) = \mathrm{proj}_{[0,H-h+1]}\left\{\max_{a_j\in\mathcal{A}_j} \overline{V}_{h,j}(s,a_j)\right\}$.

By dynamic programming we can obtain the optimistic estimate $\overline{V}_{1,j}^{*,\pi_{-j}}(s_1)$ at the initial state. Note that we only consider the deterministic strategies for player j. Thanks to the convexity of the bonus $b_h(s,\pi_h^s)$, the best response with respect to $\overline{V}_{h,j}^{\pi}(s)$ is also in the deterministic strategy class as in zero-sum Markov games. The following proposition connects Algorithm 17 and Algorithm 18:

Proposition 3.4.1. *For any strategy $\pi_{-j} \in \Pi_{-j}^{\mathrm{full}}$,$h \in [H]$ and $s \in \mathcal{S}$, we have $\overline{V}_{h,j}^{*,\pi_{-j}}(s) = \max_{\pi_j} \overline{V}_{h,j}^{\pi_j,\pi_{-j}}(s)$.*

Based on Algorithm 17 and Algorithm 18, we propose a surrogate minimization algorithm for multi-player general-sum Markov game. Suppose $\underline{V}_{1,j}^{\pi}(s_1)$ and $\overline{V}_{1,j}^{*,\pi_{-j}}(s_1)$ are pessimistic and optimistic estimates, then we have

$$\mathrm{Gap}(\pi) = \sum_{j\in[m]} V_{1,j}^{*,\pi_{-j}}(s_1) - V_{1,j}^{\pi}(s_1) \le \sum_{j\in[m]} \overline{V}_{1,j}^{*,\pi_{-j}}(s_1) - \underline{V}_{1,j}^{\pi}(s_1).$$

The RHS can serve as the surrogate function and SBSM (Algorithm 19 in Appendix) outputs the minimizer of it in Π. From the computational perspective, Algorithm 17 and Algorithm 18 are both efficient while Algorithm 19 needs to enumerate Π for the worst

case. This computational hardness agrees with the PPAD-hardness for computing approximate NE even in full information general-sum game [Daskalakis, 2013]. However, if Π is well structured, Algorithm 19 may be computationally efficient and we leave it to future work. Here we assume π^{output} is an exact solution while it is straightforward to incorporate optimization error as in the previous section.

3.4.1 Sample Complexity Guarantees for SBSM

We still begin with assumption-free bound as in the previous section. We define the uncertainty at timestep h and state s under strategy π: $\widehat{b}_h(s, \pi_h^s) = 2b_h(s, \pi_h^s) + H \sum_{\mathbf{a} \notin \mathcal{K}_h(s)} \pi_h^s(\mathbf{a})$.

Proposition 3.4.2. *Suppose π^{output} is the output of Algorithm 19. With probability $1-\delta$, we have*

$$\text{Gap}(\pi^{\text{output}}) \le \min_{\pi \in \Pi} \left[\text{Gap}(\pi) + \max_{\pi' \in \Pi^{\text{det}}} \sum_{j \in [m]} \mathbb{E}_{\pi'_j, \pi^*_{-j}} \sum_{h=1}^{H} \widehat{b}_h(s_h, \pi'^{s_h}_{h,j}, \pi^{s_h}_{h,-j}) + m\mathbb{E}_{\pi} \sum_{h=1}^{H} \widehat{b}_h(s_h, \pi_h^{s_h}) \right].$$

Proposition 3.4.2 has a similar structure as Proposition 3.3.2 with a slight difference in the expected uncertainty terms. Then we will bound using the unilateral coefficients.

Theorem 3.4.3. *Suppose π^{output} is the output of Algorithm 19. With probability $1-\delta$, we have*

$$\text{Gap}(\pi^{\text{output}}) \le \min_{\pi \in \Pi} \left[\text{Gap}(\pi) + 4mH^2 S \sqrt{\widehat{C}(\pi) \log(\mathcal{N}(\Pi))\iota / n} \right].$$

Theorem 3.4.3 directly implies the following corollary, which shows that the sample complexity of offline multi-agent RL only scales linearly with respect to the number of the players.

Corollary 3.4.4. *If $\Pi = \Pi^{\text{full}}$, with probability $1-\delta$, we have $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 S^2 \sum_{j \in [m]} A_j \widehat{C}(\pi^*)/n})$. If $\pi^* \in \Pi$, then with probability $1-\delta$, we have $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4 S^2 \log(\mathcal{N}(\Pi)) \widehat{C}(\pi^*)/n})$.*

Similarly we have the following theorem and corollary for the population unilateral coefficient.

Theorem 3.4.5. *Suppose π^{output} is the output of Algorithm 19. If $n \ge \frac{8 \log(S\Pi_{j \in [m]} A_j H/\delta)}{p_{\min}}$, with probability $1-\delta$, we have $\text{Gap}(\pi^{\text{output}}) \le \min_{\pi \in \Pi} \left[\text{Gap}(\pi) + 4mH^2 S \sqrt{2C(\pi) \log(\mathcal{N}(\Pi))\iota / n} \right]$.*

Corollary 3.4.6. *Suppose* $n \geq \frac{8\log(S\Pi_{j\in[m]}A_jH/\delta)}{p_{\min}}$. *If* $\Pi = \Pi^{\text{full}}$, *with probability* $1-\delta$, *we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4S^2\sum_{j\in[m]}A_jC(\pi^*)/n})$. *If* $\pi^* \in \Pi$, *then with probability* $1-\delta$, *we have* $\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4S^2\log(\mathcal{N}(\Pi))C(\pi^*)/n})$.

3.5 Conclusion

In this work, we studied offline MARL. With a novel strategy-wise bonus, we remove the exponential dependence on the number of players. We use different algorithm frameworks for zero-sum Markov games and general-sum Markov games due to their different properties.

Here we list several open problems for future work. One direction is to find the minimax sample complexity for offline Markov games, i.e., if the $\log(\mathcal{N}(\Pi))$ term is necessary. Another direction is to design computationally efficient algorithms for finding (coarse) correlated equilibrium in general-sum Markov games. Lastly, we only focus on the tabular setting serving as a start point. It is important to study MARL with reasonable function approximation.

Part II

PROVABLY EFFICIENT ONLINE MULTI-AGENT REINFORCEMENT LEARNING

Chapter 4

ONLINE MARKOV GAMES WITH INDEPENDENT LINEAR FUNCTION APPROXIMATION

This chapter is based on Cui et al. [2023], with Kaiqing Zhang and Simon S. Du.

4.1 Introduction

Decision-making under uncertainty in a multi-agent system has shown its potential to approach artificial intelligence, with superhuman performance in Go games [Silver et al., 2017], Poker [Brown and Sandholm, 2019], and real-time strategy games [Vinyals et al., 2019b], etc. All these successes can be generally viewed as examples of multi-agent reinforcement learning (MARL), a generalization of single-agent reinforcement learning (RL) [Sutton and Barto, 2018] where multiple RL agents interact and make sequential decisions in a common environment [Zhang et al., 2021a]. Despite the impressive empirical achievements of MARL, the theoretical understanding of MARL is still far from complete due to the complex interactions among agents.

One of the most prominent challenges in RL is the curse of *large state-action* spaces. In real-world applications, the number of states and actions is exponentially large so that the *tabular* RL algorithms are not applicable. For example, there are 3^{361} potential states in Go games, and it is impossible to enumerate all of them. In single-agent RL, plenty of works attempt to tackle this issue via function approximation so that the sample complexity only depends on the complexity of the function class, thus successfully breaking the curse of large state-action spaces [Wen and Van Roy, 2017, Jiang et al., 2017, Yang and Wang, 2020, Du et al., 2019, Jin et al., 2020, Weisz et al., 2021, Wang et al., 2020, Zanette et al., 2020, Wang et al., 2021, Jin et al., 2021a, Du et al., 2021, Foster et al., 2021].

However, it is still unclear what is the proper function approximation model for multi-agent RL. The existing theoretical analyses in MARL exclusively focus on a *global* function approximation paradigm, i.e., a function class capturing the state-joint-action value

$Q_i(s, a_1, \cdots, a_m)$ where s is the state and a_i is the action of player $i \in [m]$ [Xie et al., 2020a, Huang et al., 2021, Chen et al., 2021b, Jin et al., 2022, Chen et al., 2022, Ni et al., 2022]. Unfortunately, these algorithms would *suffer from the curse of multiagents* when specialized to tabular Markov games, one of the most canonical models in MARL. Specifically, the sample complexity depends on the number of joint actions $\prod_{i\in[m]} A_i$, where A_i is the number of actions for player i, which is exponentially worse than the best algorithms specified to the tabular Markov game whose sample complexity only depends on $\max_{i\in[m]} A_i$ [Jin et al., 2021b, Song et al., 2021a, Mao et al., 2022, Daskalakis et al., 2022].

On the other hand, empirical algorithms with *independent* function approximation such as Independent PPO have surprisingly good performance, where only the independent state-individual-action value function $Q_i(s, a_i)$ is modeled [de Witt et al., 2020a, Yu et al., 2021]. This is very surprising due to the fact that the independent state-action value function $Q_i(s, a_i)$ does not reflect the change of other players' policies, a.k.a. the non-stationarity from multiple agents, which should fail to allow learning at first glance. In addition, single-agent RL with function approximation already suffers from the non-stationarity of applying function approximation [Baird, 1995], making it even harder for MARL. This gap between theoretical and empirical research leads to the following question:

Can we design provably efficient MARL algorithms for Markov games with independent function approximation that can break the curse of multiagents?

In this paper, we provide an affirmative answer to the above question. We highlight our contributions and technical novelties below.

4.1.1 Main Contributions and Technical Novelties

1. Multi-player general-sum Markov games with independent linear function approximation. We propose independent linear Markov games, which is the first provably efficient model in MARL that allows each agent to have its own independent function approximation. We show that independent linear Markov games capture several important instances, namely tabular Markov games [Shapley, 1953], linear Markov decision processes (MDP) [Jin et al., 2020], and congestion games [Rosenthal, 1973]. Then we provide the

first provably efficient algorithm in MARL that breaks the curse of multiagents and the curse of large state and action spaces at the same time, i.e., the sample complexity only has polynomial dependence on the complexity of the independent function class complexity. See Table 4.1 for comparisons between our work and prior works.

Our algorithm design relies on two high-level technical ideas which we detail here:

- **Policy replay to tackle non-stationarity.** Different from experience replay that incrementally adds new on-policy data to a dataset, *policy replay* maintains a policy set and completely renews the dataset at each episode by collecting fresh data using the policy set. We propose a new policy replay mechanism for learning equilibria in independent linear Markov games, which allows efficient exploration while adapting to the non-stationarity induced by both multiple agents and function approximation at the same time.

- **Separating exploration and learning Markov equilibria.** States and actions in independent linear Markov games are correlated through the feature map, so we can no longer resort to adversarial bandit oracles as in algorithms for tabular Markov games [Jin et al., 2021b, Song et al., 2021a, Mao et al., 2022, Daskalakis et al., 2022]. In particular, the adversarial contextual linear bandit oracles would be a potential substitute, while the existence of such oracles remains largely an open problem (see Section 29.4 in Lattimore and Szepesvári [2020]). To tackle this issue, we exploit the fact that under the self-play setting, other players are not adversarial but *under control*, so we can sample multiple i.i.d. feedback to derive an accurate estimate instead of just a single bandit feedback. We separate the exploration in Markov games from learning equilibria so that any no-regret algorithms with *full-information feedback* are sufficient for our MARL algorithm, which is significantly weaker than the adversarial bandit oracle used in all the previous works that break the curse of multiagents in the tabular setting.

2. Learning Nash equilibria in Linear Markov potential games. We provide an algorithm to learn Markov Nash equilibria (NE) when the underlying independent linear

Markov game is also a Markov potential game. The algorithm is based on the reduction from learning NE in independent linear Markov potential games to learning the optimal policy in linear MDPs. In addition, the result directly implies a provable efficient decentralized algorithm for learning NE in congestion games, which has better sample complexity compared with the previous state-of-the-art result in Cui and Du [2022b].

3. Improved sample complexity for tabular multi-player general-sum Markov games. Aside from our contributions to Markov games with function approximation, we design an algorithm for tabular Markov games with improved sample complexity for learning Markov CCE by adapting the policy replay mechanism we proposed for the independent linear Markov games. Our sample complexity for learning Markov CCE is $\widetilde{O}(H^6S^2A_{\max}\epsilon^{-2})$, which significantly improves the prior state-of-the-art result $\widetilde{O}(H^{11}S^3A_{\max}\epsilon^{-3})$ in Daskalakis et al. [2022], where H is the time horizon, S is the number of the states, $A_{\max} = \max_{i\in[m]} A_i$ is the maximum action space and ϵ is the desired accuracy.* Furthermore, our analysis is simpler. In addition, we provide the first provably efficient algorithm for learning Markov CE with sample complexity $\widetilde{O}(H^6S^2A_{\max}^2\epsilon^{-2})$.

4.1.2 Related Work

Tabular Markov games. Markov games, also known as stochastic games, are introduced in the seminal work Shapley [1953]. We first discuss works that consider bandit feedback as in our paper. Bai and Jin [2020a] provide the first provably sample-efficient MARL algorithm for two-player zero-sum Markov games, which is later improved in Bai et al. [2020]. For multi-player general-sum Markov games, Liu et al. [2021a] provide the first provably efficient algorithm with sample complexity depending on the size of joint action space $\prod_{i\in[m]} A_i$. Jin et al. [2021b], Song et al. [2021a], Mao et al. [2022] utilize a decentralized algorithm to break the curse of multiagents. However, the output policy therein is non-Markov. Recently, Daskalakis et al. [2022] provide the first algorithm that can learn Markov CCE and break the curse of multiagents at the same time. Several other lines of research consider full-

*We use $\widetilde{O}(\cdot)$ to omit logarithmic dependence on all the parameters.

information feedback setting in Markov games and have attempted to prove convergence to NE/CE/CCE and/or sublinear individual regret [Sayin et al., 2021, Zhang et al., 2022b, Cen et al., 2022b, Yang and Ma, 2022, Erez et al., 2022, Ding et al., 2022], and offline learning setting where a dataset is given and no further interaction with the environment is permitted [Cui and Du, 2022a, Zhong et al., 2022, Yan et al., 2022, Xiong et al., 2022, Cui and Du, 2022b].

Markov games with function approximation. To tackle the curse of large state and action spaces, it is natural to incorporate existing function approximation frameworks for single-agent RL into MARL algorithms. Xie et al. [2020a], Chen et al. [2021b] consider linear function approximation in two-player zero-sum Markov games, which originate from linear MDP and linear mixture MDP in single-agent RL, respectively [Jin et al., 2020, Yang and Wang, 2020]. Huang et al. [2021], Jin et al. [2022], Chen et al. [2022], Ni et al. [2022] consider different kinds of general function approximation, which also originate from single-agent RL literature [Jiang et al., 2017, Du et al., 2019, Agarwal et al., 2020c, Wang et al., 2020, Zanette et al., 2020, Jin et al., 2021a, Foster et al., 2021, Du et al., 2021]. It is notable that all of these frameworks are based on *global* function approximation, which is centralized and suffers from the curse of multiagents when applied to tabular Markov games.

Markov potential games. Markov potential games incorporate Markovian state transition to potential games [Monderer and Shapley, 1996]. Most existing results consider full-information feedback or well-explored setting and prove fast convergence of policy gradient methods to NE [Leonardos et al., 2021, Zhang et al., 2021c, Ding et al., 2022]. Song et al. [2021a] provide a best-response type algorithm that can *explore* in tabular Markov potential games. One important class of potential games is congestion games [Rosenthal, 1973]. Cui et al. [2022] give the first non-asymptotic analysis for general congestion games with bandit feedback. We refer the readers to Cui et al. [2022] for a more detailed background about learning in potential/congestion games. It is worth noting that for congestion games, each player is in a combinatorial bandit if other players' policies are fixed, which can be directly handled by our independent linear Markov games model, while applying po-

tential game results lead to polynomial dependence on $A_{\max}$, which could be exponentially large in the number of facilities in congestion games.

Comparison with Wang et al. [2023]. Shortly after we submitted our work to arXiv, we became aware of a concurrent and independent work Wang et al. [2023]. The two works share quite a bit of results, e.g., the use of a similar function approximation model, similar algorithm design and sample complexity results for learning Markov CCE in tabular Markov games, similar discussions on the improved result by using additional communication among agents, etc. Here we highlight several differences in learning Markov CCE with linear function approximation. First, they utilize a novel second-order regret oracle and Bernstein-type concentration bounds, so that they can leverage the *single-sample* estimate instead of the *batched* estimate in our algorithm, which results in better dependence on $d_{\max}$, ϵ and H compared with our sample complexity. On the other hand, our result has no dependence on the number of actions, which is aligned with the single-agent linear MDP sample complexity, while theirs has a polynomial dependence on $A_{\max}$.[†] This difference is because they use a uniform policy to sample at the last step while we always use the on-policy samples. In fact, neither of the sample complexity bounds is strictly better than the other one and is not directly comparable as the assumptions are not the same. Second, our algorithm can use arbitrary full-information no-regret learning oracles while their results are specialized to the Expected Follow-the-Perturbed-Leader (E-FTPL) oracle [Hazan and Minasyan, 2020], which makes the policy class Π^{estimate} therein the linear argmax policy class. Our Π^{estimate} is induced by the full-information oracle being used, and the result is in this sense more agnostic. On the other hand, if we use E-FTPL, the induced Π^{estimate} has a more complicated form than the linear argmax policy class. This is because we use the optimistic estimation of the Q function in our algorithm. Third, our algorithm can work with agnostic model misspecification which is not considered in Wang et al. [2023]. Besides the differences in linear function approximation results mentioned above and the similar algorithms and sample complexity for the tabular case, we also have results for learning NE

[†]In Theorem 4.4.3, there is a $\log(A_{\max})$ factor, which can be replaced by $d_{\max}$ by using a covering argument as in adversarial linear bandits [Bubeck et al., 2012].

in Markov potential games, as well as learning Markov CE in general-sum Markov games, while they provide a policy mirror-descent-type algorithm for other function approximation settings, such as linear quadratic games and the settings with low Eluder dimension, with a weaker version of CCE called policy-class-restricted CCE.

Notation. For a finite set X, we use $\Delta(X)$ to denote the space of distributions over X. For $n \in \mathbb{N}^+$, we use $[n]$ to denote $\{1, 2, \cdots, n\}$. We use $\|\cdot\|$ to denote the Euclidean norm $\|\cdot\|_2$ and $\langle \cdot, \cdot \rangle$ to denote the Euclidean inner product. We define $\text{proj}_{[a,b]}(x) := \min\{\max\{x, a\}, b\}$ and $x \vee y := \max\{x, y\}$. An arbitrary tie-breaking rule can be used for determining $\text{argmax}_x f(x)$.

4.2 Preliminaries

Multi-player general-sum Markov games are defined by the tuple $(\mathcal{S}, \{A_i\}_{i=1}^m, H, \mathbb{P}, \{r_i\}_{i=1}^m)$, where $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$, m is the number of the players, $\mathcal{A}_i$ is the action space for player i with $|\mathcal{A}_i| = A_i$, H is the length of the horizon, $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is the collection of the transition kernels such that $\mathbb{P}_h(\cdot \mid s, \mathbf{a})$ gives the distribution of the next state given the current state s and joint action $\mathbf{a} = (a_1, a_2, \cdots, a_m)$ at step h, and $r_i = \{r_{h,i}\}_{h\in[H]}$ is the collection of random reward functions for each player such that $r_{h,i}(s, \mathbf{a}) \in [0, 1]$ is the random reward with mean $R_{h,i}(s, \mathbf{a})$ for player i given the current state s and the joint action $\mathbf{a}$ at step h. We use $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_m$ to denote the joint action space, $\mathbf{r}_h = (r_{h,1}, r_{h,2}, \cdots, r_{h,m})$ to denote the joint reward profile at step h, and $A_{\max} = \max_{i\in[m]} A_i$. In the rest of the paper, we will simplify "multi-player general-sum Markov games" to "Markov games" when it is clear from the context.

Markov games will start at a fixed initial state s_1 for each episode.[‡] At each step $h \in [H]$, each player i will observe the current state s_h and choose some action $a_{h,i}$ simultaneously, and receive their own reward realization $\widetilde{r}_{h,i} \sim r_{h,i}(s_h, \mathbf{a}_h)$ where $\mathbf{a}_h = (a_{h,1}, a_{h,2}, \cdots, a_{h,m})$. Then the state will transition according to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, \mathbf{a}_h)$. The game will terminate when state s_{H+1} is reached and the goal of each player is to maximize their own expected total reward $\mathbb{E}\left[\sum_{h=1}^H \widetilde{r}_{h,i}\right]$. We consider the bandit-feedback setting where only the reward

[‡]It is straightforward to generalize to stochastic initial state $s_1 \sim p_1(\cdot)$ by adding a dummy state s_0 instead, which will transition to $s_1 \sim p_1(\cdot)$ no matter what action is chosen.

for the chosen action is revealed, and there is no simulator and thus exploration is necessary.

Policy. A Markov joint policy is denoted by $\pi = \{\pi_h\}_{h=1}^H$ where each $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$ is the joint policy at step h. We say that a Markov joint policy is a Markov product policy if there are policies $\{\pi_i\}_{i=1}^m$ such that $\pi_h(\mathbf{a} \mid s) = \prod_{i=1}^m \pi_{h,i}(a_i \mid s)$ for each $h \in [H]$, where $\pi_i = \{\pi_{h,i}\}_{h=1}^H$ is the collection of Markov policies $\pi_{h,i} : \mathcal{S} \to \Delta(\mathcal{A}_i)$ for player i. In other words, a Markov product policy means that the policies of each player are not correlated. For a Markov joint policy π, we use π_{-i} to denote the Markov joint policy for all the players except player i. We will simplify the terminology by using "policy" instead of "Markov joint policy" when it is clear from the context as we will only focus on Markov policies.

Value function. For a policy π, it can induce a random trajectory $(s_1, \mathbf{a}_1, \mathbf{r}_1, s_2, \cdots, s_H, \mathbf{a}_H, \mathbf{r}_H, s_{H+1})$ such that $\mathbf{a}_h \sim \pi_h(\cdot \mid s_h)$, $\mathbf{r}_h \sim \mathbf{r}_h(s_h, \mathbf{a}_h)$, and $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, \mathbf{a}_h)$ for all $h \in [H]$. For simplicity, we will denote $\mathbb{E}_\pi[\cdot] = \mathbb{E}_{(s_1, \mathbf{a}_1, \mathbf{r}_1, s_2, \cdots, s_H, \mathbf{a}_H, \mathbf{r}_H, s_{H+1}) \sim \pi}[\cdot]$. We define the state value function under policy π for each player $i \in [m]$ to be

$$V_{h,i}^\pi(s_h) := \mathbb{E}_\pi \left[\sum_{t=h}^H r_{t,i}(s_t, \mathbf{a}_t) \;\middle|\; s_h \right], \forall s_h \in \mathcal{S},$$

which is the expected total reward for player i if all the players are following policy π starting from state s_h at step h.

Best response and strategy modification. Suppose all the players except player i are playing according to a fixed policy π_{-i}, then the best response of player i is the policy that can achieve the highest total reward for player i. Concretely, π_i is the best response to π_{-i} if $\pi_i = \operatorname{argmax}_{\pi_i' \in \Pi_i} V_{1,i}^{\pi_i', \pi_{-i}}(s_1)$, where Π_i consists of all the possible policies for player i. We will use $V_{h,i}^{\dagger, \pi_{-i}}(s)$ to denote the best-response value $\max_{\pi_i' \in \Pi_i} V_{h,i}^{\pi_i', \pi_{-i}}(s)$ for all $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ and $\mathbb{E}_{\dagger, \pi_{-i}}[\cdot]$ to be the expectation over the corresponding best-response policy. Note that if all the other players are playing a fixed policy, then player i is in an MDP and the best response is the corresponding optimal policy, which can always be deterministic and achieve the optimal value $\max_{\pi_i' \in \Pi_i} V_{h,i}^{\pi_i', \pi_{-i}}(s)$ for all $h \in [H]$ and $s \in \mathcal{S}$ simultaneously.

A strategy modification $\psi_i = \{\psi_{h,i}\}_{h=1}^H$ for player i is a collection of maps $\psi_{h,i} : \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i$, which will map the action chosen at any state to another action.[§] For a Markov joint policy π, we use $\psi_i \diamond \pi$ to denote the modified Markov joint policy such that

$$(\psi_i \diamond \pi)_h(\mathbf{a} \mid s) = \sum_{\mathbf{a}' : \psi_{h,i}(a_i' | s) = a_i, \mathbf{a}'_{-i} = \mathbf{a}_{-i}} \pi_h(\mathbf{a}' \mid s).$$

In words, if the policy π_h assigns action a_i to player i at state s, it will be modified to action $\psi_{h,i}(a_i \mid s)$. We use Ψ_i to denote all the possible strategy modifications for player i. As Ψ_i contains all the constant modifications, we have

$$\max_{\psi_i \in \Psi_i} V_{1,i}^{\psi_i \diamond \pi}(s_1) \geq \max_{\pi_i'} V_{1,i}^{\pi_i', \pi_{-i}}(s_1) = V_{1,i}^{\dagger, \pi_{-i}}(s_1),$$

which means that strategy modification is always stronger than the best response.

Notions of equilibria. A Markov Nash equilibrium is a Markov product policy where no player can increase their total reward by changing their own policy.

Definition 4.2.1. (Markov Nash equilibrium) A Markov product policy π is an ϵ-approximate Nash equilibrium if

$$\text{NashGap}(\pi) := \max_{i \in [m]} \left(V_{1,i}^{\dagger, \pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1) \right) \leq \epsilon.$$

In general, it is intractable to compute Nash equilibrium even in normal-form general-sum games, which are Markov games with $H = 1$ and $S = 1$ [Daskalakis et al., 2009, Chen et al., 2009]. In this paper, we will focus on the following two relaxed equilibrium notions, which allow computationally efficient learning.

Definition 4.2.2. (Markov Coarse Correlated Equilibrium) A Markov joint policy π is a Markov coarse correlated equilibrium if

$$\text{CCEGap}(\pi) := \max_{i \in [m]} \left(V_{1,i}^{\dagger, \pi_{-i}}(s_1) - V_{1,i}^{\pi}(s_1) \right) \leq \epsilon.$$

[§]We only consider deterministic strategy modification as it is known that the optimal strategy modification can always be deterministic [Jin et al., 2021b].

Definition 4.2.3. (Markov Correlated Equilibrium) A Markov joint policy π is a Markov correlated equilibrium if

$$\text{CEGap}(\pi) := \max_{i\in[m]} \left(\max_{\psi_i \in \Psi_i} V_{1,i}^{\psi_i \diamond \pi}(s_1) - V_{1,i}^{\pi}(s_1) \right) \leq \epsilon.$$

It is known that every Markov NE is a Markov CE and every Markov CE is a Markov CCE, and in two-player zero-sum Markov games, these three notions are equivalent. In this work, we will focus on Markov equilibria, which are more refined compared with non-Markov equilibria considered in Jin et al. [2021b], Song et al. [2021a], Mao et al. [2022]. For a detailed discussion regarding the difference, we refer the readers to Daskalakis et al. [2022].

Two important special cases of Markov games are two-player zero-sum Markov games and Markov potential games, which have computationally efficient algorithms for learning Markov NE. Two-player zero-sum Markov games are Markov games with the number of players $m = 2$ and reward function satisfying $r_{h,1}(s, \mathbf{a}) + r_{h,2}(s, \mathbf{a}) = 0$ for all $(s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$. Markov potential games are Markov games with a potential function $\Phi : \Pi \to [0, \Phi_{\max}]$, where Π is the set of all possible Markov product policies $\pi_1 \times \pi_2 \cdots \times \pi_m$, such that for any player $i \in [m]$, two policies π_i, π_i' of player i and policy π_{-i} for the other players, we have

$$V_{1,i}^{\pi_i, \pi_{-i}}(s_1) - V_{1,i}^{\pi_i', \pi_{-i}}(s_1) = \Phi(\pi_i, \pi_{-i}) - \Phi(\pi_i', \pi_{-i}).$$

Immediately, we have $\Phi_{\max} \leq mH$ by varying π_i for each player i for one time. One special case of Markov potential games is Markov cooperative games, where all the players share the same reward function.

4.3 MARL with Independent Linear Function Approximation

In this section, we will introduce the independent linear Markov game model and demonstrate the advantage of this model over existing Markov games with function approximation. Intuitively, independent linear Markov games assume that if other players are following some fixed Markov product policies, then player i is approximately in a linear MDP [Jin et al., 2020]. This is fundamentally different from previous global function approximation formu-

lations, which basically assume that the Markov game is a big linear MDP where the action is the joint action $\mathbf{a} = (a_1, a_2, \cdots, a_m)$.

Feature and independent linear function class. For each player i, they have access to their own feature map $\phi_i : \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}^{d_i}$ and we assume that

$$\sup_{(s,a_i)\in\mathcal{S}\times\mathcal{A}_i} \|\phi_i(s, a_i)\|_2 \leq 1.$$

For player i, given parameters $\theta_i = (\theta_{1,i}, \cdots, \theta_{H,i})$, the corresponding linear state-action value function for player i would be $f_i^\theta = (f_{1,i}^{\theta_{1,i}}, f_{2,i}^{\theta_{2,i}}, \cdots, f_{H,i}^{\theta_{H,i}})$ where $f_{h,i}^{\theta_{h,i}}(s, a_i) = \langle \phi_i(s, a_i), \theta_{h,i} \rangle$ for all $(s, a_i) \in \mathcal{S} \times \mathcal{A}_i$. We consider the following linear state-action value function class for player i:

$$\mathcal{Q}_i^{\text{lin}} = \left\{ f_i^{\theta_i} \mid \|\theta_{h,i}\|_2 \leq H\sqrt{d}, \forall h \in [H] \right\}.$$

We also define the state value function class

$$\mathcal{V} = \{(V_1, \cdots, V_{H+1}) \mid V_h(s) \in [0, H + 1 - h], \forall h \in [H + 1], s \in \mathcal{S}\}.$$

Given the state value function $V \in \mathcal{V}$ and other players' policies π_{-i}, we can define the independent state-action value function for all $h \in [H]$ and $(s_h, a_{h,i}) \in \mathcal{S} \times \mathcal{A}_i$ as:

$$Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) = \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)} \left[r_{h,i}(s_h, a_{h,i}, a_{h,-i}) + V_{h+1}(s_{h+1}) \right].$$

Now we formally define Markov games with independent linear function approximation. This definition generalizes the misspecified MDPs with linear function approximation model proposed in Zanette and Wainwright [2022] to the Markov games setting.

Definition 4.3.1. For any player i, feature map ϕ_i is ν-misspecified with policy set Π^{estimate} if for any rollout policy $\overline{\pi}$, target policy $\widetilde{\pi}$, we have for any $V \in \mathcal{V}$,

$$\max_{\pi\in\Pi^{\text{estimate}}} \left| \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[\text{proj}_{[0,H+1-h]} \left(\left\langle \phi_i(s_h, a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V} \right\rangle \right) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \right] \right| \leq \nu,$$

where Π^{estimate} is the collection of Markov product policies that need to be evaluated and

$$\theta_h^{\overline{\pi},\pi_{-i},V} = \underset{\|\theta\|\leq H\sqrt{d}}{\text{argmin}} \, \mathbb{E}_{\overline{\pi}} \left(\langle \phi_i(s_h, a_{h,i}), \theta \rangle - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \right)^2 \tag{4.1}$$

is the parameter for the best linear function fit to $Q_{h,i}^{\pi_{-i},V}$ under rollout policy $\overline{\pi}$. We say a multi-player general-sum Markov game with features $\{\phi_i\}_{i\in[m]}$ is a ν-misspecified linear Markov game with Π^{estimate} if for any player i, the feature map ϕ_i is ν-misspecified with Π^{estimate}. In addition, we define $d_{\max} := \max_{i\in[m]} d_i$ as the complexity measure of the linear Markov game.

The policy estimation set Π^{estimate} consists of policies that need to be estimated in the algorithm, which reflects the inductive bias of the algorithm. We emphasize that all of our algorithms do not require any knowledge of the policy estimation set Π^{estimate} or the misspecification error ν, which is known as the *agnostic setting* [Agarwal et al., 2020e,a]. Here we give some concrete examples to serve as the special cases of the independent linear Markov game.

Example 1. (Tabular Markov games) Let $d_i = SA_i$ and set $\phi_i(s, a_i) = e_{(s,a_i)}$ be the canonical basis in $\mathbb{R}^{d_i}$ for all $i \in [m]$. Then we recover tabular Markov game with misspecification error $\nu = 0$.

Example 2. (State abstraction Markov games) Suppose we have an abstraction function $\psi : \mathcal{S} \to \mathcal{Z}$ for all $h \in [H]$, where $\mathcal{Z}$ is a finite set as the "state abstractions" such that states with the same images have similar properties. The model misspecification is defined as

$$\epsilon_h(z) := \max_{s,s':\psi(s)=\psi(s')=z;i\in[m],h\in[H],\mathbf{a}\in\mathcal{A}} \left\{ \left| r_{h,i}(s,\mathbf{a}) - r_{h,i}(s',\mathbf{a}) \right|, \left\| \mathbb{P}_h(\cdot \mid s,\mathbf{a}) - \mathbb{P}_h(\cdot \mid s',\mathbf{a}) \right\|_1 \right\}, \forall z \in \mathcal{Z}.$$

We define ν-misspecified state abstraction Markov games to satisfy that for any policy π, we have

$$\left| \sum_{h=1}^{H} \mathbb{E}_\pi \left[\epsilon_h(\psi(s_h)) \right] \right| \leq \nu,$$

which means the misspecification error is small under any policy π.

Proposition 4.3.2. *ν-misspecified state abstraction Markov games (Example 2) are $H\nu$-misspecified independent linear Markov games with $\Pi^{\text{abstraction}} = \{\pi \mid \pi_h(\cdot \mid s) = \pi_h(\cdot \mid s'), \psi(s) = \psi(s')\}$, $d_i = |\mathcal{Z}|A_i$ for all $i \in [m]$ and feature $\phi_i(s, a_i) = e_{\psi(s),a_i}$ to be the canonical basis in $\mathbb{R}^{d_i}$.*

Example 3. (Congestion games) Congestion games are normal-form general-sum games defined by the tuple $(\mathcal{F}, \{A_i\}_{i=1}^m, \{r^f\}_{f\in\mathcal{F}})$, where $\mathcal{F}$ is the facility set with $F = |\mathcal{F}|$, $\mathcal{A}_i \subseteq 2^{\mathcal{F}}$ is the action set for player $i \in [m]$, and $r^f(n) \in [0, 1/F]$ is a random reward function with mean $R^f(n)$ for all $n \in [m]$. For a joint action $\mathbf{a} = (a_1, \cdots, a_m)$, $n^f(\mathbf{a}) = \sum_{i=1}^m \mathbf{1}\{f \in a_i\}$ is the number of players choosing facility f and the reward collected for player i is $r_i(\mathbf{a}) = \sum_{f\in a_i} r^f(n^f(\mathbf{a}))$, which is sum of the reward from the facilities they choose.

Proposition 4.3.3. *Congestion games (Example 3) are independent linear Markov games with $S = 1$, $H = 1$ and $d_i = F$ for all $i \in [m]$ and misspecification error $\nu = 0$.*

The proofs for Proposition 4.3.2 and Proposition 4.3.3 are deferred to Appendix C.1. These examples demonstrate the generality of the linear Markov games we defined. We want to emphasize that the complexity of tabular Markov games would be $d = S\prod_{i\in[m]} A_i$ if we apply the global function approximation models in Chen et al. [2022], Ni et al. [2022], which is exponentially larger than $d_{\max} = S\max_{i\in[m]} A_i$, as in the tabular setting when model-based approaches are used [Bai and Jin, 2020a, Zhang et al., 2020a, Liu et al., 2021a]. See Table 4.1 for a detailed comparison.

4.4 Algorithms and Analyses for Linear Markov Games

4.4.1 Experience Replay and Policy Replay

Before getting into the details of our algorithm, we will first review two popular exploration paradigms in single-agent RL, namely *experience replay* and *policy replay*. Experience replay is utilized in most empirical and theoretical algorithms, which adds new on-policy data to a dataset and then uses the dataset to retrain a new policy [Mnih et al., 2013, Azar et al., 2017, Jin et al., 2020]. By carefully designing how to train the new policy to strategically explore the underlying MDP, the dataset will contain more and more information about the MDP and thus we can learn the optimal policy without any simulator.

Another popular approach is called policy replay, which is also known as policy cover. Instead of incrementally maintaining a dataset, the algorithm will maintain a policy set,

and at each episode renew the dataset by drawing fresh samples using the policies in this policy set. As the dataset is completely refreshed at each episode, policy replay is able to tackle non-stationarity and enjoy better robustness in many different settings. In Agarwal et al. [2020a], it is used to address the "catastrophic forgetting" problem in policy gradient methods while being robust to the so-called transfer error. In Zanette and Wainwright [2022], Daskalakis et al. [2022], it is used to tackle the non-stationarity in Q-learning with function approximation and non-stationarity of multiple agents in tabular Markov games, respectively.

In independent linear Markov games, non-stationarity comes from both multiple agents and function approximation. In particular, the change in other players' policies will lead to a different independent state-action value function to estimate, and the change in the next-step value function estimate will lead to changing targets for regression. In our algorithm, we will show that policy replay can tackle both types of non-stationarity at the same time as we use it to create a stationary environment with fixed regression targets, which leads to provably efficient algorithms for independent linear Markov games. Policy replay also guarantees that if each player has a misspecified feature, the final guarantee will only have a linear dependence on the misspecification error. In addition, we will provide a carefully designed policy-replay-type algorithm for tabular Markov games which has significant improvement over Daskalakis et al. [2022] in Section 4.6.

4.4.2 Algorithm

One technical difficulty in designing algorithms for linear Markov games is that we can no longer resort to adversarial bandits oracles, which is utilized in all algorithms that can break the curse of multiagents [Jin et al., 2021b, Song et al., 2021a, Mao et al., 2022, Daskalakis et al., 2020]. This is because adversarial contextual linear bandits oracle is necessary to avoid dependence on S and A_i. However, to the best of our knowledge, the only relevant result considering i.i.d. context with known covariance is Neu and Olkhovskaya [2020], which can not fit into Markov games. Indeed, adversarial linear bandits with changing action set is still an open problem (See Section 29.4 in Lattimore and Szepesvári [2020]).

Protocol 1 No-regret Learning Algorithm

Initialize: Action set $\mathcal{B}$, and p_1 to be the uniform distribution over $\mathcal{B}$.

for $t = 1, 2, \ldots, T$ **do**

Adversary chooses loss l_t.

Observe loss l_t.

Update $p_{t+1} \leftarrow$ No_Regret_Update(l_t).

end for

Perhaps surprisingly, our algorithms only require no-regret learning with full-information feedback oracle (Protocol 1). This oracle is considerably easier than the previous (weighted) high-probability adversarial bandit with noisy bandit feedback oracles [Jin et al., 2021b, Daskalakis et al., 2022]. The intuition is that as all the players are using the same algorithm, the environment is not completely adversarial and we can take multiple i.i.d. samples so that the full-information feedback can be constructed with the batched data.

No_Regret_Update subroutine. Consider the expert problem with B experts [Freund and Schapire, 1997]. We use $\mathcal{B}$ to denote the action set with $|\mathcal{B}| = B$, and the policy $p \in \Delta(\mathcal{B})$. At round t, the adversary chooses some loss l_t (also known as the "expert advice"). Then the learner observes the loss l_t and updates the policy to p_{t+1}, which is denoted as $p_{t+1} \leftarrow$ No_Regret_Update(l_t).

For learning CCE and CE, the no-regret learning oracle needs to satisfy the following no-external-regret and no-swap-regret properties, respectively. We will use the minimax optimal no-external-regret and no-swap-regret algorithms while any other no-regret algorithms are eligible. Assumption 4.4.1 and Assumption 4.4.2 can be achieved by EXP3 [Freund and Schapire, 1997] and BM-EXP3 [Blum and Mansour, 2007], respectively.

Assumption 4.4.1. (No-external-regret with full-information feedback) For any loss sequence $l_1, \ldots, l_T \in \mathbb{R}^B$ bounded between $[0, 1]$, the no-regret learning oracle (Protocol 1) enjoys external-regret [Freund and Schapire, 1997]:

$$\max_{b \in \mathcal{B}} \sum_{t=1}^{T} (\langle p_t, l_t \rangle - l_t(b)) \leq \mathrm{Reg}(T) := O(\sqrt{\log(B)T}).$$

Assumption 4.4.2. (No-swap-regret with full-information feedback) For any loss sequence $l_1, \ldots, l_T \in \mathbb{R}^B$ bounded between $[0, 1]$, the no-regret learning oracle (Protocol 1) enjoys swap-regret [Blum and Mansour, 2007, Ito, 2020]:

$$\max_{\psi \in \Psi} \sum_{t=1}^{T} (\langle p_t, l_t \rangle - \langle \psi \diamond p_t, l_t \rangle) \leq \text{SwapReg}(T) := O(\sqrt{B \log(B) T}),$$

where Ψ denote the set $\{\psi : \mathcal{B} \to \mathcal{B}\}$ which consists of all possible strategy modifications.

We will explain the algorithm for learning Markov CCE and the only difference in learning Markov CE is to use the no-swap-regret oracle to replace the no-external-regret one. The algorithm has two main components: learning Markov CCE with policy cover and policy cover update. For the first part, given a policy cover Π, we will compute an approximate optimistic CCE under the distribution induced by the policy cover. Specifically, we use a value-iteration-type algorithm that computes the CCE and the corresponding value function from step H to 1 (Line 5). At each step h, each player will run a no-regret algorithm for T steps (Line 8). In this inner loop, we will generate a dataset by using policies in the policy cover concatenated with the current policies from the no-regret oracle (Line 11). Then we compute an optimistic local Q function $\overline{Q}_{h,i}^{k,t}$ via constrained least squares and feed it into the no-regret algorithm as the full-information feedback (Line 19 and Line 23). At the end of the no-regret loop, we will compute the optimistic value function, which will be an upper bound of the best response value with high probability (Line 26).

For the policy cover update part, we utilize a lazy update to ensure that the algorithm will end within $K \leq K_{\max} := \widetilde{O}(mHd_{\max})$ episodes with high probability, which can significantly improve the final sample complexity bound, similar to the single-agent MDP case studied in Zanette and Wainwright [2022]. We maintain a counter $T_{h,i}$ for each player i at each step h, which estimates the information gained by adding the current policy π^k to the existing policy cover (Line 38). Whenever there is a counter satisfying $T_{h,i} \geq T_{\text{Trig}}$ for some carefully chosen parameter T_{Trig}, we will add (π^k, n^k) to the policy cover, where n^k is the number of times that π^k should be repeated in data collection. In addition, the algorithm will terminate when the dataset size reaches N (Line 42 and Line 29) so that the sample complexity is always upper bounded by $O(mHTK_{\max}N)$.

Algorithm 3 **P**olicy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (**PReFI**)

Input: ϵ, δ, $d_{\max}$, λ, β, T_{Trig}, $K_{\max}$, T, N

Initialization: Policy Cover $\Pi = \emptyset$. $n^{\mathrm{tot}} = 0$.

for episode $k = 1, 2, \ldots, K_{\max}$ **do**

  Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$, $n^k = 0$.

  for $h = H, H-1, \ldots, 1$ **do** ▷ Retrain policy with the current policy cover

    Initialize $\pi^{k,1}_{h,i}$ to be uniform policy for all player i. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.

    Each player i initializes a no-regret learning instance (Protocol 1) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for which we will use $\textsc{No\_Regret\_Update}_{h,i,s}(\cdot)$ to denote the update.

    for $t = 1, 2, \ldots, T$ **do**

      for $i \in [m]$ **do**

        Set Dataset $\mathcal{D}^{k,t}_{h,i} = \emptyset$.

        for $l = 1, 2, \ldots, \sum_{j=1}^{k-1} n^j$ **do**

          Sample $\pi^l \in \Pi = \{\pi^j\}_{j=1}^{k-1}$ with probability $n^l / \sum_{j=1}^{k-1} n^j$.

          Draw a joint trajectory $(s^l_1, \mathbf{a}^l_1, r^l_{1,i}, \ldots, s^l_h, \mathbf{a}^l_h, r^l_{h,i}, s^l_{h+1})$ from $\pi^l_{1:h-1} \circ \left(\pi^l_{h,i}, \pi^{k,t}_{h,-i}\right)$, which is the policy that follows π^l for the first $h-1$ steps and follows $\pi^l_{h,i}, \pi^{k,t}_{h,-i}$ for step h.

          Add $(s^l_h, a^l_{h,i}, r^l_{h,i}, s^l_{h+1})$ to $\mathcal{D}^{k,t}_{h,i}$.

        end for

        Set $\Sigma^{k,t}_{h,i} = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \phi_i(s,a)\phi_i(s,a)^\top$.

        Set $\overline{\theta}^{k,t}_{h,i} = \mathrm{argmin}_{\|\theta\| \le H\sqrt{d_{\max}}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \overline{V}^k_{h+1,i}(s')\right)^2$.

        Set $\underline{\theta}^{k,t}_{h,i} = \mathrm{argmin}_{\|\theta\| \le H\sqrt{d_{\max}}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \underline{V}^k_{h+1,i}(s')\right)^2$.

        Set $\overline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \overline{\theta}^{k,t}_{h,i} \right\rangle + \beta \left\|\phi_i(\cdot,\cdot)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.

        Set $\underline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \mathrm{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \underline{\theta}^{k,t}_{h,i} \right\rangle - \beta \left\|\phi_i(\cdot,\cdot)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.

        Update $\overline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\overline{V}^k_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\overline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.

        Update $\underline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\underline{V}^k_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\underline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.

        Update the no-regret learning instance for all state s at step h: $\pi^{k,t+1}_{h,i}(\cdot \mid s) \leftarrow \textsc{No\_Regret\_Update}_{h,i,s}(1 - \overline{Q}^{k,t}_{h,i}(s,\cdot)/H)$.

      end for

    end for

Algorithm 4 **P**olicy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (**PReFI**) (Part 2)

Set $\overline{V}^k_{h,i}(s) \leftarrow \text{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot (\text{Swap})\text{Reg}(T)\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.

end for

Set π^k to be the Markov joint policy such that $\pi^k_h(\mathbf{a}|s) = \frac{1}{T}\sum_{t=1}^{T}\prod_{i\in[m]}\pi^{k,t}_{h,i}(a_i|s)$.

if $n^{\text{tot}} = N$ **then**

Output $\pi^{\text{output}} = \pi^{k^{\text{output}}}$, where $k^{\text{output}} = \text{argmin}_{k'\in[k]}\max_{i\in[m]}\overline{V}^{k'}_{1,i}(s_1) - \underline{V}^{k'}_{1,i}(s_1)$.

end if

Set $T_{h,i} = 0$, for all $h \in [H], i \in [m]$.

repeat ▷ Update policy cover

Reset to $s = s_1$, $n^k = n^k + 1$, $n^{\text{tot}} = n^{\text{tot}} + 1$.

for $h = 1, 2, \ldots, H$ **do**

Play $\mathbf{a} = \pi^k_h(\cdot|s)$.

for $i \in [m]$ **do**

$T_{h,i} \to T_{h,i} + \|\phi_i(s, a_i)\|^2_{[\Sigma^{k,1}_{h,i}]^{-1}}$.

end for

Get next state s', $s \to s'$.

end for

until $\exists h \in [H], i \in [m]$ such that $T_{h,i} \geq T_{\text{Trig}}$ or $n^{\text{tot}} = N$.

Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.

end for

We also have a policy certification part, where similar ideas have been utilized in Dann et al. [2019], Liu et al. [2021a], Ni et al. [2022] to convert regret-based analysis to sample complexity. Specifically, we maintain a pessimistic value estimate $\underline{V}_{1,i}^k(s_1)$, which satisfies $\underline{V}_{1,i}^k(s_1) \le V_{1,i}^{\pi^k}(s_1)$ with high probability (Line 22). Thus the output policy is the best approximation of Markov CCE in the policy cover. This technique can be applied to most no-regret algorithms in RL to transform regret bounds to sample complexity bounds with a better dependence on the failure probability δ.[¶]

4.4.3 Decentralized Implementation

Now we discuss the implementation details of the algorithm. Our algorithm can be implemented in a decentralized manner as specified below:

1. All players know the input parameters of the algorithm.

2. Each player only knows their own features $\phi_i(\cdot,\cdot)$ and observes the states, individual actions, and individual rewards in each sample trajectory.

3. All players have shared random seeds to sample from the output Markov joint policy π^{output}.

4. All players have shared random seeds to sample from the Markov joint policy π^k, which is the policy learned at episode k.

5. All players can communicate $O(1)$ bit at each episode $k \in [K]$.

V-learning [Jin et al., 2021b, Song et al., 2021a, Mao et al., 2022] can be implemented with (1), (2) and (3), and SPoCMAR [Daskalakis et al., 2022] can be implemented with (1), (2), (3) and (4). Similar to the algorithm proposed in Daskalakis et al. [2022], our algorithm can be implemented in a decentralized way with shared random seeds to enable

[¶]In Jin et al. [2018], they show how to transform regret bounds to sample complexity bounds while the dependence on failure probability becomes $1/\delta$. This technique can improve it to $\log(1/\delta)$.

sampling from the Markov joint policy π^k. In details, when the players want to sample $\mathbf{a} \sim \pi_h^k(\mathbf{a} \mid s) = \frac{1}{T}\sum_{t=1}^{T}\prod_{i\in[m]}\pi_{h,i}^{k,t}(a_i \mid s)$, each player samples $t \sim \mathrm{Unif}(T)$ with the shared random seed and then independently samples $a_i \sim \pi_{h,i}^{k,t}(a_i \mid s)$. Our algorithm also requires $O(1)$ communication for broadcasting the policy cover update (Line 42) and the output policy (Line 30) at each episode.[‖] The total communication complexity is bounded by $O(K_{\max}) = \widetilde{O}(mHd_{\max})$ with only polylog dependence on the accuracy ϵ.

In Appendix C.3, we present another algorithm for MARL in independent linear Markov games without communication, which can be implemented with (1), (2), (3) and (4). To remove communication, we utilize agile policy cover update and the number of episodes becomes $K = \widetilde{O}(m^2H^4d_{\max}^2\epsilon^{-2})$. As a result, the final sample complexity will be worse than Algorithm 3. It would be an interesting future direction to study this tradeoff between communication and sample complexity.

4.4.4 Guarantees

Our algorithm, **PReFI**, has the following guarantees for learning Markov CCE and Markov CE in linear Markov games. The sample complexity only has polynomial dependence on $d_{\max}$, which exponentially improves all the previous results for Markov games with function approximation. Note that the $\widetilde{O}(\cdot)$ notation here only hide polylog dependence on $m, H, d_{\max}, \epsilon, \delta$, and the $\log(A_{\max})$ factor in the bound can be replaced by $d_{\max}$ as in adversarial linear bandits [Bubeck et al., 2012].

Theorem 4.4.3. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4\nu)$-approximate Markov CCE. The sample complexity is*

$$O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4\log(A_{\max})\epsilon^{-4}),$$

where $d_{\max} = \max_{i\in[m]} d_i$ and $A_{\max} = \max_{i\in[m]} A_i$.

[‖] Line 30 can be implemented with $O(1)$ communication at each episode by maintaining the best index and corresponding value up to the current episode k.

Theorem 4.4.4. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.2. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4\nu)$-approximate Markov CE. The sample complexity is*

$$O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4 A_{\max}\log(A_{\max})\epsilon^{-4}).$$

The choice of input parameters and the proofs are deferred to Appendix C.2. As Markov CCE is equivalent to Markov NE in two-player zero-sum Markov games, we directly have the following Corollary.

Corollary 4.4.5. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. Then for ν-misspecified independent linear two-player zero-sum Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4\nu)$-approximate Markov NE. The sample complexity is $O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4\log(A_{\max})\epsilon^{-4})$.*

By Proposition 4.3.2, we have the following corollary for state abstraction Markov games. Note that the feature is the same $\phi_i(s,a_i) = \phi_i(s',a_i)$ if $\psi(s) = \psi(s')$, so $\pi^{k,t} \in \Pi^{\text{abstraction}}$ for all $(k,t) \in [K]\times[T]$ as the full-information feedback would be the same for s and s' mapped to the same abstraction and then the policy would be same as well.

Corollary 4.4.6. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. Then for ν-misspecified state abstraction Markov games, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4H\nu)$-approximate Markov NE. The sample complexity is $O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}|\mathcal{Z}|^4A_{\max}^4\log(A_{\max})\epsilon^{-4})$.*

4.5 Learning Markov NE in Independent Linear Markov Potential Games

In this section, we will focus on a special class of independent linear Markov games, namely *independent linear Markov potential games*. The existence of the potential function guarantees that the stationary points of the potential function are NE [Leonardos et al., 2021], which means the iterative best-response dynamic can converge to NE as it is similar to coordinate descent [Durand, 2018]. Specifically, we will provide an iterative best-response-type

algorithm that can learn pure Markov NE in independent linear Markov potential games, which generalizes the algorithm for tabular Markov potential games in Song et al. [2021a].

As when the other players are fixed, player $i \in [m]$ will be in an approximate linear MDP, existing algorithms for misspecified linear MDP can all serve as the best-response oracle. The algorithm will use the following oracle LinearMDP_Solver that can solve misspecified linear MDPs. Here misspecified linear MDPs are the degenerated cases of misspecified independent linear Markov games with only one player and thus no Π^{estimate} is included, which is similar to the model in Zanette and Wainwright [2022].

Definition 4.5.1. Feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is ν-misspecified if for any rollout policy $\overline{\pi}$, target policy $\widetilde{\pi}$, we have for any $V \in \mathcal{V}$,

$$\left| \sum_{h=1}^{H} \mathbb{E}_{\widetilde{\pi}} \left[\text{proj}_{[0,H+1-h]} \left(\left\langle \phi(s_h, a_h), \theta_h^{\overline{\pi},V} \right\rangle \right) - Q_h^V(s_h, a_h) \right] \right| \leq \nu,$$

where

$$\theta_h^{\overline{\pi},V} = \underset{\|\theta\| \leq H\sqrt{d}}{\text{argmin}} \, \mathbb{E}_{\overline{\pi}} \left(\langle \phi(s_h, a_h), \theta \rangle - Q_h^V(s_h, a_h) \right)^2, Q_h^V(s_h, a_h) = \mathbb{E} \left[r_h(s_h, a_h) + V_{h+1}(s_{h+1}) \right].$$

We say a Markov decision process with feature ϕ is a ν-misspecified linear MDP if the feature map ϕ is ν-misspecified.

Assumption 4.5.2. For any ν-misspecified linear MDP with feature $\phi(s, a) \in \mathbb{R}^d$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, LinearMDP_Solver takes features $\phi(\cdot, \cdot)$ as input and can interact with the underlying linear MDP. Then it can output an $(\epsilon + O(\nu))$-approximate optimal policy with sample complexity LinearMDP_SC(ϵ, δ, d) with probability at least $1 - \delta$. Without loss of generality, we assume that LinearMDP_SC(ϵ, δ, d) is non-decreasing w.r.t. d.

In Appendix C.4, we will adapt Algorithm 3 to the single-agent case to serve as LinearMDP_Solver with LinearMDP_SC$(\epsilon, \delta, d) = \widetilde{O}(H^6 d^4 \epsilon^{-2})$ and output an $(\epsilon + 4\nu)$-optimal policy (See Algorithm 22). With the best-response oracle, we provide our MARL algorithm for linear Markov potential games (Algorithm 5). It is easy to see that Algorithm 5 can be implemented in the same decentralized way as Algorithm 3. Below we provide the sample complexity guarantees.

Algorithm 5 Nash **C**oordinate **A**scent for Independent **Lin**ear Markov Potential Games (**Lin-Nash-CA**)

Input: ϵ, δ, $K = 5mH\epsilon^{-1}$
Initialization: π^1 to be an arbitrary deterministic policy.
for episode $k = 1, 2, \ldots, K$ **do**
    Execute policy π^k for $\widetilde{O}(H^2\epsilon^{-2})$ episodes and obtain $\widehat{V}_{1,i}^{\pi^k}(s_1)$ as the empirical average of the total reward for all player $i \in [m]$.
    for $i \in [m]$ **do**
        Fix all the players except player i to follow policy π_{-i}^k and player i runs LinearMDP_Solver with feature $\phi_i(\cdot,\cdot)$, accuracy $\epsilon/8$ and failure probability $\delta/(2mK)$. Set $\widehat{\pi}_i^{k+1}$ to be the output of LinearMDP_Solver.
        Execute policy $(\widehat{\pi}_i^{k+1}, \pi_{-i}^k)$ for $\widetilde{O}(H^2\epsilon^{-2})$ episodes and obtain $\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1)$ as the empirical average of the total reward.
        Set $\Delta_i \leftarrow \widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) - \widehat{V}_{1,i}^{\pi^k}(s_1)$.
    end for
    if $\max_{i\in[m]} \Delta_i > \epsilon/2$ **then**
        Set $\pi^{k+1} : \pi_i^{k+1} = \pi_i^k, \pi_j^{k+1} = \widehat{\pi}_j^k$ for $i \neq j$ and $j = \operatorname{argmax}_{i\in[m]} \Delta_i$.
    else
        Output $\pi^{\text{output}} = \pi^k$.
    end if
end for

Theorem 4.5.3. *For ν-misspecified independent linear Markov potential games with $\Pi^{\text{estimate}} = \{\pi^k\}_{k=1}^K$, with probability at least $1-\delta$, Algorithm 5 will output an $(\epsilon + O(\nu))$-approximate pure Markov NE. The sample complexity is*

$$O(m^2 H\epsilon^{-1} \cdot \text{LinearMDP\_SC}(\epsilon/8, \delta/(10m^2H\epsilon^{-1}), d_{\max})).$$

As the congestion game is a special case of linear Markov potential game (Proposition 4.3.3), we have the following corollary if we replace the linear MDP solver with a linear bandit solver with LinearBandit_SC(ϵ, δ, d)) sample complexity.

Corollary 4.5.4. *For congestion games, with probability at least* $1-\delta$*, Algorithm 5 will output an* ϵ*-approximate pure NE. The sample complexity is*

$$O(m^2\epsilon^{-1}\cdot \text{LinearBandit\_SC}(\epsilon/8, \delta/(10m^2H\epsilon^{-1}), F)).$$

If we use Algorithm 22 as the oracle, the sample complexity for linear Markov potential games would be $\widetilde{O}(m^2H^7d_{\max}^4\epsilon^{-3})$. For linear bandits, it is easy to adapt the $\widetilde{O}(d\sqrt{K})$ algorithm in Abbasi-Yadkori et al. [2011] to sample complexity $\widetilde{O}(d^2\epsilon^{-2})$, which leads to $\widetilde{O}(m^2F^2\epsilon^{-3})$ sample complexity for congestion games.** Our algorithm significantly improves the previous result for the decentralized algorithm, which has sample complexity $\widetilde{O}(m^{12}F^6\epsilon^{-6})$ [Cui et al., 2022].

4.6 Improved Sample Complexity in Tabular Case

In this section, we will present an algorithm specialized to tabular Markov games based on the policy cover technique in Algorithm 3. The sample complexity for learning an ϵ-approximate Markov CCE is $\widetilde{O}(H^6S^2A_{\max}\epsilon^{-2})$, which significantly improves the previous state-of-the-art result $\widetilde{O}(H^{11}S^3A_{\max}\epsilon^{-3})$ [Daskalakis et al., 2022], and is only worse than learning an ϵ-approximate *non-Markov* CCE by a factor of HS [Jin et al., 2021b]. In addition, our algorithm can learn an ϵ-approximate Markov CE with $\widetilde{O}(H^6S^2A_{\max}^2\epsilon^{-2})$ sample complexity, which is the first provably efficient result for learning Markov CE in tabular Markov games.

Adv_Bandit_Update subroutine. Consider the adversarial multi-armed bandit problem with B arms. At round t, the adversary chooses some loss l_t and the learner chooses some action $b_t\sim p_t$, where $p_t\in\Delta(\mathcal{B})$ is the policy at round t. Then the learner observes a noisy bandit-feedback $\widetilde{l}_t(b_t)\in[0,1]$ such that $\mathbb{E}[\widetilde{l}_t(b_t)\mid l_t, b_t] = l_t(b_t)$. The player will update the policy to p_{t+1} for round $t+1$, which is denoted as $p_{t+1}\leftarrow \textsc{Adv\_Bandit\_Update}(b_t, \widetilde{l}_t(b_t))$.

For learning CCE and CE, the adversarial bandit algorithm (Protocol 2) needs to satisfy the following no-external-regret and no-swap-regret properties, respectively. The following

**E.g., we can use policy certification as in Algorithm 3 to find the best policy among all the policies played with no additional sample complexity.

Protocol 2 Adversarial Bandit Algorithm

Initialize: Action set $\mathcal{B}$, and p_1 to be the uniform distribution over $\mathcal{B}$.

for $t = 1, 2, \ldots, T$ **do**

Adversary chooses loss l_t.

Player take action $b_t \sim p_t$ and observe noisy bandit-feedback $\widetilde{l}_t(b_t)$.

Update $p_{t+1} \leftarrow$ ADV_BANDIT_UPDATE$(b_t, \widetilde{l}_t(b_t))$.

end for

two assumptions can be achieved by leveraging the results in Neu [2015] and Blum and Mansour [2007], which is shown in Jin et al. [2021b].[††]

Assumption 4.6.1. (No-external-regret with bandit-feedback) For any loss sequence $l^1, \ldots, l^T \in \mathbb{R}^B$ bounded between $[0, 1]$, the adversarial bandit oracle satisfies that with probability at least $1 - \delta$, for all $t \leq T$,

$$\max_{b \in \mathcal{B}} \sum_{i=1}^{t} (\langle p_i, l_i \rangle - l_i(b)) \leq \mathrm{BReg}(t) := O\left(\sqrt{Bt} \log(Bt/\delta)\right).$$

Assumption 4.6.2. (No-swap-regret with bandit-feedback) For any loss sequence $l^1, \ldots, l^T \in \mathbb{R}^B$ bounded between $[0, 1]$, the adversarial bandit oracle satisfies that with probability at least $1 - \delta$, for all $t \leq T$,

$$\max_{\psi \in \Psi} \sum_{i=1}^{t} (\langle p_i, l_i \rangle - \langle \psi \diamond p_i, l_i \rangle) \leq \mathrm{BSwapReg}(t) := O\left(B\sqrt{t} \log(Bt/\delta)\right).$$

where Ψ denotes the set $\{\psi : \mathcal{B} \to \mathcal{B}\}$ which consist of all possible strategy modifications.

Here we emphasize several major differences between Algorithm 6 and Algorithm 3. The choice of input parameters and the proofs are deferred to Appendix C.6.

1. The states and actions are no longer entangled through the feature map as in independent linear Markov games. As a result, we can use the adversarial bandit oracle to explore *individual action space* while using policy cover to explore the *shared state space*. Then there will be no inner loop for estimating the full-information feedback and saving $\widetilde{O}(\epsilon^{-2})$ factors.

[††]They proved a stronger version for weighted regret while we only require the unweighted version.

Algorithm 6 **P**olicy **Re**ply with **B**andit **O**racle in Tabular Markov Games (**PReBO**)

Input: ϵ, δ, β, T_{Trig}, $K_{\max}$, $N_{\max}$

Initialization: Policy Cover $\Pi = \emptyset$. $n^{\mathrm{tot}} = 0$.

for episode $k = 1, 2, \ldots, K_{\max}$ **do**

  Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$, $n^k = 0$, $n^k_h(s) = 0$ for all $h \in [H]$ and $s \in \mathcal{S}$.

  for $h = H, H-1, \ldots, 1$ **do** ▷ Retrain policy with the current policy cover

    Initialize $\pi^{k,1}_{h,i}$ to be uniform policy for all player i. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.

    Each player i initializes an adversarial bandit instance (Protocol 2) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for which we will use $\textsc{No\_Regret\_Update}_{h,i,s}(\cdot)$ to denote the update.

    for $t = 1, 2, \ldots, \sum_{j=1}^{k-1} n^j$ **do**

      Sample $\pi^l \in \Pi$ with probability $n^l / \sum_{j=1}^{k-1} n^j$.

      Draw a joint trajectory $(s_1, \mathbf{a}_1, \mathbf{r}_1, \ldots, s_h, \mathbf{a}_h, \mathbf{r}_h, s_{h+1})$ from $\pi^l_{1:h-1} \circ \pi^{k,t}_h$,which is the policy that follows π^l for the first $h-1$ steps and follows $\pi^{k,t}_h$ for step h.

      Update $n^k_h(s_h) \leftarrow n^k_h(s_h) + 1$.

      Update the adversarial bandit instance for player i at step h and state s_h: $\pi^{k,t+1}_{h,i}(\cdot|s_h) \leftarrow \textsc{Adv\_Bandit\_Update}_{h,i,s_h}(a_{h,i}, 1 - \left(r_{h,i} + \overline{V}^k_{h+1,i}(s_{h+1})\right)/H)$.

      Update policy $\pi^{k,t+1}_{h,i}(\cdot|s) \leftarrow \pi^{k,t+1}_{h,i}(\cdot|s)$ for $s \neq s_h$.

      Update $\overline{V}^k_{h,i}(s_h) \leftarrow \frac{n^k_h(s_h)-1}{n^k_h(s_h)} \overline{V}^k_{h,i}(s_h) + \frac{1}{n^k_h(s_h)}(r_{h,i} + \overline{V}^k_{h+1,i}(s_{h+1}))$.

      Update $\underline{V}^k_{h,i}(s_h) \leftarrow \frac{n^k_h(s_h)-1}{n^k_h(s_h)} \underline{V}^k_{h,i}(s_h) + \frac{1}{n^k_h(s_h)}(r_{h,i} + \underline{V}^k_{h+1,i}(s_{h+1}))$.

    end for

    Set $\overline{V}^k_{h,i}(s) \leftarrow \mathrm{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot \mathrm{B(Swap)Reg}(n^k_h(s_h)) + \beta_{n^k_h(s)}\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.

    Set $\underline{V}^k_{h,i}(s) \leftarrow \mathrm{proj}_{[0,H+1-h]}\left(\underline{V}^k_{h,i}(s) - \beta_{n^k_h(s)}\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.

  end for

  Set π^k to be the Markov joint policy such that $\pi^k_h(\mathbf{a}|s) = \frac{1}{n^k_h(s)} \sum_{j=1}^{n^k_h(s)} \prod_{i \in [m]} \pi^{k,t^k_h(j;s)}_{h,i}(a_i|s)$, where $t^k_h(j;s)$ is the time t such that state s is visited for the j-th time in episode k at step h.

  if $\max_{i \in [m]} \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \leq \epsilon$ **then** ▷ Policy certification

    Output: $\pi^{\mathrm{output}} = \pi^t$.

  end if

  Set $T^k_h(s) = 0$ for all $h \in [H]$, $s \in \mathcal{S}$.

  repeat ▷ Update policy cover

Algorithm 7 **P**olicy **Re**ply with **B**andit **O**racle in Tabular Markov Games (**PReBO**) (Part 2)

Reset $s = s_1$, $n^k = n^k + 1$, $n^{\text{tot}} = n^{\text{tot}} + 1$.
for $i \in [m]$ **do**
  for $h = 1, 2, \ldots, H$ **do**
    Play $\mathbf{a}_h = \pi_h^k(\cdot|s)$.
    $T_h^k(s_h) \leftarrow T_h^k(s_h) + 1$.
    Get next state s', $s \to s'$.
  end for
end for
until $\exists h \in [H]$ such that $T_h^k(s_h) = n_h^k(s_h) \vee T_{\text{Trig}}$ or $n^{\text{tot}} = N_{\max}$.
Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.
end for

2. For independent linear Markov games, each player has its own feature space so that the exploration progress is different and communication is required to synchronize. However, in tabular Markov games, all the players explore in the shared state space, which means the exploration progress is inherently synchronous and no communication is required. The triggering event is that whenever a state visitation is approximately doubled, the policy cover will update, which guarantees that with high probability, the number of episodes is bounded by $\widetilde{O}(HS)$.

Theorem 4.6.3. *Suppose Algorithm 6 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 4.6.1. Then for tabular Markov games, with probability at least $1 - \delta$, Algorithm 6 will output an ϵ-approximate Markov CCE. The sample complexity is $\widetilde{O}(HK_{\max}N_{\max}) = \widetilde{O}(H^6S^2A_{\max}\epsilon^{-2})$.*

Theorem 4.6.4. *Suppose Algorithm 6 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 4.6.2. Then for tabular Markov games, with probability at least $1 - \delta$, Algorithm 6 will output an ϵ-approximate Markov CE. The sample complexity is $\widetilde{O}(HK_{\max}N_{\max}) = \widetilde{O}(H^6S^2A_{\max}^2\epsilon^{-2})$.*

4.7 Conclusion

In this paper, we propose the independent function approximation model for Markov games and provide algorithms for different types of Markov games that can break the curse of multiagents in a large state space. We hope this work can serve as the first step towards understanding the empirical success of MARL with independent function approximation. Below we list some interesting open problems for future research.

1. Sharpen the sample complexity. The sample complexity for independent linear sample complexity is far from optimal. For example, it would be a significant improvement if the dependence on ϵ could be improved to the optimal rate of $\widetilde{O}(\epsilon^{-2})$.

2. Incorporate general function approximation. We study independent linear function approximation as an initial attempt. There is a huge body of general function approximation results for single-agent RL and it would be interesting to study them in the context of independent function approximation for Markov games.

3. Different data collection oracles. In this work, we study the online setting where exploration is necessary. It would be interesting to extend our results to other settings, such as the offline setting or the simulator setting where specific new challenges might occur or the tightest sample complexity is preferred.

Algorithms	Game	Equilibrium	Sample complexity	Sample complexity (tabular)	BCM
[Liu et al., 2021a]	MG	NE/CE/CCE	$H^4S^2\prod_{i=1}^m A_i\epsilon^{-2}$	-	$\times$
[Jin et al., 2021b]	ZSMG	NE	$H^5SA_{\max}\epsilon^{-2}$	-	-
[Jin et al., 2021b]	MG	NM-CCE	$H^5SA_{\max}\epsilon^{-2}$	-	✓
[Jin et al., 2021b]	MG	NM-CE	$H^5SA_{\max}^2\epsilon^{-2}$	-	✓
[Daskalakis et al., 2022]	MG	CCE	$H^{11}S^3A_{\max}\epsilon^{-3}$	-	✓
[Xie et al., 2020a]	ZSMG	NE	$H^4d^3\epsilon^{-2}$	$d=SA_1A_2$	-
[Chen et al., 2021b]	ZSMG	NE	$H^3d^2\epsilon^{-2}$	$d=SA_1A_2$	-
[Huang et al., 2021]	ZSMG	NE	$H^3W^2A_{\max}\epsilon^{-2}$	$W=SA_1A_2$	-
[Jin et al., 2022]	ZSMG	NE	$H^2d^2\epsilon^{-2}$	$d=SA_1A_2$	-
[Chen et al., 2022]	MG	NE/CE/CCE	$S^3(\prod_{i\in[m]}A_i)^2H^3\epsilon^{-2}$	-	$\times$
[Ni et al., 2022]	MG	NE/CE/CCE	$H^6d^4(\prod_{i=1}^m A_i)^2\log(\lvert\Phi\rvert\lvert\Psi\rvert)\epsilon^{-2}$	$d=S\prod_{i\in[m]}A_i$	$\times$
[Ni et al., 2022]	MG	NE/CE/CCE	$m^4H^6d^{2(L+1)^2}A_{\max}^{2(L+1)}\epsilon^{-2}$	$d=S\prod_{i\in[m]}A_i$	$\times$
Algorithm 3 (**PReFI**)	MG	CCE	$m^4H^{10}d_{\max}^4\epsilon^{-4}$	$d_{\max}=SA_{\max}$	✓
Algorithm 3 (**PReFI**)	MG	CE	$m^4H^{10}d_{\max}^4A_{\max}\epsilon^{-4}$	$d_{\max}=SA_{\max}$	✓
Algorithm 6 (**PReBO**)	MG	CCE	$H^6S^2A_{\max}\epsilon^{-2}$	-	✓
Algorithm 6 (**PReBO**)	MG	CE	$H^6S^2A_{\max}^2\epsilon^{-2}$	-	✓

Table 4.1: Comparison of the models and the most related sample complexity results for MARL in Markov games. S is the number of states, m is the number of players, A_i is the number of actions for player i with $A_{\max}=\max_{i\in[m]}A_i$, ϵ is the target accuracy, and d or W is the complexity of the corresponding function class. We use **MG** to denote multi-player general-sum Markov games, **ZSMG** to denote two-player zero-sum Markov games, **NE/CE/CCE** to denote Markov Nash equilibria, Markov correlated equilibria, and Markov coarse correlated equilibria, respectively. We use the prefix (NM-) to denote non-Markov equilibria. For algorithms with function approximation, we show the parameters when applied to the tabular setting and whether breaking the curse of multiagents (**BCM**) or not in the last two columns. Polylog dependence on relevant parameters is omitted in the sample complexity results.

Algorithms	Game type	Sample complexity
[Leonardos et al., 2021]	Markov potential game	$\text{poly}(\kappa, m, A_{\max}, S, H, \epsilon)$
[Ding et al., 2022]	Markov potential game	$\text{poly}(\kappa, m, A_{\max}, d, H, \epsilon)$
[Song et al., 2021a]	Markov potential game	$m^2 H^4 S A_{\max} \epsilon^{-3}$
[Cui et al., 2022] (Centralized)	Congestion game	$m^2 F \epsilon^{-2}$
[Cui et al., 2022] (Decentralized)	Congestion game	$m^{12} F^6 \epsilon^{-6}$
Algorithm 5 (**Lin-Nash-CA**)	Linear Markov potential game	$m^2 H^7 d_{\max}^4 \epsilon^{-3}$
Algorithm 5 (**Lin-Nash-CA**)	Congestion game	$m^2 F^2 \epsilon^{-3}$

Table 4.2: Comparison of algorithms for learning NE in Markov potential games. κ is the distribution mismatch coefficient, S is the number of states, m is the number of players, A_i is the number of actions for player i, $A_{\max} = \max_{i \in [m]} A_i$, F is the number of facilities in congestion games, ϵ is accuracy, and $d_{\max}$ is the complexity of the function class. For Leonardos et al. [2021], Ding et al. [2022], κ can be arbitrarily large as no exploration is considered.

Chapter 5

NON-STATIONARY MARKOV GAMES

This chapter is based on Jiang et al. [2023], with haozhe Jiang, Zhihan Xiong, Maryam Fazel and Simon S. Du.

5.1 Introduction

Multi-agent reinforcement learning (MARL) studies the interactions of multiple agents in an unknown environment with the aim of maximizing their long-term returns [Zhang et al., 2021a]. This field has applications in diverse areas such as computer games [Vinyals et al., 2019b], robotics [de Witt et al., 2020b], and smart manufacturing [Kim et al., 2020]. Although various algorithms have been developed for MARL, it is typically assumed that the underlying repeated game is stationary throughout the entire learning process. However, this assumption often fails to represent real-world scenarios where the environment is evolving throughout the learning process.

The task of learning within a non-stationary multi-agent system, while crucial, poses additional challenges when attempts are made to generalize non-stationary single-agent reinforcement learning (RL), especially for the bandit feedback case where minimal information is revealed to the agents [Anagnostides et al., 2023]. In addition, the various multi-agent settings, such as zero-sum, potential, and general-sum games, along with normal-form and extensive-form games, and fully observable or partially observable Markov games, further complicate the design of specialized algorithms.

In this work, we take the first step towards understanding non-stationary MARL with bandit feedback. First, we point out several challenges that differentiate non-stationary MARL from non-stationary single-agent RL, and bandit feedback from full-information feedback. Subsequently, we propose black-box algorithms with sub-linear dynamic regret in arbitrary non-stationary games, provided there is access to learning algorithms in the

corresponding (near-)stationary environment. This versatile approach allows us to leverage existing algorithms for various stationary games, while facilitating seamless adaptation to future algorithms that may offer improved guarantees.

5.1.1 Main Contributions and Novelties

1. Identifying challenges in non-stationary games with bandit feedback (Section 5.3). First, we point out that bandit feedback is incompatible with online-learning based algorithms as the gradient of reward is hard to estimate. Then, we show that bandit feedback complicates the application of test-based algorithms as testing an arbitrary small gap can incur $O(1)$ regret each term. Non-uniqueness of equilibria makes replay-based test difficult as well. Additionally, we point out that it is non-trivial to generalize an algorithm for non-stationary Markov games to a parameter-free version since the objective for games is very different from that of multi-armed bandits.

2. Generic black-box approach for non-stationary games. Our approach is a black-box reduction that can transform any base algorithm designed for (near-)stationary games into an algorithm capable of learning in a non-stationary environment. This approach inherits favorable properties of the base algorithm, like breaking the curse of multi-agents, and directly adapts to future algorithmic advances.

3. Restart-based algorithm when non-stationarity budget is known (Section 5.4). When we know a bound on the degree of non-stationarity, often measured by number of switches or total variation (which from here on, we refer to as the "nonstationarity budget"), we design a simple restart-based algorithm achieving sublinear dynamic equilibrium regret of $\widetilde{O}\left(L^{1/4}T^{3/4}\right)$ or $\widetilde{O}\left(\Delta^{1/4}T^{3/4}\right)$, where L is the switching number and Δ is the total variation non-stationarity budget. In words, this result implies that all the players follow a near-equilibrium strategy in most episodes.

4. Multi-scale testing algorithm when non-stationarity budget is unknown (Section 5.5). We also propose a multi-scale testing algorithm to optimize the regret when the non-stationarity budget is unknown, which can adaptively avoid the strategy deviating from equilibrium for too many rounds. The algorithm achieves the same $\widetilde{O}\left(L^{1/4}T^{3/4}\right)$ regret for

unknown switching number L, and a marginally higher $\widetilde{O}\left(\Delta^{1/5}T^{4/5}\right)$ regret for unknown total variation budget Δ. The testing algorithms are newly designed and the scheduling is specially designed for the PAC assumptions, which is different from that in Wei and Luo [2021] where regret assumptions are made.

While the ultimate goal is to design no-regret algorithms for each agent, i.e. achieving no-regret no matter what policy other players adopt (like Panageas et al. [2023]), our setting is already applicable in various real-world cases even without yet achieving this desired property, this is discussed with a concrete example below. We leave the problem of finding no-regret algorithms for each individual for future work.

Example (traffic routing with navigation). In traffic routing using navigation applications (Guo et al. [2023]), being able to track Nash Equilibrium is advantageous. Assume all drivers use the same navigation application which runs our algorithm. It is reasonable to assume that drivers adhere to the application's suggestions. After following the route recommended by the application, the drivers all find that their routes are not improvable because all drivers are committing to the equilibrium; this makes drivers satisfied with the algorithm's recommendation.

5.1.2 Related Work

(Stationary) Multi-agent reinforcement learning. Numerous works have been devoted to learning equilibria in (stationary) multi-agent systems, including zero-sum Markov games [Bai et al., 2020, Liu et al., 2021a], general-sum Markov games [Jin et al., 2021b, Mao et al., 2022, Song et al., 2021a, Daskalakis et al., 2022, Wang et al., 2023, Cui et al., 2023], Markov potential games [Leonardos et al., 2021, Song et al., 2021a, Ding et al., 2022, Cui et al., 2023], congestion games [Cui et al., 2022], extensive-form games [Kozuno et al., 2021, Bai et al., 2022, Song et al., 2022], and partially observable Markov games [Liu et al., 2022b]. These works aim to learn equilibria with bandit feedback efficiently, measured by either regret or sample complexity. There also exists a rich literature on asymptotic convergence of different learning dynamics in known games and non-asymptotic convergence with full-information feedback, which are not listed here due to space limitations.

Non-stationary (single-agent) reinforcement learning. The study of non-stationary reinforcement learning originated from non-stationary bandits [Auer et al., 2002, Besbes et al., 2014, Chen et al., 2019, Zhao et al., 2020, Wei and Luo, 2021, Cheung et al., 2022, Garivier and Moulines, 2011]. Auer et al. [2019b] and Chen et al. [2019] first achieve near-optimal dynamic regret without knowing the non-stationary budget for bandits. The most relevant work is Wei and Luo [2021], which also proposes a black-box approach with multi-scale testing and achieves optimal regret in various single-agent settings. We refer readers to Wei and Luo [2021] for a more comprehensive literature review on non-stationary reinforcement learning.

Non-stationary multi-agent reinforcement learning. Most of the previous works have been focused on the full-information feedback setting, which is considerably easier than the bandit feedback setting as testing becomes unnecessary [Cardoso et al., 2019, Zhang et al., 2022a, Anagnostides et al., 2023, Duvocelle et al., 2022, Poveda et al., 2022]. For two-player zero-sum matrix games, Zhang et al. [2022a] proposes a meta-algorithm over a group of base algorithms to tackle with unknown parameters. Anagnostides et al. [2023] studies the convergence of no-regret learning dynamics in non-stationary matrix games, including zero-sum, general-sum and potential games, and shares a similar dynamic regret notion as ours. Notably, Cardoso et al. [2019] also studies the bandit feedback case and aims to minimize NE-regret, while the regret is comparing with the best NE in hindsight instead of a dynamic regret.

5.2 Preliminaries

We consider the multi-player general-sum Markov games framework, which covers a wide range of problems. A multi-agent general-sum Markov game is described by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m, H, \mathbb{P}, \{r_i\}_{i=1}^m)$, where $\mathcal{S}$ is the state space with cardinality S, m is the number of the players, $\mathcal{A}_i$ is the action space for player i with cardinality A_i, H is the length of the horizon, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is the collection of the transition kernels such that $\mathbb{P}_h(\cdot \mid s, \mathbf{a})$ is the next state distribution given the current state s and joint action $\mathbf{a} = (a_1, \cdots, a_m)$ at step h, and $r_i = \{r_{h,i}\}_{h=1}^H$ is the collection of random reward functions for player i with support $[0, 1]$ and mean $\{R_{h,i}\}_{h=1}^H$.

At the beginning of each episode, the players start at a fixed initial state s_1.* At each step $h \in [H]$, each player observes the current state s_h and chooses action $a_{h,i}$ simultaneously. Then player $i \in [m]$ will receive her own reward realization $\widetilde{r}_{h,i} \sim r_{h,i}(s_h, \mathbf{a}_h)$ where $\mathbf{a}_h = (a_{h,1}, \cdots, a_{h,m})$ and the state will transition according to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, \mathbf{a}_h)$. The game terminates when $h = H+1$. We consider the bandit feedback setting where only the reward of the chosen action is revealed to the player.

Here we discuss the generality of Markov games. When the horizon $H = 1$, multi-player general-sum Markov games degenerate to multi-player general-sum matrix games, which include zero-sum games, potential games, congestion games, etc [Nisan et al., 2007b]. If we posit different assumptions on the Markov game structure, we can obtain zero-sum Markov games [Bai et al., 2020], Markov potential games [Leonardos et al., 2021], extensive-form games [Kozuno et al., 2021]. If the state s_h is not directly observable, the Markov games are modeled by partially observable Markov games [Liu et al., 2022b]. A detailed preliminary for different games is deferred to the appendix.

Policy. A Markov joint policy is defined by $\pi = \{\pi_h\}_{h=1}^H$ where $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$ is the policy at step h. We will use π_{-i} to denote that all players except player i are following policy π. A special case of Markov joint policy is Markov product policy, which satifies that there exist policies $\{\pi_i\}_{i=1}^m$ such that for all $h \in [H]$ and $(s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$, we have $\pi_h(\mathbf{a} \mid s) = \prod_{i=1}^m \pi_{h,i}(a_i \mid s)$, where $\pi_i = \{\pi_{h,i}\}_{h=1}^H$ is the collection of Markov policies $\pi_{h,i} : \mathcal{S} \to \Delta(\mathcal{A}_i)$ for player i. In words, a Markov product policy can be factorized into individual policies such that they are uncorrelated.

Value function. Given a Markov game $M \in \mathcal{M}$ and a policy π, the value function for player i is defined as $V_i^M(\pi) := \mathbb{E}_\pi \left[\sum_{h=1}^H r_{h,i}(s_h, \mathbf{a}_h) \mid M\right]$, where the expectation is over the randomness in both the policy and the environment.

Best response and strategy modification. Given a policy π and model M, the best response value for player i is $V_i^M(\dagger, \pi_{-i}) := \max_{\pi_i' \in \Pi_i} V_i^M(\pi_i', \pi_{-i})$, which is the maximum achievable expected return for player i if all the other players are following π_{-i}. Equivalently,

*It is straightforward to generalize to stochastic initial state by adding a dummy state s_0 that transition to the random initial state.

best response is the optimal policy in the induced Markov decision process (MDP), i.e., Markov games with only one player.

A strategy modification $\psi_i = \{\psi_{h,i}\}_{h=1}^H$ is a collection of mappings $\psi_{h,i} : \mathcal{S} \times \mathcal{A}_i \to \mathcal{A}_i$ that maps the joint state-action space to the action space.[†] For policy π, $\psi_i \diamond \pi$ is the modified policy such that

$$(\psi_i \diamond \pi)_h(\mathbf{a} \mid s) = \sum_{\mathbf{a}':\psi_{h,i}(a_i'|s)=a_i, \mathbf{a}'_{-i}=\mathbf{a}_{-i}} \pi_h(\mathbf{a}' \mid s).$$

In other words, $\psi_i \diamond \pi$ is a policy such that if π assigns each player j a random action a_j at state s and step h, then $\psi_i \diamond \pi$ assigns action $\psi_{h,i}(a_i \mid s)$ to player i while all the other players are following the action assigned by policy. We will use Ψ_i to denote all the possible strategy modifications for player i. As Ψ_i contains all the constant strategy modifications, we have

$$\max_{\psi_i \in \Psi_i} V_i^M(\psi_i \diamond \pi) \geq \max_{\pi_i'} V_i^M(\pi_i', \pi_{-i}) = V_i^M(\dagger, \pi_{-i}),$$

which means that the best strategy modification is always no worse than the best response.

Notions of equilibria.

Definition 5.2.1. For Markov game M, policy π is an ϵ-approximate Nash equilibrium (NE) if it is a product policy and

$$\mathrm{NEGap}^M(\pi) = \max_{i \in [m]} \left(V_i^M(\dagger, \pi_{-i}) - V_i^M(\pi) \right) \leq \epsilon.$$

Learning Nash equilibrium (NE) is neither computationally nor statistically efficient for general-sum normal-form games [Chen et al., 2009], while it is tractable for games with special structures, such as potential games [Monderer and Shapley, 1996] and two-player zero-sum games [Adler, 2013].

Definition 5.2.2. For Markov game M, policy π is an ϵ-approximate coarse correlated equilibrium (CCE) if

$$\mathrm{CCEGap}^M(\pi) = \max_{i \in [m]} \left(V_i^M(\dagger, \pi_{-i}) - V_i^M(\pi) \right) \leq \epsilon.$$

[†]We only consider deterministic strategy modification as the optimal strategy modification can always be deterministic [Jin et al., 2021b].

The only difference between CCE and NE is that CCE is not required to be a product policy. This relaxation allows tractable algorithms for learning CCE.

Definition 5.2.3. For Markov game M, policy π is an ϵ-approximate correlated equilibrium (CE) if

$$\text{CEGap}^M(\pi) = \max_{i\in[m]} \left(\max_{\psi_i \in \Psi_i} V_i^M(\psi_i \diamond \pi) - V_i^M(\pi) \right) \leq \epsilon.$$

Correlated equilibrium generalizes the best response used in CCE to best strategy modification. It is known that each NE is a CE and each CE is a CCE. For conciseness, we use ϵ-EQ to denote ϵ-approximate NE/CE/CCE.

Non-stationarity measure. Here we formalize the non-stationary Markov game. There are T total episodes and at each episode t, the players follow some policy π^t an unknown Markov game M^t. The non-stationarity degree of the environment is measured by the cumulative difference between two consecutive models, defined as follows.

Definition 5.2.4. The non-stationarity degree of Markov games $(M^1, M^2, \cdots, M^T)$ is measured by total variation Δ or number of switches L, which are respectively defined as

$$\Delta = \sum_{t=1}^{T-1} \left\| M^{t+1} - M^t \right\|_1, \quad L = \sum_{t=1}^{T-1} \mathbb{1}[M^t \neq M^{t+1}].$$

Here, the total variation distance between two Markov games is defined as

$$\|M - M'\|_1 := \sum_{h=1}^{H} \left(\left\| \mathbb{P}_h^M - \mathbb{P}_h^{M'} \right\|_1 + \left\| R_h^M - R_h^{M'} \right\|_1 \right).$$

We also define

$$\Delta_{[t_1,t_2]} = \sum_{t=t_1}^{t_2-1} \left\| M^{t+1} - M^t \right\|_1, \quad L_{[t_1,t_2]} = \sum_{t=t_1}^{t_2-1} \mathbb{1}[M^t \neq M^{t+1}].$$

Dynamic regret. We generalize the standard dynamic regret in non-stationary single-agent RL to non-stationary MARL.

Definition 5.2.5. The dynamic equilibrium regret is defined as

$$\text{Regret}(T) = \sum_{t=1}^{T} \text{Gap}^{M^t}(\pi^t),$$

where M^t is the Markov game at episode t, π^t is the policy at episode t and $\text{Gap}(\cdot)$ can be NEGap, CCEGap or CEGap.

A small dynamic regret implies that for most episodes $t \in [T]$, the policy π^t is an approximate equilibrium for model M^t. The same dynamic regret is used in Anagnostides et al. [2023] for matrix games. In the literature, Cardoso et al. [2019] and Zhang et al. [2022a] propose NE-regret and dynamic NE-regret for two-player zero-sum games where the comparator is the best NE value in hindsight and the best dynamic NE value. However, these regret notions can not be generalized to general-sum games as the NE/CE/CCE values become non-unique. Zhang et al. [2022a] also considers duality gap as a performance measure, which coincides with our dynamic regret where Gap is NEGap.

Base algorithms. Our algorithm uses black-box oracles that can learn and test equilibria in (near-)stationary environments. Details of the base algorithms are shown in Appendix.

Assumption 5.2.6. (PAC guarantee for learning equilibrium) We assume that we have access to an oracle LEARN_EQ such that with probability $1-\delta$, in an environment with non-stationarity Δ as defined in Definition 5.2.4, it can output an $\left(\epsilon + c_1^{\Delta}\Delta\right)$-EQ of a game with at most $C_1(\epsilon, \delta)$ samples.

Assumption 5.2.7. (PAC guarantee for testing equilibrium) We assume that we have access to an oracle TEST_EQ such that given a policy π, with probability $1-\delta$, in an environment with non-stationarity Δ as defined in Definition 5.2.4, it outputs False when π is not a $(2\epsilon + c_2^{\Delta}\Delta)$-EQ for all $t = 1, \ldots, C_2(\epsilon, \delta)$ and outputs True when π is an $\left(\epsilon - c_2^{\Delta}\Delta\right)$-EQ for all $t = 1, \ldots, C_2(\epsilon, \delta)$.

There exist various algorithms (see Table 5.1) providing PAC guarantees for learning equilibrium in stationary games, which satisfies Assumption 5.2.6 when non-stationarity degree $\Delta = 0$. We will show that most of these algorithms enjoy an additive error w.r.t. non-stationarity degree Δ in the appendix and discuss how to construct oracles satisfying Assumption 5.2.7 in Section 5.5.1. For simplicity, We will omit δ in $C_1(\epsilon, \delta)$ and $C_2(\epsilon, \delta)$ as they only have polylogarithmic dependence on δ for all the oracle realizations in this work. Furthermore, since the dependence of $C_1(\epsilon), C_2(\epsilon)$ on ϵ are all polynomial, we denote $C_1(\epsilon) = c_1\epsilon^{\alpha}, C_2(\epsilon) = c_2\epsilon^{-2}$. Here c_1, c_2 does not depend on ϵ and α is a constant depending on the oracle algorithm. In Table 5.1, $\alpha = -2$ or $\alpha = -3$, where α is the exponent in $C_1(\epsilon)$.

Types of Games	LEARN_EQ	TEST_EQ	Dynamic Regret
Zero-sum (NE)	$(A+B)\epsilon^{-2}$	$(A+B)\epsilon^{-2}$	$((A+B)\Delta)^{1/4}T^{3/4}$
General-sum (CCE)	$A_{\max}\epsilon^{-2}$	$mA_{\max}\epsilon^{-2}$	$(A_{\max}\Delta)^{1/4}T^{3/4}$
General-sum (CE)	$A_{\max}^2\epsilon^{-2}$	$mA_{\max}^2\epsilon^{-2}$	$(A_{\max}^2\Delta)^{1/4}T^{3/4}$
Potential (NE)	$m^2A_{\max}\epsilon^{-3}$	$mA_{\max}\epsilon^{-2}$	$(m^2A_{\max}\Delta)^{1/5}T^{4/5}$
Congestion (NE)	$m^2F^3\epsilon^{-2}$	$mF^2\epsilon^{-2}$	$(m^3F^4\Delta)^{1/4}T^{3/4}$
Zero-sum Markov (NE)	$H^5S(A+B)\epsilon^{-2}$	$H^3S(A+B)\epsilon^{-2}$	$(H^7S(A+B)\Delta)^{1/4}T^{3/4}$
General-sum Markov (CCE)	$H^6S^2A_{\max}\epsilon^{-2}$	$mH^3SA_{\max}\epsilon^{-2}$	$(H^7S^3A_{\max}\Delta)^{1/4}T^{3/4}$
General-sum Markov (CE)	$H^6S^2A_{\max}^2\epsilon^{-2}$	$mH^3SA_{\max}^2\epsilon^{-2}$	$(H^7S^3A_{\max}^2\Delta)^{1/4}T^{3/4}$
Markov Potential (NE)	$m^2H^4SA_{\max}\epsilon^{-3}$	$mH^3SA_{\max}\epsilon^{-2}$	$(m^2H^6SA_{\max}\Delta)^{1/5}T^{4/5}$

Table 5.1: A, B are the size of action spaces for two-player zero-sum games. X_i and A_i are the number of information sets and actions for player i. $A_{\max} = \max_{j\in[m]} A_j$. S is the size of the state space, H is the horizon of the Markov games and T is the number of episodes. The second and third column is the sample complexity for learning and testing an equilibrium in a stationary game. The last column shows the regret bounds for Algorithm 8.

5.3 Challenges in Non-Stationary Games

In this section, we discuss the major difficulties generalizing single-agent non-stationary algorithms to non-stationary Markov games. There are two major lines of work in the single-agent setting. The first line of work uses online learning techniques to tackle non-stationarity. There exist works generalizing online learning algorithms to the multi-agent setting. However all of them apply only to the full-information setting. In the bandit feedback setting, it is hard to estimate the gradient of the objective function. The other line of work uses explicit tests to determine notable changes of the environment and restart the whole algorithm accordingly. This paper also adpots this paradigm.

The first type of test is to play a sub-optimal action a consecutively to determine whether it has become optimal [Auer et al., 2019b, Chen et al., 2019]. For simplicity, let us think of learning NE in the environment with abrupt changes (switching number as the non-stationary measure). In order to assure a has not become a new optimal action, one needs to spend $1/D^2$ steps to play a and secure its value up to D confidence bound where D is the suboptimality. The regret incurred in this testing process is $D \cdot 1/D^2 = 1/D$. In the multi-agent setting, if one wants to repeat the process by testing (a_i', a_{-i}) to assure $\boldsymbol{a}$ is a NE, the timesteps needed is still $1/D^2$ where D is the empirical reward difference of (a_i', a_{-i}) and $\boldsymbol{a}$. However, the gap of (a_i', a_{-i}) depends on its own unilateral deviations, which can be $O(1)$ in general. Hence the regret incurred can be $1/D^2$, sabotaging the test process (example in Figure 5.1).

The second type of test restarts the learning algorithm for a small amount of time and checks for abnormality in the replay [Wei and Luo, 2021]. In the multi-agent setting, since equilibrium is not unique in all games, different runs of the same algorithm can converge to different equilibria even in a stationary environment. Hence test of this type fails to detect abnormality in the base algorithm.

Another method worth mentioning was invented in Garivier and Moulines [2011]. This method proposes to forget old history through putting a discount weight on old feedback or imposing a sliding window based on which we calculate the empirical estimate of value of actions. There is no obvious obstacle in generalizing it to the multi-agent setting but

	a	b
a	1	0
b	0	ε

Figure 5.1: Consider a two-player cooperative game. Both players have access to action space $\{a, b\}$ and the corresponding rewards are shown in the picture. Assume we have found NE (b, b). If we want to make sure (a, b) has not become a best response for player 1, we have to play (a, b) for $1/\varepsilon^2$ times. However the regret of (a, b) is 1, so this process induces $1/\varepsilon^2$ regret.

it is hard to derive a parameter-free version. Cheung et al. [2020] uses the Bandit-Over-RL technique to get a parameter-free version for the single-agent setting based on the sliding-window idea. However, the Bandit-Over-RL technique does not generalize to the multi-agent setting as the learning objectives are totally different. A more detailed version of the challanges mentioned is presented in the Appendix D.1.

5.4 Warm-Up: Known Non-Stationary Budget

We first present an algorithm for MARL against non-stationary environments with known non-stationarity budget to serve as a starting point.

Algorithm 8 Restarted Explore-then-Commit for Non-stationary MARL

1: **Input:** number of episodes T; non-stationarity budget Δ; confidence level δ; parameter T_1
2: **while** episode T is not reached **do**
3: Run LEARN_EQ with accuracy ϵ and confidence level δ, and receive the output π.
4: Execute π for T_1 episodes.
5: **end while**

Initially, the algorithm starts a LEARN_EQ algorithm, intending to learn an ϵ-EQ policy π. After that, it commits to π for T_1 episodes. Subsequently, the algorithm repeats this

learn-then-commit pattern until the end. The restart mechanism guarantees that the non-stationarity in the environment can at most affect T_1 episodes. By carefully tuning T_1, we can achieve a sublinear regret. This algorithm admits a performance guarantee as follows.

Proposition 5.4.1. *With probability $1 - T\delta$, the regret of Algorithm 8 satisfies*

$$\text{Regret}(T) \leq \frac{4TC_1(\epsilon)}{T_1} + T\epsilon + 2\max\left\{c_1^{\Delta}, H\right\}T_1\Delta.$$

Remark 5.4.2. Let us look at the meaning of each term in this bound. The first term comes from all LEARN_EQ. The second and third terms come from committing to the learned policy.

Corollary 5.4.3. *With probability $1 - T\delta$, the regret of Algorithm 8 satisfies*

$$\text{Regret}(T) \leq \begin{cases} 13\left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}\right)^{1/4} T^{3/4}, & \alpha = -2, \\ 13\left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}\right)^{1/5} T^{4/5}, & \alpha = -3, \end{cases}$$

by setting

$$T_1 = \left\lceil \sqrt{\frac{TC_1(\epsilon)}{\max\left\{c_1^{\Delta}, H\right\}\Delta}} \right\rceil, \quad \epsilon = \begin{cases} \left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}/T\right)^{1/4}, & \alpha = -2, \\ \left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}/T\right)^{1/5}, & \alpha = -3. \end{cases}$$

Example 4. As a concrete example, for learning CCE in general-sum Markov games, Algorithm 8 achieves $O\left(A_{\max}^{1/4}\Delta^{1/4}T^{3/4}\right)$ regret. We can see that this algorithm breaks the curse of multi-agents (dependence on the number of players) which is a nice property inherited from the base algorithm. In addition, as long as the base algorithm is decentralized, Algorithm 8 will also be decentralized.

5.5 Unknown Non-Stationarity Budget

In this section, we generalize Algorithm 8 to a parameter-free version, which achieves a similar regret bound without the knowledge of the non-stationarity budget and the time horizon T. If the non-stationarity budget is unknown, we cannot determine the appropriate rate to restart in advance as in Algorithm 8. Hence, we use multi-scale testing to monitor the performance of the committed policy and restart adaptively.

5.5.1 *Black-box Algorithms for Testing Equilibria*

In this section, we present the construction of the testing algorithms TEST_EQ that satisfies Assumption 5.2.7 by a black-box reduction to single-agent algorithms, which is able to test whether a policy is an equilibrium in a (near-)stationary game. We make the following assumption on the single-agent learning oracle.

Assumption 5.5.1. (PAC guarantee for single-agent RL) We assume that we have access to an oracle LEARN_OP such that with probability $1-\delta$, in a single-agent environment with non-stationarity Δ, it can output an $\left(\epsilon + c_3^{\Delta}\Delta\right)$-optimal policy with $C_3(\epsilon, \delta)$ samples.

The construction of TEST_EQ is described in Protocol 3. We first illustrate how Protocol 3 test NE/CCE in a stationary environment. Note that here we only consider Markov policies and the best response to a Markov policy is the optimal policy in the induced single-agent MDP. First, we sample $\widetilde{O}\left(\epsilon^{-2}\right)$ trajectories following π to get an estimate of $V_i(\pi)$ for all i up to an error bound of $\epsilon/6$ by standard concentration inequalities. Then, for each player i, we run LEARN_OP and by Assumption 5.5.1, π_i' is an $\epsilon/6$-optimal policy in the MDP induced by other players following π_{-i}. In other words, π_i' is an $\epsilon/6$-best response to π_{-i}. After that we run (π_i', π_{-i}) for $\widetilde{O}\left(\epsilon^{-2}\right)$ episodes and estimate the policy value $\widehat{V}_i(\pi_i', \pi_{-i})$ for players i up to $\epsilon/6$ error bound. Finally the algorithm decides the output according to the empirical estimate of the gap. If the policy is not a 2ϵ-EQ, with high probability the empirical gap is larger than $3\epsilon/2$, which leads to a False output. Meanwhile, if the policy is an ϵ-EQ , with high probability the empirical gap is smaller than $3\epsilon/2$, which leads to a True output.

To test a CE, we need to learn the best strategy modification in the induced MDP. While there are many algorithms in prior works that can serve as LEARN_OP, no algorithm is designed for learning the best strategy modification as far as we know. Interestingly, by constructing an MDP with an extended state space, we can reduce learning the best strategy modification to learning the optimal policy in the new MDP. Specifically, here we design an MDP M' such that learning the best strategy modification with random recommendation policy π in MDP $M = (\mathcal{S}, \mathcal{A}, P, r, H)$ is equivalent to learning the optimal policy in M', where the randomness in π could be correlated with the transition. In M', the state space

is $\mathcal{S}' = \mathcal{S} \times \mathcal{A}$, the action space is $\mathcal{A}$, the transition is $P'_h((s_{h+1}, b_{h+1}) \mid (s_h, b_h), a_h) = \mathbb{P}_h(s_{h+1} \mid s_h, \pi_h(s_h) = b_h, a_h) \cdot \pi_{h+1}(b_{h+1} \mid s_{h+1})$ and the reward is $r'_h(\cdot \mid (s_h, b_h), a_h) = r_h(\cdot \mid s_h, \pi_h(s_h) = b_h, a_h)$. The following proposition shows that learning the best strategy modification to recommendation policy π in MDP M is equivalent to learning the optimal policy in MDP M'.

Proposition 5.5.2. *Suppose MDP M' is induced by MDP M and recommendation policy π. Then the optimal policy in MDP M' corresponds to a best strategy modification to recommendation policy π in MDP M.*

Note that the state space in M' is enlarged by a factor of A, which means the sample complexity for testing CE is A times larger than CCE, which coincides with the fact that the minimax swap regret is $\sqrt{A}$ times larger than the minimax external regret [Ito, 2020].

Proposition 5.5.3. *As long as* LEARN_OP *satisfies Assumption 5.5.1, Protocol 3 satisfies Assumption 5.2.7.*

5.5.2 Multi-scale Test Scheduling

In this section, we introduce how to schedule TEST_EQ during the committing phase. The scheduling is motivated by MALG in Wei and Luo [2021], with modifications to the multi-agent setting.

We consider a block with length 2^n for some integer n. The block starts with a LEARN_EQ with $\epsilon = 2^{-n/4}$ and is followed by the committing phase. During the committing phase, TEST_EQ starts randomly for different gaps with different probabilities at each step. That is, we intend to test larger changes more quickly by testing for them more frequently (by setting the probability higher) so that the detection is adaptive to the severity of changes. Denote the episode index in this block by τ. In the committing phase, if τ is an integer multiple of 2^{c+q} for some $q \in \{0, 1, \cdots, Q\}$, with probability $p(q) = 1/(\epsilon(q)2^{n/2})$ we start a test for gap $\epsilon(q) = \sqrt{c_2/2^q}$ so that the length of test is 2^q, where the value of c_2 comes from the testing oracle and Q, c are defined as

$$Q = \min\left\{\left\lfloor \log_2\left(c_2 2^{n/2-1}\right)\right\rfloor, n-c\right\}_+, c = \left\lceil 1 + \log_2 \max\left\{5\sqrt{c_2}, 2\log\frac{1}{\delta}\right\}\right\rceil.$$

Protocol 3 TEST_EQ

Input: Joint Markov policy π, failure probability δ, tolerance ϵ.

Run π for $\widetilde{O}(\epsilon^{-2})$ episodes and estimate the policy value $\widehat{V}_i(\pi)$ with confidence level $\epsilon/6$ for all players $i \in [m]$.

for $i = 1, 2, \ldots, m$ **do**

    Let players $[m]/\{i\}$ follow π_{-i} and player i run LEARN_OP with δ and $\epsilon/6$. Receive best reponse policy π_i' or best strategy modification $\psi_i \diamond \pi$ for (NE,CCE) or CE.

    Run (π_i', π_{-i}) or $\psi_i \diamond \pi$ for $\widetilde{O}(\epsilon^{-2})$ episodes and estimate the best response value $\widehat{V}_i(\pi_i', \pi_{-i})$ or the best strategy modification value $\widehat{V}_i(\psi_i \diamond \pi)$ with confidence level $\epsilon/6$ for players i.

end for

if $\max_{i\in[m]} \left(\widehat{V}_i(\pi_i', \pi_{-i}) - \widehat{V}_i(\pi)\right) \leq 3\epsilon/2$ or $\max_{i\in[m]} \left(\widehat{V}_i(\psi_i \diamond \pi) - \widehat{V}_i(\pi)\right) \leq 3\epsilon/2$ **then**

    return True

else

    return False

end if

The gaps we intend to test are approximately $\left\{\sqrt{2}\epsilon, 2\epsilon, 2\sqrt{2}\epsilon, \cdots\right\}$. It is possible that TEST_EQ for different $\epsilon(q)$ are overlapped. In this case, we prioritize the running of TEST_EQ for larger $\epsilon(q)$ and pause those for smaller $\epsilon(q)$. After the shorter TEST_EQ ends, we resume the longer ones until they are completed. In addition, if a TEST_EQ for $\epsilon(q)$ spans for more than 2^{c+q} episodes, it is aborted. To better illustrate the scheduling, we construct an example shown in Figure 5.2. It can be proved that with high probability no TEST_EQ is aborted (Lemma D.3.2), i.e. the 2^c multiplication in length reserves enough space for all TEST_EQ. Note that the original MALG (Wei and Luo [2021]) does not work here because the length of each scheduled TEST_EQ can be reduced greatly and there is no guarantee how a TEST_EQ with reduced length would work. The scheduling is formally stated in Protocol 5.

Lemma 5.5.4. *With probability* $1 - 3QT\delta$*, the regret inside this block*

$$\text{Regret} = \widetilde{O}\left(2^{3n/4} + c_2 \min\left\{2^{2n/3}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/3}, 2^{5n/8}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/2}\right\} + 2^{n/2}c_2^{3/2} + 2^{-\alpha n/4}c_1\right) \tag{5.1}$$

Remark 5.5.5. The common way to bound the regret with total variation is to divide the block into several near-stationary intervals $[C_1(\epsilon)+1, 2^n] = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_K$. In each interval the near-stationarity ensure all Test_EQ to work properly and hence the regret is bounded. This is because if the regret is to big for a long time Test_EQ would detect it. After that we bound K and finally bound the regret of a block using Hölder's inequality. While prior works [Chen et al., 2019] partition the intervals according to $\Delta_{\mathcal{I}_k} = O\left(|\mathcal{I}_k|^{-1/2}\right)$, we set $\Delta_{\mathcal{I}_k} = O\left(\max\left\{|I_k|^{-1/2}, 2^{-n/4}\right\}\right)$. This greatly change the subsequent calculations and makes the regret better in our case, please refer to the appendix for more details.

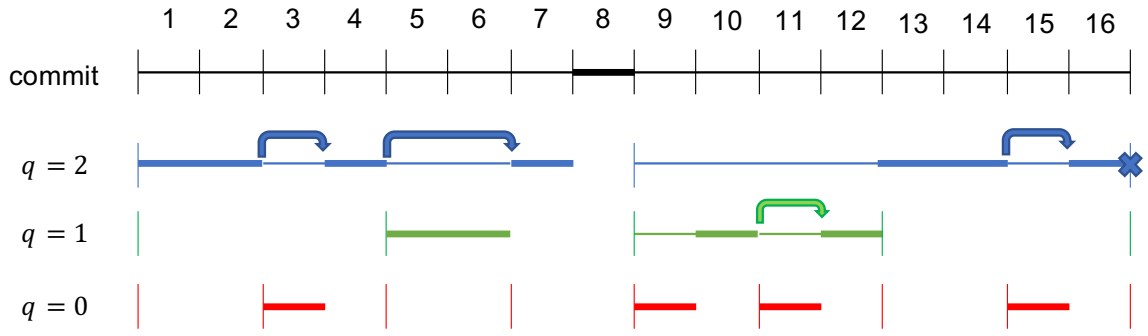


Figure 5.2: This is an example of the scheduling for committing phase with length $16, Q = 2, c = 1$. The horizontal lines represent the scheduled Test_EQ except for the black line on the top which represent the time horizon. Different colors represent Test_EQ for different $\epsilon(q)$. The bold parts of a line represent the active parts and the other parts are the paused parts. The colored vertical lines represent the possible starting points of Test_EQ for each level. The cross at the last episode indicates the Test_EQ is aborted because it spans $2^{c+q} = 8$ episodes but has only run $3 < 2^q$ episodes. The bold part of the black line indicates that at this episode we commit to the learned policy and there is no Test_EQ running.

5.5.3 Main Algorithm

The main algorithm consists of blocks with doubling lengths. The first block is the shortest block that can accomodate a whole LEARN_EQ in it. The doubling structure is not only important to making the algorithm parameter free of Δ, but also to that of T (see Appendix for more details). The performance guarantee of this algorithm is stated in Theorem 5.5.6. For simplicity, let $\widetilde{\Delta}_{\mathcal{J}} = c_2^{\Delta}\Delta_{\mathcal{J}}$ and $\check{\Delta}_{\mathcal{J}} = \max\left\{c_1^{\Delta}, c_2^{\Delta}\right\}\Delta_{\mathcal{J}}$

Algorithm 9 Multi-scale Testing for Non-stationary MARL

1: **Input:** failure probability δ.
2: $N \leftarrow \min\left\{n \mid 2^n \geq C_1(2^{n/2})\right\}$
3: **for** $n = N, N+1, \cdots$ **do**
4: Schedule a block sized 2^n according to Section 5.5.2.
5: Run LEARN_EQ with accuracy $\epsilon = 2^{-n/4}$ and receive π.
6: Run the committing phase according to the schedule. If any TEST_EQ returns False, go to Line 2 immediately.
7: **end for**

Theorem 5.5.6. *With probability $1 - 3QT\delta$, the regret of Algorithm 9 is*

$$\text{Regret}(T) = \begin{cases} \widetilde{O}\left(\check{\Delta}^{1/5}T^{4/5} + c_2 \min\left\{\widetilde{\Delta}^{1/3}T^{2/3}, \widetilde{\Delta}^{1/2}T^{5/8}\right\} + \left(c_1 + c_2^{3/2}\right)\check{\Delta}^{2/5}T^{3/5}\right) & \alpha = -2 \\ \widetilde{O}\left(c_1\check{\Delta}^{1/5}T^{4/5} + c_2 \min\left\{\widetilde{\Delta}^{1/3}T^{2/3}, \widetilde{\Delta}^{1/2}T^{5/8}\right\} + c_2^{3/2}\check{\Delta}^{2/5}T^{3/5}\right) & \alpha = -3 \end{cases}$$

Remark 5.5.7. The main idea of the proof is as follows. The restarts divide the whole time horizon into consecutive segments $[1, T] = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \cdots \cup \mathcal{J}_J$. In each segment $\mathcal{J}_j$ between restarts, the regret can be bounded by adding up Formula 5.1 for all blocks as

$$\text{Regret}\left(\mathcal{J}_j\right) = \widetilde{O}\left(|\mathcal{J}_j|^{3/4} + c_2 \min\left\{|\mathcal{J}_j|^{2/3}\,\widetilde{\Delta}_{\mathcal{J}_j}^{1/3}, |\mathcal{J}_j|^{5/8}\,\widetilde{\Delta}_{\mathcal{J}_j}^{1/2}\right\} + c_2^{3/2}\,|\mathcal{J}_j|^{1/2} + c_1\,|\mathcal{J}_j|^{-\alpha/4}\right).$$

It can be proved that the number of segments is bounded by $J = O\left(T^{1/5}\check{\Delta}^{4/5}\right)$. Using Hölder's inequality, we get the conclusion.

Meanwhile, the following theorem can be obtained if we only consider L, the number of switches.

Theorem 5.5.8. *With probability* $1 - 3QT\delta$*, the regret of Algorithm 9 is*

$$\text{Regret}(T) = \begin{cases} \widetilde{O}\left(L^{1/4}T^{3/4} + \left(c_1 + c_2^{3/2}\right) L^{1/2}T^{1/2}\right) & \alpha = -2 \\ \widetilde{O}\left(c_1 L^{1/4}T^{3/4} + c_2^{3/2} L^{1/2}T^{1/2}\right) & \alpha = -3 \end{cases}$$

Remark 5.5.9. Algorithm 9 breaks the curse of multi-agent as long as the base algorithms do. If the base algorithm is decentralized, all players are informed to restart when a change is detected and no further communication is needed. In this sense very few extra communications are needed in Algorithm 9.

5.6 Conclusions

In this work, we propose black-box reduction approaches for learning the equilibria in non-stationary multi-agent reinforcement learning, both with and without knowledge of parameters. These algorithms offer favorable performance guarantees in terms of the non-stationarity measure, while preserving the advantages of breaking curse of multi-agent and decentralization found in the base algorithms. We conclude this paper by posing two open questions. Firstly, we assume that all oracles with PAC guarantees may have regret as large as $O(1)$ in the proofs. However, it remains unknown how to design algorithms such that the oracles themselves are also no-regret, which would further minimize the regret in learning. Secondly, the lower bound of regret for learning in non-stationary multi-agent systems is currently unknown, despite extensive investigations into lower bounds for single-agent systems [Besbes et al., 2014, Garivier and Moulines, 2011].

Part III

PROVABLY EFFICIENT LEARNING IN CONGESTION GAMES

Chapter 6

LEARNING EQUILIBRIUM IN CONGESTION GAMES

This chapter is based on Cui et al. [2022], with Zhihan Xiong, Maryam Fazel and Simon S. Du.

6.1 Introduction

Nash equilibrium (NE) is a widely adopted concept in game theory community, used to describe the behavior of multi-agent systems with selfish players [Roughgarden, 2010]. At the Nash equilibrium, no player has the incentive to change its own strategy unilaterally, which implies it is a steady state of the game dynamics. For a general-sum game, computing the Nash equilibrium is PPAD-hard [Daskalakis, 2013] and the query complexity is exponential in the number of players [Rubinstein, 2016]. To help address these issues, a natural approach is to consider games with special structures. In this paper, we focus on congestion games.

Congestion games are general-sum games with *facilities* (resources) shared among non-cooperative players [Rosenthal, 1973]. During the game, each player will decide what combination of facilities to utilize, and popular facilities will become congested, which results in a possibly higher cost on each user. One example of congestion game is the routing game [Fotakis et al., 2002], where each player needs to travel from a given starting point to a destination point through some shared routes. These routes are represented as a traffic graph and the facilities are the edges. Each player will decide her path to go, and the more players use the same edge, the longer the edge travel time will be. Congestion games also have wide applications in electrical grids [Ibars et al., 2010], internet routing [Al-Kashoash et al., 2017] and rate allocation [Johari and Tsitsiklis, 2004]. In many real-world scenarios, players can only have (semi-)bandit feedback, i.e., players know only the payoff of the facilities they choose. This kind of learning under uncertainty has been widely studied in bandits and in

reinforcement learning for the single-agent setting, while theoretical understanding for the multi-agent case is still largely missing.

There are two types of algorithms in multi-agent systems, namely centralized algorithms and decentralized algorithms. For centralized algorithms, there exists a central authority that can control and receive feedback from all players in the game. As we have global coordination, centralized algorithms usually have favorable performance. On the other hand, such a central authority may not always be available in practice, and thus people turn to decentralized algorithms, i.e., each player makes decisions individually and can only observe her own feedback. However, decentralized algorithms are vulnerable to *nonstationarity* because each player is making decisions in a nonstationary environment as others' strategies are changing [Zhang et al., 2021a]. In this paper, we will study both centralized and decentralized algorithms in congestion games with bandit feedback, and we will provide motivating scenarios for both algorithms in Section 6.1.2.

The main challenge in designing algorithms for m-player congestion games with bandit feedback is the curse of exponential action set, i.e., the number of actions can be exponential in the number of facilities F because every subset of facilities can be an action. As a result, an efficient algorithm should have sample complexity polynomial in m and F and has no dependence on the size of the action space. One closely related type of general-sum game is the potential game, in which each individual's payoff changes, resulting from strategy modification, can be quantified by a common potential function. It is well-known that all congestion games are potential games, and each potential game has an equivalent congestion game formulation [Monderer and Shapley, 1996]. However, existing algorithms designed for potential games all have sample complexity scaling at least linearly in the number of actions [Leonardos et al., 2021, Ding et al., 2022], which is inefficient for congestion games. This motivates the following question:

Can we design provably sample-efficient centralized and decentralized learning algorithms for congestion games with bandit feedback?

We provide an affirmative answer to this question. To be precise, we use Nash-regret minimization (formally defined in Section 6.3) as our objective for learning in congestion

games. This regret-like objective commonly appears in the literature of online learning and reinforcement learning [Orabona, 2019, Ding et al., 2022, Liu et al., 2021a], which focuses on finite-time analysis and accumulative rewards throughout the learning process instead of the asymptotic behavior. In general, a sublinear Nash regret implies a best-iterate convergence, meaning that the algorithm has reached the approximate Nash equilibrium at least once, while the converse does not hold.

We highlight our contributions below and compare our results with previous algorithms in Table 6.1. Our algorithms are shaded and we prove sublinear Nash regrets for all of them. In Table 6.1, sample complexity refers to the number of samples required to reach best-iterate convergence to an ϵ-approximate Nash equilibrium and the results are obtained by standard online-to-batch conversion as in Section 3.1 of [Jin et al., 2018].

6.1.1 Main Novelties and Contributions

1. Centralized algorithm for congestion game. We adapt the principle of optimism in the face of uncertainty in stochastic bandits to ensure sufficient exploration in congestion games. We begin with congestion games with semi-bandit feedback, in which each player can observe the reward of every facility in the action. Instead of estimating the action reward as in stochastic multi-armed bandits, we estimate the facility rewards directly, which *removes the dependence on the size of action space*. Furthermore, we consider congestion games with bandit feedback, in which each player can only observe the overall reward. In this setting, we borrow ideas from linear bandits to estimate the reward function and analyze the algorithm. The algorithm is provably sample efficient in both cases.

2. Decentralized algorithm for congestion game. Our decentralized algorithm is a Frank-Wolfe method with exploration, in which each player only observes her own actions and rewards. To efficiently explore in the congestion game, we utilize G-optimal design allocation for bandit feedback and a specific distribution for semi-bandit feedback. As a result, the sample complexity does not depend on the number of actions. In addition, the L_1 smoothness parameter of the potential function does not depend on the number of actions, which is exploited by the Frank-Wolfe method. With the help of these two specific

Table 6.1: Comparison of algorithms for congestion games in terms of sample complexity and Nash regret, where "IPPG" stands for "independent projected policy gradient", "IPGA" stands for "independent policy gradient ascent", "I" represents the setting of semi-bandit feedback and "II" represents the setting of bandit feedback. Bandit feedback is assumed for algorithms from previous work. Here, A_i is the size of player i's action space, m is the number of players, $A_{\max} = \max_{i\in[m]} A_i$, F is the number of facilities and T is the number of samples collected. Our algorithms are shaded.

Algorithms	Sample complexity	Nash regret	Decentralized
Nash-VI [Liu et al., 2021a]	$(\prod_{i=1}^m A_i)F/\epsilon^2$	$\sqrt{(\prod_{i=1}^m A_i)FT}$	No
V-learning [Jin et al., 2021b]	$A_{\max}F/\epsilon^2$ (CCE)	NA	Yes
IPPG [Leonardos et al., 2021]	$A_{\max}mF/\epsilon^6$	NA	Yes
IPGA [Ding et al., 2022]	$A_{\max}^2m^3F^5/\epsilon^5$	$mF^{4/3}\sqrt{A_{\max}}T^{4/5}$	Yes
Nash-UCB I	mF^2/ϵ^2	$F\sqrt{mT}$	No
Nash-UCB II	m^2F^3/ϵ^2	$mF^{3/2}\sqrt{T}$	No
Frank-Wolfe with Exploration I	$m^{12}F^9/\epsilon^6$	$m^2F^{3/2}T^{5/6}$	Yes
Frank-Wolfe with Exploration II	$m^{12}F^{12}/\epsilon^6$	$m^2F^2T^{5/6}$	Yes

algorithmic designs for congestion games, we give the first decentralized algorithm for both semi-bandit feedback and bandit feedback that has no dependence on the size of the action space in congestion games.

3. Centralized algorithm for independent Markov congestion game. We extend the formulation of congestion game into a Markov setting and propose the independent Markov congestion game (IMCG), in which each facility has its own internal state and state transition happens independently among all the facilities. In Section 6.1.2, we give some examples that fit in this model. By utilizing techniques from factored MDPs, we extend our centralized algorithms for congestion games to efficiently solve IMCGs, with both semi-bandit and bandit feedback.

6.1.2 Motivating Examples

We provide an exmple here to motivate our proposed models. See Section 6.3 for the formal definition of (semi-)bandit feedback and (Markov) congestion games and Appendix E.1 for additional examples.

Example 5 (**Routing Games)**. For a routing game, there are multiple players in a traffic graph travelling from starting points to destination points, and the facilities are the edges (roads). The cost of each edge is the waiting time, which depends on the number of players using that edge.

• **Centralized algorithm for routing games:** Imagine each player is using Google Maps to navigate. Then Google Maps can serve as a center that knows the starting points and the destination points, as well as the real-time feedback of the waiting time on each edge of all the players. Google Maps itself also has the incentive to assign paths according to the Nash equilibrium strategy as then each player will find out that deviating from the navigation has no benefit and thus sticks to the app.

• **Decentralized algorithm for routing games:** Consider the case where players are still using Google Maps but due to privacy concerns or limited bandwidth, they only use the offline version, which has access only to the information of each single user. Then Google Maps needs to use decentralized algorithms so that it can still assign Nash equilibrium strategy to each user after repeated plays.

• **Markov routing games:** For Markov routing games, the time cost on each edge will change between different timesteps, which is a more accurate model of the real-world. For instance, some roads are prone to car accidents, which will result in an increasing cost on the next timestep, and the chance of accidents also depends on the number of players using that edge currently. This is modeled by the Markovian facility state transition in independent Markov congestion games.

6.2 Related Work

Potential Games. Potential games are general-sum games that admit a common potential function to quantify the changes in individual's payoff [Monderer and Shapley, 1996].

Algorithmic game theory community has studied how different dynamics converge to the Nash equilibium, e.g., best response dynamics [Durand, 2018, Swenson et al., 2018a] and no-regret dynamics [Heliou et al., 2017b, Cheung and Piliouras, 2020], while usually they provide only asymptotic convergence, with either full information setting or bandit feedback setting. Recently, reinforcement learning community studied Markov potential games with bandit feedback, which can be applied to standard potential games. See the Markov Games part below for more details.

Congestion Games. Congestion games are developed in the seminal work [Rosenthal, 1973], and later Monderer and Shapley [1996] builds a close connection between congestion games and potential games. Congestion games are divided into atomic and non-atomic congestion games depending on whether each player is separable. Many papers consider non-atomic congestion games with non-decreasing cost function, which implies a convex potential function [Roughgarden and Tardos, 2004]. We consider the more difficult atomic congestion game where the potential function can be non-convex. For online non-atomic case, [Krichene et al., 2015] considers partial information setting while they provide convergence in the sense of Cesaro means. [Kleinberg et al., 2009, Krichene et al., 2014] show that some no-regret online learning algorithms asymptotically converges to Nash equilibrium. [Chen and Lu, 2015, 2016] are two closely related works that consider bandit feedback in atomic congestion games and provide non-asymptotic convergence. However, they still assume a convex potential function and the sample complexity has exponential dependence on the number of facilities, which is far from ideal.

Markov Games. Markov games are widely studied since the seminal work [Shapley, 1953]. Recently, the topic has received much attention due to advances in reinforcement learning theory. Liu et al. [2021a] provides a centralized algorithm for learning the Nash equilibrium in general-sum Markov games, and [Jin et al., 2021b, Song et al., 2021a] provide decentralized algorithms for learning the (coarse) correlated equilibrium. One closely related line of research is on Markov potential games [Leonardos et al., 2021, Zhang et al., 2021c, Fox et al., 2021, Cen et al., 2022a, Ding et al., 2022]. However, applying their algorithms to congestion games leads to explicit dependence on the number of actions, which would be exponentially worse than our algorithms. See Table 6.1 for comparisons. Our independent

Markov congestion game is motivated by the state-based potential games studied in Marden [2012] and Macua et al. [2018], and its transition kernel is closely related to the factored MDPs, for which single agent algorithms are studied in [Osband and Van Roy, 2014, Chen et al., 2020, Xu and Tewari, 2020, Tian et al., 2020, Rosenberg and Mansour, 2021].

Learning in Games. Different from our paper, learning in games in traditional literature of game theory mainly considers players' asymptotic behavior [Leslie and Collins, 2005, Cominetti et al., 2010, Coucheney et al., 2015]. In early literature, Leslie [2004] investigates actor-critic learning and Q-learning algorithms in games with bandit feedback and their connection to best-response dynamics. Leslie and Collins [2005] proposes individual Q-learning algorithm and shows that it converges to the NE almost surely in two-player zero-sum game and Leslie and Collins [2006] studies learning the NE from the perspective of a fictitious play-like process. Later, Cominetti et al. [2010] considers payoff-based learning rules and shows convergence to NE in traffic games, while another payoff-based learning model for continuous games is developed in Bervoets et al. [2020]. Coucheney et al. [2015] derives a new penalty-regulated dynamics and proposes a corresponding learning algorithms that converges to NE in potential games with bandit feedback. Bravo et al. [2018] proposes that in monotone games with bandit feedback, as long as all players are using some no-regret learning algorithm, the dynamics will converge to the NE, and an improved analysis of the same derivative-free algorithm is given in Drusvyatskiy et al. [2022]. In contrast, our learning objective focuses on finite-time cumulative rewards, which is more widely used in current multi-agent reinforcement learning literature [Ding et al., 2022, Liu et al., 2021a].

6.3 Preliminaries

General-sum Matrix Games. We consider the model of general-sum matrix games, defined by the tuple $\mathcal{G} = (\{\mathcal{A}_i\}_{i=1}^m, R)$, where m is the number of players, $\mathcal{A}_i$ is the action space of player i and $R(\cdot|\mathbf{a})$ is the reward distribution on $[0, r_{\max}]^m$ with mean $\mathbf{r}(\mathbf{a})$. Let $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ be the whole action space and denote an element as $\mathbf{a} = (a_1, \ldots, a_m) \in \mathcal{A}$. After all players take actions $\mathbf{a} \in \mathcal{A}$, a reward vector is sampled $\mathbf{r} \sim R(\cdot|\mathbf{a})$ and player i will receive reward $r_i \in [0, r_{\max}]$ with mean $r_i(\mathbf{a})$. Each player's objective is to maximize her own reward.

A general policy π is defined as a vector in $\Delta(\mathcal{A})$, the probability simplex over the action space $\mathcal{A}$. A product policy $\pi = (\pi_1, \ldots, \pi_m)$ is defined as a tuple in $\Delta(\mathcal{A}_1) \times \cdots \times \Delta(\mathcal{A}_m)$, in which $\mathbf{a} = (a_1, \ldots, a_m) \sim \pi$ represents $a_i \overset{\text{i.i.d.}}{\sim} \pi_i$. The value of policy π for player i is $V_i^{\pi} = \mathbb{E}_{\mathbf{a}\sim\pi}[r_i(\mathbf{a})]$.

Nash Equilibrium and Nash Regret. Given a general policy π, let π_{-i} be the marginal joint policy of players $1, \ldots, i-1, i+1, \ldots, m$. Then, the best response of player i under policy π is $\pi_i^{\dagger} = \text{argmax}_{\mu\in\Delta(\mathcal{A}_i)} V_i^{\mu,\pi_{-i}}$ and the corresponding value is $V_i^{\dagger,\pi_{-i}} := V_i^{\pi_i^{\dagger},\pi_{-i}}$. Our goal is to find the approximate Nash equilibrium of the matrix game, which is defined below.

Definition 6.3.1. A product policy π is an *ϵ-approximate Nash equilibrium* if $\max_i(V_i^{\dagger,\pi_{-i}} - V_i^{\pi}) \leq \epsilon$.

An ϵ-approximate Nash equilibrium can be obtained by achieving a sublinear Nash regret, which is defined below. See Section 3 in Ding et al. [2022] for a more detailed discussion.

Definition 6.3.2. With π^k being the policy at k-th episode, the *Nash regret* after K episodes is define as

$$\text{Nash-Regret}(K) = \sum_{k=1}^{K} \max_{i\in[m]} \left(V_i^{\dagger,\pi_{-i}^k} - V_i^{\pi^k} \right).$$

Remark 6.3.3. Here, if we replace $\max_{i\in[m]}$ by $\sum_{i=1}^m$ in the definition of Nash regret, the single-step Nash regret at episode k will become the Nikaido-Isoda (NI) function evaluated at π^k, which is a popular objective for equilibrium computation [Nikaidô and Isoda, 1955, Raghunathan et al., 2019]. Replacing $\max_{i\in[m]}$ by $\sum_{i=1}^m$ will multiply our regret bounds by a factor of m, while our conclusion will not be affected.

Potential Games. A potential game is a general-sum game such that there exists a potential function $\Phi : \Delta(\mathcal{A}) \to [0, \Phi_{\max}]$ such that for any player $i \in [m]$ and policies π_i, π_i', π_{-i}, it satisfies

$$\Phi(\pi_i, \pi_{-i}) - \Phi(\pi_i', \pi_{-i}) = V_i^{\pi_i,\pi_{-i}} - V_i^{\pi_i',\pi_{-i}}.$$

We can immediately see that a policy that maximizes the potential function is a Nash equilibrium.

Congestion Games. A congestion game is defined by $\mathcal{G} = (\mathcal{F}, \{\mathcal{A}_i\}_{i=1}^m, \left\{R^f\right\}_{f\in\mathcal{F}})$, where $\mathcal{F} = [F]$ is called the facility set and $R^f(\cdot|n) \in [0,1]$ is the reward distribution for facility f with mean $r^f(n)$, where $n \in [m]$. Each action $a_i \in \mathcal{A}_i$ is a subset of $\mathcal{F}$ (i.e., $a_i \subseteq \mathcal{F}$). Suppose the joint action chosen by all the players is $\mathbf{a} \in \mathcal{A}$, then a random reward is sampled $r^f \sim R^f(\cdot|n^f(\mathbf{a}))$ for each facility f, where $n^f(\mathbf{a}) = \sum_{i=1}^m \mathbb{1}\{f \in a_i\}$ is the number of players using facility f. The reward collected by player i is $r_i = \sum_{f\in a_i} r^f$ with mean $r_i(\mathbf{a}) = \sum_{f\in a_i} r^f(n^f(\mathbf{a})) \in [0, F]$.

Connection to Potential Games [Monderer and Shapley, 1996]. As a special class of potential game, all congestion games have the potential function: $\Phi(\mathbf{a}) = \sum_{f\in\mathcal{F}} \sum_{i=1}^{n^f(\mathbf{a})} r^f(i)$. To see this, we can easily verify that $\Phi(a_i, a_{-i}) - \Phi(a_i', a_{-i}) = r_i(a_i, a_{-i}) - r_i(a_i', a_{-i})$ holds. Then, by defining $\Phi(\pi) = \mathbb{E}_{\mathbf{a}\sim\pi}[\Phi(\mathbf{a})]$, we can have $\Phi(\pi_i, \pi_{-i}) - \Phi(\pi_i', \pi_{-i}) = V_i^{\pi_i,\pi_{-i}} - V_i^{\pi_i',\pi_{-i}}$.

Types of feedback. There are in general two types of reward feedback for the congestion games, semi-bandit feedback and bandit feedback, both of which are reasonable under different scenarios. In semi-bandit feedback, after taking the action, player i will receive reward information r^f for each $f \in a_i$; in bandit feedback, after taking the action, player i will only receive the reward $r_i = \sum_{f\in a_i} r^f$ with no knowledge about each r^f. In this paper, we will address both of them, with more focus on the bandit feedback, which can be directly generalized to semi-bandit feedback.

6.4 Centralized Algorithms for Congestion Games

In this section, we introduce two centralized algorithms for congestion games – one for the semi-bandit feedback and one for the bandit feedback. We will see that both of them can achieve sublinear Nash regret with polynomial dependence on both m and F.

6.4.1 Algorithm for Semi-bandit Feedback

Summarized in Algorithm 10, Nash upper confidence bound (Nash-UCB) for congestion games is developed based on optimism in the face of uncertainty. In particular, the algorithm estimates the reward matrices optimistically in line 4, computes its Nash equilibrium policy

in line 6 and then follows this policy.

For convenience, we define the empirical counter $N^{k,f}(n) = \sum_{k'=1}^{k} \mathbb{1}\left\{n^f(\boldsymbol{a}^{k'}) = n\right\}$ and $\widetilde{\iota} = 2\log(4(m+1)K/\delta)$. Then, the reward estimator for f and the bonus term are defined as

$$\hat{r}^{k,f}(n) = \frac{\sum_{k'=1}^{k} r^{k',f}\mathbb{1}\left\{n^f(\boldsymbol{a}^{k'}) = n\right\}}{N^{k,f}(n) \vee 1}, \quad b_i^{k,\mathrm{r}}(\mathbf{a}) = \sum_{f \in a_i} \sqrt{\frac{\widetilde{\iota}}{N^{k,f}(n^f(\mathbf{a})) \vee 1}}, \tag{6.1}$$

where $r^{k,f} \in [0,1]$ is the random reward realization of $r^f(n^f(\mathbf{a}^k))$. Naturally, the reward estimator for player i is $\hat{r}_i^k(\mathbf{a}) = \sum_{f \in a_i} \hat{r}^{k,f}(n^f(\mathbf{a}))$.

Algorithm 10 Nash-UCB for Congestion Games

1: **Input:** ϵ, accuracy parameter for Nash equilibrium computation
2: **for** episode $k = 1, \ldots, K$ **do**
3: **for** player $i = 1, \ldots, m$ **do**
4: Compute $\overline{Q}_i^k(\boldsymbol{a}) \leftarrow \hat{r}_i^k(\boldsymbol{a}) + b_i^{k,\mathrm{r}}(\boldsymbol{a})$ for all $\boldsymbol{a} \in \mathcal{A}$
5: **end for**
6: Compute $\pi^k \leftarrow \epsilon$-NASH$(\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot))$ (Algorithm 11)
7: Take action $\boldsymbol{a}^k \sim \pi^k$ and observe reward $r^{k,f}$
8: Update reward estimators $\hat{r}_i^k$ and bonus term $b_i^{k,\mathrm{r}}$
9: **end for**

Algorithm 10 is motivated by the Nash-VI algorithm in [Liu et al., 2021a] plus a deliberate utilization of the special reward structure in the congestion games. Moreover, notice that a matrix game with reward functions $\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot)$ forms a potential game (see Lemma E.2.1). As a result, in line 6, we can *efficiently compute* the ϵ-approximate Nash equilibrium π^k for that matrix game by utilizing Algorithm 11, (see Lemma E.2.2). It is a simple greedy algorithm such that in each round, it modifies one player's policy whose modification can increase the potential function most. In addition, Algorithm 11 always outputs a deterministic product policy.

Algorithm 11 ϵ-approximate Nash Equilibrium for Potential Games

Input: ϵ, accuracy parameter; full information potential game $\{(\mathcal{A}_i)_{i=1}^m, (r_i)_{i=1}^m\}$ such that $r_i \in [0, r_{\max}]$ for all $i \in [m]$
Initialize: $\pi^1 = \mathbf{a}^1$, arbitrary deterministic product policy
for round $k = 1, \ldots, \lceil \frac{mr_{\max}}{\epsilon} \rceil$ **do**
    for player $i = 1, \ldots, m$ **do**
        $\Delta_i \leftarrow \max_{a_i \in \mathcal{A}_i} r_i(a_i, \pi_{-i}^k) - r_i(\pi^k)$
        $a_i^{k+1} \leftarrow \operatorname{argmax}_{a \in \mathcal{A}_i} r_i(a_i, \pi_{-i}^k) - r_i(\pi^k)$
    end for
    if $\max_{i \in [m]} \Delta_i \le \epsilon$ **then**
        return π^k
    end if
    $j \leftarrow \operatorname{argmax}_{i \in [m]} \Delta_i$
    Update $\pi^{k+1}(j) \leftarrow a_j^{k+1}$ and $\pi^{k+1}(i) \leftarrow \pi^k(i)$ for all $i \neq j$
end for

6.4.2 Algorithm for Bandit Feedback

When the players can only receive bandit feedback, estimating $\hat{r}^{k,f}$ directly for each $f \in \mathcal{F}$ is no longer feasible. However, notice that the reward function $r_i(\mathbf{a}) = \sum_{f \in a_i} r^f(n^f(\mathbf{a}))$ can be seen as an inner product between vectors characterized by action $\mathbf{a}$ and reward function $r^f(\cdot)$. Therefore, under bandit feedback, we can treat it as a linear bandit and use ridge regression to build the reward estimator $\tilde{r}_i^k$ and corresponding bonus term $\tilde{b}^{k,\mathrm{r}}$, whose index i is dropped since it is the same for all players. The new algorithm will use these two terms to replace $\hat{r}_i^k$ and $b_i^{k,\mathrm{r}}$ in line 4 of Algorithm 10.

In particular, define $\theta \in [0,1]^{\tilde{d}}$ with $\tilde{d} = mF$ to be the vector such that $r^f(n) = \theta_{n+m(f-1)}$. Meanwhile, for player $i \in [m]$, define $A_i : \mathcal{A} \mapsto \{0,1\}^{\tilde{d}}$ to be the vector-valued function such that

$$[A_i(\mathbf{a})]_j = \mathbb{1}\left\{j = n + m(f-1), f \in a_i, n = n^f(\mathbf{a})\right\}.$$

In other words, $A_i(\mathbf{a})$ is a 0-1 vector with element 1 only at indices corresponding to those in θ that represents $r^f(n)$ for $f \in a_i$ and $n = n^f(\mathbf{a})$. Now, with these definitions, the reward

function can be written as $r_i(\mathbf{a}) = \langle A_i(\mathbf{a}), \theta \rangle$. Then, we build the reward estimator and the bonus term through ridge regression and corresponding confidence bound, which are defined as the following:

$$\tilde{r}_i^k(\mathbf{a}) = \left\langle A_i(\mathbf{a}), \widehat{\theta}^k \right\rangle, \quad \tilde{b}^{k,\mathrm{r}}(\mathbf{a}) = \max_{i \in [m]} \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \sqrt{\tilde{\beta}_k}, \tag{6.2}$$

where $\widehat{\theta}^k = \left(V^k\right)^{-1} \sum_{k'=1}^{k-1} \sum_{i=1}^{m} A_i(\mathbf{a}^{k'}) r_i^{k'}$, $V^k = I + \sum_{k'=1}^{k-1} \sum_{i=1}^{m} A_i(\mathbf{a}^{k'}) A_i(\mathbf{a}^{k'})^\top$ and $\sqrt{\tilde{\beta}_k} = \sqrt{\tilde{d}} + \sqrt{F\tilde{d}\log\left(1 + \frac{mkF}{\tilde{d}}\right) + F\tilde{\iota}}$. Note that we cannot bound the sum of this bonus terms by directly applying the elliptical potential lemma. We instead prove its variant in Lemma E.3.2.

6.4.3 Regret Analysis

The Nash regret bounds for the two versions of Algorithm 10 are formally presented in Theorem 6.4.1. The proof details are deferred to Appendix E.3.

Theorem 6.4.1. *Let $\epsilon = 1/K$. For congestion games with semi-bandit feedback, by running Algorithm 10 with reward estimator and bonus term in (6.1), with probability at least $1-\delta$, we can achieve that*

$$\textit{Nash-Regret}(K) \leq \widetilde{\mathcal{O}}\left(F\sqrt{mK}\right).$$

Furthermore, if we only have bandit feedback, then by running Algorithm 10 with reward estimator and bonus term in (6.2), with probability at least $1-\delta$, we can achieve that

$$\textit{Nash-Regret}(K) \leq \widetilde{\mathcal{O}}\left(mF^{3/2}\sqrt{K}\right).$$

Remark 6.4.2. Since each action is a subset of $\mathcal{F}$, the size of each player's action space can be 2^F. As a result, directly applying Nash-VI in [Liu et al., 2021a] leads to a regret bound exponential in F.

Remark 6.4.3. Note that we assume $r^f \in [0,1]$, which implies $r_i \in [0,F]$ for each player $i \in [m]$.

6.5 Decentralized Algorithms for Congestion Games

In this section, we present a decentralized algorithm for congestion games. Due to limited space, we only introduce the version of bandit feedback as in Section 6.4.2. The algorithmic details for the semi-bandit feedback setting are deferred into Appendix E.4.3. We will show that under both settings, even though each player can only observe her own actions and rewards, our decentralized algorithm still enjoys sublinear Nash regret with polynomial dependence on m and F.

We first define the vector-valued function $\phi_i : \mathcal{A}_i \mapsto \{0,1\}^{F_i}$ to be the feature map of player i such that $[\phi_i(a_i)]_f = \mathbb{1}\{f \in a_i\}$ for $a_i \in \mathcal{A}_i$ and $f \in \bigcup_{a_i \in \mathcal{A}_i} a_i$. Here, F_i is the size of $\bigcup_{a_i \in \mathcal{A}_i} a_i \subseteq \mathcal{F}$ and we can immediately see that $F_i \leq F$ for any $i \in [m]$.

The core idea of our algorithm is that the Nash equilibrium can be found by reaching the stationary points of the potential function since all congestion games are potential games. Here, the UCB-like algorithms used in the centralized setting are not applicable because their policy computation requires value functions for all players (e.g., line 6 of Algorithm 10), which are not available in the decentralized setting. Summarized in Algorithm 12, the decentralized algorithm is developed based on the Frank-Wolfe method and has the following three major components.

Gradient Estimator. In line 8, the algorithm builds the estimator $\widehat{\nabla}_i^k \Phi$ defined in (6.4) by using the τ reward samples collected from line 5. Here, $\widehat{\nabla}_i^k \Phi$ estimates the gradient of potential function Φ with respect to the policy π_i^k. Recall that for a congestion game, we have $\Phi(\mathbf{a}) = \sum_{f \in \mathcal{F}} \sum_{i=1}^{n^f(\mathbf{a})} r^f(i)$ and $\Phi(\pi) = \mathbb{E}_{\mathbf{a} \sim \pi}[\Phi(\mathbf{a})]$. Then we can define $\nabla_i \Phi := \nabla_{\pi_i} \Phi$ as a vector of dimension $|\mathcal{A}_i|$. For the component indexed by some $a_i \in \mathcal{A}_i$, we can see that $\Phi(\pi) = \pi_i(a_i) \mathbb{E}_{a_{-i} \sim \pi_{-i}}[r_i(a_i, a_{-i})] + \text{const}$, where const does not depend on $\pi_i(a_i)$. Therefore, we have

$$\nabla_i \Phi(a_i) = \mathbb{E}_{a_{-i} \sim \pi_{-i}}[r_i(a_i, a_{-i})] = \mathbb{E}_{a_{-i} \sim \pi_{-i}} \left[\sum_{f \in a_i} r^f(n^f(a_i, a_{-i})) \right] = \langle \phi_i(a_i), \theta_i(\pi) \rangle, \quad (6.3)$$

Algorithm 12 Frank-Wolfe with Exploration for Congestion Game

Input: γ, ν, mixture weights; π_i^1, initial policy
Initialize: ρ_i, the G-optimal design for player i, defined in (6.5)
for episode $k = 1, \dots, K$ **do**
  for round $t = 1, \dots, \tau$ **do**
    Each player takes action $a_i^{k,t} \sim \pi_i^k$, observes reward $r_i^{k,t}$
  end for
  for player $i = 1, \dots, m$ **do**
    Compute $\widehat{\nabla}_i^k \Phi(a_i)$ by the formula in (6.4) for all $a_i \in \mathcal{A}_i$
    Compute $\widetilde{\pi}_i^{k+1} \leftarrow \operatorname{argmax}_{\pi_i \in \Delta(\mathcal{A}_i)} \langle \pi_i, \widehat{\nabla}_i^k \Phi \rangle$
    Update

$$\pi_i^{k+1} \leftarrow (1-\gamma)\left(\nu \widetilde{\pi}_i^{k+1} + (1-\nu)\pi_i^k\right) + \gamma \rho_i$$

  end for
end for

where $[\theta_i(\pi)]_f = \mathbb{E}_{a_{-i} \sim \pi_{-i}}\left[r^f(n^f(a_{-i}) + 1)\right]$. Meanwhile, the mean of the t-th reward that player i received at episode k satisfies

$$\mathbb{E}\left[r_i^{k,t} \mid \mathbf{a}^{k,t}\right] = r_i(\mathbf{a}^{k,t}) = \sum_{f \in a_i^{k,t}} r^f(n^f(\mathbf{a}^{k,t})) = \left\langle \phi_i(a_i^{k,t}), \theta_i^{k,t}(a_{-i}^{k,t}) \right\rangle,$$

where $[\theta_i^{k,t}(a_{-i}^{k,t})]_f = r^f(n^f(a_{-i}^{k,t}) + 1)$ and its mean is $[\theta_i(\pi^k)]_f$. Therefore, we can use linear regression to estimate $\theta_i(\pi^k)$. In particular, we have $\widehat{\theta}_i^k(\pi^k) = \frac{1}{\tau}\sum_{t=1}^{\tau}\left(\Sigma_i^k\right)^{-1}\phi_i(a_i^{k,t}) r_i^{k,t}$, with the covariance matrix $\Sigma_i^k = \mathbb{E}_{a_i \sim \pi_i^k}\left[\phi_i(a_i)\phi_i(a_i)^\top\right]$. Then, we have the unbiased gradient estimate

$$\widehat{\nabla}_i^k \Phi(a_i) = \left\langle \phi_i(a_i), \widehat{\theta}_i^k(\pi^k) \right\rangle = \frac{1}{\tau}\sum_{t=1}^{\tau} \phi_i(a_i)^\top \left(\Sigma_i^k\right)^{-1} \phi_i(a_i^{k,t}) r_i^{k,t}. \tag{6.4}$$

Remark 6.5.1. One difference between Algorithm 12 (decentralized) and Algorithm 10 (centralized) is that in the decentralized algorithm, each player is required to play the same policy for τ times before an update can be applied. An episode is thus defined for convenience as the time period during which the players' policies are fixed. We make this artificial

design mainly for controlling the variance of the gradient estimator $\widehat{\nabla}_i^k \Phi(a_i)$. However, we conjecture that with more careful design and analysis, it should be possible to improve Algorithm 12 so that only one sample is required per episode [Zhang et al., 2020c].

G-optimal Design. In line 9 and 10, the algorithm performs standard Frank-Wolfe update and mixes the updated policy with an exploration policy ρ_i, which is defined as the G-optimal allocation for features $\{\phi_i(a_i)\}_{a_i \in \mathcal{A}_i}$. To be specific, we have

$$\rho_i = \operatorname*{argmin}_{\lambda \in \Delta(\mathcal{A}_i)} \max_{a_i \in \mathcal{A}_i} \|\phi_i(a_i)\|^2_{\mathbb{E}_{a_i' \sim \lambda}\left[\phi_i(a_i')\phi_i(a_i')^\top\right]^{-1}}. \tag{6.5}$$

Here ρ_i guarantees that Σ_i^k is invertible and the variance of $\widehat{\nabla}_i^k \Phi(a_i) = \left\langle \phi_i(a_i), \widehat{\theta}_i^k(\pi^k) \right\rangle$ depends only on F instead of the size of action space (Lemma E.4.3) because by the famous Kiefer-Wolfowitz theorem, we have $\max_{a_i \in \mathcal{A}_i} \|\phi_i(a_i)\|^2_{\mathbb{E}_{a_i' \sim \rho_i}\left[\phi_i(a_i')\phi_i(a_i')^\top\right]^{-1}} = F_i \le F$ [Lattimore and Szepesvári, 2020].

Frank-Wolfe Update. Finally, we emphasize that it is crucial to use Frank-Wolfe update because it is compatible with L_1 *norm* and we can show that Φ is mF-smooth with respect to the L_1 norm (Lemma E.4.5). In contrast, its smoothness for L_2 norm will depend on the size of the action space.

Before the game starts, each player i can compute her ρ_i based on her own action set $\mathcal{A}_i$. During the game, all players only have access to their own actions and rewards, which means that Algorithm 12 is fully decentralized. The Nash regret bound for this algorithm is formally stated in Theorem 6.5.2 and the proof details are given in Appendix E.4.1 and E.4.2.

Theorem 6.5.2. *Let $T = K\tau$. For congestion game with bandit feedback, by running Algorithm 12 with gradient estimator $\widehat{\nabla}_i^k \Phi$ in (6.4) and exploration distribution ρ_i in (6.5), if $K \ge \frac{2F}{m}$, then with probability at least $1-\delta$, we have*

$$\textit{Nash-Regret}(T) := \sum_{k=1}^{K} \tau \max_{i \in [m]} \left(V_i^{\dagger, \pi_{-i}^k} - V_i^{\pi^k} \right) \le \widetilde{\mathcal{O}}\left(m^2 F^2 T^{5/6} + m^3 F^3 T^{2/3} \right).$$

For congestion game with semi-bandit feedback, by running Algorithm 12 with gradient estimator $\widetilde{\nabla}_i^k \Phi(a_i)$ and exploration distribution $\tilde{\rho}_i$ defined in Appendix E.4.3, if $K \ge \frac{2\sqrt{F}}{m}$, then

with probability at least $1-\delta$*, we have*

$$\textit{Nash-Regret}(T) \le \widetilde{\mathcal{O}}\left(m^2F^{3/2}T^{5/6} + m^3F^2T^{2/3}\right).$$

6.6 Extension to Independent Markov Congestion Games

In this section, we propose and analyze a Markov extension of the congestion games, called the independent Markov congestion games (IMCGs).

6.6.1 Problem Formulation

General-sum Markov Games. A finite-horizon time-inhomogeneous tabular general-sum Markov game is defined by $\mathcal{M} = \{\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^m, H, P, R, s_0\}$, where $\mathcal{S}$ is the state space, m is the number of players, $\mathcal{A}_i$ is the action space of player i, $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ is the whole action space, H is the time horizon, s_0 is the initial state*, $P = (P_1, P_2, \cdots, P_H)$ with $P_h \in [0,1]^{S\times A\times S}$ as the transition kernel at timestep h, $R = \{R_h(\cdot|s_h, \mathbf{a}_h)\}_{h=1}^H$ with $R_h(\cdot|s_h, \mathbf{a}_h)$ as the reward distribution on $[0, r_{\max}]^m$ with mean $\mathbf{r}_h(s_h, a_h) \in [0, r_{\max}]^m$ at timestep $h \in [H]$. At timestep h, all players choose their actions simultaneously and a reward vector is sampled $\mathbf{r}_h \sim R_h(\cdot|s_h, \mathbf{a}_h)$, where s_h is the current state and $\mathbf{a}_h = (a_{h,1}, a_{h,2}, \cdots, a_{h,m})$ is the joint action. Each player i receives reward $r_{h,i}$ and the state transits to $s_{h+1} \sim P_h(\cdot|s_h, \mathbf{a}_h)$. The objective for each player is to maximize her own total reward. We assume that the initial state s_1 is fixed.

A (Markov) policy π is a collection of H functions $\{\pi_h : \mathcal{S} \mapsto \Delta(\mathcal{A})\}_{h=1}^H$, each of which maps a state to a distribution over the action space. π is a product policy if $\pi_h(\cdot \mid s)$ is a product policy for each $(h, s) \in [H] \times \mathcal{S}$. The value function and Q-value function of player i at timestep h under policy π are defined as

$$V_{h,i}^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \mathbf{a}_{h'}) \mid s_h = s\right],\ Q_{h,i}^{\pi}(s, \mathbf{a}) = \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h',i}(s_{h'}, \mathbf{a}_{h'}) \mid s_h = s, \mathbf{a}_h = a\right].$$

The best responses and Nash regret can be defined similarly as those for matrix games. In particular, given a policy π, player i's best response policy is $\pi_{h,i}^{\dagger}(\cdot \mid s) = \operatorname{argmax}_{\mu\in\Delta(\mathcal{A}_i)} V_{h,i}^{\mu,\pi_{-i}}(s)$ and the corresponding value function is denoted as $V_{h,i}^{\dagger,\pi_{-i}}$.

*An episode is defined as running H steps from the initial state s_0, which is common for the episodic MDP.

Definition 6.6.1. With π^k being the policy at kth episode, the *Nash regret* after K episodes is define as

$$\text{Nash-Regret}(K) = \sum_{k=1}^{K} \max_{i\in[m]} \left(V_{1,i}^{\dagger,\pi_{-i}^k} - V_{1,i}^{\pi^k}\right)(s_1).$$

Independent Markov Congestion Game. A general-sum Markov game is an independent Markov congestion game (IMCG) if there exists a facility set $\mathcal{F}$ such that $a_i \subseteq \mathcal{F}$ for any $a_i \in \mathcal{A}_i$, a state space $\mathcal{S} = \prod_{f\in\mathcal{F}} \mathcal{S}^f$, a set of facility reward distributions $\{R_h^f\}_{h\in[H],f\in\mathcal{F}}$ such that if the joint action at s_h is $\mathbf{a}$, we have $r_{h,i} = \sum_{f\in a_i} r_h^f$, where $r_h^f \sim R_h^f(\cdot|s_h, n^f(\mathbf{a}))$ with support on $[0,1]$ and mean $r_h^f(s_h, n^f(\mathbf{a}))$, and a set of transition matrices $\{P_h^f\}_{h\in[H],f\in\mathcal{F}}$ such that $P_h(s'|s,\mathbf{a}) = \prod_{f\in\mathcal{F}} P_h^f(s'^f|s^f, n^f(\mathbf{a}))$. In other words, at each timestep h and state $s \in \mathcal{S}$, the players are in a congestion game. Meanwhile, each facility has its own state and independent state transition, which only depends on its current state and number of players using that facility. This transition kernel can be viewed as a special case of that in factored MDPs. The IMCG also admits two types of feedback, semi-bandit feedback and bandit feedback, just like the congestion game. In this paper, we will consider both types of feedback.

6.6.2 Theoretical Guarantee

Summarized in Algorithm 25, our centralized algorithm for IMCGs is naturally extended from the Nash-UCB (Algorithm 10) by incorporating transition kernel estimators, corresponding bonus terms and Bellman backward update. The key idea is to utilize the independent transition structure to remove the dependence on the exponential size of the state space $S = \prod_{f\in\mathcal{F}} S^f$. We tackle this issue by adapting technique from factored MDP [Chen et al., 2020]. The algorithmic details for both types of feedback are deferred into Appendix E.5. The Nash regret bounds for the two versions of Algorithm 25 are stated in Theorem 6.6.2 and the proof details are deferred to Appendix E.6.

Theorem 6.6.2. *For independent Markov congestion game with semi-bandit feedback, by running the centralized Algorithm 25, with probability at least $1-\delta$, we can achieve that*

$$\textit{Nash-Regret}(K) \le \widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} FS^f\sqrt{mH^3T}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right).$$

Furthermore, if we only have bandit feedback, then by running Algorithm 25 with reward estimator and bonus term in (E.7) and (E.8), with probability at least $1-\delta$*, we can achieve that*

$$\textit{Nash-Regret}(K) \leq \widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} FS^f\sqrt{m^2H^3T}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right).$$

The regret bound in [Liu et al., 2021a] is $\widetilde{O}(\sqrt{H^3S^2(\Pi_{i=1}^m A_i)T})$, where both A_i and $S=\prod_{f\in\mathcal{F}} S^f$ can be exponential in F. Our bounds have polynomial dependence on all the parameters.

6.7 Conclusion

In this paper, we study sample-efficient learning in congestion games by utilizing the special reward structure. We propose both centralized and decentralized algorithms for congestion games with two types of feedback, all achieving sample complexities only polynomial in the number of facilities. To the best of our knowledge, each one of them is the first sample-efficient learning algorithm for congestion games in its own setting. We further define the independent Markov congestion game (IMCG) as a natural extension of the congestion game into the Markov setting together with a sample-efficient centralized algorithm for both types of feedback.

One promising future direction is to find a sample-efficient decentralized algorithm such that from each player's own perspective, the algorithm is still no-regret. In other words, diminishing regret is guaranteed for the player by running this algorithm even though other players may use policies from different algorithms. Another important future direction is to find sample-efficient centralized/decentralized algorithms that can explicitly find an approximate Nash equilibrium policy.

Chapter 7

LEARNING OPTIMAL TAX DESIGN IN CONGESTION GAMES

This chapter is based on Cui et al. [2024], with Maryam Fazel and Simon S. Du.

7.1 Introduction

In modern society, large-scale systems often consist of many self-interested players with shared resources, such as transportation and communication networks. Importantly, the objectives of individual players are not always aligned with the system efficiency, and the system designer should take this into consideration. A widely known example is Braess's paradox, where adding more roads to a network can make the network more congested [Braess, 1968]. Price of anarchy is a notion that measures the inefficiency caused by selfish behavior compared with optimal centralized behavior [Koutsoupias and Papadimitriou, 1999]. Characterizing such inefficiency has been an active research area with applications in resource allocation [Marden and Roughgarden, 2014], traffic congestion [Roughgarden and Tardos, 2004], and others. The inefficiency motivates research on how to design mechanisms to improve performance even when the players are still behaving selfishly.

Tax mechanisms are a standard approach to resolving the inefficiency issue, which are widely studied in economics, operations research, and game theory. The goal of tax mechanisms is to incentivize self-interested players to follow socially optimal behavior by applying tax/subsidy. Congestion game is a widely studied class of game theory models characterizing the interactions between players sharing facilities, where the cost of each facility depends on the "congestion" level [Wardrop, 1952, Rosenthal, 1973]. As a motivating example, in traffic routing games, each facility corresponds to an edge in a network, and each player chooses a path that connects her source node and target node. The cost of each facility corresponds to the latency of each edge, which depends on the number of players using that edge. Then, the tax can be interpreted as the toll collected by the road owner or the

government to improve overall traffic efficiency [Bergendorff et al., 1997].

Most existing works on congestion game tax design focus on the computation complexity of the optimal tax [Nisan et al., 2007a, Caragiannis et al., 2010]. They assume the tax designer has full knowledge of the underlying game, which is unrealistic in many applications. As Nash equilibrium is the only stable state of the system, we study a partial information feedback setting named "equilibrium feedback", where the tax designer can only observe information about the Nash equilibrium. The limited feedback information brings new challenges to the tax designer, and strategic exploration is necessary to learn or design the optimal tax. In this work, we aim to take the first step in learning optimal tax design for congestion games, and we study the following problem:

How can we learn the optimal tax design in congestion games with equilibrium feedback?

Below we highlight our contributions.

7.1.1 Main Contributions and Technical Novelties

1. The first algorithm for learning optimal tax design in congestion games. To the best of our knowledge, this is the first result for learning optimal tax in congestion games with partial information feedback. Our algorithm enjoys $O(F^2\beta/\epsilon)$ sample complexity for learning an ϵ-optimal tax, where F is the number of facilities and β is the smoothness coefficient of the cost function. The sample complexity has no dependence on the number of actions, which could be exponential in F. In addition, we provide an efficient implementation for network congestion games with $\widetilde{O}(\text{poly}(V, E, \epsilon))$ computational complexity, where V and E are the numbers of the vertexes and edges in the network. Due to space limitation, we defer the computation analysis and experiments to Appendix F.3 and Appendix F.5.

2. Piece-wise linear function approximation. We only assume the cost functions are smooth and make no parameterization assumptions as they are too strong to be satisfied in real-world applications. To tackle this challenge, we use piece-wise linear functions to approximate the optimal tax function. While only the values of the cost functions can be observed, we show that a carefully designed piece-wise linear function can approximate the unobservable optimal tax function well.

3. Strongly convex potential function. One challenge in tax design is controlling the sensitivity of Nash equilibrium w.r.t. tax perturbation. We always enforce tax functions with subgradient lower bounded by some positive value, which leads to a strongly convex potential function. As a result, the Nash equilibrium will be unique and Lipschitz with respect to tax perturbation. As the potential function for optimal tax is not necessarily strongly convex, we carefully choose the strong-convexity coefficient to balance the induced bias.

4. Exploratory tax design. Given the equilibrium feedback, the tax designer can only indirectly query the cost function by applying tax. Consequently, exploration in tax design becomes much more difficult than that in standard bandit problems where the player can directly query the value of an action [Lattimore and Szepesvári, 2020]. We design an exploratory tax that pushes the equilibrium to the "boundary", where an additional tax perturbation will change the equilibrium and reveal information about at least one unknown facility.

In this work, we focus on the well-known nonatomic congestion games. We hope our algorithm and analysis provide new insight on the intriguing structure of nonatomic congestion games. In addition, the tax design algorithm might find applications in real-world problems such as toll design in traffic networks. Due to space limitation, most proofs are deferred to the appendix.

Notations. $[m] = \{1, 2, \cdots, m\}$. For a set of real numbers $\mathcal{K}$ and a real number $x : \min\{\mathcal{K}\} \le x \le \max\{\mathcal{K}\}$, we define $[x]^+_{\mathcal{K}} := \min_{y\in\mathcal{K}:y\ge x} y$ and $[x]^-_{\mathcal{K}} := \max_{y\in\mathcal{K}:y\le x} y$. The clip operation $\text{clip}(a, l, r) := \min\{\max\{a, l\}, r\}$ clips a into the interval $[l, r]$. We use $O(\cdot)$ to hide absolute constants and $\widetilde{O}(\cdot)$ to hide polylog terms as well. A function $f : \mathcal{X} \mapsto \mathbb{R}$ is α-strongly convex if $f(y) \ge f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2} \|y - x\|_2^2, \forall x, y \in \mathcal{X}$. f is β-smooth if $\|\nabla f(x) - \nabla f(y)\|_2 \le \beta \|x - y\|_2, \forall x, y \in \mathcal{X}$.

7.2 Related Work

Learning in congestion games. We refer the readers to the textbook [Nisan et al., 2007a] for a general introduction to congestion games, the price of anarchy and tax mechanisms. Nonatomic congestion games were first studied in [Pigou, 1912] and formalized

by [Wardrop, 1952]. Atomic congestion games were introduced by [Rosenthal, 1973] and the connection with potential games is developed by [Monderer and Shapley, 1996]. In contrast to general-sum games without structures, (approximate) Nash equilibrium can be computed efficiently in congestion games due to the existence of the potential function. Recently, various algorithms are developed to learn the Nash equilibrium in congestion games with different feedback oracles [Krichene et al., 2015, Chen and Lu, 2016, Cui et al., 2022, Jiang et al., 2022, Panageas et al., 2023, Dong et al., 2023, Dadi et al., 2024]. These algorithms are derived from the perspective of the players in the system, while our algorithm is essentially different in that it is utilized by the system designer to induce better equilibrium.

Optimal tax design in congestion games. For nonatomic congestion games, optimal tax design has a closed-form solution known as the marginal cost mechanism [Nisan et al., 2007a]. For atomic congestion games, the marginal cost mechanism can no longer improve the efficiency [Paccagnan et al., 2021]. Instead, other mechanisms are proposed for optimal local/global and congestion dependent/independent tax in atomic congestion games [Caragiannis et al., 2010, Bilò and Vinci, 2019, Paccagnan et al., 2021, Paccagnan and Gairing, 2021, Harks et al., 2015]. Notably, all of these mechanisms assume full knowledge of the game while we consider learning with partial information feedback.

Stackelberg games. Stackelberg game [Von Stackelberg, 2010] models the interactions between leaders and followers such that leaders take actions first and the followers make decisions after observing leaders' actions. Tax design can be formulated as a Stackelberg game where the designer is the leader and the game players are the followers. Equipped with a best response oracle to predict followers' actions, Letchford et al. [2009], Blum et al. [2014], Peng et al. [2019] propose algorithms for learning Stackelberg equilibrium. Recently, Bai et al. [2021], Zhong et al. [2021], Zhao et al. [2023] generalize these results to learning Skackelberg equilibrium with bandit feedback, under finite actions or linear function approximation assumptions. For tax design, the search space is an exponentially large function space with complicated dependence on the objective. Consequently, existing results for Stackelberg games become vacuous when specialized to our problem.

Mathematical programming under equilibrium constraint. Tax design can be formulated as minimizing social cost with respect to tax under the constraint that players are following the equilibrium. This is known as mathematical programs with equilibrium constraints (MPEC). MPEC is a bilevel optimization problem and is NP-hard in general [Luo et al., 1996]. Existing approaches use specific inner loop algorithms to approach the equilibrium so that the gradient can be propagated to the outer loop [Li et al., 2020, Liu et al., 2022a, Li et al., 2022c, Maheshwari et al., 2023, Li et al., 2023, Grontas et al., 2024], relying on a unique and differentiable equilibrium [Colson et al., 2007]. However, such an approach requires many strong assumptions, such as the tax designer can control the algorithm of the agents, convex objective function and parameterized tax function. In contrast, our results make none of these assumptions.

7.3 Preliminaries

Nonatomic congestion games. A weighted nonatomic congestion game (congestion game) is described by the tuple $(\mathcal{F}, \mathcal{A}_{[m]}, w_{[m]}, c_{\mathcal{F}})$, where $\mathcal{F}$ is the set of facilities with cardinality F, m is the number of commodities, $\mathcal{A}_i$ is the action set for commodity $i \in [m]$, $w_i \in [0,1]$ is the weight for commodity $i \in [m]$ such that $\sum_{i\in[m]} w_i = 1$, and $c_f : [0,1] \mapsto [0,1]$ is the cost function for facility $f \in \mathcal{F}$. Each commodity consists of infinite number of infinitesimal players with a total load to be w_i. Each individual player is self-interested and has a negligible effect on the game.

In congestion games, action $a \in \mathcal{A}_i, i \in [m]$ is a subset of $\mathcal{F}$, i.e. $a \subseteq \mathcal{F}$, which denotes the facilities utilized by action a. For commodity $i \in [m]$, we use strategy $x_i = (x_{i,a})_{a\in\mathcal{A}_i} \in [0, w_i]^{|\mathcal{A}_i|}$ with constraint $\sum_{a\in\mathcal{A}_i} x_{i,a} = w_i$ to denote how the load is distributed over all the actions. The joint strategy for the game is represented by $x = (x_1, x_2, \cdots, x_m) \in [0,1]^A$, where $A = \sum_{i\in[m]} |\mathcal{A}_i|$. We use $\mathcal{X}$ to denote the set of all feasible strategies.

A decentralized perspective of strategy x_i for commodity i is that each self-interested infinitesimal player follows a randomized strategy that chooses $a \in \mathcal{A}_i$ with probability proportional to $x_{i,a}$. With the law of large number, the load on action a would be $x_{i,a}$.

Cost function. For a strategy x, the cost of a facility is $c_f(l_f(x))$, where $l_f(x) = \sum_{i\in[m]} \sum_{a\in\mathcal{A}_i: f\in a} x_{i,a}$ is the load on facility f. The cost of an action a is the sum of the facility cost that a utilizes: $c_a(x) := \sum_{f\in a} c_f(l_f(x))$.

We make the following assumption on the cost function. Monotonicity is a standard congestion game assumption, which is also observed in many real-world applications as more players sharing one facility, each player will have less gain or more cost [Nisan et al., 2007a]. Smoothness is a standard technical assumption for analysis.

Assumption 7.3.1. We assume the cost function satisfies:

1. Monotonicity: $c_f(\cdot)$ is non-decreasing for all $f \in \mathcal{F}$,

2. Smoothness: $c_f(\cdot)$ is β-smooth for all $f \in \mathcal{F}$.

Nash equilibrium. Nash equilibrium in nonatomic congestion games, also known as the Wardrop equilibrium [Wardrop, 1952], is the strategy that no player has the incentive to deviate from its strategy as formalized in Definition 7.3.2. In other words, Nash equilibrium is a stable state for a system with selfish players.

Definition 7.3.2. A Nash equilibrium strategy x is a joint strategy such that each player is choosing the best action: for any commodity $i \in [m]$ and actions $a, a' \in \mathcal{A}_i$, we have

$$c_a(x) \le c_{a'}(x), \text{ if } x_{i,a} > 0.$$

Similarly, an ϵ-approximate Nash equilibrium x satisfies that

$$\forall i \in [m], a, a' \in \mathcal{A}_i, c_a(x) \le c_{a'}(x) + \epsilon, \text{ if } x_{i,a} > 0.$$

For a strategy x and commodity i, actions $a \in \mathcal{A}_i$ such that $x_{i,a} > 0$ are named as the "in-support" actions and the others are "off-support" actions. For a Nash equilibrium, in-support actions must all have the same cost and off-support actions are no better than in-support actions. It is well known that Nash equilibrium always exists in congestion games [Beckmann et al., 1956].

Potential Function. An important concept in congestion games is the potential function:

$$\Phi(x) := \sum_f \int_0^{l_f(x)} c_f(u)du.$$

If Assumption 7.3.1 is satisfied, then $\Phi(x)$ is a convex function and Nash equilibrium is equivalent to the minimizer of the potential function [Beckmann et al., 1956].

Network congestion games. Network congestion games are congestion games with multicommodity network structure, which are also known as the selfish routing games [Roughgarden, 2005]. A multicommodity network is described by a directed graph $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge (facility) set. In addition, each commodity $i \in [m]$ corresponds to a pair of source and target vertex (s_i, t_i), and actions are all feasible paths connecting s_i and t_i. Each edge is associated with a nondecreasing cost (latency) function.

7.4 Tax Design for Congestion Games

In this section, we introduce tax design in congestion games. Before we get into the details, we will first introduce some notions to simplify the problem.

7.4.1 Polytope Description for Congestion Games

For a strategy $x_i \in \mathbb{R}^{|\mathcal{A}_i|}$, the dimension $|\mathcal{A}_i|$ can be as large as 2^F. Instead, it would be convenient to consider the facility load $y_i \in \mathbb{R}^F$ such that $y_{i,f} = \sum_{a \in \mathcal{A}_i : f \in a} x_{i,a}$. In addition, we define $y = \sum_{i \in [m]} y_i \in \mathbb{R}^F$ to be the total facility load. We use $\phi_{(i)}(\cdot)$ to denote the reparameterization mapping:

$$\phi(x) = y, \phi_i(x_i) = y_i, \forall i \in [m],$$

and we set $\mathcal{Y} = \{y \in \mathbb{R}^F : \exists x \in \mathcal{X}, y = \phi(x)\}$ to be the set of all feasible loads. Note that ϕ is not necessarily a bijection, i.e., there could exist multiple strategies sharing the same load. We use $\phi^{-1}(y) := \{x \in \mathcal{X} : \phi(x) = y\}$ to denote the set of strategies that are mapped to load y. The potential function can be defined after the repameterization as well:

$$\Phi^{\text{repa}}(y) := \sum_f \int_0^{y_f} c_f(u)du = \Phi(x), \forall x \in \Phi^{-1}(y).$$

Importantly, $\Phi^{\text{repa}}(y)$ does not depend on the choice of strategy $x \in \phi^{-1}(y)$. For the reparameterized potential function, we have the following lemma showing that it is almost equivalent to the original potential function. When it is clear from the context, we will simplify Φ^{repa} as Φ.

Lemma 7.4.1. *Φ^{repa} is convex under Assumption 7.3.1. If $y^* = \operatorname{argmin}_y \Phi^{\text{repa}}(y)$, then for any $x \in \phi^{-1}(y^*)$, x is a Nash equilibrium.*

For any Nash equilibrium strategy x, we call $y = \phi(x)$ the Nash equilibrium load (Nash load).

7.4.2 Optimal Tax for Congestion Games

Nash equilibrium is a stable state for a system with self-interested players, as no player has the incentive to deviate unilaterally. However, Nash equilibrium does not efficiently utilize the facilities, which is measured by the social cost:

$$\Psi(y) := \sum_f y_f c_f(y_f).$$

Price of anarchy is a concept that measures the efficiency of selfish agents in a system, defined as the ratio between the worst-case social cost for equilibria and the optimal social cost:

$$\text{PoA} = \frac{\max_{y \text{ is a Nash equilibrium load}} \Psi(y)}{\min_{y \in \mathcal{Y}} \Psi(y)}$$

For example, in nonatomic congestion games with polynomial cost functions, the price of anarchy grows as $\Theta(d/\ln d)$ where d is the degree of the polynomials [Nisan et al., 2007a].

To reduce the price of anarchy, one standard approach is to enforce a tax on each facility to change the behavior of the self-interested players. Formally, a taxed congestion game is described by $(\mathcal{F}, \mathcal{A}_{[m]}, w_{[m]}, c_{\mathcal{F}}, \tau_{\mathcal{F}})$ with an additional tax function $\tau_f : [0,1] \to \mathbb{R}$ on facility $f \in \mathcal{F}$. The cost of facility f with load u under tax becomes $c_f(u) + \tau_f(u)$. Correspondingly, we define the potential function with tax τ as

$$\Phi(y; \tau) := \sum_f \int_0^{y_f} [c_f(u) + \tau_f(u)]\, du,$$

and the Nash load would satisfy $y^* \in \operatorname{argmin}_y \Phi(y;\tau)$.

The optimal tax is defined as the tax that can induce optimal social behavior for self-interested players. We want to note that tax is not included in social cost following the convention in tax design.

Definition 7.4.2. A tax τ is an optimal tax if all Nash equilibria under tax τ can minimize the social cost:

$$\operatorname*{argmin}_{y\in\mathcal{Y}} \Phi(y;\tau) \subseteq \operatorname*{argmin}_{y\in\mathcal{Y}} \Psi(y).$$

In addition, a tax τ is an ϵ-optimal tax if we have

$$\Psi(y) \leq \min_{y'\in\mathcal{Y}} \Psi(y') + \epsilon, \forall y \in \operatorname*{argmin}_{y''\in\mathcal{Y}} \Phi(y'';\tau).$$

The marginal cost tax is defined as

$$\tau^* : \tau_f^*(u) = uc_f'(u), \forall f \in \mathcal{F}.$$

As $\Phi(y;\tau^*) = \Psi(y)$, the Nash equilibrium under tax τ^* will minimize the social cost and τ^* is an optimal tax [Nisan et al., 2007a]. We will make the following assumption so that the cost combined with tax $c+\tau^*$ is still non-decreasing. In many real world problems, $c_f'(u)$ is non-decreasing due to the law of diminishing marginal utility, which guarantees Assumption 7.4.3.

Assumption 7.4.3. Marginal cost tax $\tau_f^*(u) = uc_f'(u)$ is non-decreasing for all $f \in \mathcal{F}$.

7.4.3 Tax Design for Congestion Games

In this paper, we consider the case where the system designer (e.g. government) wants to enforce an (approximate) optimal tax to induce optimal social behavior and maximize social welfare. However, the cost function is unknown so the optimal tax function cannot be computed directly via the marginal cost mechanism. On the other hand, the designer can enforce several taxes and observe the feedback. As Nash equilibrium is the stable state of the system, we assume the designer can observe the equilibrium feedback.

Formally, tax design is formulated as an online learning problem as shown in Protocol 4. At round t, the designer can choose a tax τ^t and observe the corresponding Nash equilibrium

Protocol 4 Online Tax Design for Congestion Games

Initialize: Facility set $\mathcal{F}$.

for $t = 1, 2, \ldots, T$ **do**

Designer chooses tax τ^t.

Designer observes Nash load $y^t = \operatorname{argmin}_{y \in \mathcal{Y}} \Phi(y; \tau)$ and Nash cost $c^t = [c_f(y_f^t)]_{f \in \mathcal{F}}$ for $f \in \mathcal{F}$.

end for

load $y^t \in \mathbb{R}^F$ and Nash equilibrium cost $c^t \in \mathbb{R}^F$. The sample complexity of a tax design algorithm is the number of rounds for designing an ϵ-optimal tax.

A naive approach is the designer first enumerates all of the ϵ-approximations of τ^* and chooses the tax with minimal social cost. However, such an approach would require $O((1/\epsilon)^{F\beta/\epsilon})$ samples as the complexity of using piece-wise linear function to approximate τ_f^* (a β-smooth function) with ϵ error is $O((1/\epsilon)^{\beta/\epsilon})$, resulting in exponential dependence on the parameters β, $1/\epsilon$ and F.

Another approach is applying algorithms for mathematical programming under equilibrium constraints. Specifically, we can formulate tax design as solving

$$\min_{\tau} \Psi(y(\tau)), \text{ s.t. } y(\tau) = \operatorname*{argmin}_{y \in \mathcal{Y}} \Phi(y; \tau).$$

However, $y(\tau)$ can be non-differentiable or even discontinuous w.r.t. τ, and $\Psi(y(\tau))$ can be non-convex w.r.t. τ (Lemma F.1.2). As a result, previous results do not apply to our problem as they apply gradient-based methods and make convexity assumptions [Li et al., 2020, Liu et al., 2022a].

7.5 Learning Optimal Tax in Nonatomic Congestion Games

In this section, we describe our algorithm that can learn an ϵ-optimal tax with $O(F^2\beta/\epsilon)$ samples. First, we introduce piece-wise linear functions as a nonparametric way to approximate the marginal cost tax τ^* [Takezawa, 2005].

Definition 7.5.1. (Piece-wise Linear Function) We use a dictionary* $d = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

*In this dictionary, key is x_i and value is y_i. For readers unfamiliar with the dictionary data structure, it can be regarded as a set with a special update operation.

for $x_i \neq x_j, \forall i \neq j$ (w.l.o.g. we let $x_1 < x_2 < \cdots < x_n$) to represent a piece-wise linear function $d(\cdot)$ on $[x_1, x_n]$ such that

$$d(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} y_i + \frac{x_i - x}{x_i - x_{i+1}} y_{i+1}, \forall x \in [x_i, x_{i+1}].$$

In addition, we use $\bigcup$ to represent the update method for dictionary. I.e., $d \bigcup (x, y)$ is the piece-wise linear function interpolating one more point (x, y) if $(x, d(x))$ is not already in d, otherwise it will update $d(x)$ to y.

We will maintain the piece-wise function on a grid Ł $= \{0, \Delta, 2\Delta, \cdots, K\Delta = 1\}$ with $K = \left\lceil \frac{2\beta}{\epsilon} \right\rceil$ and $\Delta = 1/K$. The time complexity for computing $d(x)$ is $O(\log K)$ for any $x \in [0, 1]$.

7.5.1 Main Algorithm

In this section, we introduce our main algorithm. At each round t, we will maintain a known index set $\mathcal{K}_f^t \subseteq$ Ł where the marginal cost tax can be accurately estimated (Lemma F.2.2), and use a piece-wise linear function to approximate the tax function by interpolating the values at the known indexes. The piece-wise linear function takes the form $\tau_f^t = \{(x_i^t, y_i^t)\}_i$ and the known index set $\mathcal{K}_f^t$ satisfies $\{x_i^t\}_i = \mathcal{K}_f^t \bigcup \{1\}$ and $\mathcal{K}_f^t \subseteq$ Ł. Here 1 is a special case as it is not in the known index set initially but it is needed as the boundary for the piece-wise linear function τ_f^t. Initially, the tax is set to be $\tau_f^1(u) = \{(0, 0), (1, \beta + \epsilon)\}$ and the auxiliary tax is $\widehat{\tau}_f^1 = \{(0, 0), (1, \beta)\}$ for $f \in \mathcal{F}$ (Line 2). Here we set $\widehat{\tau}_f^1(1) = \beta$ as β is always an upper bound on $\tau_f^*(1)$. The auxiliary tax $\widehat{\tau}_f^t$ is a non-decreasing piece-wice linear approximation of τ_f^* and we always set tax $\tau_f^t(u) = \widehat{\tau}_f^t(u) + \epsilon u$ to ensure that the subgradient of the tax enforced is lower bounded by ϵ.

At round t, after observing Nash equilibrium load $y^t \in \mathbb{R}^F$ and Nash equilibrium cost $c^t \in \mathbb{R}^F$, the facilities are split into two sets: known facilities and unknown facilities.

Definition 7.5.2. For each round t, facility f is known if the Nash load $y_f^t \in [0, 1]$ satisfies $[y_f^t]_{\text{Ł}}^- \in \mathcal{K}_f^t$ and $[y_f^t]_{\text{Ł}}^+ \in \mathcal{K}_f^t$. Otherwise, facility f is unknown for round t.

For a known facility f, the Nash load is either in the known index set or sandwiched by two consecutive known indexes. As a result, the tax estimate for the Nash load $\tau_f^t(y_f^t)$

Algorithm 13 Tax Design for Congestion Game

1: **Initialize**: Facility set $\mathcal{F}$, number of rounds T, tolerance ϵ, smoothness β, perturbation $\delta = \epsilon\Delta^2/8$.
2: Set initial tax $\tau^1 : \tau_f^1 = \{(0,0),(1,\beta+\epsilon)\}$ for all $f \in \mathcal{F}$. Set $\mathcal{K}_f^1$ to be $\{0\}$ for all $f \in \mathcal{F}$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Observe Nash load $y^t \in \mathbb{R}^F$ and Nash cost $c^t \in \mathbb{R}^F$ under tax τ^t.
5: Set $\bar{\mathcal{F}}$ to be the unknown facility set (Definition 7.5.2).
6: Set $l_f = \tau_f^t([y_f^t]^-_{\mathcal{K}_f^t}) + \epsilon(y_f^t - [y_f^t]^-_{\mathcal{K}_f^t})$ and $r_f = \tau_f^t([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) + \epsilon(y_f^t - [y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}})$ for each $f \in \bar{\mathcal{F}}$.
7: Run Algorithm 14 with input y^t, c^t, $\tau^t = [\tau_f^t(y_f^t)]_f$, $\bar{\mathcal{F}}$ and $[l_f, r_f]_{f\in\bar{\mathcal{F}}}$.
8: **if** Algorithm 14 return False **then**
9: **return** τ^t
10: **else**
11: Algorithm 14 return $\widetilde{\tau} \in \mathbb{R}^F, \widetilde{f} \in \bar{\mathcal{F}}, \text{sign} \in \{-1, 1\}$.
12: Apply tax $\dot{\tau}^t : \dot{\tau}^t_{\widetilde{f}} = \tau^t_{\widetilde{f}} \bigcup (y^t_{\widetilde{f}}, \widetilde{\tau}^t_{\widetilde{f}}) + \text{sign} \cdot \delta$ and $\dot{\tau}^t_f = \tau^t_f \bigcup (y^t_f, \widetilde{\tau}^t_f)$ for $f \neq \widetilde{f}$.
13: Observe $\dot{y}^t, \dot{c}^t \in \mathbb{R}^F$ as the Nash load and the Nash cost of each facility.
14: Update τ_{t+1} and $\mathcal{K}_{t+1}$ according to (7.1).
15: **end if**
16: **end for**

will be close to the true optimal tax $\tau_f^*(y_f^t)$ with error 2ϵ (Lemma F.2.3). We will apply Algorithm 14 to find the exploratory tax to gather information about unknown facilities (Line 7).

Proposition 7.5.3. *If Algorithm 14 return* False *at round t, then tax τ^t is an $6\epsilon F$-optimal tax. If Algorithm 14 output $\widetilde{\tau}^t, \widetilde{f}^t, \text{sign}^t$ at round t, then we have*

$$0 < \left| y^t_{\widetilde{f}^t} - \dot{y}^t_{\widetilde{f}^t} \right| \leq \Delta.$$

If Algorithm 14 output $\widetilde{\tau}^t, \widetilde{f}^t, \text{sign}^t$ at round t, we update the tax and the known index set by the following rule. For $u \in \{[y^t_{\widetilde{f}^t}]^+_{\mathbf{L}}, [y^t_{\widetilde{f}^t}]^-_{\mathbf{L}}\} \backslash \mathcal{K}^t_{\widetilde{f}^t}$ (this set is not empty as $\widetilde{f}^t$ is an

unknown facility), we set

$$\widehat{\tau}^{t+1}_{\widetilde{f}^t} = \widehat{\tau}^{t}_{\widetilde{f}^t} \bigcup \Big(u, \mathrm{clip}(u \cdot \frac{c^t_{\widetilde{f}^t} - \dot{c}^t_{\widetilde{f}^t}}{y^t_{\widetilde{f}^t} - \dot{y}^t_{\widetilde{f}^t}}, \widehat{\tau}^t_{\widetilde{f}^t}([y^t_{\widetilde{f}^t}]^-_{\mathcal{K}^t_{\widetilde{f}^t}}), \widehat{\tau}^t_{\widetilde{f}^t}([y^t_{\widetilde{f}^t}]^+_{\mathcal{K}^t_{\widetilde{f}^t}} \bigcup \{1\}))\Big), \tag{7.1}$$

$$\mathcal{K}^{t+1}_{\widetilde{f}^t} = \mathcal{K}^t_{\widetilde{f}^t} \bigcup \{u\}. \tag{7.2}$$

and $\widehat{\tau}^{t+1}_f = \widehat{\tau}^t_f, \mathcal{K}^{t+1}_f = \mathcal{K}^t_f$ for $f \neq \widetilde{f}^t$. Then we set $\tau^{t+1}_f(u) = \widehat{\tau}^{t+1}_f(u) + \epsilon u$ for all $f \in \mathcal{F}$ and $u \in [0,1]$.

In words, we clip the two-point estimate $u \cdot \frac{c^t_{\widetilde{f}^t} - \dot{c}^t_{\widetilde{f}^t}}{y^t_{\widetilde{f}^t} - \dot{y}^t_{\widetilde{f}^t}}$ on the left and right known index of $\widehat{\tau}^t_{\widetilde{f}^t}(u)$ so that $\widehat{\tau}^{t+1}_{\widetilde{f}^t}(u)$ is still a non-decreasing piece-wise linear approximation of the marginal cost tax τ^*_f. τ^{t+1}_f is added with an extra linear term to guarantee a strongly convex potential function (Lemma 7.5.7). As $0 < \left|y^t_f - \dot{y}^t_f\right| \leq \Delta$, the two point estimate of the gradient $\frac{c^t_f - \dot{c}^t_f}{y^t_f - \dot{y}^t_f}$ is accurate enough for $c'_f(u)$ such that $\left|\tau^{t+1}_{\widetilde{f}^t}(u) - \tau_f(u)\right| \leq \epsilon$ (Lemma F.2.1).

As $|\mathcal{K}^t_{\widetilde{f}^t}|$ increases by 1 at round t and there are F such sets with size bounded by $O(\beta/\epsilon)$, Algorithm 14 will output False within at most $O(F\beta/\epsilon)$ rounds, which implies τ^t is an ϵF-optimal tax (Proposition 7.5.3). With proper rescaling, the sample complexity for learning ϵ-optimal tax is $O(F^2\beta/\epsilon)$.

Theorem 7.5.4. *Under Assumption 7.3.1 and Assumption 7.4.3, Algorithm 13 will output a $6\epsilon F$ tax within $T \leq 2F\beta/\epsilon$ rounds. In addition, each round has at most two tax realizations.*

Remark 7.5.5. To uniformly approximate a β-smooth function, we have to know its value at $O(\beta/\epsilon)$ points [Takezawa, 2005]. For an ϵ-optimal tax, we need to estimate τ^*_f with ϵ/F accuracy as the error accumulates with all the facilities. As a result, we conjecture that $O(F^2\beta/\epsilon)$ sample complexity is tight and we leave the lower bound to future work.

Remark 7.5.6. Our algorithm can be easily adapted to the case where we have feedback other than only the equilibrium feedback. Specifically, when the tax designer obtain a non-equilibrium feedback, she can still update the optimal tax estimate if the feedback provides new information. It is possible for our algorithm to find the optimal tax even if no equilibrium feedback is provided. In addition, as long as the equilibrium can be reached after applying a tax, the algorithm can always find the optimal tax.

7.5.2 Subroutine for Finding Exploratory Tax

In this section, we describe Algorithm 14, which can find an exploratory tax that satisfies Proposition 7.5.3. The idea is we can observe another similar but different Nash equilibrium load by perturbing the tax. However, there are two challenges:

1. Perturbing the tax might change the Nash equilibrium load drastically.

2. Perturbing the tax might not change the Nash equilibrium load at all.

To resolve the first issue, we always apply taxes that have (sub)gradient lower bounded by $\epsilon > 0$. The feasible range $[l_f, r_f]$ for updating tax τ_f^t with $(y_f^t, \cdot)$ guarantees that the updated tax still maintains the subgradient lower bound. By Lemma 7.5.7, the potential function is always ϵ-strongly convex. As a result, the Nash load for any feasible tax is unique and Lipschitz w.r.t. tax perturbation. To resolve the second issue, we find the tax that makes the current Nash equilibrium on the "boundary". I.e., an additional perturbation will make the Nash equilibrium change. Intuitively, this is similar to removing the slackness in a constrained optimization problem. By Lemma 7.5.8, we can observe a different Nash load on f if we make the additional perturbation.

Lemma 7.5.7. *If the subgradient of the cost function c_f is lower bounded by $\epsilon > 0$ for all $f \in \mathcal{F}$, then the potential function $\Phi^{\text{repa}}(y)$ is ϵ-strongly convex. However, $\Phi(x)$ is not necessarily strongly convex.*

Lemma 7.5.8. *If two taxes τ and $\dot{\tau}$ only differ in facility f and the Nash loads y and $\dot{y}$ are different, then $y_f \neq \dot{y}_f$.*

Definition 7.5.9. The gap for a strategy $x \in \mathcal{X}$ with cost $c \in \mathbb{R}^F$ is defined as

$$\text{Gap}_i(x, c) = \min_{a: x_{i,a}=0} \sum_{f: f \in a} c_f - \max_{a: x_{i,a} \neq 0} \sum_{f: f \in a} c_f. \tag{7.3}$$

In the algorithm, we use $\text{Gap}_i(x, c)$ to measure the cost gap between in-support actions and off-support actions for commodity i and strategy x. If x is a Nash equilibrium and c is the Nash cost, then all of the in-support actions have the same minimal cost and

Algorithm 14 Test Tax Design (Part 1)

Initialize: Nash flow $y \in \mathbb{R}^F$, tax $\tau \in \mathbb{R}^F$, cost $c \in \mathbb{R}^F$, unknown facility set $\bar{\mathcal{F}}$, unknown facility range $[l_f, r_f]$ for $f \in \bar{F}$.
Set strategy $x \in \phi^{-1}(y)$. Compute commodity load $y_i = \phi_i(x_i) \in \mathbb{R}^F$ for $i \in [m]$.
Set $I =$ False.
if Exists $f \in \bar{\mathcal{F}}$ and $i \in [m]$ such that $0 < y_i(f) < w_i$ **then**
    return $\tau, f, 1$.
end if
for Commodity $i \in [m]$ **do**
    Let $\bar{\mathcal{F}}_i = \{f \in \bar{\mathcal{F}} : \sum_{a:f\in a} x_{i,a} = w_i\}$ and $\bar{\mathcal{F}}'_i = \{f \in \bar{\mathcal{F}} : \sum_{a:f\in a} x_{i,a} = 0\}$.
    Set $\bar{\tau} : \bar{\tau}_{\bar{\mathcal{F}}_i} = r_{\bar{\mathcal{F}}_i}, \bar{\tau}_{\bar{\mathcal{F}}'_i} = l_{\bar{\mathcal{F}}'_i}, \bar{\tau}_{\mathcal{F}\setminus(\bar{\mathcal{F}}_i \bigcup \bar{\mathcal{F}}'_i)} = \tau_{\mathcal{F}\setminus(\bar{\mathcal{F}}_i \bigcup \bar{\mathcal{F}}'_i)}$.
    if $\mathrm{Gap}_i(x, c + \bar{\tau}) < 0$ **then**
        Set $I =$ True.
        break
    end if
end for
if $I =$ False **then**
    return False.
end if
Set $\tau' = \tau$
for $f \in \bar{\mathcal{F}}_i$ **do**
    Set $\widetilde{\tau}^u : \widetilde{\tau}^u_f = u, \widetilde{\tau}^u_{\mathcal{F}\setminus\{f\}} = \tau'_{\mathcal{F}\setminus\{f\}}$.
    Set $u = \mathrm{argmax}\{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j\}$.
    if $u \leq r_f$ **then**
        return $\widetilde{\tau}^u, f, 1$.
    end if
    Set $\tau' = \widetilde{\tau}^{r_f}$.
end for
for $f \in \bar{\mathcal{F}}'_i$ **do**
    Set $\widetilde{\tau}^u : \widetilde{\tau}^u_f = u, \widetilde{\tau}^u_{\mathcal{F}\setminus\{f\}} = \tau'_{\mathcal{F}\setminus\{f\}}$.
    Set $u = \mathrm{argmin}\{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j\}$.

Algorithm 15 Test Tax Design (Part 2)

30: **if** $u \geq l_f$ **then**

31: **return** $\widetilde{\tau}^u, f, -1$.

32: **end if**

33: Set $\tau' = \widetilde{\tau}^{l_f}$.

34: **end for**

$\text{Gap}_i(x, c) \geq 0$ holds. Informally, "boundary" tax τ means that $\text{Gap}_i(x, c+\tau) = 0$ for a Nash equilibrium x and perturbing τ results in $\text{Gap}_i(x, c+\tau) < 0$, so the Nash equilibrium under the perturbed tax will be different from x.

Now we discuss how Algorithm 14 finds the "boundary" tax in detail. The input to the algorithm is the Nash flow y, the Nash cost c, the Nash tax τ, the unknown facility set $\bar{\mathcal{F}}$ and the feasible tax range $[l_f, r_f]$ for each unknown facility $f \in \bar{\mathcal{F}}$. We emphasize that here the Nash cost/tax are the values of the cost/tax function on the Nash load and they are vectors in $\mathbb{R}^F$ instead of functions. $[l_f, r_f]$ is the feasible range for the perturbed tax value at facility f. By the definition of l_f and r_f, current tax τ_f^t updated with $(y_f^t, u), u \in [l_f, r_f]$ is still a tax with subgradient lower bounded by ϵ.

For the first step, the algorithm will compute strategy $x \in \phi^{-1}(y)$ as the Nash equilibrium strategy (Line 2). If there exists an unknown facility f and commodity i such that not all load of commodity i is using f or not using f (Line 4), then perturbing the tax at facility f will make x not longer a Nash equilibrium as in-support actions have different costs.

Otherwise, for each unknown facility f and commodity i, either all of the load is using f or all of the load is not using f. As a result, in-support actions always have the same cost after perturbing the tax. For the next step, we verify if there exists a tax within the feasible ranges for unknown facilities that makes x not a Nash equilibrium. However, there does not exist a universal worst-case tax that can verify if x is always a Nash equilibrium or not.

Fortunately, the worst-case tax has a closed form for each commodity separately: the taxes for facilities used by all of the Nash load would be the upper bound r_f and the taxes for facilities used by none of the Nash load would be the lower bound l_f, thus maximizing

the cost for in-support actions and minimizing the cost for off-support actions (Line 9). For each commodity i, we apply the corresponding worst-case tax and check if the in-support actions are still the optimal actions (Line 10). If for all commodities, the in-support actions are optimal under the worst-case tax, then for any tax within the feasible range, y is the Nash load and the algorithm will output False (Line 16). As τ^* is approximately within the range, y^t approximately minimizes the social cost (Lemma F.2.5).

Otherwise, the algorithm finds commodity i such that x is not the Nash equilibrium under the worst-case tax (Line 11). For the last step, we gradually transform the initial tax to this worst-case tax and stop when x is not the Nash equilibrium for some commodity. Specifically, the algorithm iteratively changes the tax in the unknown facility set $\bar{\mathcal{F}} = \bar{\mathcal{F}}_i \bigcup \bar{\mathcal{F}}_i'$ (Line 19 and Line 27) to the worst-case tax.

For facility $f \in \bar{\mathcal{F}}_i$, the algorithm finds the boundary tax for facility f that satisfies

$$u = \operatorname*{argmax}_u \{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j \in [m]\}.$$

If $u \leq r_f$, we will output $\widetilde{\tau}^u, f, 1$. By the definition of u, if we further increase u, one of the gaps will become negative and x is no longer the Nash equilibrium. Otherwise, all feasible taxes for f have a nonnegative gap for all commodities, which means x is still the Nash equilibrium, and we continue for the next facility. After enumerating all the facilities in $\bar{\mathcal{F}}_i$, we enumerate $\bar{\mathcal{F}}_i'$ in the same way. Eventually, the tax is transformed into the worst-case tax with negative gap for commodity i, so this process will end and output $\widetilde{\tau}^u, f, \mathrm{sign}$ such that $\widetilde{\tau}^u$ is the tax that makes the Nash equilibrium on the boundary, f is the facility to perturb and sign is the direction to perturb the tax at f.

7.6 Conclusion

We proposed the first algorithm with polynomial sample complexity for learning optimal tax in nonatomic congestion games. The algorithm leverages several novel designs to exploit the special structure of congestion games, which can also be implemented efficiently. Below we list a few potential future research directions:

1. Relaxing the Nash equilibrium assumption to players following no-regret dynamics or quantal response equilibrium.

2. Design algorithms that do not require prior knowledge of the smoothess coefficient.

3. Generalize the algorithm to atomic congestion games.

BIBLIOGRAPHY

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Kenshi Abe and Yusuke Kaneko. Off-policy exploitability-evaluation in two-player zero-sum markov games. *arXiv preprint arXiv:2007.02141*, 2020.

Ilan Adler. The equivalence of linear programs and zero-sum games. *International Journal of Game Theory*, 42(1):165, 2013.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in neural information processing systems*, 33:13399–13412, 2020a.

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning, 2020b.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20095–20107. Curran Associates, Inc., 2020c. URL `https://proceedings.neurips.cc/paper/2020/file/e894d787e2fd6c133af47140aa156f00-Paper.pdf`.

Alekh Agarwal, Sham Kakade, and Lin F Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *Conference on Learning Theory*, pages 67–83. PMLR, 2020d.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020e.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift, 2020f.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Alekh Agarwal, Yujia Jin, and Tong Zhang. Vo q l: Towards optimal regret in model-free rl with nonlinear function approximation. *arXiv preprint arXiv:2212.06069*, 2022.

Hayder AA Al-Kashoash, Maryam Hafeez, and Andrew H Kemp. Congestion control for 6lowpan networks: A game theoretic framework. *IEEE internet of things journal*, 4(3): 760–771, 2017.

Ioannis Anagnostides, Ioannis Panageas, Gabriele Farina, and Tuomas Sandholm. On the convergence of no-regret learning dynamics in time-varying games. *arXiv preprint arXiv:2301.11241*, 2023.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Peter Auer, Yifang Chen, Pratik Gajane, Chung-Wei Lee, Haipeng Luo, Ronald Ortner, and Chen-Yu Wei. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *Conference on Learning Theory*, pages 159–163. PMLR, 2019a.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158. PMLR, 2019b.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*, pages 551–560. PMLR, 2020a.

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning*, pages 551–560. PMLR, 2020b.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34: 25799–25811, 2021.

Yu Bai, Chi Jin, Song Mei, and Tiancheng Yu. Near-optimal learning of extensive-form games with imperfect information. In *International Conference on Machine Learning*, pages 1337–1382. PMLR, 2022.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.

Pia Bergendorff, Donald W Hearn, and Motakuri V Ramana. *Congestion toll pricing of traffic networks*. Springer, 1997.

Sebastian Bervoets, Mario Bravo, and Mathieu Faure. Learning with minimal information in continuous games. *Theoretical Economics*, 15(4):1471–1508, 2020.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27, 2014.

Vittorio Bilò and Cosimo Vinci. Dynamic taxes for polynomial congestion games. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–36, 2019.

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014.

Dietrich Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12: 258–268, 1968.

Mario Bravo, David Leslie, and Panayotis Mertikopoulos. Bandit learning in concave n-person games. *Advances in Neural Information Processing Systems*, 31, 2018.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365 (6456):885–890, 2019.

Noam Brown, Tuomas Sandholm, and Strategic Machine. Libratus: The Superhuman AI for No-Limit Poker. In *IJCAI*, pages 5226–5228, 2017.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Jacob Buckman, Carles Gelada, and Marc G. Bellemare. The Importance of Pessimism in Fixed-Dataset Policy Optimization. *arXiv:2009.06799 [cs]*, November 2020. URL `http://arxiv.org/abs/2009.06799`. arXiv: 2009.06799.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization, 2020.

Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Taxes for linear atomic congestion games. *ACM Transactions on Algorithms (TALG)*, 7(1):1–31, 2010.

Adrian Rivera Cardoso, Jacob Abernethy, He Wang, and Huan Xu. Competing against nash equilibria in adversarially changing zero-sum games. In *International Conference on Machine Learning*, pages 921–930. PMLR, 2019.

Shicong Cen, Fan Chen, and Yuejie Chi. Independent natural policy gradient methods for potential games: Finite-time global convergence with entropy regularization. *arXiv preprint arXiv:2204.05466*, 2022a.

Shicong Cen, Yuejie Chi, Simon S Du, and Lin Xiao. Faster last-iterate convergence of policy optimization in zero-sum markov games. *arXiv preprint arXiv:2210.01050*, 2022b.

Fan Chen, Song Mei, and Yu Bai. Unified algorithms for rl with decision-estimation coefficients: No-regret, pac, and reward-free learning. *arXiv preprint arXiv:2209.11745*, 2022.

Jinglin Chen and Nan Jiang. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. *arXiv preprint arXiv:2203.13935*, 2022.

Minshuo Chen, Yan Li, Ethan Wang, Zhuoran Yang, Zhaoran Wang, and Tuo Zhao. Pessimism meets invariance: Provably efficient offline mean-field multi-agent rl. *Advances in Neural Information Processing Systems*, 34, 2021a.

Po-An Chen and Chi-Jen Lu. Playing congestion games with bandit feedbacks. In *AAMAS*, pages 1721–1722, 2015.

Po-An Chen and Chi-Jen Lu. Generalized mirror descents in congestion games. *Artificial Intelligence*, 241:217–243, 2016.

Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272. IEEE, 2006.

Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.

Xiaoyu Chen, Jiachen Hu, Lihong Li, and Liwei Wang. Efficient reinforcement learning in factored mdps with application to constrained rl. *arXiv preprint arXiv:2008.13319*, 2020.

Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *Conference on Learning Theory*, pages 696–726. PMLR, 2019.

Zixiang Chen, Dongruo Zhou, and Quanquan Gu. Almost optimal algorithms for two-player markov games with linear function approximation. *arXiv preprint arXiv:2102.07404*, 2021b.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism. In *International Conference on Machine Learning*, pages 1843–1854. PMLR, 2020.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Hedging the drift: Learning to optimize under nonstationarity. *Management Science*, 68(3):1696–1713, 2022.

Yun Kuen Cheung and Georgios Piliouras. Chaos, extremism and optimism: Volume analysis of learning in games. *Advances in Neural Information Processing Systems*, 33:9039–9049, 2020.

Yong Huat Chew, Boon-Hee Soong, et al. *Potential game theory*. Springer, 2016.

Alon Cohen, Haim Kaplan, Tomer Koren, and Yishay Mansour. Online markov decision processes with aggregate bandit feedback. In *Conference on Learning Theory*, pages 1301–1329. PMLR, 2021a.

Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021b.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153:235–256, 2007.

Roberto Cominetti, Emerson Melo, and Sylvain Sorin. A payoff-based learning procedure and its application to traffic games. *Games and Economic Behavior*, 70(1):71–83, 2010.

Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. Penalty-regulated dynamics and robust learning procedures in games. *Mathematics of Operations Research*, 40(3): 611–633, 2015.

Qiwen Cui and Simon S Du. When is offline two-player zero-sum markov game solvable? *arXiv preprint arXiv:2201.03522*, 2022a.

Qiwen Cui and Simon S Du. Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. *arXiv preprint arXiv:2206.00159*, 2022b.

Qiwen Cui and Lin F Yang. Minimax sample complexity for turn-based stochastic game. *arXiv preprint arXiv:2011.14267*, 2020.

Qiwen Cui and Lin F Yang. Minimax sample complexity for turn-based stochastic game. In *Uncertainty in Artificial Intelligence*, pages 1496–1504. PMLR, 2021.

Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S Du. Learning in congestion games with bandit feedback. *Advances in Neural Information Processing Systems*, 35:11009–11022, 2022.

Qiwen Cui, Kaiqing Zhang, and Simon S Du. Breaking the curse of multiagents in a large state space: Rl in markov games with independent linear function approximation. *arXiv preprint arXiv:2302.03673*, 2023.

Qiwen Cui, Maryam Fazel, and Simon S Du. Learning optimal tax design in nonatomic congestion games. *arXiv preprint arXiv:2402.07437*, 2024.

Leello Dadi, Ioannis Panageas, Stratis Skoulakis, Luca Viano, and Volkan Cevher. Polynomial convergence of bandit no-regret dynamics in congestion games. *arXiv preprint arXiv:2401.09628*, 2024.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019.

Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 9(3):1–35, 2013.

Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 235–254. SIAM, 2011.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Constantinos Daskalakis, Dylan J Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33:5527–5540, 2020.

Constantinos Daskalakis, Dylan J. Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning, 2021.

Constantinos Daskalakis, Noah Golowich, and Kaiqing Zhang. The complexity of markov equilibrium in stochastic games. *arXiv preprint arXiv:2204.03991*, 2022.

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020a.

Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 19, 2020b.

Zhihong Deng, Zuyue Fu, Lingxiao Wang, Zhuoran Yang, Chenjia Bai, Zhaoran Wang, and Jing Jiang. SCORE: Spurious COrrelation REduction for Offline Reinforcement Learning. *arXiv:2110.12468 [cs]*, October 2021. URL `http://arxiv.org/abs/2110.12468`. arXiv: 2110.12468.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo R. Jovanović. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence, 2022.

Jing Dong, Jingyu Wu, Siwei Wang, Baoxiang Wang, and Wei Chen. Taming the exponential action set: Sublinear regret and fast convergence to nash equilibrium in online congestion games. *arXiv preprint arXiv:2306.13673*, 2023.

Zehao Dou, Zhuoran Yang, Zhaoran Wang, and Simon S Du. Gap-dependent bounds for two-player markov games. *arXiv preprint arXiv:2107.00685*, 2021.

Dmitriy Drusvyatskiy, Maryam Fazel, and Lillian J Ratliff. Improved rates for derivative free gradient play in strongly monotone games. In *Proc. IEEE Conference on Decision and Control*, 2022.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019.

Simon S. Du, Sham M. Kakade, Jason D. Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl, 2021.

Yaqi Duan, Chi Jin, and Zhiyuan Li. Risk Bounds and Rademacher Complexity in Batch Reinforcement Learning. *arXiv:2103.13883 [cs, stat]*, March 2021. URL `http://arxiv.org/abs/2103.13883`. arXiv: 2103.13883.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly Robust Policy Evaluation and Optimization. *Statistical Science*, 29(4), November 2014. ISSN 0883-4237. doi: 10.1214/14-STS500. URL `http://arxiv.org/abs/1503.02834`. arXiv: 1503.02834.

Stéphane Durand. *Analysis of Best Response Dynamics in Potential Games*. PhD thesis, Université Grenoble Alpes, 2018.

Benoit Duvocelle, Panayotis Mertikopoulos, Mathias Staudigl, and Dries Vermeulen. Multi-agent online learning in time-varying games. *Mathematics of Operations Research*, 2022.

Editor. Hyphenation exception log. *TUGboat*, 7(3):145, 1986.

Yonathan Efroni, Lior Shani, Aviv Rosenberg, and Shie Mannor. Optimistic policy optimization with bandit feedback. *arXiv preprint arXiv:2002.08243*, 2020.

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7288–7295, 2021.

Liad Erez, Tal Lancewicki, Uri Sherman, Tomer Koren, and Yishay Mansour. Regret minimization and convergence to equilibria in general-sum markov games. *arXiv preprint arXiv:2207.14211*, 2022.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More Robust Doubly Robust Off-policy Evaluation. *arXiv:1802.03493 [cs]*, May 2018. URL `http://arxiv.org/abs/1802.03493`. arXiv: 1802.03493.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Dimitris Fotakis, Spyros Kontogiannis, Elias Koutsoupias, Marios Mavronicolas, and Paul Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *International Colloquium on Automata, Languages, and Programming*, pages 123–134. Springer, 2002.

Roy Fox, Stephen McAleer, Will Overman, and Ioannis Panageas. Independent natural policy gradient always converges in markov potential games. *arXiv preprint arXiv:2110.10614*, 2021.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 174–188, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24412-4.

Angeliki Giannou, Emmanouil Vasileios Vlatakis-Gkaragkounis, and Panayotis Mertikopoulos. The convergence rate of regularized learning in games: From bandits and uncertainty to optimism and beyond. In *NeurIPS 2021-35th International Conference on Neural Information Processing Systems*, pages 1–28, 2021.

Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. *Advances in neural information processing systems*, 33:20766–20778, 2020.

Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, 1994.

Panagiotis D Grontas, Giuseppe Belgioioso, Carlo Cenedese, Marta Fochesato, John Lygeros, and Florian Dörfler. Big hype: Best intervention in games via distributed hypergradient descent. *IEEE Transactions on Automatic Control*, 2024.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates. *arXiv:1610.00633 [cs]*, November 2016. URL `http://arxiv.org/abs/1610.00633`. arXiv: 1610.00633.

Yingya Guo, Qi Tang, Yulong Ma, Han Tian, and Kai Chen. Distributed traffic engineering in hybrid software defined networks: A multi-agent reinforcement learning framework, 2023.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Ben Hambly, Renyuan Xu, and Huining Yang. Policy gradient methods find the nash equilibrium in n-player general-sum linear-quadratic games, 2021.

Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

Tobias Harks, Ingo Kleinert, Max Klimm, and Rolf H Möhring. Computing network tolls with support constraints. *Networks*, 65(3):262–285, 2015.

Elad Hazan and Satyen Kale. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In *Conference on Learning Theory*, pages 1877–1893. PMLR, 2020.

Amélie Heliou, Johanne Cohen, and Panayotis Mertikopoulos. Learning with bandit feedback in potential games. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/39ae2ed11b14a4ccb41d35e9d1ba5d11-Paper.pdf`.

Amélie Heliou, Johanne Cohen, and Panayotis Mertikopoulos. Learning with bandit feedback in potential games. *Advances in Neural Information Processing Systems*, 30, 2017b.

Yu-Guan Hsieh, Kimon Antonakopoulos, Volkan Cevher, and Panayotis Mertikopoulos. No-regret learning in games with noisy feedback: Faster rates and adaptivity via learning rate separation. *arXiv preprint arXiv:2206.06015*, 2022.

Baihe Huang, Jason D Lee, Zhaoran Wang, and Zhuoran Yang. Towards general function approximation in zero-sum markov games. *arXiv preprint arXiv:2107.14702*, 2021.

Jiawei Huang and Nan Jiang. From Importance Sampling to Doubly Robust Policy Gradi-

ent. *arXiv:1910.09066 [cs, stat]*, June 2020. URL http://arxiv.org/abs/1910.09066. arXiv: 1910.09066.

Christian Ibars, Monica Navarro, and Lorenza Giupponi. Distributed demand management in smart grid with a congestion game. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 495–500. IEEE, 2010.

Shinji Ito. A tight lower bound and efficient reduction for swap regret. *Advances in Neural Information Processing Systems*, 33:18550–18559, 2020.

Zeyu Jia, Lin F Yang, and Mengdi Wang. Feature-based q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*, 2019.

Haozhe Jiang, Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S Du. Offline congestion games: How feedback type affects data coverage requirement. *arXiv preprint arXiv:2210.13396*, 2022.

Haozhe Jiang, Qiwen Cui, Zhihan Xiong, Maryam Fazel, and Simon S Du. A black-box approach for non-stationary multi-agent reinforcement learning. *arXiv preprint arXiv:2306.07465*, 2023.

Jiechuan Jiang and Zongqing Lu. Offline decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:2108.01832*, 2021.

Nan Jiang and Jiawei Huang. Minimax Value Interval for Off-Policy Evaluation and Policy Optimization. *arXiv:2002.02081 [cs, math, stat]*, November 2020. URL http://arxiv.org/abs/2002.02081. arXiv: 2002.02081.

Nan Jiang and Lihong Li. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. *arXiv:1511.03722 [cs, stat]*, May 2016. URL http://arxiv.org/abs/1511.03722. arXiv: 1511.03722.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021a.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021b.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning – a simple, efficient, decentralized algorithm for multiagent rl, 2021c.

Chi Jin, Qinghua Liu, and Tiancheng Yu. The power of exploiter: Provable multi-agent rl in large state spaces. In *International Conference on Machine Learning*, pages 10251–10279. PMLR, 2022.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021d.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is Pessimism Provably Efficient for Offline RL? *arXiv:2012.15085 [cs, math, stat]*, May 2021e. URL `http://arxiv.org/abs/2012.15085`. arXiv: 2012.15085.

Ramesh Johari and John N Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

Gregory Kahn, Pieter Abbeel, and Sergey Levine. Badgr: An autonomous self-supervised learning-based navigation system. *IEEE Robotics and Automation Letters*, 6(2):1312–1319, 2021. Publisher: IEEE.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning.* University of London, University College London (United Kingdom), 2003.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and others. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

Yun Geon Kim, Seokgi Lee, Jiyeon Son, Heechul Bae, and Byung Do Chung. Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. *Journal of Manufacturing Systems*, 57:440–450, 2020.

Robert Kleinberg, Georgios Piliouras, and Éva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 533–542, 2009.

Donald E. Knuth. *The TEX book.* Addison-Wesley, 1984.

Donald E. Knuth. *Computer Modern Typefaces.* Addison-Wesley, 1986a.

Donald E. Knuth. *The Metafont book.* Addison-Wesley, 1986b.

Donald E. Knuth. *TEX: The Program.* Addison-Wesley, 1986c.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. Publisher: SAGE Publications Sage UK: London, England.

Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pages 404–413. Springer, 1999.

Tadashi Kozuno, Pierre Ménard, Rémi Munos, and Michal Valko. Model-free learning for two-player zero-sum partially observable markov games with perfect recall. *arXiv preprint arXiv:2106.06279*, 2021.

Walid Krichene, Benjamin Drighès, and Alexandre Bayen. On the convergence of no-regret learning in selfish routing. In *International Conference on Machine Learning*, pages 163–171. PMLR, 2014.

Walid Krichene, Benjamin Drighès, and Alexandre M Bayen. Online learning of nash equilibria in congestion games. *SIAM Journal on Control and Optimization*, 53(2):1056–1081, 2015.

Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 2nd edition, 1994.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020.

Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in markov potential games, 2021.

David S Leslie. *Reinforcement learning in games*. PhD thesis, University of Bristol, 2004.

David S Leslie and Edmund J Collins. Individual q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

David S Leslie and Edmund J Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory: Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings 2*, pages 250–262. Springer, 2009.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv:2005.01643 [cs, stat]*, November 2020. URL `http://arxiv.org/abs/2005.01643`. arXiv: 2005.01643.

Gen Li, Laixi Shi, Yuxin Chen, Yuantao Gu, and Yuejie Chi. Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Gen Li, Yuejie Chi, Yuting Wei, and Yuxin Chen. Minimax-optimal multi-agent rl in zero-sum markov games with a generative model. *arXiv preprint arXiv:2208.10458*, 2022a.

Gen Li, Laixi Shi, Yuxin Chen, Yuejie Chi, and Yuting Wei. Settling the sample complexity of model-based offline reinforcement learning. *arXiv preprint arXiv:2204.05275*, 2022b.

Jiayang Li, Jing Yu, Yu Nie, and Zhaoran Wang. End-to-end learning and intervention in games. *Advances in Neural Information Processing Systems*, 33:16653–16665, 2020.

Jiayang Li, Jing Yu, Qianni Wang, Boyi Liu, Zhaoran Wang, and Yu Marco Nie. Differentiable bilevel programming for stackelberg congestion games. *arXiv preprint arXiv:2209.07618*, 2022c.

Jiayang Li, Jing Yu, Boyi Liu, Yu Nie, and Zhaoran Wang. Achieving hierarchy-free approximation for bilevel programs with equilibrium constraints. In *International Conference on Machine Learning*, pages 20312–20335. PMLR, 2023.

Lihong Li, Remi Munos, and Csaba Szepesvari. On Minimax Optimal Offline Policy Evaluation. *arXiv:1409.3653 [cs]*, September 2014. URL `http://arxiv.org/abs/1409.3653`. arXiv: 1409.3653.

Boyi Liu, Jiayang Li, Zhuoran Yang, Hoi-To Wai, Mingyi Hong, Yu Nie, and Zhaoran Wang. Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence. *Advances in Neural Information Processing Systems*, 35:29001–29013, 2022a.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation. *arXiv:1810.12429 [cs, stat]*, October 2018. URL `http://arxiv.org/abs/1810.12429`. arXiv: 1810.12429.

Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based re-

inforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021a.

Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021b.

Qinghua Liu, Csaba Szepesvári, and Chi Jin. Sample-efficient reinforcement learning of partially observable markov games. *arXiv preprint arXiv:2206.01315*, 2022b.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo. Learning parametric closed-loop policies for markov potential games. *arXiv preprint arXiv:1802.00899*, 2018.

Chinmay Maheshwari, S Shankar Sasty, Lillian Ratliff, and Eric Mazumdar. Convergent first-order methods for bi-level optimization and stackelberg games. *arXiv preprint arXiv:2302.01421*, 2023.

Weichao Mao, Tamer Basar, Lin F Yang, and Kaiqing Zhang. Decentralized cooperative multi-agent reinforcement learning with exploration. *arXiv preprint arXiv:2110.05707*, 2021a.

Weichao Mao, Kaiqing Zhang, Ruihao Zhu, David Simchi-Levi, and Tamer Basar. Near-optimal model-free reinforcement learning in non-stationary episodic mdps. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7447–7458. PMLR, 18–24 Jul 2021b. URL `https://proceedings.mlr.press/v139/mao21b.html`.

Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049. PMLR, 2022.

Jason R Marden. State based potential games. *Automatica*, 48(12):3075–3088, 2012.

Jason R Marden and Tim Roughgarden. Generalized efficiency bounds in distributed resource allocation. *IEEE Transactions on Automatic Control*, 59(3):571–584, 2014.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Linghui Meng, Muning Wen, Yaodong Yang, Chenyang Le, Xiyun Li, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, and Bo Xu. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *arXiv preprint arXiv:2112.02845*, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020.

Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14 (1):124–143, 1996.

Ofir Nachum and Bo Dai. Reinforcement Learning via Fenchel-Rockafellar Duality. *arXiv:2001.01866 [cs, stat]*, January 2020. URL `http://arxiv.org/abs/2001.01866`. arXiv: 2001.01866.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

Gergely Neu and Julia Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Conference on Learning Theory*, pages 3049–3068. PMLR, 2020.

Chengzhuo Ni, Yuda Song, Xuezhou Zhang, Chi Jin, and Mengdi Wang. Representation learning for general-sum low-rank markov games. *arXiv preprint arXiv:2210.16976*, 2022.

Hukukane Nikaidô and Kazuo Isoda. Note on non-cooperative convex games. *Pacific Journal of Mathematics*, 5(S1):807–815, 1955.

Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007a.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge university press, 2007b.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Dario Paccagnan and Martin Gairing. In congestion games, taxes achieve optimal approximation. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 743–744, 2021.

Dario Paccagnan, Rahul Chandan, Bryce L Ferguson, and Jason R Marden. Optimal taxes in atomic congestion games. *ACM Transactions on Economics and Computation (TEAC)*, 9(3):1–33, 2021.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. *arXiv preprint arXiv:2111.11188*, 2021.

Ioannis Panageas, Stratis Skoulakis, Luca Viano, Xiao Wang, and Volkan Cevher. Semi bandit dynamics in congestion games: Convergence to nash equilibrium and no-regret guarantees. In *International Conference on Machine Learning*, pages 26904–26930. PMLR, 2023.

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2149–2156, 2019.

Julien Perolat, Bruno Scherrer, Bilal Piot, and Olivier Pietquin. Approximate dynamic programming for two-player zero-sum markov games. In *International Conference on Machine Learning*, pages 1321–1329. PMLR, 2015.

Julien Pérolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning nash equilibrium for general-sum markov games from batch data. In *Artificial Intelligence and Statistics*, pages 232–241. PMLR, 2017.

Arthur Cecil Pigou. *Wealth and welfare*. Macmillan and Company, limited, 1912.

Jorge I Poveda, Miroslav Krstić, and Tamer Basar. Fixed-time seeking and tracking of time-varying nash equilibria in noncooperative games. In *2022 American Control Conference (ACC)*, pages 794–799. IEEE, 2022.

Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

Arvind Raghunathan, Anoop Cherian, and Devesh Jha. Game theoretic optimization via

gradient-based nikaido-isoda function. In *International Conference on Machine Learning*, pages 5291–5300. PMLR, 2019.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 26, 2013.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34, 2021a.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism. *arXiv:2103.12021 [cs, math, stat]*, March 2021b. URL `http://arxiv.org/abs/2103.12021`. arXiv: 2103.12021.

Tongzheng Ren, Jialian Li, Bo Dai, Simon S Du, and Sujay Sanghavi. Nearly horizon-free offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021a.

Tongzheng Ren, Jialian Li, Bo Dai, Simon S. Du, and Sujay Sanghavi. Nearly Horizon-Free Offline Reinforcement Learning. *arXiv:2103.14077 [cs, stat]*, October 2021b. URL `http://arxiv.org/abs/2103.14077`. arXiv: 2103.14077.

Aviv Rosenberg and Yishay Mansour. Oracle-efficient regret minimization in factored mdps with unknown structure. *Advances in Neural Information Processing Systems*, 34, 2021.

Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

Tim Roughgarden. *Selfish routing and the price of anarchy*. MIT press, 2005.

Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7):78–86, 2010.

Tim Roughgarden and Éva Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and economic behavior*, 47(2):389–403, 2004.

Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 258–265. IEEE, 2016.

Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. *ACM SIGecom Exchanges*, 15(2):45–49, 2017.

Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized Q-learning in zero-sum markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

William Shakespeare. *Hamlet*. F.S. Crofts & Co., Inc., NY, 1946. Act I, Scene 3, Lines 70-72, are apropos.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

Aaron Sidford, Mengdi Wang, Lin Yang, and Yinyu Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pages 2992–3002. PMLR, 2020.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. ISSN 1476-4687. doi: 10.1038/nature16961. URL `https://doi.org/10.1038/nature16961`.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Ziang Song, Song Mei, and Yu Bai. When can we learn general-sum markov games with a large number of players sample-efficiently? *arXiv preprint arXiv:2110.04184*, 2021a.

Ziang Song, Song Mei, and Yu Bai. When can we learn general-sum markov games with a large number of players sample-efficiently?, 2021b.

Ziang Song, Song Mei, and Yu Bai. Sample-efficient learning of correlated equilibria in extensive-form games. *arXiv preprint arXiv:2205.07223*, 2022.

Spivak, M.D., Ph.D. *PCTEX Manual*. Personal TEX, Inc., CA, 1985.

Spivak, M.D., Ph.D. *The Joy of TEX*. American Mathematical Society, RI, 1986.

Jayakumar Subramanian, Amit Sinha, and Aditya Mahajan. Robustness and sample complexity of model-based marl for general-sum markov games. *arXiv preprint arXiv:2110.02355*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Brian Swenson, Ryan Murray, and Soummya Kar. On best-response dynamics in potential games. *SIAM Journal on Control and Optimization*, 56(4):2734–2767, 2018a.

Brian Swenson, Ryan Murray, and Soummya Kar. On best-response dynamics in potential games, 2018b.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 28, 2015.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887, 2005.

Kunio Takezawa. *Introduction to nonparametric regression*. John Wiley & Sons, 2005.

Philip S. Thomas and Emma Brunskill. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. *arXiv:1604.00923 [cs]*, April 2016. URL `http://arxiv.org/abs/1604.00923`. arXiv: 1604.00923.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-Confidence Off-Policy Evaluation. page 7.

Yi Tian, Jian Qian, and Suvrit Sra. Towards minimax optimal reinforcement learning in factored markov decision processes. *Advances in Neural Information Processing Systems*, 33:19896–19907, 2020.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021a.

Masatoshi Uehara and Wen Sun. Pessimistic Model-based Offline Reinforcement Learning under Partial Coverage. *arXiv:2107.06226 [cs, stat]*, October 2021b. URL `http://arxiv.org/abs/2107.06226`. arXiv: 2107.06226.

Masatoshi Uehara, Jiawei Huang, and Nan Jiang. Minimax Weight and Q-Function Learning for Off-Policy Evaluation. *arXiv:1910.12809 [cs, stat]*, October 2020. URL `http://arxiv.org/abs/1910.12809`. arXiv: 1910.12809.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation Learning for Online and Offline RL in Low-rank MDPs. *arXiv:2110.04652 [cs, stat]*, November 2021. URL `http://arxiv.org/abs/2110.04652`. arXiv: 2110.04652.

Graduate School University of Washington. Format guidelines for theses and dissertations, 2012.

Dirck Van Vliet. The frank-wolfe algorithm for equilibrium traffic assignment viewed as a variational inequality. *Transportation Research Part B: Methodological*, 21(1):87–89, 1987.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, and

others. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2, 2019a.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019b.

Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*, pages 22430–22456. PMLR, 2022.

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

Yuanhao Wang, Ruosong Wang, and Sham Kakade. An exponential lower bound for linearly realizable mdp with constant suboptimality gap. *Advances in Neural Information Processing Systems*, 34:9521–9533, 2021.

Yuanhao Wang, Qinghua Liu, Yu Bai, and Chi Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent rl with function approximation. *arXiv preprint arXiv:2302.06606*, 2023.

John Glen Wardrop. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362, 1952.

Chen-Yu Wei and Haipeng Luo. Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *Conference on Learning Theory*, pages 4300–4354. PMLR, 2021.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. *arXiv preprint arXiv:2006.09517*, 2020.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive markov games, 2021.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021.

Zheng Wen and Benjamin Van Roy. Efficient reinforcement learning in deterministic systems with value function generalization. *Mathematics of Operations Research*, 42(3):762–782, 2017.

Lin Xiao. On the convergence rates of policy gradient methods, 2022.

Qiaomin Xie, Yudong Chen, Zhaoran Wang, and Zhuoran Yang. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, pages 3674–3682. PMLR, 2020a.

Qiaomin Xie, Yudong Chen, Zhaoran Wang, and Zhuoran Yang. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on Learning Theory*, pages 3674–3682. PMLR, 2020b.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021a.

Tengyang Xie and Nan Jiang. Batch Value-function Approximation with Only Realizability. *arXiv:2008.04990 [cs, stat]*, June 2021b. URL `http://arxiv.org/abs/2008.04990`. arXiv: 2008.04990.

Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards Optimal Off-Policy Evaluation for

Reinforcement Learning with Marginalized Importance Sampling. *arXiv:1906.03393 [cs, stat]*, March 2020c. URL `http://arxiv.org/abs/1906.03393`. arXiv: 1906.03393.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021a.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34, 2021b.

Wei Xiong, Han Zhong, Chengshuai Shi, Cong Shen, Liwei Wang, and Tong Zhang. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent mdp and markov game. *arXiv preprint arXiv:2205.15512*, 2022.

Haike Xu, Tengyu Ma, and Simon Du. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. In *Conference on Learning Theory*, pages 4438–4472. PMLR, 2021.

Ziping Xu and Ambuj Tewari. Reinforcement learning in factored mdps: Oracle-efficient algorithms and tighter regret bounds for the non-episodic setting. *Advances in Neural Information Processing Systems*, 33:18226–18236, 2020.

Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. Model-based reinforcement learning is minimax-optimal for offline zero-sum markov games. *arXiv preprint arXiv:2206.04044*, 2022.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

Yuepeng Yang and Cong Ma. $O(T^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum markov games. *arXiv preprint arXiv:2209.12430*, 2022.

Ming Yin and Yu-Xiang Wang. Towards instance-optimal offline reinforcement learning with pessimism. *Advances in neural information processing systems*, 34, 2021a.

Ming Yin and Yu-Xiang Wang. Towards Instance-Optimal Offline Reinforcement Learning with Pessimism. *arXiv:2110.08695 [cs, stat]*, October 2021b. URL `http://arxiv.org/abs/2110.08695`. arXiv: 2110.08695.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. *arXiv preprint arXiv:2007.03760*, 2020a.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-Optimal Provable Uniform Convergence in Offline Policy Evaluation for Reinforcement Learning. *arXiv:2007.03760 [cs, stat]*, December 2020b. URL `http://arxiv.org/abs/2007.03760`. arXiv: 2007.03760.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal offline reinforcement learning via double variance reduction. *Advances in neural information processing systems*, 34, 2021a.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-Optimal Offline Reinforcement Learning via Double Variance Reduction. *arXiv:2102.01748 [cs, stat]*, February 2021b. URL `http://arxiv.org/abs/2102.01748`. arXiv: 2102.01748.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Andrea Zanette. Exponential Lower Bounds for Batch Reinforcement Learning: Batch RL can be Exponentially Harder than Online RL. *arXiv:2012.08005 [cs]*, June 2021. URL `http://arxiv.org/abs/2012.08005`. arXiv: 2012.08005.

Andrea Zanette and Martin J Wainwright. Stabilizing q-learning with linear architectures for provably efficient learning. *arXiv preprint arXiv:2206.00796*, 2022.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pages 10978–10989. PMLR, 2020.

Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation, 2021a.

Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021b.

Andrea Zanette, Martin J. Wainwright, and Emma Brunskill. Provable Benefits of Actor-Critic Methods for Offline Reinforcement Learning. *arXiv:2108.08812 [cs]*, August 2021c. URL `http://arxiv.org/abs/2108.08812`. arXiv: 2108.08812.

Kaiqing Zhang, Sham Kakade, Tamer Basar, and Lin Yang. Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity. *Advances in Neural Information Processing Systems*, 33:1166–1178, 2020a.

Kaiqing Zhang, Sham M Kakade, Tamer Başar, and Lin F Yang. Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461*, 2020b.

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021a.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents. *IEEE Transactions on Automatic Control*, 66(12):5925–5940, 2021b.

Mengxiao Zhang, Peng Zhao, Haipeng Luo, and Zhi-Hua Zhou. No-regret learning in time-varying zero-sum games. In *International Conference on Machine Learning*, pages 26772–26808. PMLR, 2022a.

Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020c.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. GenDICE: Generalized Offline Estimation of Stationary Values. *arXiv:2002.09072 [cs, stat]*, February 2020d. URL `http://arxiv.org/abs/2002.09072`. arXiv: 2002.09072.

Runyu Zhang, Zhaolin Ren, and Na Li. Gradient play in stochastic games: stationary points, convergence, and sample complexity. *arXiv preprint arXiv:2106.00198*, 2021c.

Runyu Zhang, Qinghua Liu, Huan Wang, Caiming Xiong, Na Li, and Yu Bai. Policy optimization for markov games: Unified framework and faster convergence. *arXiv preprint arXiv:2206.02640*, 2022b.

Siyuan Zhang and Nan Jiang. Towards Hyperparameter-free Policy Selection for Offline Reinforcement Learning. page 19.

Geng Zhao, Banghua Zhu, Jiantao Jiao, and Michael Jordan. Online learning in stackelberg games with an omniscient follower. In *International Conference on Machine Learning*, pages 42304–42316. PMLR, 2023.

Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 746–755. PMLR, 2020.

Yulai Zhao, Yuandong Tian, Jason D. Lee, and Simon S. Du. Provably efficient policy gradient methods for two-player zero-sum markov games, 2021.

Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopic followers? *arXiv preprint arXiv:2112.13521*, 2021.

Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran

Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. *arXiv preprint arXiv:2202.07511*, 2022.

Appendix A

DEFERRED CONTENTS FROM CHAPTER ??

A.1 Algorithm

Algorithm 16 Pessimistic Nash Value Iteration with Reference Advantage Decomposition

Input: Dataset $\mathcal{D} = \left\{(s_h^k, a_h^k, b_h^k, r_h^k, s_{h+1}^k)\right\}_{k,h=1}^{n,H}$; Failure Probability δ

Initialization: Randomly split the dataset $\mathcal{D}$ into $\mathcal{D}_{\text{re}}$, $\mathcal{D}_0$, and $\{\mathcal{D}_{h,1}\}_{h=1}^H$ with $|\mathcal{D}_{\text{re}}| = n/3$, $|\mathcal{D}_0| = n/3$, $|\mathcal{D}_{h,1}| = n/(3H)$ for all $h \in [H]$

Set $\underline{v}_{H+1} = \overline{v}_{H+1} = 0$

Learn the reference value function $\underline{v}_{\text{re}}, \overline{v}_{\text{re}} \leftarrow \text{PNVI}(\mathcal{D}_{\text{re}})$ (Algorithm 1)

Set $\widehat{P}_{h,0}$ and $\widehat{r}_{h,0}$ as (2.2) using the dataset $\mathcal{D}_0$ for all $h \in [H]$

Set $\widehat{P}_{h,1}$ and $\widehat{r}_{h,1}$ as (2.2) using the dataset $\mathcal{D}_{h,1}$ for all $h \in [H]$

Set $\underline{b}_{h,0}$ and $\overline{b}_{h,0}$ as (2.7) using the dataset $\mathcal{D}_0$ for all $h \in [H]$

for $h = H, H-1, \ldots, 1$ **do**

  Set $\underline{b}_{h,1}$ and $\overline{b}_{h,1}$ as (2.8) using the dataset $\mathcal{D}_{h,1}$

  Set $\underline{q}_h(\cdot,\cdot,\cdot)$ as (2.5)

  Compute the NE of $\underline{q}_h(\cdot,\cdot,\cdot)$ as $(\underline{m}_h(\cdot), \underline{n}_h(\cdot))$

  Compute $\underline{v}_h(\cdot) = \mathbb{E}_{a\sim\underline{m}_h, b\sim\underline{n}_h} \underline{q}_h(\cdot, a, b)$

  Set $\overline{q}_h(\cdot,\cdot,\cdot)$ as (2.6)

  Compute the NE of $\overline{q}_h(\cdot,\cdot,\cdot)$ as $(\overline{m}_h(\cdot), \overline{n}_h(\cdot))$

  Compute $\overline{v}_h(\cdot) = \mathbb{E}_{a\sim\overline{m}_h, b\sim\overline{n}_h} \overline{q}_h(\cdot, a, b)$

end for

Output: $\underline{m} = (\underline{m}_1, \underline{m}_2, \ldots, \underline{m}_H)$, $\overline{n} = (\overline{n}_1, \overline{n}_2, \ldots, \overline{n}_H)$

A.2 Proofs in Section 2.4.1

Lemma A.2.1. *(Concentration) With probability $1-\delta$, we have*

$$\left|r_h(s,a,b)-\widehat{r}_h(s,a,b)+\left\langle P_h(\cdot|s,a,b)-\widehat{P}_h(\cdot|s,a,b),\underline{V}_{h+1}(\cdot)\right\rangle\right|\leq \underline{b}_h(s,a,b),$$

$$\left|r_h(s,a,b)-\widehat{r}_h(s,a,b)+\left\langle P_h(\cdot|s,a,b)-\widehat{P}_h(\cdot|s,a,b),\overline{V}_{h+1}(\cdot)\right\rangle\right|\leq \overline{b}_h(s,a,b),$$

$$\frac{1}{n_h(s,a,b)\vee 1}\leq \frac{8H\iota}{nd_h^{\rho}(s,a,b)}.$$

holds for all $h\in[H]$, $s\in\mathcal{S}$, $a\in\mathcal{A}$ and $b\in\mathcal{B}$. We define this as the good event $\mathcal{G}$.

Proof. We provide the proof for the first argument and the proof for the second argument holds similarly. For all s,a,b,h, we have

$$|r_h(s,a,b)-\widehat{r}_h(s,a,b)|\leq H\sqrt{\frac{1}{n_h(s,a,b)\vee 1}},$$

as whenever $n_h(s,a,b)\geq 1$, $\widehat{r}_h(s,a,b)=r_h(s,a,b)$. For the concentration on $\left\langle \widehat{P}(\cdot|s,a,b),\underline{V}_{h+1}(\cdot)\right\rangle$, note that $\underline{V}_{h+1}$ only depends on the dataset $\{\mathcal{D}_t\}_{t=h+1}^{H}$ while $\widehat{P}_h(\cdot|s,a,b)$ only depends on the dataset $\mathcal{D}_h$, which means they are independent and then Hoeffding's inequality can be applied:

$$\left\langle P_h(\cdot|s,a,b)-\widehat{P}_h(\cdot|s,a,b),\underline{V}_{h+1}(\cdot)\right\rangle\leq 2\sqrt{\frac{H^2\iota}{n_h(s,a,b)\vee 1}}.$$

The second argument holds similarly. For the third argument, the proof is from Lemma B.1 in Xie et al. [2021b]. □

Lemma A.2.2. *(Pessimism) Under the good event $\mathcal{G}$, we have that $\underline{V}_h(s)\leq V_h^{\underline{\mu},*}(s)$ and $\overline{V}_h(s)\geq V_h^{*,\overline{\nu}}(s)$ hold for all $h\in[H]$ and $s\in\mathcal{S}$.*

Proof. We prove this lemma by induction. The inequalities trivially hold for $h=H+1$. If the inequalities hold for timestep $h+1$, now we consider timestep h. By the definition of $\overline{Q}_h(s,a,b)$, we have

$$\begin{aligned}\underline{Q}_h(s,a,b)&=\left(\widehat{r}_h(s,a,b)+(\widehat{P}_h\cdot\underline{V}_{h+1})(s,a,b)-\underline{b}_h(s,a,b)\right)\vee 0\\&\leq\left(r(s,a,b)+(P\cdot V_{h+1}^{\underline{\mu},*})(s,a,b)\right)\vee 0\end{aligned}$$

$$
\begin{aligned}
=&r(s,a,b) + (P \cdot V_{h+1}^{\underline{\mu},*})(s,a,b) \\
=&Q_h^{\underline{\mu},*}(s,a,b),
\end{aligned}
$$

where the inequality is from Lemma A.2.1. With the pessimism on the state-action value function, we can prove the pessimism on the state value function.

$$
\begin{aligned}
\underline{V}_h(s) =&\mathbb{E}_{\underline{\mu}_h,\underline{\nu}_h}\underline{Q}_h(s,a,b) \\
\leq&\mathbb{E}_{\underline{\mu}_h,\mathrm{br}(\underline{\mu}_h)}\underline{Q}_h(s,a,b) \\
\leq&\mathbb{E}_{\underline{\mu}_h,\mathrm{br}(\underline{\mu}_h)}Q_h^{\underline{\mu},*}(s,a,b) \\
=&V_h^{\underline{\mu},*}(s,a,b),
\end{aligned}
$$

where the first inequality is from the definition of NE and the second inequality is from the pessimism of the state-action value function. The arguments for $\overline{V}_h$ hold similarly. Then by mathematical induction we can prove the lemma. □

Lemma A.2.3. *Under the good event $\mathcal{G}$, for all $h \in [H]$ and $s_h \in \mathcal{S}$, we have*

$$
V_h^*(s_h) - V_h^{\underline{\mu},*}(s_h) \leq V_h^{\mu^*,\underline{\nu}}(s_h) - \underline{V}_h(s_h) \leq 2\mathbb{E}_{\mu^*,\underline{\nu}}\left[\sum_{t=h}^{H}\underline{b}_t(s_t,a_t,b_t)|s_h\right],
$$

$$
V_h^{*,\overline{\nu}}(s_h) - V_h^*(s_h) \leq \overline{V}_h(s_h) - V_h^{\overline{\mu},\nu^*}(s_h) \leq 2\mathbb{E}_{\overline{\mu},\nu^*}\left[\sum_{t=h}^{H}\overline{b}_t(s_t,a_t,b_t)|s_h\right].
$$

Proof. We prove the first argument and the second argument can be proven similarly. By the definition of NE, we have $V_h^* \leq V_h^{\mu^*,\underline{\nu}}$. Combined with Lemma A.2.2, we have the first inequality. For the second inequality, we have

$$
\begin{aligned}
&V_h^{\mu^*,\underline{\nu}}(s_h) - \underline{V}_h(s_h) \\
=&\mathbb{E}_{\mu_h^*,\underline{\nu}_h}Q_h^{\mu^*,\underline{\nu}}(s_h,a_h,b_h) - \mathbb{E}_{\underline{\mu}_h,\underline{\nu}_h}\underline{Q}_h(s_h,a_h,b_h) \\
\leq&\mathbb{E}_{\mu_h^*,\underline{\nu}_h}Q_h^{\mu^*,\underline{\nu}}(s_h,a_h,b_h) - \mathbb{E}_{\mu_h^*,\underline{\nu}_h}\underline{Q}_h(s_h,a_h,b_h) \\
=&\mathbb{E}_{\mu_h^*,\underline{\nu}_h}\left[Q_h^{\mu^*,\underline{\nu}}(s_h,a_h,b_h) - \underline{Q}(s_h,a_h,b_h)\right] \\
=&\mathbb{E}_{\mu_h^*,\underline{\nu}_h}\left[r_h(s_h,a_h,b_h) + \left\langle P_h(\cdot|s_h,a_h,b_h), V_{h+1}^{\mu^*,\underline{\nu}}(\cdot)\right\rangle - \widehat{r}_h(s_h,a_h,b_h)\right.
\end{aligned}
$$

$$
\begin{aligned}
&-\left\langle \widehat{P}_h(\cdot|s_h,a_h,b_h), \underline{V}_{h+1}(\cdot)\right\rangle + \underline{b}_h(s_h,a_h,b_h)\Big] \\
\leq &\mathbb{E}_{\mu_h^*,\underline{\nu}_h}\left[\left\langle P_h(\cdot|s_h,a_h,b_h), V_{h+1}^{\mu^*,\underline{\nu}}(\cdot) - \underline{V}_{h+1}(\cdot)\right\rangle + 2\underline{b}_h(s_h,a_h,b_h)\right] \qquad \text{(Lemma A.2.1)} \\
= &\mathbb{E}_{\mu_h^*,\underline{\nu}_h}\left[V_{h+1}^*(s_{h+1}) - \underline{V}_{h+1}^*(s_{h+1})|s_h\right] + 2\mathbb{E}_{\mu_h^*,\underline{\nu}_h^*}\underline{b}_h(s_h,a_h,b_h) \\
\leq &2\mathbb{E}_{\mu^*,\underline{\nu}}\left[\sum_{t=h}^{H}\underline{b}_h(s_t,a_t,b_t)|s_h\right].
\end{aligned}
$$

□

Theorem A.2.4. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$, with probability $1-\delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 1 satisfies*

$$
V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq 64\sqrt{\frac{C^*SABH^5\iota^2}{n}}, V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq 64\sqrt{\frac{C^*SABH^5\iota^2}{n}}.
$$

As a result, we have

$$
\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^*SABH^5}{n}}\right).
$$

Proof. By Lemma A.2.3, with probability at least $1-\delta$, we have

$$
\begin{aligned}
&V_1^{\mu^*,*}(s_1) - V_1^{\underline{\mu},*}(s_1) \\
\leq &2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\underline{b}_h(\mathrm{s_h},\mathrm{a_h},\mathrm{b_h}) \\
= &2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[4\sqrt{\frac{H^2\iota}{n_h(s,a,b)\vee 1}}\right] \\
\leq &2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[32\sqrt{\frac{H^3\iota^2}{nd_h^\rho(s,a,b)}}\right] \qquad \text{(Lemma A.2.1)} \\
= &2\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[32\sqrt{\frac{H^3\iota^2}{nd_h^\rho(s,a,b)}}\right] \\
\leq &64\sum_{h=1}^{H}\sum_{(s,a,b)}\left[\sqrt{\frac{d_h^{\mu^*,\underline{\nu}}(s,a,b)C^*H^3\iota^2}{n}}\right] \\
\leq &64\sqrt{SABH}\cdot\sqrt{\frac{\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)C^*H^3\iota^2}{n}} \qquad \text{(Cauchy-Schwarz Inequality)} \\
= &64\sqrt{\frac{C^*SABH^5\iota^2}{n}}.
\end{aligned}
$$

Similarly we have

$$V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq 64\sqrt{\frac{C^*SABH^5\iota^2}{n}}.$$

As a result, we have

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) + V_1^{\mu^*,*}(s_1) - V_1^{\underline{\mu},*}(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^*SABH^5}{n}}\right).$$

□

Theorem A.2.5. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$ and strategy μ, ν, with probability $1-\delta$, the pessimistic values $\underline{V}_h$ and $\overline{V}_h$ of Algorithm 1 satisfy*

$$\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \leq 64\sqrt{\frac{C^*SABH^5\iota^2}{n}},$$

$$\mathbb{E}_{\mu,\nu^*}\left[\overline{V}_h(s_h) - V_h^*(s_h)\right] \leq 64\sqrt{\frac{C^*SABH^5\iota^2}{n}},$$

where s_h is sampled from the trajectory following the strategy in the expectation at timestep h.

Proof. We prove the first argument and the second argument can be proven similarly. By Lemma A.2.3, under good event $\mathcal{G}$ for all state s we have

$$\begin{aligned} &V_h^*(s) - V_h^{\underline{\mu},*}(s) \\ \leq &2\sum_{t=h}^{H} \mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t, a_t, b_t) | s_h = s\right]. \end{aligned}$$

We define $\nu' = (\nu_1, \cdots, \nu_{h-1}, \underline{\nu}_h, \cdots, \underline{\nu}_H)$. Then we have

$$\begin{aligned} \mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \leq &\mathbb{E}_{\mu^*,\nu}\left[2\sum_{t=h}^{H} \mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t, a_t, b_t) | s_h = s\right] | s\right] \\ = &2\sum_{t=h}^{H} \mathbb{E}_{\mu^*,\nu'}\left[\underline{b}_h(s_t, a_t, b_t)\right]. \end{aligned}$$

Then following the proof of Theorem A.2.4, we can prove the argument.

□

A.3 Proofs in Section 2.4.2

For simplicity, we only provide the guarantee for the max player and the guarantee for the min player can be proven in a similar manner.

Lemma A.3.1. *(Concentration) There exists some absolute constant $c > 0$ such that the concentration event $\mathcal{G}'$ holds with probability at least $1-\delta$, i.e.,*

$$\begin{aligned}
&\left|\widehat{r}_{h,0}(s,a,b) - r_{h,0}(s,a,b) + \left[\left(\widehat{P}_{h,0} - P_h\right)\underline{V}_{h+1}^{\mathrm{ref}}\right](s,a,b)\right| \\
\leq & c\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,0}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,0}(s,a,b)\vee 1}\right),
\end{aligned}$$

$$\begin{aligned}
&\left|\left[\left(\widehat{P}_{h,1} - P_h\right)\left(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}}\right)\right](s,a,b)\right| \\
\leq & c\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,1}(s,a,b)}(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,1}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,1}(s,a,b)\vee 1}\right),
\end{aligned}$$

$$\frac{1}{n_{h,0}(s,a,b)\vee 1} \leq c\frac{\iota}{nd_h^{\rho}(s,a,b)}, \frac{1}{n_{h,1}(s,a,b)\vee 1} \leq c\frac{H\iota}{nd_h^{\rho}(s,a,b)}.$$

Proof. The proof is a direct application of Lemma C.1 in Xie et al. [2021b] with s, a replaced by s, a, b. □

Lemma A.3.2. *For all $h \in [H]$ and $s \in \mathcal{S}$, we have $\underline{V}_h(s) \geq \underline{V}_h^{\mathrm{ref}}(s)$.*

Proof. By the update rule (2.5), we have $\underline{Q}_h(s,a,b) \geq \underline{Q}_h^{\mathrm{ref}}(s,a,b)$ for $h \in [H]$ and $s, a, b \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}$. Then by the definition of NE, we have

$$\underline{V}_h(s) = \mathbb{E}_{\underline{\mu}_h,\underline{\nu}_h}\underline{Q}_h(s,a,b) \geq \mathbb{E}_{\underline{\mu}_h^{\mathrm{ref}},\underline{\nu}_h}\underline{Q}_h(s,a,b) \geq \mathbb{E}_{\underline{\mu}_h^{\mathrm{ref}},\underline{\nu}_h}\underline{Q}_h^{\mathrm{ref}}(s,a,b) \geq \mathbb{E}_{\underline{\mu}_h^{\mathrm{ref}},\underline{\nu}_h^{\mathrm{ref}}}\underline{Q}_h^{\mathrm{ref}}(s,a,b) = \underline{V}_h^{\mathrm{ref}}(s).$$

□

Lemma A.3.3. *(Pessimism) Under the good event $\mathcal{G}'$, we have that $\underline{V}_h(s) \leq V_h^{\underline{\mu},*}(s)$ holds for all $h \in [H]$ and $s \in \mathcal{S}$.*

Proof. We prove this lemma by induction. The inequalities trivially hold for $h = H+1$. If the inequalities hold for $h+1$, now we consider h.

$\underline{Q}_h(s,a,b)$

$$
\begin{aligned}
&= \left\{ \widehat{r}_{h,0}(s,a,b) + (\widehat{P}_{h,0} \cdot \underline{V}_{h+1}^{\mathrm{ref}})(s,a,b) - \underline{b}_{h,0}(s,a,b) + (\widehat{P}_{h,1} \cdot (\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}}))(s,a,b) - \underline{b}_{h,1}(s,a,b) \right\} \\
&\quad \vee \underline{Q}_h^{\mathrm{ref}}(s,a,b) \\
&\leq \max \left\{ r_h(s,a,b) + (P_h \cdot \underline{V}_{h+1}^{\mathrm{ref}})(s,a,b) + \left(P_h \cdot \left(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}} \right) \right)(s,a,b), \underline{Q}_h^{\mathrm{ref}}(s,a,b) \right\} \\
&= \max \left\{ r_h(s,a,b) + (P_h \cdot \underline{V}_{h+1})(s,a,b), \underline{Q}_h^{\mathrm{ref}}(s,a,b) \right\} \\
&\leq \max \left\{ r_h(s,a,b) + (P_h \cdot \underline{V}_{h+1})(s,a,b), r_h(s,a,b) + (P_h \cdot \underline{V}_{h+1}^{\mathrm{ref}})(s,a,b) \right\} \quad \text{(Lemma A.2.2)} \\
&\leq r_h(s,a,b) + (P_h \cdot \underline{V}_{h+1})(s,a,b) \quad \text{(Lemma A.3.2)} \\
&\leq r_h(s,a,b) + (P_h \cdot V_{h+1}^{\underline{\mu},*})(s,a,b) \quad \text{(Induction hypothesis)} \\
&= Q_h^{\underline{\mu},*}(s,a,b).
\end{aligned}
$$

Then by the definition of NE, we have

$$
\begin{aligned}
\underline{V}_h(s) =& \mathbb{E}_{\underline{\mu}_h, \underline{\nu}_h} \underline{Q}_h(s,a,b) \\
\leq& \mathbb{E}_{\underline{\mu}_h, \mathrm{br}(\underline{\mu}_h)} \underline{Q}_h(s,a,b) \\
\leq& \mathbb{E}_{\underline{\mu}_h, \mathrm{br}(\underline{\mu}_h)} Q_h^{\underline{\mu},*}(s,a,b) \\
=& V_h^{\underline{\mu},*}(s).
\end{aligned}
$$

With mathematical induction we can prove the lemma. □

Lemma A.3.4. *Under the good event $\mathcal{G}'$, we have*

$$
V_1^{\mu^*, \underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq 2\mathbb{E}_{\mu^*, \underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,0}(s_h, a_h, b_h) + 2\mathbb{E}_{\mu^*, \underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,1}(s_h, a_h, b_h)
$$

Proof.

$$
\begin{aligned}
& V_1^{\mu^*, \underline{\nu}}(s_1) - \underline{V}_1(s_1) \\
=& \mathbb{E}_{\mu_1^*, \underline{\nu}_1} Q_1^{\mu^*, \underline{\nu}}(s_1, a_1, b_1) - \mathbb{E}_{\underline{\mu}_1, \underline{\nu}_1} \underline{Q}_1(s_1, a_1, b_1) \\
\leq& \mathbb{E}_{\mu_1^*, \underline{\nu}_1} Q_1^{\mu^*, \underline{\nu}}(s_1, a_1, b_1) - \mathbb{E}_{\mu_1^*, \underline{\nu}_1} \underline{Q}_1(s_1, a_1, b_1) \\
=& \mathbb{E}_{\mu_1^*, \underline{\nu}_1} \left[Q_1^{\mu^*, \underline{\nu}}(s_1, a_1, b_1) - \underline{Q}_1(s_1, a_1, b_1) \right] \\
=& \mathbb{E}_{\mu_1^*, \underline{\nu}_1} \left[r_1(s_1, a_1, b_1) + \left\langle P_1(\cdot | s_1, a_1, b_1), V_2^{\mu^*, \underline{\nu}}(\cdot) \right\rangle - \underline{V}_1^{\mathrm{ref}}(s_1) \vee \left\{ \widehat{r}_{1,0}(s_1, a_1, b_1) \right. \right. \\
& \left. \left. + (\widehat{P}_{1,0} \underline{V}_2^{\mathrm{ref}})(s_1, a_1, b_1) - \underline{b}_{1,0}(s_1, a_1, b_1) + (\widehat{P}_{1,1}(\underline{V}_2 - \underline{V}_2^{\mathrm{ref}}))(s_1, a_1, b_1) - \underline{b}_{1,1}(s_1, a_1, b_1) \right\} \right]
\end{aligned}
$$

$$\leq \mathbb{E}_{\mu_1^*,\underline{\nu}_1}\left[\left\langle P_1(\cdot|s_1,a_1,b_1), V_2^{\mu^*,\underline{\nu}}(\cdot) - \underline{V}_2(\cdot)\right\rangle + 2\underline{b}_{1,0}(s_1,a_1,b_1) + 2\underline{b}_{1,1}(s_1,a_1,b_1)\right]$$

(Lemma A.3.1)

$$= \mathbb{E}_{\mu_1^*,\underline{\nu}_1}\left[V_2^{\mu^*,\underline{\nu}}(s_2) - \underline{V}_2^*(s_2)\right] + 2\mathbb{E}_{\mu_1^*,\underline{\nu}_1^*}\underline{b}_{1,0}(s_1,a_1,b_1) + 2\mathbb{E}_{\mu_1^*,\underline{\nu}_1^*}\underline{b}_{1,1}(s_1,a_1,b_1)$$
$$\leq 2\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) + 2\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,1}(s_h,a_h,b_h),$$

where the last inequality is from telescoping the timestep H. □

Lemma A.3.5. *For any strategy ν, we have*

$$\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\nu}(s,a,b)\operatorname*{Var}_{P_h(s,a,b)}(V_{h+1}^{\mu^*,\nu}) \leq H^2.$$

Proof. This is the standard total variance lemma.

$$\begin{aligned}
&\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\nu}(s,a,b)\operatorname*{Var}_{P_h(s,a,b)}(V_h^{\mu^*,\nu}) \\
=&\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\nu}\left[\operatorname{Var}\left[V_{h+1}^*(s_{h+1})|s_h,a_h,b_h\right]\right] \\
=&\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\nu}\left[\mathbb{E}\left[\left(V_{h+1}^*(s_{h+1}) + r_h(s_h,a_h,b_h) - V_h^*(s_h)\right)^2|s_h,a_h,b_h\right]\right] \\
=&\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\nu}\left[\left(V_{h+1}^*(s_{h+1}) + r_h(s_h,a_h,b_h) - V_h^*(s_h)\right)^2\right] \\
=&\mathbb{E}_{\mu^*,\nu}\left[\left(\sum_{h=1}^{H}\left(V_{h+1}^*(s_{h+1}) + r_h(s_h,a_h,b_h) - V_h^*(s_h)\right)\right)^2\right] \\
=&\mathbb{E}_{\mu^*,\nu}\left[\left(\sum_{h=1}^{H} r_h(s_h,a_h,b_h) - V_1^*(s_1)\right)^2\right] \\
=&\operatorname*{Var}_{\mu^*,\nu}\left(\sum_{h=1}^{H} r_h(s_h,a_h,b_h)\right) \\
\leq&H^2.
\end{aligned}$$

□

Lemma A.3.6. *The output strategy $\pi = (\underline{\mu}, \overline{\nu})$ and the pessimistic estimate $\underline{V}$ of Algorithm 1 satisfy*

$$V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \geq \mathbb{E}_{\mu^*,\underline{\nu}}\left[V_h^{\mu^*,\underline{\nu}}(s_h) - \underline{V}_h(s_h)\right].$$

Proof. We prove the argument for $h = 2$ first.

$$
\begin{aligned}
&V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \\
\geq &\mathbb{E}_{\mu^*,\underline{\nu}}[Q_1^{\mu^*,\underline{\nu}}(s_1,a_1,b_1) - \underline{Q}_1(s_1,a_1,b_1)] \\
\geq &\mathbb{E}_{\mu^*,\underline{\nu}}\left[r_1(s_1,a_1,b_1) + \left\langle P_1(\cdot|s_1,a_1,b_1), V_2^{\mu^*,\underline{\nu}}(\cdot)\right\rangle\right] - \mathbb{E}_{\mu^*,\underline{\nu}}\left[\widehat{r}_{1,0}(s_1,a_1,b_1) + (\widehat{P}_{1,0}\underline{V}_2^{\mathrm{ref}})(s_1,a_1,b_1)\right. \\
&\left. -\underline{b}_{1,0}(s_1,a_1,b_1) + (\widehat{P}_{1,1}(\underline{V}_2 - \underline{V}_2^{\mathrm{ref}}))(s_1,a_1,b_1) - \underline{b}_{1,1}(s_1,a_1,b_1)\right] \\
\geq &\mathbb{E}_{\mu^*,\underline{\nu}}\left[r_1(s_1,a_1,b_1) + \left\langle P_1(\cdot|s_1,a_1,b_1), V_2^{\mu^*,\underline{\nu}}(\cdot)\right\rangle\right] - \mathbb{E}_{\mu^*,\underline{\nu}}\left[r_1(s_1,a_1,b_1) + \langle P_1(\cdot|s_1,a_1,b_1), \underline{V}_2(\cdot)\rangle\right] \\
= &\mathbb{E}_{\mu^*,\underline{\nu}}\left[V_2^{\mu^*,\underline{\nu}}(s_2) - \underline{V}_2(s_2)\right].
\end{aligned}
$$

We can prove the lemma for arbitrary h by telescoping the argument to timestep h.

□

Lemma A.3.7. *For $n \geq C^*SABH^3$, we have*

$$
\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \leq \widetilde{O}\left(\sqrt{\frac{C^*SABH^3}{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{C^*SABH^3}{n}}\right).
$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \\
= &c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,0}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,0}(s,a,b)\vee 1}\right) \\
\leq &c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{c\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^{\rho}(s,a,b)}} + \frac{cH\iota}{nd_h^{\rho}(s,a,b)} + \frac{cH\iota}{nd_h^{\rho}(s,a,b)}\right) \\
= &c^2\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left(\sqrt{\frac{\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^{\rho}(s,a,b)}} + \frac{H\iota}{nd_h^{\rho}(s,a,b)}\right) \\
\leq &c^2\sum_{h=1}^{H}\sum_{(s,a,b)}\left(\sqrt{\frac{C^*d_h^{\mu^*,\underline{\nu}}(s,a,b)\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n}} + \frac{C^*H\iota}{n}\right) \\
\leq &c^2\sqrt{SABH}\cdot\sqrt{\frac{C^*\iota\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})}{n}} + \frac{c^2SABC^*H\iota}{n} \\
\leq &c^2\sqrt{C^*SABH\iota}\cdot\sqrt{\frac{\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\right]}{n}} + \frac{c^2SABC^*H\iota}{n}
\end{aligned}
$$

$$\leq c^2\sqrt{C^*SABH\iota}\cdot\sqrt{\frac{\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(V_{h+1}^{\mu^*,\underline{\nu}}) + 2H[P_h(V_{h+1}^{\mu^*,\underline{\nu}} - \underline{V}_{h+1}^{\mathrm{ref}})](s,a,b)\right]}{n}} + \frac{c^2SABC^*H\iota}{n}$$

(Lemma A.5.4)

$$\leq c^2\sqrt{C^*SABH\iota}\cdot\sqrt{\frac{H^2 + 2H\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1}) - \underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{n}} + \frac{c^2SABC^*H\iota}{n}$$

(Lemma A.3.5)

$$= c^2\sqrt{C^*SABH\iota}\cdot\sqrt{\frac{H^2 + 2H\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1}) - V_{h+1}^*(s_{h+1}) + V_{h+1}^*(s_{h+1}) - \underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{n}}$$
$$+ \frac{c^2SABC^*H\iota}{n}$$
$$\leq c^2\sqrt{C^*SABH\iota}\cdot\sqrt{\frac{H^2 + 2H^2(V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)) + 128H\sqrt{\frac{C^*SABH^5\iota^2}{n_{\mathrm{ref}}}}}{n}} + \frac{c^2SABC^*H\iota}{n}$$

(Lemma A.3.6 and Theorem A.2.5)

$$\leq \frac{c^2\sqrt{C^*SABH^3\iota}}{\sqrt{n}} + \frac{c^2\sqrt{384C^*SABH^2\iota\sqrt{C^*SABH^5\iota^2}}}{n^{3/4}} + \frac{c\sqrt{2C^*SABH^3\iota}}{\sqrt{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}$$
$$+ \frac{c^2SABC^*H\iota}{n}$$
$$\leq \widetilde{O}\left(\sqrt{\frac{C^*SABH^3}{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{C^*SABH^3}{n}}\right). \qquad (n \geq C^*SABH^3)$$

□

Lemma A.3.8. *For $n \geq C^*SABH^4$, we have*

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H \underline{b}_{h,1}(s_h,a_h,b_h) \leq \widetilde{O}\left(\sqrt{\frac{C^*SABH^3}{n}}\right).$$

Proof.

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H \underline{b}_{h,1}(s_h,a_h,b_h)$$
$$= c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,1}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,1}(s,a,b)\vee 1}\right)$$
$$\leq c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{cH\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^\rho(s,a,b)}} + \frac{cH^2\iota}{nd_h^\rho(s,a,b)} + \frac{cH^2\iota}{nd_h^\rho(s,a,b)}\right)$$

$$\leq c^2 \mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1} - \underline{V}^{\mathrm{ref}}_{h+1})^2\right](s,a,b)\iota}{n d_h^{\rho}(s,a,b)}} + \frac{H^2\iota}{n d_h^{\rho}(s,a,b)} \right)$$

$$= c^2 \sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) \left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1} - \underline{V}^{\mathrm{ref}}_{h+1})^2\right](s,a,b)\iota}{n d_h^{\rho}(s,a,b)}} + \frac{H^2\iota}{n d_h^{\rho}(s,a,b)} \right)$$

$$\leq c^2 \sum_{h=1}^{H} \sum_{(s,a,b)} \left(\sqrt{\frac{C^* H d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(\underline{V}_{h+1} - \underline{V}^{\mathrm{ref}}_{h+1})^2\right](s,a,b)\iota}{n_1}} + \frac{H^2 C^*\iota}{n_1} \right)$$

(Cauchy-Schwarz Inequality)

$$\leq c^2\sqrt{SABH\iota} \sqrt{\frac{C^* H \sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(\underline{V}_{h+1} - \underline{V}^{\mathrm{ref}}_{h+1})^2\right](s,a,b)}{n}} + \frac{c^2 SABH^3 C^*\iota}{n}$$

$$\leq c^2\sqrt{SABH\iota} \sqrt{\frac{C^* H\iota \sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(V^*_{h+1} - \underline{V}^{\mathrm{ref}}_{h+1})^2\right](s,a,b)}{n}} + \frac{c^2 C^* SABH^3\iota}{n}$$

$(V^*_{h+1} \geq \underline{V}_{h+1} \geq \underline{V}^{\mathrm{ref}}_{h+1})$

$$= c^2\sqrt{SABH\iota} \sqrt{\frac{H^2 C^* \sum_{h=1}^{H}\sum_{s} d_{h+1}^{\mu^*,\underline{\nu}}(s)(V^*_{h+1}(s) - \underline{V}^{\mathrm{ref}}_{h+1}(s))}{n}} + \frac{c^2 C^* SABH^3\iota}{n}$$

$$\leq c^2\sqrt{SABH\iota} \sqrt{\frac{H^2 C^* 64\sqrt{\frac{C^* SABH^5\iota^2}{n_{\mathrm{ref}}}}}{n}} + \frac{c^2 SABH^3 C^*\iota}{n} \qquad \text{(Theorem A.2.5)}$$

$$= c^2 \sqrt{\frac{192 C^* SABH^3\iota\sqrt{C^* SABH^5\iota^2}}{n^{3/2}}} + \frac{c^2 C^* SABH^3\iota}{n}$$

$$\leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right). \qquad (n \geq C^* SABH^4)$$

□

Theorem A.3.9. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$ and $n \geq C^* SABH^4$, with probability $1-\delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 1 satisfies*

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right),$$

$$V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right).$$

As a result, we have

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right).$$

Proof.

$$\begin{aligned}
&V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \\
\leq & 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,0}(s_h, a_h, b_h) + 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,1}(s_h, a_h, b_h) && \text{(Lemma A.3.4)} \\
\leq & \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}} \sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right) \\
& && \text{(Lemma A.3.7 and Lemma A.3.8)} \\
\leq & \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right) + \widetilde{O}\left(\frac{C^* SABH^3}{n}\right) && \text{(Lemma A.5.5)} \\
= & \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right).
\end{aligned}$$

By the definition of NE, we have

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right).$$

The second argument can be proven in a similar manner. Combining these two argument and we can prove that

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^* SABH^3}{n}}\right).$$

□

A.4 Proofs in Section 2.4.3

A.4.1 Uniform Coverage

Theorem A.4.1. *Suppose* $d_m = \min\left\{d_h^\rho(s,a,b) : h \in [H], (s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}\right\}$ *and Assumption 2.2.2 holds. For any* $0 < \delta < 1$*, with probability* $1-\delta$*, the output policy* $\pi = (\underline{\mu}, \overline{\nu})$ *of Algorithm 1 satisfies*

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq 64\sqrt{\frac{H^5\iota^2}{nd_m}}, V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq 64\sqrt{\frac{H^5\iota^2}{nd_m}}.$$

As a result, we have

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{H^5}{nd_m}}\right).$$

Proof. By Lemma A.2.3, with probability $1-\delta$ we have

$$\begin{aligned}
&V_1^{\mu^*,*}(s_1) - V_1^{\underline{\mu},*}(s_1) \\
\leq& 2\sum_{h=1}^{H} \mathbb{E}_{\mu^*,\underline{\nu}} \underline{b}_h(\mathrm{s_h}, \mathrm{a_h}, \mathrm{b_h}) \\
=& 2\sum_{h=1}^{H} \mathbb{E}_{\mu^*,\underline{\nu}} \left[4\sqrt{\frac{H^2\iota}{n_h(s,a,b)\vee 1}}\right] \\
\leq& 2\sum_{h=1}^{H} \mathbb{E}_{\mu^*,\underline{\nu}} \left[32\sqrt{\frac{H^3\iota^2}{nd_h^\rho(s,a,b)}}\right] && \text{(Lemma A.2.1)} \\
=& 2\sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) \left[32\sqrt{\frac{H^3\iota^2}{nd_h^\rho(s,a,b)}}\right] \\
\leq& 64\sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) \left[\sqrt{\frac{H^3\iota^2}{nd_m}}\right] \\
\leq& 64\sqrt{\sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)} \cdot \sqrt{\frac{\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) C^* H^3\iota^2}{nd_m}} \\
& && \text{(Cauchy-Schwarz Inequality)} \\
=& \sqrt{H}\cdot\sqrt{\frac{H^4\iota^2}{nd_m}} \\
=& 64\sqrt{\frac{H^5\iota^2}{nd_m}}.
\end{aligned}$$

□

Theorem A.4.2. *Suppose* $d_m = \min\left\{d_h^\rho(s,a,b) : h\in[H], (s,a,b)\in\mathcal{S}\times\mathcal{A}\times\mathcal{B}\right\}$ *and Assumption 2.2.2 holds. For any* $0<\delta<1$ *and strategy* μ,ν*, with probability* $1-\delta$*, the pessimistic value* $\underline{V}_h$ *and optimistic estimate* $\overline{V}_h$ *of Algorithm 1 satisfies*

$$\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \leq 64\sqrt{\frac{H^5\iota^2}{nd_m}}, \mathbb{E}_{\mu,\nu^*}\left[\overline{V}_h(s_h) - V_h^*(s_h)\right] \leq 64\sqrt{\frac{H^5\iota^2}{nd_m}},$$

where s_h *is sampled from the trajectory following the strategy in the expectation at timestep* h*.*

Proof. By Lemma A.2.3, under good event $\mathcal{G}$ for all state s we have

$$V_h^*(s) - V_h^{\underline{\mu},*}(s) \leq 2\sum_{t=h}^{H} \mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t,a_t,b_t)|s_h = s\right]$$

We define $\nu' = (\nu_1, \cdots, \nu_{h-1}, \underline{\nu}_h, \cdots, \underline{\nu}_H)$. Then we have

$$\begin{aligned}\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \leq & \mathbb{E}_{\mu^*,\nu}\left[2\sum_{t=h}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t,a_t,b_t)|s_h = s\right]|s\right] \\ =& 2\sum_{t=h}^{H}\mathbb{E}_{\mu^*,\nu'}\left[\underline{b}_h(s_t,a_t,b_t)\right].\end{aligned}$$

Then following the proof of Theorem A.4.1, we can prove the argument.

□

Lemma A.4.3. *Suppose* $d_m = \min\left\{d_h^\rho(s,a,b) : h \in [H], (s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\right\}$ *and Assumption 2.3.1 holds. For* $n \geq H^3/d_m$*, we have*

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \\
=& c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,0}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,0}(s,a,b)\vee 1}\right) \\
\leq& c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{c\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^\rho(s,a,b)}} + \frac{cH\iota}{nd_h^\rho(s,a,b)} + \frac{cH\iota}{nd_h^\rho(s,a,b)}\right) \\
\leq& c^2\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left(\sqrt{\frac{\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_m}} + \frac{H\iota}{nd_m}\right) \\
\leq& c^2\sqrt{\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)}\left(\sqrt{\frac{\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_m}} + \frac{H\iota}{nd_m}\right) \\
&\hfill\text{(Cauchy-Schwarz inequality)} \\
\leq& c^2\sqrt{H}\cdot\sqrt{\frac{\iota\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})}{nd_m}} + \frac{c^2H\iota}{nd_m}
\end{aligned}$$

$$\leq c^2\sqrt{H\iota}\cdot\sqrt{\frac{\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\right]}{nd_m}}+\frac{c^2H\iota}{nd_m}$$

$$\leq c^2\sqrt{H\iota}\cdot\sqrt{\frac{\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(V_{h+1}^{\mu^*,\underline{\nu}})+2H[P_h(V_{h+1}^{\mu^*,\underline{\nu}}-\underline{V}_{h+1}^{\mathrm{ref}})](s,a,b)\right]}{nd_m}}+\frac{c^2H\iota}{nd_m}$$

(Lemma A.5.4)

$$\leq c^2\sqrt{H\iota}\cdot\sqrt{\frac{H^2+2H\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1})-\underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{nd_m}}+\frac{c^2H\iota}{nd_m}\qquad\text{(Lemma A.3.5)}$$

$$=c^2\sqrt{H\iota}\cdot\sqrt{\frac{H^2+2H\sum_{h=1}^H \mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1})-V_{h+1}^*(s_{h+1})+V_{h+1}^*(s_{h+1})-\underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{nd_m}}+\frac{c^2H\iota}{nd_m}$$

$$\leq c^2\sqrt{H\iota}\cdot\sqrt{\frac{H^2+2H^2(V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1))+128H\sqrt{\frac{H^5\iota^2}{n_{\mathrm{ref}}d_m}}}{nd_m}}+\frac{c^2H\iota}{nd_m}$$

(Lemma A.3.6 and Theorem A.4.2)

$$\leq\frac{c^2\sqrt{H^3\iota}}{\sqrt{nd_m}}+\frac{c^2\sqrt{384H^2\iota\sqrt{H^5\iota^2}}}{(nd_m)^{3/4}}+\frac{c\sqrt{2H^3\iota}}{\sqrt{nd_m}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1)}+\frac{c^2H\iota}{nd_m}$$

$$\leq\widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1)}\right)+\widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).\qquad(n\geq H^3/d_m)$$

□

Lemma A.4.4. *For $n\geq H^4/d_m$, we have*

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H \underline{b}_{h,1}(s_h,a_h,b_h)\leq\widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).$$

Proof.

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H \underline{b}_{h,1}(s_h,a_h,b_h)$$

$$=c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,1}(s,a,b)\vee 1}}+\frac{H\iota}{n_{h,1}(s,a,b)\vee 1}\right)$$

$$\leq c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{cH\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^\rho(s,a,b)}}+\frac{cH^2\iota}{nd_h^\rho(s,a,b)}+\frac{cH^2\iota}{nd_h^\rho(s,a,b)}\right)$$

$$\leq c^2\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)\iota}{nd_h^\rho(s,a,b)}}+\frac{H^2\iota}{nd_h^\rho(s,a,b)}\right)$$

$$\leq c^2 \sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) \left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)\iota}{nd_m}} + \frac{H^2\iota}{nd_m} \right)$$

$$\leq c^2 \sqrt{\sum_{h=1}^{H} \sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)} \left(\sqrt{\frac{\sum_{h=1}^{H}\sum_{(s,a,b)} H d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)\iota}{nd_m}} + \frac{H^2\iota}{nd_m} \right)$$

(Cauchy-Schwarz Inequality)

$$\leq c^2\sqrt{H}\sqrt{\frac{H\iota\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)}{nd_m}} + \frac{c^2H^3\iota}{nd_m}$$

$$\leq c^2\sqrt{H\iota}\sqrt{\frac{H\iota\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[P_h(V_{h+1}^* - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)}{nd_m}} + \frac{c^2H^3\iota}{nd_m}$$

$(V_{h+1}^* \geq \underline{V}_{h+1} \geq \underline{V}_{h+1}^{\mathrm{ref}})$

$$= c^2\sqrt{H\iota}\sqrt{\frac{H^2\sum_{h=1}^{H}\sum_{s} d_{h+1}^{\mu^*,\underline{\nu}}(s)(V_{h+1}^*(s) - \underline{V}_{h+1}^{\mathrm{ref}}(s))}{nd_m}} + \frac{c^2H^3\iota}{nd_m}$$

$$\leq c^2\sqrt{H\iota}\sqrt{\frac{H^2 64\sqrt{\frac{H^5\iota^2}{n_{\mathrm{ref}}d_m}}}{nd_m}} + \frac{c^2H^3\iota}{nd_m} \qquad \text{(Theorem A.4.2)}$$

$$= c^2\sqrt{\frac{192H^3\iota\sqrt{H^5\iota^2}}{(nd_m)^{3/2}}} + \frac{c^2H^3\iota}{nd_m}$$

$$\leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right). \qquad (n \geq H^4/d_m)$$

□

Theorem A.4.5. *Suppose* $d_m = \min\{d_h^\rho(s,a,b) : h \in [H], (s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\}$ *and Assumption 2.3.1 holds. For any* $0 < \delta < 1$ *and* $n \geq H^4/d_m$*, with probability* $1-\delta$*, the output policy* $\pi = (\underline{\mu}, \overline{\nu})$ *of Algorithm 16 satisfies*

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right), V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).$$

As a result, we have

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).$$

Proof.

$$V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)$$

$$
\begin{aligned}
&\leq 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,0}(s_h, a_h, b_h) + 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,1}(s_h, a_h, b_h) && \text{(Lemma A.3.4)} \\
&\leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}} \sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right) && \text{(Lemma A.4.3 and Lemma A.4.4)} \\
&\leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right) + \widetilde{O}\left(\frac{H^3}{nd_m}\right) && \text{(Lemma A.5.5)} \\
&= \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).
\end{aligned}
$$

By the definition of NE, we have

$$
V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).
$$

The second argument can be proven in a similar manner. Combining two arguments together and we can derive that

$$
\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{H^3}{nd_m}}\right).
$$

□

A.4.2 Turn-based Markov Games

For turn-based Markov games, there always exists a pure (deterministic) NE equilibrium strategy. As a result, we can have that μ^*, ν^*, $\underline{\mu}$, $\underline{\nu}$, $\overline{\mu}$, $\overline{\nu}$ are all pure strategy.

Theorem A.4.6. *Suppose Assumption 2.2.2 holds. For any $0 < \delta < 1$, with probability $1 - \delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 1 satisfies*

$$
V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq 64\sqrt{\frac{C^* S H^5 \iota^2}{n}}, V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq 64\sqrt{\frac{C^* S H^5 \iota^2}{n}}.
$$

As a result, we have

$$
\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^* S H^5}{n}}\right).
$$

Proof. By Lemma A.2.3, with probability $1 - \delta$ we have

$$
V_1^{\mu^*,*}(s_1) - V_1^{\underline{\mu},*}(s_1)
$$

$$
\begin{aligned}
&\leq 2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\underline{b}_h(\mathrm{s_h},\mathrm{a_h},\mathrm{b_h})\\
&= 2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[4\sqrt{\frac{H^2\iota}{n_h(s,a,b)\vee 1}}\right]\\
&\leq 2\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[32\sqrt{\frac{H^3\iota^2}{nd_h^{\rho}(s,a,b)}}\right] && \text{(Lemma A.2.1)}\\
&= 2\sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b)\left[32\sqrt{\frac{H^3\iota^2}{nd_h^{\rho}(s,a,b)}}\right]\\
&\leq 64\sum_{h=1}^{H}\sum_{(s,a,b)}\left[\sqrt{\frac{d_h^{\mu^*,\underline{\nu}}(s,a,b)C^*H^3\iota^2}{n}}\right]\\
&= 64\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}\left[\sqrt{\frac{d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))C^*H^3\iota^2}{n}}\right] && (\mu^*,\underline{\nu}\text{ are deterministic strategy.})\\
&\leq 64\sqrt{SH}\cdot\sqrt{\frac{\sum_{h=1}^{H}\sum_{s\in\mathcal{S}} d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))C^*H^3\iota^2}{n}} && \text{(Cauchy-Schwarz Inequality)}\\
&= 64\sqrt{\frac{C^*SH^5\iota^2}{n}}.
\end{aligned}
$$

□

Theorem A.4.7. *Suppose Assumption 2.2.2 holds. For any $0<\delta<1$ and policy μ,ν, with probability $1-\delta$, the pessimistic value $\underline{V}_h$ of Algorithm 1 satisfies*

$$\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h)-\underline{V}_h(s_h)\right]\leq 64\sqrt{\frac{C^*SH^5\iota^2}{n}},$$

$$\mathbb{E}_{\mu,\nu^*}\left[\overline{V}_h(s_h)-V_h^*(s_h)\right]\leq 64\sqrt{\frac{C^*SH^5\iota^2}{n}},$$

where s_h is sampled from the trajectory following the strategy in the expectation at timestep h.

Proof. By Lemma A.2.3, under good event $\mathcal{G}$ for all state s we have

$$
\begin{aligned}
&V_h^*(s)-V_h^{\underline{\mu},*}(s)\\
\leq& 2\sum_{t=h}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t,a_t,b_t)|s_h=s\right]
\end{aligned}
$$

We define $\nu' = (\nu_1, \cdots, \nu_{h-1}, \underline{\nu}_h, \cdots, \underline{\nu}_H)$. Then we have

$$\begin{aligned}\mathbb{E}_{\mu^*,\nu}\left[V_h^*(s_h) - \underline{V}_h(s_h)\right] \leq & \mathbb{E}_{\mu^*,\nu}\left[2\sum_{t=h}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\underline{b}_h(s_t,a_t,b_t)|s_h=s\right]|s\right] \\ = & 2\sum_{t=h}^{H}\mathbb{E}_{\mu^*,\nu'}\left[\underline{b}_h(s_t,a_t,b_t)\right].\end{aligned}$$

Then following the proof of Theorem A.4.6, we can prove the argument.

□

Lemma A.4.8. *For $n \geq C^*SH^3$, we have*

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \leq \widetilde{O}\left(\sqrt{\frac{C^*SH^3}{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{C^*SH^3}{n}}\right).$$

Proof.

$$\begin{aligned}&\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\underline{b}_{h,0}(s_h,a_h,b_h) \\ =&c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,0}(s,a,b)\vee 1}} + \frac{H\iota}{n_{h,0}(s,a,b)\vee 1}\right) \\ \leq&c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^{H}\left(\sqrt{\frac{c\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^{\rho}(s,a,b)}} + \frac{cH\iota}{nd_h^{\rho}(s,a,b)} + \frac{cH\iota}{nd_h^{\rho}(s,a,b)}\right) \\ =&c^2\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))\left(\sqrt{\frac{\mathrm{Var}_{P_h(s,\mu^*(s),\underline{\nu}(s))}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^{\rho}(s,\mu^*(s),\underline{\nu}(s))}} + \frac{H\iota}{nd_h^{\rho}(s,\mu^*(s),\underline{\nu}(s))}\right) \\ \leq&c^2\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}\left(\sqrt{\frac{C^*d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))\,\mathrm{Var}_{P_h(s,\mu^*(s),\underline{\nu}(s))}(\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n}} + \frac{C^*H\iota}{n}\right)\end{aligned}$$

(μ^*, $\underline{\nu}$ are deterministic strategies.)

$$\begin{aligned}\leq&c^2\sqrt{SH}\cdot\sqrt{\frac{C^*\iota\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))\,\mathrm{Var}_{P_h(s,\mu^*(s),\underline{\nu}(s))}(\underline{V}_{h+1}^{\mathrm{ref}})}{n}} + \frac{c^2SC^*H\iota}{n} \\ \leq&c^2\sqrt{C^*SH\iota}\cdot\sqrt{\frac{\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}^{\mathrm{ref}})\right]}{n}} + \frac{c^2SC^*H\iota}{n} \\ \leq&c^2\sqrt{C^*SH\iota}\cdot\sqrt{\frac{\sum_{h=1}^{H}\mathbb{E}_{\mu^*,\underline{\nu}}\left[\mathrm{Var}_{P_h(s,a,b)}(V_{h+1}^{\mu^*,\underline{\nu}}) + 2H[P_h(V_{h+1}^{\mu^*,\underline{\nu}} - \underline{V}_{h+1}^{\mathrm{ref}})](s,a,b)\right]}{n}} + \frac{c^2SC^*H\iota}{n}\end{aligned}$$

(Lemma A.5.4)

$$\leq c^2\sqrt{C^*SH\iota}\cdot\sqrt{\frac{H^2+2H\sum_{h=1}^H\mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1})-\underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{n}}+\frac{c^2SC^*H\iota}{n}$$

(Lemma A.3.5)

$$=c^2\sqrt{C^*SH\iota}\cdot\sqrt{\frac{H^2+2H\sum_{h=1}^H\mathbb{E}_{\mu^*,\underline{\nu}}\left[V_{h+1}^{\mu^*,\underline{\nu}}(s_{h+1})-V_{h+1}^{*}(s_{h+1})+V_{h+1}^{*}(s_{h+1})-\underline{V}_{h+1}^{\mathrm{ref}}(s_{h+1})\right]}{n}}+\frac{c^2SC^*H\iota}{n}$$

$$\leq c^2\sqrt{C^*SH\iota}\cdot\sqrt{\frac{H^2+2H^2(V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1))+128H\sqrt{\frac{C^*SH^5\iota^2}{n_{\mathrm{ref}}}}}{n}}+\frac{c^2SC^*H\iota}{n}$$

(Lemma A.3.6 and Theorem A.4.7)

$$\leq\frac{c^2\sqrt{C^*SH^3\iota}}{\sqrt{n}}+\frac{c^2\sqrt{384C^*SH^2\iota\sqrt{C^*SH^5\iota^2}}}{n^{3/4}}+\frac{c\sqrt{2C^*SH^3\iota}}{\sqrt{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1)}+\frac{c^2SC^*H\iota}{n}$$

$$\leq\widetilde{O}\left(\sqrt{\frac{C^*SH^3}{n}}\sqrt{V_1^{\mu^*,\underline{\nu}}(s_1)-\underline{V}_1(s_1)}\right)+\widetilde{O}\left(\sqrt{\frac{C^*SH^3}{n}}\right).\qquad(n\geq C^*SH^3)$$

□

Lemma A.4.9. *For $n\geq C^*SH^4$, we have*

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\underline{b}_{h,1}(s_h,a_h,b_h)\leq\widetilde{O}\left(\sqrt{\frac{C^*SH^3}{n}}\right).$$

Proof.

$$\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\underline{b}_{h,1}(s_h,a_h,b_h)$$

$$=c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{\mathrm{Var}_{\widehat{P}_{h,0}(s,a,b)}(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})\iota}{n_{h,1}(s,a,b)\vee 1}}+\frac{H\iota}{n_{h,1}(s,a,b)\vee 1}\right)$$

$$\leq c\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{cH\,\mathrm{Var}_{P_h(s,a,b)}(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})\iota}{nd_h^\rho(s,a,b)}}+\frac{cH^2\iota}{nd_h^\rho(s,a,b)}+\frac{cH^2\iota}{nd_h^\rho(s,a,b)}\right)$$

$$\leq c^2\mathbb{E}_{\mu^*,\underline{\nu}}\sum_{h=1}^H\left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)\iota}{nd_h^\rho(s,a,b)}}+\frac{H^2\iota}{nd_h^\rho(s,a,b)}\right)$$

$$=c^2\sum_{h=1}^H\sum_{s\in\mathcal{S}}d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s))\left(\sqrt{\frac{H\left[P_h(\underline{V}_{h+1}-\underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,\mu^*(s),\underline{\nu}(s))\iota}{nd_h^\rho(s,\mu^*(s),\underline{\nu}(s))}}+\frac{H^2\iota}{nd_h^\rho(s,\mu^*(s),\underline{\nu}(s))}\right)$$

$$\leq c^2 \sum_{h=1}^{H} \sum_{s\in\mathcal{S}} \left(\sqrt{\frac{C^* H d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s)) \left[P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,\mu^*(s),\underline{\nu}(s))\iota}{n_1}} + \frac{H^2 C^* \iota}{n_1} \right)$$

(Cauchy-Schwarz Inequality)

$$\leq c^2\sqrt{SH\iota}\sqrt{\frac{C^* H \sum_{h=1}^{H}\sum_{s\in\mathcal{S}} d_h^{\mu^*,\underline{\nu}}(s,\mu^*(s),\underline{\nu}(s)) \left[P_h(\underline{V}_{h+1} - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,\mu^*(s),\underline{\nu}(s))}{n}} + \frac{c^2 SH^3 C^* \iota}{n}$$

$$\leq c^2\sqrt{SH\iota}\sqrt{\frac{C^* H \iota \sum_{h=1}^{H}\sum_{(s,a,b)} d_h^{\mu^*,\underline{\nu}}(s,a,b) \left[P_h(V_{h+1}^* - \underline{V}_{h+1}^{\mathrm{ref}})^2\right](s,a,b)}{n}} + \frac{c^2 C^* SH^3 \iota}{n}$$

$(V_{h+1}^* \geq \underline{V}_{h+1} \geq \underline{V}_{h+1}^{\mathrm{ref}})$

$$= c^2\sqrt{SH\iota}\sqrt{\frac{H^2 C^* \sum_{h=1}^{H}\sum_{s} d_{h+1}^{\mu^*,\underline{\nu}}(s)(V_{h+1}^*(s) - \underline{V}_{h+1}^{\mathrm{ref}}(s))}{n}} + \frac{c^2 C^* SH^3 \iota}{n}$$

$$\leq c^2\sqrt{SH\iota}\sqrt{\frac{H^2 C^* 64\sqrt{\frac{C^* SH^5\iota^2}{n_{\mathrm{ref}}}}}{n}} + \frac{c^2 SH^3 C^* \iota}{n} \qquad \text{(Theorem A.4.7)}$$

$$= c^2\sqrt{\frac{192 C^* SH^3 \iota \sqrt{C^* SH^5 \iota^2}}{n^{3/2}}} + \frac{c^2 C^* SH^3 \iota}{n}$$

$$\leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right). \qquad (n \geq C^* SH^4)$$

□

Theorem A.4.10. *Suppose Assumption 2.2.2 holds for a turn-based Markov game and $n \geq C^* SH^4$. For any $0 < \delta < 1$, with probability $1-\delta$, the output policy $\pi = (\underline{\mu}, \overline{\nu})$ of Algorithm 1 satisfies*

$$V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right),$$

$$V_1^{*,\overline{\nu}}(s_1) - V_1^*(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right).$$

As a result, we have

$$\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right).$$

Proof.

$$V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)$$

$$
\begin{aligned}
&\leq 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,0}(s_h, a_h, b_h) + 2\mathbb{E}_{\mu^*,\underline{\nu}} \sum_{h=1}^{H} \underline{b}_{h,1}(s_h, a_h, b_h) && \text{(Lemma A.3.4)} \\
&\leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}} \sqrt{V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1)}\right) + \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right) \\
& && \text{(Lemma A.4.8 and Lemma A.4.9)} \\
&\leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right) + \widetilde{O}\left(\frac{C^* SH^3}{n}\right) && \text{(Lemma A.5.5)} \\
&= \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right).
\end{aligned}
$$

By the definition of NE, we have

$$
V_1^*(s_1) - V_1^{\underline{\mu},*}(s_1) \leq V_1^{\mu^*,\underline{\nu}}(s_1) - \underline{V}_1(s_1) \leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right).
$$

The second argument can be proven in a similar manner. Combining these two arguments and we can derive that

$$
\mathrm{Gap}(\underline{\mu}, \overline{\nu}) \leq \widetilde{O}\left(\sqrt{\frac{C^* SH^3}{n}}\right).
$$

□

A.5 Auxiliary Lemmas

Lemma A.5.1. *(Multiplicative Chernoff bound). Let X be a binomial random variable with parameter p, n. For any $1 \geq \theta > 0$, we have that*

$$
\mathbb{P}[(1-\theta)pn < X < (1+\theta)pn] < 2e^{-\frac{\theta^2 pn}{2}}
$$

Lemma A.5.2. *For all $(\mathrm{s_h}, \mathrm{a_h}, \mathrm{b_h}) \in \mathcal{K}_h$ and any $\|V\|_\infty \leq H$, with probability $1-\delta$ we have*

$$
\sqrt{\underset{\widehat{P}^\dagger_{\mathrm{s_h},\mathrm{a_h},\mathrm{b_h}}}{\mathrm{Var}}(V)} \leq \sqrt{\underset{P^\dagger_{\mathrm{s_h},\mathrm{a_h},\mathrm{b_h}}}{\mathrm{Var}}(V)} + cH\sqrt{\frac{\iota}{nd_h^\mu(\mathrm{s_h},\mathrm{a_h},\mathrm{b_h})}}.
$$

Proof. The is a direct application of Lemma A.5.3 with a union bound. □

Lemma A.5.3. *(Empirical Berstein Inequality [Maurer and Pontil, 2009]) Let $n \geq 2$ and $V \in \mathbb{R}^S$ be any functions with $\|V\|_\infty \leq H$, P be any S-dimensional distribution and $\widehat{P}$ be*

its empirical version using n samples. Then with probability $1-\delta$,

$$\left|\sqrt{\operatorname*{Var}_{\widehat{P}}(V)}-\sqrt{\frac{n-1}{n}\operatorname*{Var}_{P}(V)}\right|\leq 2H\sqrt{\frac{\log(2/\delta)}{n-1}}.$$

Lemma A.5.4. *For $0\leq V\leq V'\leq H$, we have*

$$\operatorname*{Var}_{P_h(s,a,b)}(V)\leq\operatorname*{Var}_{P_h(s,a,b)}(V')+2H[P_h(V'-V)](s,a,b).$$

Proof.

$$\begin{aligned}
&\operatorname*{Var}_{P_h(s,a,b)}(V)-\operatorname*{Var}_{P_h(s,a,b)}(V')\\
\leq&\left[P_h(V)^2-(P_hV)^2-P_h(V')^2+(P_hV')^2\right](s,a,b)\\
=&\left[P_h(V+V')(V-V')+[P_h(V'-V)][P_h(v'+v)]\right](s,a,b)\\
\leq&2H[P_h(V'-V)](s,a,b).
\end{aligned}$$

□

Lemma A.5.5. *If $x\leq a\sqrt{x}+b$ for $a,b>0$, then we have*

$$x\leq 2a^2+2b.$$

Proof. We have

$$(\sqrt{x}-\frac{a}{2})^2\leq b+\frac{a^2}{4}.$$

If $\sqrt{x}<\frac{a}{2}$, the argument holds directly. Otherwise we have

$$\sqrt{x}-\frac{a}{2}\leq\sqrt{b+\frac{a^2}{4}}\leq\sqrt{b}+\frac{a}{2}.$$

So we have $\sqrt{x}\leq\sqrt{b}+a$, which implies $x\leq 2(a^2+b)$. □

Appendix B

DEFERRED CONTENTS FROM CHAPTER ??

B.1 Algorithms

Algorithm 17 Value Estimation

Input: Offline dataset $\mathcal{D}$, player index j, and strategy π

Initialization: $\underline{v}^{\pi}_{H+1,j}(s) = \overline{v}^{\pi}_{H+1,j}(s) = 0$ for all $s \in \mathcal{S}$

for $h = H, H-1, \ldots, 1$ **do**

Set

$$\underline{q}^{\pi}_{h,j}(s,\mathbf{a}) = \widehat{r}_{h,j}(s,\mathbf{a}) + \langle \widehat{P}_h(s,\mathbf{a}), \underline{v}^{\pi}_{h+1,j} \rangle$$

Set

$$\underline{v}^{\pi}_{h,j}(s) = \mathrm{proj}_{[0,H-h+1]}\left(\mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\underline{q}^{\pi}_{h,j}(s,\mathbf{a}) - b_h(s,\pi^s_h)\right)$$

Set

$$\overline{q}^{\pi}_{h,j}(s,\mathbf{a}) = \widehat{r}_{h,j}(s,\mathbf{a}) + \langle \widehat{P}_h(s,\mathbf{a}), \overline{v}^{\pi}_{h+1,j} \rangle + H\mathbf{1}\{\mathbf{a} \notin \mathcal{K}_h(s)\}$$

Set

$$\overline{v}^{\pi}_{h,j}(s) = \mathrm{proj}_{[0,H-h+1]}\left(\mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\overline{q}^{\pi}_{h,j}(s,\mathbf{a}) + b_h(s,\pi^s_h)\right)$$

end for

Output: $\underline{v}^{\pi}_{1,j}(s_1)$ and $\overline{v}^{\pi}_{1,j}(s_1)$

B.2 Technical Lemmas

B.2.1 Covering Number of Strategy Classes

Lemma B.2.1. *For the no prior knowledge setting ($\Pi = \Pi^{\mathrm{full}}$), we have*

$$\log \mathcal{N}(\Pi) = \widetilde{O}\left(\sum_{j\in[m]} A_j \log(1/\epsilon_{\mathrm{cover}})\right).$$

Algorithm 18 Best Response Estimation

Input: Offline dataset $\mathcal{D}$, player index j, and strategy π_{-j}

Initialization: $\overline{v}_{H+1,j}^{*,\pi_{-j}}(s) = 0$ for all $s \in \mathcal{S}$

for $h = H, H-1, \ldots, 1$ **do**

Set

$$\overline{q}_{h,j}^{*,\pi_{-j}}(s,\mathbf{a}) = \widehat{r}_{h,j}(s,\mathbf{a}) + \langle \widehat{P}_h(s,\mathbf{a}), \overline{v}_{h+1,j}^{*,\pi_{-j}} \rangle + H\mathbf{1}\{\mathbf{a} \notin \mathcal{K}_h(s)\}$$

Set

$$\overline{v}_{h,j}(s,a_j) = \mathbb{E}_{\mathbf{a}_{-j}\sim\pi_{h,-j}(\cdot|s)}\overline{q}_{h,j}^{*,\pi_{-j}}(s,\mathbf{a}) + b_h(s,a_j,\pi_{h,-j}^s)$$

Set

$$\overline{v}_{h,j}^{*,\pi_{-j}}(s) = \mathrm{proj}_{[0,H-h+1]}\left(\max_{a_j\in\mathcal{A}_j} \overline{v}_{h,j}(s,a_j)\right)$$

end for

Output: $\overline{v}_{1,j}^{*,\pi_{-j}}(s_1)$

Algorithm 19 **S**trategy-wise **B**onus + **S**urrogate **M**inimization (SBSM)

Input: Offline dataset $\mathcal{D}$

Compute

$$\pi^{\mathrm{output}} = \operatorname*{argmin}_{\pi\in\Pi} \sum_{j\in[m]} \overline{v}_{1,j}^{*,\pi_{-j}}(s_1) - \underline{v}_{1,j}^{\pi}(s_1)$$

where $\overline{v}_{1,j}^{*,\pi_{-j}}(s_1)$ and $\underline{v}_{1,j}^{\pi}(s_1)$ are computed via Algorithm 18 and Algorithm 17.

Output: π^{output}

Proof. If $\Pi = \Pi^{\text{full}}$, by Lemma B.5.1 we have

$$\begin{aligned}
\log \mathcal{N}(\Pi) =& \log \left(\sum_{s\in\mathcal{S}, h\in[H]} \prod_{j\in[m]} |\mathcal{C}(\Pi_{h,j}(s), \epsilon_{\text{cover}})| \right) \\
=& \log \left(SH \prod_{j\in[m]} |\mathcal{C}(\Delta(\mathcal{A}_j), \epsilon_{\text{cover}})| \right) \\
=& \sum_{j\in[m]} \log \left(\mathcal{C}(\Delta(\mathcal{A}_j), \epsilon_{\text{cover}}) \right) + \log(SH) \\
\leq& \sum_{j\in[m]} A_j \log \left(3A_j/\epsilon_{\text{cover}} \right) + \log(SH) && \text{(Lemma B.5.1)} \\
=& \widetilde{O}(\sum_{j\in[m]} A_j \log(1/\epsilon_{\text{cover}})).
\end{aligned}$$

□

Lemma B.2.2. *If Π is a finite set, we have*

$$\log(\mathcal{N}(\Pi)) \leq m \log(|\Pi|) + \log(SH).$$

Proof. We have $|\mathcal{C}(\Pi_{h,j}(s), \epsilon_{\text{cover}})| \leq |\Pi_{h,j}(s)| \leq |\Pi|$ for all $h \in [H]$ and $j \in [m]$. Plug it into the definition of $\mathcal{N}(\Pi)$ and we can prove the argument. □

B.2.2 Convexity in Two-player Zero-sum Games

In this section, we prove that $\underline{V}_h^{\mu_h^s, \nu_h^s}(s)$ is concave and $\overline{V}_h^{\mu_h^s, \nu_h^s}(s)$ is convex for both μ_h^s and ν_h^s. In addition, we show that (3.10) and (3.11) can be achieved efficiently.

Lemma B.2.3. *For any coefficient $c(a_i, b_j)$ s.t. $c(a_i, b_j) \geq 0$ for all $a_i \in \mathcal{A}$ and $b_j \in \mathcal{B}$, function $f(\mu, \nu) = \sqrt{\sum_{a_i\in\mathcal{A}, b_j\in\mathcal{B}} c(a_i, b_j)\mu(a_i)^2\nu(b_j)^2}$ defined on $\mu \in \Delta(\mathcal{A})$ and $\nu \in \Delta(\mathcal{B})$ is a convex function and $\sqrt{\sum_{a_i\in\mathcal{A}} \sum_{b_j\in\mathcal{B}} c(a_i, b_j)}$-Lipschitz continuous function with respect to ν. In addition, it is convex and $\sqrt{\sum_{a_i\in\mathcal{A}} \sum_{b_j\in\mathcal{B}} c(a_i, b_j)}$-Lipschitz continuous with respect to μ by symmetry.*

Proof. We use the convention that $\frac{0}{0} = 0$. We first compute the first-order derivatives

$$\frac{\partial f}{\partial \nu(b_j)} = \frac{\sum_{a_i\in\mathcal{A}} c(a_i, b_j)\mu(a_i)\nu(b_j)^2}{\sqrt{\sum_{a_i\in\mathcal{A}, b\in\mathcal{B}} c(a_i, b)\mu(a_i)^2\nu(b)^2}}. \tag{B.1}$$

By Cauchy-Schwarz inequality, we have

$$\begin{aligned}\frac{\partial f}{\partial \nu(b_j)} =& \frac{\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)^2}{\sqrt{\sum_{a_i\in\mathcal{A},b\in\mathcal{B}} c(a_i,b)\mu(a_i)^2\nu(b)^2}}\\ \leq& \frac{\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)}{\sqrt{\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2}}\\ \leq& \sqrt{\sum_{a_i\in\mathcal{A}} c(a_i,b_j)}.\end{aligned}$$

Then we have

$$\left\|\frac{\partial f}{\partial \nu}\right\|_2 \leq \sqrt{\sum_{a_i\in\mathcal{A}}\sum_{b_j\in\mathcal{B}} c(a_i,b_j)},$$

which implies $f(\mu,\cdot)$ is $\sqrt{\sum_{a_i\in\mathcal{A}}\sum_{b_j\in\mathcal{B}} c(a_i,b_j)}$-Lipschitz continuous.

The second-order derivatives are

$$\frac{\partial^2 f}{\partial\nu(b_j)\partial\nu(b_k)} = -\frac{\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)^2\right)\cdot\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_k)\mu(a_i)\nu(b_k)^2\right)}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}, j\neq k,$$

$$\frac{\partial^2 f}{\left(\partial\nu(b_j)\right)^2} = \frac{\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\nu(b_j)^2}{\sqrt{\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2}} - \frac{\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)^2\right)^2}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}.$$

Then for arbitrary $x\in\mathbb{R}^B$, we have

$$\begin{aligned}&\sum_{j,k\in[B]} x_j x_k \frac{\partial^2 f}{\partial\nu(b_j)\partial\nu(b_k)}\\ =&\sum_{j\in[B]} \frac{x_j^2\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\nu(b_j)^2}{\sqrt{\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2}}\\ &-\sum_{j,k\in[B]} \frac{x_j x_k\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)^2\right)\cdot\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_k)\mu(a_i)\nu(b_k)^2\right)}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}\\ =&\frac{\sum_{j\in[B]}\left(x_j^2\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\nu(b_j)^2\right)\cdot\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}\\ &-\frac{\left(\sum_{j\in[B]} x_j\left(\sum_{a_i\in\mathcal{A}} c(a_i,b_j)\mu(a_i)\nu(b_j)^2\right)\right)^2}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}} c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}\end{aligned}$$

$$
\begin{aligned}
=&\frac{\sum_{j\in[B]}\left(x_j^2\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\nu(b_j)^2\right)\cdot\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}}c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}}c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}\\
&-\frac{\left(\sum_{j\in[B]}x_j\left(\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\mu(a_i)\nu(b_j)^2\right)\right)^2}{\left(\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}}c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\right)^{3/2}}.
\end{aligned}
$$

By Cauchy-Schwarz's inequality, we have

$$
\begin{aligned}
&\sum_{j\in[B]}\left(x_j^2\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\nu(b_j)^2\right)\cdot\sum_{a_i\in\mathcal{A},b_j\in\mathcal{B}}c(a_i,b_j)\mu(a_i)^2\nu(b_j)^2\\
=&\left(\sum_{j\in[B]}x_j^2\nu(b_j)^2\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\right)\cdot\left(\sum_{j\in[B]}\nu(b_j)^2\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\mu(a_i)^2\right)\\
\geq&\left(\sum_{j\in[B]}x_j\nu(b_j)^2\sqrt{\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\mu(a_i)^2}\right)^2\\
\geq&\left(\sum_{j\in[B]}x_j\nu(b_j)^2\sum_{a_i\in\mathcal{A}}c(a_i,b_j)\mu(a_i)\right)^2\\
\geq&0.
\end{aligned}
$$

Thus for arbitrary $x\in\mathbb{R}^B$, we have

$$
\sum_{j,k\in[B]}x_jx_k\frac{\partial^2 f}{\partial\nu(b_j)\partial\nu(b_k)}\geq 0,
$$

which implies f is convex with respect to ν. □

Proposition B.2.4. *For all $h\in[H]$ and $s\in\mathcal{S}$, $\underline{V}_h^{\mu_h^s,\nu_h^s}$ is concave and $H+H\sqrt{\log(\mathcal{N}(\Pi))\iota}$-Lipschitz with respect to μ_h^s and ν_h^s. Similarly, $\overline{V}_h^{\mu_h^s,\nu_h^s}$ is convex with respect to μ_h^s and ν_h^s. As a result, (3.10) and (3.11) can be achieved with $(H+H\sqrt{\log(\mathcal{N}(\Pi))\iota})^2/\epsilon_{\text{opt}}^2$ iterations by projected gradient descent.*

Proof. Recall that

$$
\underline{V}_h^{\mu_h^s,\nu_h^s}(s)=\mathbb{E}_{a\sim\mu_h^s,b\sim\nu_h^s}\underline{Q}_h(s,a,b)-H\sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h^s(a)^2\nu_h^s(b)^2}{n_h(s,a,b)}\log(\mathcal{N}(\Pi))\iota}-\sqrt{\iota}/n.
$$

The first term is linear with respect to μ_h^s, The second term is convex by Lemma B.2.3 and the last term is a constant. As a result, $\underline{V}_h^{\mu_h^s,\nu_h^s}$ is concave with respect to μ_h^s. By symmetry, it is also concave with respect to ν_h^s. The proof for $\overline{V}_h^{\mu_h^s,\nu_h^s}$ is the same. The Lipschitz constant is a direct implication of Lemma B.2.3. The iteration complexity of projected gradient descent is from Section 3.1 in Bubeck et al. [2015]. Note that in each iteration we only need to compute the gradient (B.1) and a projection onto the probability simplex. □

B.2.3 Convexity in Multi-player General-sum Games

In this section, we will show that the bonus $b_h(s, \pi_h^s)$ in multi-player general-sum game is also convex with respect to $\pi_{h,j}^s$ for all $j \in [m]$.

Lemma B.2.5. *For any $h \in [H]$ and $s \in \mathcal{S}$, $b_h(s, \pi_h^s)$ is convex with respect to $\pi_{h,j}^s$.*

Proof. Recall that

$$b_h(s, \pi_h^s) = H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi_h^s(\mathbf{a})^2}{n_h(s,\mathbf{a})}\log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n.$$

As we have

$$\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi_h^s(\mathbf{a})^2}{n_h(s,\mathbf{a})} = \sum_{a_j\in\mathcal{A}_j} \sum_{\mathbf{a}_{-j}:(a_j,\mathbf{a}_{-j})\in\mathcal{K}_h(s)} \frac{\pi_{h,j}^s(a_j)^2\pi_{h,-j}^s(\mathbf{a}_{-j})^2}{n_h(s,\mathbf{a})},$$

by Lemma B.2.3 we have that $b_h(s, \pi_h^s)$ is convex with respect to $\pi_{h,j}^s$. □

One direction implication is that $\max_{\pi_{h,j}^s} \overline{V}_{h,j}^{\pi}(s)$ can be achieved by a deterministic strategy $\pi_{h,j}^s \in D(\mathcal{A}_j)$, which will be utilized in Appendix B.4.

B.3 Proofs in Section 3.3

Lemma B.3.1. *Fix $h \in [H]$ and $s \in \mathcal{S}$, $\mu_h'(\cdot|s) \in \Delta(\mathcal{A})$, $\nu_h'(\cdot|s) \in \Delta(\mathcal{B})$, with probability $1-\delta$ we have*

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h'(a|s)\nu_h'(b|s)\left(r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle\right)\right| \leq H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}\log(2/\delta)},$$

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h'(a|s)\nu_h'(b|s)\left(r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle-\widehat{r}_h(s,a,b)-\left\langle\widehat{P}_h(s,a,b),\overline{V}_{h+1}\right\rangle\right)\right| \leq H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}\log(2/\delta)}.$$

Proof. We use $k_h^i(s,a,b)$ to denote the index of (s,a,b) appears in the dataset at timestep h for ith time. We prove the first argument and the second argument holds similarly. With probability $1-\delta$, we have

$$\begin{aligned}
&\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h'(a|s)\nu_h'(b|s)\left(r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle-\widehat{r}_h(s,a,b)-\left\langle\widehat{P}_h(s,a,b),\underline{V}_{h+1}\right\rangle\right)\right| \\
=&\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\sum_{i=1}^{n_h(s,a,b)}\frac{\mu_h'(a|s)\nu_h'(b|s)}{n_h(s,a,b)}\left(r_h^{k_h^i(s,a,b)}-r_h(s,a,b)\right)\right. \\
&\left.+\sum_{(a,b)\in\mathcal{K}_h(s)}\sum_{i=1}^{n_h(s,a,b)}\frac{\mu_h'(a|s)\nu_h'(b|s)}{n_h(s,a,b)}\left(\underline{V}_{h+1}(s_{h+1}^{k_h^i(s,a,b)})-\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right)\right| \\
\leq&\sqrt{\frac{1}{2}\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}\log(2/\delta)}+H\sqrt{\frac{1}{2}\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}\log(2/\delta)} \\
\leq&H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}\log(2/\delta)},
\end{aligned}$$

where the first inequality is from Hoeffding's inequality and the fact that $\underline{V}_{h+1}$ has no dependence on the dataset at timestep h. □

Lemma B.3.2. *With probability $1-\delta$, for all $h\in[H], s\in\mathcal{S}, \mu_h^s\in\Pi_h^{\max}(s)$, $\nu_h^s\in D(\mathcal{B})$, we have*

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h^s(a)\nu_h^s(b)\left(r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle-\widehat{r}_h(s,a,b)-\left\langle\widehat{P}_h(s,a,b),\underline{V}_{h+1}\right\rangle\right)\right| \leq b_h(s,\mu_h^s,\nu_h^s),$$

and for $\mu_h^s\in D(\mathcal{A})$, $\nu_h^s\in\Pi_h^{\min}(s)$, we have

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h^s(a)\nu_h^s(b)\left(r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle-\widehat{r}_h(s,a,b)-\left\langle\widehat{P}_h(s,a,b),\overline{V}_{h+1}\right\rangle\right)\right|$$

$$\leq b_h(s, \mu_h^s, \nu_h^s).$$

Denote this event as $\mathcal{G}$.

Proof. We prove the first argument and the second argument holds similarly. First, using a union bound for all $h \in [H], s \in \mathcal{S}, \mu_h'^s \in \mathcal{C}(\Pi_h^{\max}(s))$, $\nu_h'^s \in D(\mathcal{B})$ on Lemma B.3.1, with probability $1-\delta$, we have

$$\begin{aligned}
&\left|\sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h'^s(a)\nu_h'^s(b)\left(r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle\right)\right| \\
\leq & H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h'^s(a)^2\nu_h'^s(b)^2}{n_h(s,a,b)} \log(2\sum_{s\in\mathcal{S},h\in[H]} |(\mathcal{C}(\Pi_h^{\max}(s))|B + |\mathcal{C}(\Pi_h^{\min}(s))|A)/\delta)} \\
\leq & H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h'^s(a)^2\nu_h'^s(b)^2}{n_h(s,a,b)} \log(2\mathcal{N}(\Pi)ABSH\delta)}. \qquad \text{(See Definition 3.2.2)}
\end{aligned}$$

Note that $r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle$ is bounded in $[-H, H]$ as $r_h(s,a,b) \in [0,1]$ and $\underline{V}_{h+1} \in [0, H-h]$. For any $\mu_h(\cdot|s) \in \Pi_h^{\max}(s)$ and $\nu_h(\cdot|s) \in D(\mathcal{B})$, there exists $\mu_h'(\cdot|s) \in \mathcal{C}(\Pi_h^{\max}(s))$ and $\nu_h'(\cdot|s) \in D(\mathcal{B})$ such that $\|\mu_h(\cdot|s) - \mu_h'(\cdot|s)\| \leq \epsilon_{\text{cover}}$ and $\|\nu_h(\cdot|s) - \nu_h'(\cdot|s)\| = 0 \leq \epsilon_{\text{cover}}$. So with Lemma B.5.2, we have

$$\begin{aligned}
&\left|\sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h'(a|s)\nu_h'(b|s)\left(r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle\right)\right. \\
&\left. - \sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle\right)\right| \\
\leq & 2\epsilon_{\text{cover}}H.
\end{aligned}$$

By Lemma B.5.3, we have

$$\left|\sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h'(a|s)^2\nu_h'(b|s)^2}{n_h(s,a,b)}} - \sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h(a|s)^2\nu_h(b|s)^2}{n_h(s,a,b)}}\right| \leq 2\sqrt{\epsilon_{\text{cover}}}.$$

Combining all these parts together and then with probability $1-\delta$, we have

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)} \mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b) + \langle P_h(s,a,b), \underline{V}_{h+1}\rangle - \widehat{r}_h(s,a,b) - \left\langle \widehat{P}_h(s,a,b), \underline{V}_{h+1}\right\rangle\right)\right|$$

$$\leq H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h(a|s)^2\nu_h(b|s)^2}{n_h(s,a,b)}\log(2\mathcal{N}(\Pi,\epsilon_{\text{cover}})ABSH/\delta)}+2\epsilon_{\text{cover}}H$$
$$+2H\sqrt{2\epsilon_{\text{cover}}\log(2\mathcal{N}(\Pi,\epsilon_{\text{cover}})ABSH/\delta)}.$$

Set $\epsilon_{\text{cover}}=\frac{1}{(A+B)H^2n^2}$ and with some algebra we can get

$$\left|\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle-\widehat{r}_h(s,a,b)-\left\langle\widehat{P}_h(s,a,b),\underline{V}_{h+1}\right\rangle\right)\right|$$
$$\leq H\sqrt{2\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h(a|s)^2\nu_h(b|s)^2}{n_h(s,a,b)}\log(2\mathcal{N}(\Pi)ABSHn/\delta)}+\sqrt{32\log(2ABSHn/\delta)}/n$$
$$\leq H\sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)}\frac{\mu_h(a|s)^2\nu_h(b|s)^2}{n_h(s,a,b)}\log(\mathcal{N}(\Pi))\iota}+\sqrt{\iota}/n,$$

where $\iota=32\log(2ABSHn/\delta)$. □

Lemma B.3.3. *Under event $\mathcal{G}$, for all $s\in\mathcal{S}$, $h\in[H]$, $\mu_h(\cdot|s)\in\Pi_h^{\max}(s)$ and $\nu_h(\cdot|s)\in D(\mathcal{B})$, we have*

$$\underline{V}_h^{\mu_h^s,\nu_h^s}(s)\leq\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right],$$

and for $\mu_h^s\in D(\mathcal{A})$, $\nu_h^s\in\Pi_h^{\min}(s)$, we have

$$\overline{V}_h^{\mu_h^s,\nu_h^s}(s)\geq\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle\right].$$

Proof. Under the good event $\mathcal{G}$, we have

$$\begin{aligned}
&\underline{V}_h^{\mu_h^s,\nu_h^s}(s)\\
=&\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\underline{Q}_h(s,a,b)-b_h(s,\mu_h^s,\nu_h^s)\\
=&\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\left(\widehat{r}_h(s,a,b)+\left\langle\widehat{P}_h(s,a,b),\underline{V}_{h+1}\right\rangle\right)-b_h(s,\mu_h^s,\nu_h^s)\\
\leq&\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right) && \text{(Lemma B.3.2)}\\
\leq&\sum_{a\in\mathcal{A},b\in\mathcal{B}}\mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right) && (\underline{V}_{h+1}\geq 0)\\
=&\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right].
\end{aligned}$$

Similarly we have

$$
\begin{aligned}
&\overline{V}_h^{\mu_h^s,\nu_h^s}(s)\\
=&\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\overline{Q}_h(s,a,b)+b_h(s,\mu_h^s,\nu_h^s)\\
=&\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\left(\widehat{r}_h(s,a,b)+\left\langle\widehat{P}_h(s,a,b),\overline{V}_{h+1}\right\rangle\right)+H\sum_{(a,b)\notin\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\\
&+b_h(s,\mu_h^s,\nu_h^s)\\
\geq&\sum_{(a,b)\in\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle\right)+H\sum_{(a,b)\notin\mathcal{K}_h(s)}\mu_h(a|s)\nu_h(b|s)\\
&\hfill\text{(Lemma B.3.2)}\\
\geq&\sum_{a\in\mathcal{A},b\in\mathcal{B}}\mu_h(a|s)\nu_h(b|s)\left(r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle\right)\qquad(\overline{V}_{h+1}\leq H-h)\\
=&\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\left\langle P_h(s,a,b),\overline{V}_{h+1}\right\rangle\right].
\end{aligned}
$$

□

Lemma B.3.4. *Under event $\mathcal{G}$, for all $s\in\mathcal{S}$ and $h\in[H]$, with probability $1-\delta$, we have*

$$\underline{V}_h(s)\leq V_h^{\underline{\mu},*}(s),\overline{V}_h(s)\geq V_h^{*,\overline{\nu}}(s).$$

Proof. We prove the first argument and the second argument holds similarly. We prove this argument by induction. It holds trivially for $h=H+1$ as both sides are equal to zero. Suppose the argument holds for timestep $h+1$. Then for any $s\in\mathcal{S}$, we have

$$
\begin{aligned}
\underline{V}_h(s)&=\mathrm{proj}_{[0,H-h+1]}\left\{\underline{V}_h^{\underline{\mu}_h^s,\underline{\nu}_h^s}(s)\right\}\\
&=\mathrm{proj}_{[0,H-h+1]}\left\{\min_{\nu_h^s\in D(\mathcal{B})}\underline{V}_h^{\underline{\mu}_h^s,\nu_h^s}(s)\right\}\\
&\leq\mathrm{proj}_{[0,H-h+1]}\left\{\min_{\nu_h^s\in D(\mathcal{B})}\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\langle P_h(s,a,b),\underline{V}_{h+1}\rangle\right]\right\}\\
&\hfill\text{(Lemma B.3.3)}\\
&\leq\mathrm{proj}_{[0,H-h+1]}\left\{\min_{\nu_h^s\in D(\mathcal{B})}\mathbb{E}_{a\sim\mu_h(\cdot|s),b\sim\nu_h(\cdot|s)}\left[r_h(s,a,b)+\left\langle P_h(s,a,b),V_{h+1}^{\underline{\mu},*}\right\rangle\right]\right\}\\
&\hfill\text{(Induction hypothesis)}\\
&=\mathrm{proj}_{[0,H-h+1]}\left\{V_h^{\underline{\mu},*}(s)\right\}\qquad\text{(There always exists a best response in } D(\mathcal{B}))
\end{aligned}
$$

$$=V_h^{\underline{\mu},*}(s).$$

By induction, the argument holds for all $h \in [H]$. The proof for $\overline{V}_h(s)$ is the same. □

For any $\mu_h^s \in \Delta(\mathcal{A})$, with a slight abuse of notation, we define

$$\underline{\nu}_h^s(\mu_h^s) := \operatorname*{argmin}_{\nu_h^s \in D(\mathcal{B})} \underline{V}_h^{\mu_h^s, \nu_h^s}.$$

Note that $\underline{\nu}_h^s = \underline{\nu}_h^s(\underline{\mu}_h^s)$. We use $\underline{\nu}(\mu) \in \Pi^{\min,\det}$ to denote a strategy for player 2 such that she use $\underline{\nu}_h^s(\mu_h^s)$ at state s and timestep h.

Lemma B.3.5. *Under the good event $\mathcal{G}$, for any $\widetilde{\mu} \in \Pi^{\max}$ and $\widetilde{\nu} \in \Pi^{\min}$, we have*

$$V_1^{\widetilde{\mu},*}(s_1) - V_1^{\underline{\mu},*}(s_1) \le \mathbb{E}_{\widetilde{\mu},\underline{\nu}(\widetilde{\mu})} \sum_{h=1}^{H} \widehat{b}_h(s_h, \widetilde{\mu}_h^{s_h}, \underline{\nu}_h^{s_h}(\widetilde{\mu}_h^{s_h})) + H\epsilon_{\text{opt}},$$

$$V_1^{*,\overline{\nu}}(s_1) - V_1^{*,\widetilde{\nu}}(s_1) \le \mathbb{E}_{\overline{\mu}(\widetilde{\nu}),\widetilde{\nu}} \sum_{h=1}^{H} \widehat{b}_h(s_h, \overline{\mu}_h^{s_h}(\widetilde{\nu}_h^{s_h}), \widetilde{\nu}_h^{s_h}) + H\epsilon_{\text{opt}}.$$

Proof. We prove the first argument and the second argument holds similarly. By Lemma B.3.4, we have

$$V_1^{\widetilde{\mu},*}(s_1) - V_1^{\underline{\mu},*}(s_1) \le V_1^{\widetilde{\mu},*}(s_1) - \underline{V}_1(s_1).$$

Now we work on the difference between the NE value and the pessimistic estimate.

$$\begin{aligned}
&V_1^{\widetilde{\mu},*}(s_1) - \underline{V}_1(s_1) \\
=&\min_{\nu_1^{s_1}} \mathbb{E}_{\widetilde{\mu}_1^{s_1},\nu_1^{s_1}} Q_1^{\widetilde{\mu},*}(s_1,a_1,b_1) - \text{proj}_{[0,H]}\left\{\underline{V}_1^{\underline{\mu}_1^{s_1},\underline{\nu}_1^{s_1}}(s_1)\right\} \\
\le&\min_{\nu_1^{s_1}} \mathbb{E}_{\widetilde{\mu}_1^{s_1},\nu_1^{s_1}} Q_1^{\widetilde{\mu},*}(s_1,a_1,b_1) - \underline{V}_1^{\underline{\mu}_1^{s_1},\underline{\nu}_1^{s_1}}(s_1) \qquad (\underline{V}_1^{\underline{\mu}_1^{s_1},\underline{\nu}_1^{s_1}}(s_1) \le H \text{ by (3.4) and (3.5)}) \\
\le&\mathbb{E}_{\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})} Q_1^{\widetilde{\mu},*}(s_1,a_1,b_1) - \underline{V}_1^{\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1)}(s_1) + \epsilon_{\text{opt}} \\
=&\mathbb{E}_{\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1)}\left[Q_1^{\widetilde{\mu},*}(s_1,a_1,b_1) - \underline{Q}_1(s_1,a_1,b_1)\right] + b_1(s_1,\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})) + \epsilon_{\text{opt}} \\
=&\mathbb{E}_{\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})}\left[r_1(s_1,a_1,b_1) + \left\langle P_1(s_1,a_1,b_1), V_2^{\widetilde{\mu},*}\right\rangle - \widehat{r}_1(s_1,a_1,b_1) - \left\langle \widehat{P}_1(s_1,a_1,b_1), \underline{V}_2\right\rangle\right] \\
&+ b_1(s_1,\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})) + \epsilon_{\text{opt}} \\
\le&\mathbb{E}_{\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})}\left[V_2^{\widetilde{\mu},*}(s_2) - \underline{V}_2(s_2)\right] + 2b_1(s_1,\widetilde{\mu}_1^{s_1},\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})) \\
&+ H \sum_{(a_1,b_1)\notin\mathcal{K}_1(s_1)} \widetilde{\mu}_1^{s_1}(a_1)\underline{\nu}_1^{s_1}(\widetilde{\mu}_1^{s_1})(b_1) + \epsilon_{\text{opt}} \qquad \text{(Lemma B.3.2)}
\end{aligned}$$

$$\leq \mathbb{E}_{\widetilde{\mu},\underline{\nu}(\widetilde{\mu})} \sum_{h=1}^{H} \left(2b_h(s_h, \widetilde{\mu}_h^{s_h}, \underline{\nu}_h^{s_h}(\widetilde{\mu}_h^{s_h})) + H \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \widetilde{\mu}_h^{s_h}(a_h)\underline{\nu}_h^{s_h}(\widetilde{\mu}_h^{s_h})(b_h) \right) + H\epsilon_{\text{opt}},$$

where the last inequality is from telescoping from $h = 1$ to $h = H$. □

Proposition B.3.6. *Under the good event $\mathcal{G}$, we have*

$$\text{Gap}(\pi^{\text{output}}) \leq$$

$$\min_{\pi=(\mu,\nu)\in\Pi} \max_{\pi'=(\mu',\nu')\in\Pi^{\text{det}}} \left[\text{Gap}(\pi) + \mathbb{E}_{\mu,\nu'} \sum_{h=1}^{H} \widehat{b}_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) + \mathbb{E}_{\mu',\nu} \sum_{h=1}^{H} \widehat{b}_h(s_h, \mu_h'^{s_h}, \nu_h^{s_h}) \right] + 2H\epsilon_{\text{opt}}.$$

Proof. This is a direct deduction of Lemma B.3.5. Note that $(\underline{\nu}(\widetilde{\mu}), \overline{\mu}(\widetilde{\nu})) \in \Pi^{\text{det}}$. □

B.3.1 Dataset-dependent Bound

Lemma B.3.7. *Suppose $\widehat{C}(\mu,\nu)$ is finite. For any $h \in [H]$ and strategy μ' and ν', we have*

$$\mathbb{E}_{\mu,\nu'} b_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) \leq 2H\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\mu,\nu)\iota/n},$$

$$\mathbb{E}_{\mu',\nu} b_h(s_h, \mu_h'^{s_h}, \nu_h^{s_h}) \leq 2H\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\mu,\nu)\iota/n}.$$

Proof. We prove the first argument and the second argument holds similarly.

$$\begin{aligned}
&\mathbb{E}_{\mu,\nu'} b_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) \\
=&\mathbb{E}_{\mu,\nu'} \left[H\sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h^{s_h}(a)^2\nu_h'^{s_h}(b)^2}{n_h(s,a,b)}\log(\mathcal{N}(\Pi))\iota} + \frac{\sqrt{\iota}}{n} \right] \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota} \sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} \frac{d_h^{\mu,\nu'}(s_h,a_h,b_h)^2}{n_h(s_h,a_h,b_h)}} + \frac{\sqrt{\iota}}{n} \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota} \sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} \frac{d_h^{\mu,\nu'}(s_h,a_h,b_h)^2}{n\cdot\widehat{d}_h(s_h,a_h,b_h)}} + \frac{\sqrt{\iota}}{n} \\
\leq&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota} \sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} d_h^{\mu,\nu'}(s_h,a_h,b_h)\widehat{C}(\mu,\nu)/n} + \frac{\sqrt{\iota}}{n} \\
\leq&H\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\mu,\nu)\iota/n} + \frac{\sqrt{\iota}}{n} \\
\leq&2H\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\mu,\nu)\iota/n}.
\end{aligned}$$

□

Lemma B.3.8. *Suppose $\widehat{C}(\mu,\nu)$ is finite. For any $h\in[H]$ and strategy μ' and ν', we have*

$$\mathbb{E}_{\mu,\nu'}\sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)}\mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h)=0,$$

$$\mathbb{E}_{\mu',\nu}\sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)}\mu_h'^{s_h}(a_h)\nu_h^{s_h}(b_h)=0.$$

Proof. We prove the first argument and the second argument holds similarly.

$$\begin{aligned}
&\mathbb{E}_{\mu,\nu'}\sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)}\mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h)\\
=&\mathbb{E}_{\mu,\nu'}\sum_{(a_h,b_h):\widehat{d}_h(s_h,a_h,b_h)=0}\mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h)\\
=&\sum_{(a_h,b_h):\widehat{d}_h(s_h,a_h,b_h)=0}d_h^{\mu,\nu'}(s_h,a_h,b_h)\\
\le&\sum_{(a_h,b_h):\widehat{d}_h(s_h,a_h,b_h)=0}C(\mu,\nu')\widehat{d}_h(s_h,a_h,b_h)\\
=&0.
\end{aligned}$$

□

Lemma B.3.9. *For any strategy $(\mu,\nu)\in\Pi$, we have*

$$\begin{aligned}
&\max_{\nu'\in\Pi^{\min,\det}}\mathbb{E}_{\mu,\nu'}\sum_{h=1}^{H}\widehat{b}_h(s_h,\mu_h^{s_h},\nu_h'^{s_h})+\max_{\mu'\in\Pi^{\max,\det}}\mathbb{E}_{\mu',\nu}\sum_{h=1}^{H}\widehat{b}_h(s_h,\mu_h'^{s_h},\nu_h^{s_h})\\
&\le 4H^2\sqrt{S\log(|\mathcal{N}(\Pi)|)\widehat{C}(\mu,\nu)\iota/n}.
\end{aligned}$$

Proof. If $\widehat{C}(\mu,\nu)$ is infinite, the argument holds immediately. Otherwise we can prove it by Lemma B.3.7 and Lemma B.3.8. □

Theorem B.3.10. *Suppose π^{output} is the output of Algorithm 2. With probability $1-\delta$, we have*

$$\mathrm{Gap}(\pi^{\text{output}})\le\min_{\pi=(\mu,\nu)\in\Pi}\left[\mathrm{Gap}(\pi)+4H^2\sqrt{S\log(\mathcal{N}(\Pi))\widehat{C}(\pi)\iota/n}\right].$$

Proof. This can be derived from Lemma B.3.5, Lemma B.3.9 directly. □

B.3.2 Dataset-independent Bound

Lemma B.3.11. *With probability $1-\delta$, for all h, s, a, b, we have*

$$n_h(s,a,b) \geq \left(1 - \sqrt{\frac{2\log(SABH/\delta)}{np_{\min}}}\right) nd_h(s,a,b).$$

As a result, if $n \geq \frac{8\log(SABH/\delta)}{p_{\min}}$, for any strategy π we have

$$2C(\pi) \geq \widehat{C}(\pi).$$

Proof. For a fixed s, a, b, h, for any $\epsilon > 0$ we have

$$\mathbb{P}(n_h(s,a,b) < (1-\epsilon)nd_h(s,a,b)) \leq \exp\left(-\frac{\epsilon^2 nd_h(s,a,b)}{2}\right) \leq \exp\left(-\frac{\epsilon^2 np_{\min}}{2}\right).$$

With a union bound, we have

$$\mathbb{P}(\exists h,s,a,b : \mathbb{P}(n_h(s,a,b) < (1-\epsilon)nd_h(s,a,b))) \leq SABH\exp\left(-\frac{\epsilon^2 np_{\min}}{2}\right).$$

The RHS is smaller than δ if we set

$$\epsilon = \sqrt{\frac{2\log(SABH/\delta)}{np_{\min}}}.$$

If $n \geq \frac{8\log(SABH/\delta)}{p_{\min}}$, we have

$$\widehat{d}_h(s,a,b) = \frac{n_h(s,a,b)}{n} \geq \frac{d_h(s,a,b)}{2}.$$

By Definition 3.2.4 and Definition 3.2.3, we have

$$2C(\pi) \geq \widehat{C}(\pi).$$

□

The following Lemma is from Lemma A.1 in Xie et al. [2021b]. For completeness we provide a proof here.

Lemma B.3.12. *With probability at least $1-\delta$, for all $h \in [H]$, $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$, we have*

$$n_h(s,a,b) \vee 1 \geq \frac{nd_h(s,a,b)}{\iota}.$$

Proof. For fixed $h \in [H]$, $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $b \in \mathcal{B}$, $n_h(s,a,b)$ is a binomial random variable following $\mathrm{Bin}(n, d_h(s,a,b))$. We show that with probability $1-\delta$, we have

$$n_h(s,a,b) \vee 1 \geq \frac{n d_h(s,a,b)}{8\log(1/\delta)}.$$

If $d_h(s,a,b) \leq 8\log(1/\delta)/n$, the argument holds directly. Otherwise by the multiplicative Chernoff bound, we have

$$P(n_h(s,a,b) < n d_h(s,a,b)/2) \leq \exp(-n d_h(s,a,b)/8) \leq \delta.$$

So with probability $1-\delta$, we have $n_h(s,a,b) \geq n d_h(s,a,b)/2 \geq n d_h(s,a,b)/8\log(1/\delta)$. Then with union bound we can prove the lemma. □

Lemma B.3.13. *With probability $1-\delta$ for any $h \in [H]$ we have*

$$\mathbb{E}_{\mu,\nu'} b_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) \leq 2H\sqrt{S\log(\mathcal{N}(\Pi))C(\mu,\nu)\iota^2/n},$$

$$\mathbb{E}_{\mu',\nu} b_h(s_h, \mu_h'^{s_h}, \nu_h^{s_h}) \leq 2H\sqrt{S\log(\mathcal{N}(\Pi))C(\mu,\nu)\iota^2/n}.$$

Proof. From Lemma B.3.12, with probability $1-\delta$, for all h, s, a, b, we have

$$n_h(s,a,b) \vee 1 \geq \frac{n d_h(s,a,b)}{\iota}.$$

For $(a,b) \in \mathcal{K}_h(s)$, we have $n_h(s,a,b) \geq 1$ and thus $n_h(s,a,b) \geq \frac{n d_h(s,a,b)}{\iota}$.

$$\begin{aligned}
&\mathbb{E}_{\mu,\nu'} b_h(s_h, \mu_h^{s_h}, \nu_h'^{s_h}) \\
=&\mathbb{E}_{\mu,\nu'}\left[H\sqrt{\sum_{(a,b)\in\mathcal{K}_h(s)} \frac{\mu_h^{s_h}(a)^2 \nu_h'^{s_h}(b)^2}{n_h(s,a,b)}\log(\mathcal{N}(\Pi))\iota} + \frac{\sqrt{\iota}}{n}\right] \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota}\sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} \frac{d_h^{\mu,\nu'}(s_h,a_h,b_h)^2}{n_h(s_h,a_h,b_h)}} + \frac{\sqrt{\iota}}{n} \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota^2}\sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} \frac{d_h^{\mu,\nu'}(s_h,a_h,b_h)^2}{n\cdot d_h(s_h,a_h,b_h)}} + \frac{\sqrt{\iota}}{n} \\
\leq&\sum_{s_h\in\mathcal{S}} H\sqrt{\log(\mathcal{N}(\Pi))\iota^2}\sqrt{\sum_{(a_h,b_h)\in\mathcal{K}_h(s_h)} d_h^{\mu^*,\underline{\nu}(\mu^*)}(s_h,a_h,b_h)C^*/n} + \frac{\sqrt{\iota}}{n} \\
\leq&H\sqrt{S\log(\mathcal{N}(\Pi))C^*\iota^2/n} + \frac{\sqrt{\iota}}{n}
\end{aligned}$$

$$\leq 2H\sqrt{S\log(\mathcal{N}(\Pi))C^*\iota^2/n}.$$

□

Lemma B.3.14. *With probability $1-\delta$ for any $\mu' \in \Pi^{\max,\det}$, $\nu' \in \Pi^{\min,\det}$, $h \in [H]$ and $t \in [0,1]$ we have*

$$\mathbb{E}_{\mu,\nu'} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h) \leq \left(SAC(\mu,\nu)\iota/n\right)^t,$$

$$\mathbb{E}_{\mu',\nu} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h'^{s_h}(a_h)\nu_h^{s_h}(b_h) \leq \left(SBC(\mu,\nu)\iota/n\right)^t.$$

In addition, if $\mu \in \Pi^{\max,\det}$ and $\nu \in \Pi^{\min,\det}$, we have

$$\mathbb{E}_{\mu,\nu'} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h) \leq \left(SC(\mu,\nu)\iota/n\right)^t,$$

$$\mathbb{E}_{\mu',\nu} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h'^{s_h}(a_h)\nu_h^{s_h}(b_h) \leq \left(SC(\mu,\nu)\iota/n\right)^t.$$

Proof. We prove the first argument and the second one holds similarly. From Lemma B.3.12, with probability $1-\delta$, for all h, s, a, b, we have

$$n_h(s,a,b) \vee 1 \geq \frac{nd_h(s,a,b)}{\iota}.$$

For $(a,b) \notin \mathcal{K}_h(s)$, we have $n_h(s,a,b) = 0$ and thus $\iota \geq nd_h(s,a,b)$. Then for any $t \in [0,1]$, we have

$$\begin{aligned}
&\mathbb{E}_{\mu,\nu'} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h) \\
\leq &\mathbb{E}_{\mu,\nu'} \sum_{(a_h,b_h)\in\mathcal{A}\times\mathcal{B}} \frac{\mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h)\iota^t}{(nd_h(s_h,a_h,b_h))^t} \\
= &\sum_{s_h\in\mathcal{S}} \sum_{a_h\in\mathcal{A},b_h=\nu_h'(s_h)} \frac{d_h^{\mu,\nu'}(s_h,a_h,b_h)\iota^t}{(nd_h(s_h,a_h,b_h))^t} \\
\leq &\sum_{s_h\in\mathcal{S}} \sum_{a_h\in\mathcal{A},b_h=\nu_h'(a_h)} \frac{C^t(\mu,\nu)\iota^t}{n^t} \left(d_h^{\mu,\nu'}(s_h,a_h,b_h)\right)^{1-t} \\
\leq &\left(SAC(\mu,\nu)\iota/n\right)^t. \qquad \text{(Cauchy-Schwarz Inequality)}
\end{aligned}$$

If we have $\mu \in M^{\text{det}}$, then we have

$$\begin{aligned}
&\mathbb{E}_{\mu,\nu'} \sum_{(a_h,b_h)\notin\mathcal{K}_h(s_h)} \mu_h^{s_h}(a_h)\nu_h'^{s_h}(b_h) \\
&\leq \sum_{s_h\in\mathcal{S}} \sum_{a_h=\mu_h(s_h),b_h=\nu_h'(s_h)} \frac{C^t(\mu,\nu)\iota^t}{n^t}\left(d_h^{\mu,\nu'}(s_h,a_h,b_h)\right)^{1-t} \\
&\leq \left(SC(\mu,\nu)\iota/n\right)^t. \qquad \text{(Cauchy-Schwarz Inequality)}
\end{aligned}$$

□

Theorem B.3.15. *With probability $1-\delta$, we have*

$$\text{Gap}(\pi^{\text{output}}) \leq \min_{\pi=(\mu,\nu)\in\Pi}\left[\text{Gap}(\pi) + 4H^2\sqrt{S\log(\mathcal{N}(\Pi))C(\pi)\iota^2/n} + 2HC(\pi)S(A+B)\iota/n\right].$$

In additon, if $n \geq \frac{8\log(SABH/\delta)}{p_{\min}}$, we have

$$\text{Gap}(\pi^{\text{output}}) \leq \min_{\pi=(\mu,\nu)\in\Pi}\left[\text{Gap}(\pi) + 8H^2\sqrt{S\log(\mathcal{N}(\Pi))C(\pi)\iota^2/n}\right].$$

Proof. The first argument can be derived by Lemma B.3.13 and Lemma B.3.14 with $t=1$. The second argument can be derived by Theorem B.3.10 and Lemma B.3.11. □

Corollary B.3.16. *If $\Pi = \Pi^{\text{full}}$, then with probability $1-\delta$ we have*

$$\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4S(A+B)C(\pi^*)/n}).$$

In addition, for turn-based two-player zero-sum Markov games, we can set $\Pi = \Pi^{\text{det}}$ and we have

$$\text{Gap}(\pi^{\text{output}}) = \widetilde{O}(\sqrt{H^4SC(\pi^*)/n}).$$

Proof. The first argument can be derived by Lemma B.2.1 and Theorem B.3.15 with $t=1/2$. The second argument can be derived by Lemma B.2.2, Lemma B.3.13 and Lemma B.3.14 with $t=1/2$. □

B.4 Proofs in Section 3.4

Lemma B.4.1. *For any strategy $\pi\in\Pi$, $h\in[H]$ and $s_h\in\mathcal{S}$, we have*

$$\overline{V}_{h,j}^{*,\pi_{-j}}(s_h) = \max_{\pi_j}\overline{V}_{h,j}^{\pi}(s_h).$$

Proof. We prove this argument by induction. It holds trivially for $H+1$ as $\overline{V}^{*,\pi_{-j}}_{H+1,j}(s) = \max_{\pi_j} \overline{V}^{\pi}_{H+1,j}(s) = 0$ for any $s \in \mathcal{S}$. Suppose the argument holds for $h+1$ and now we consider h.

Consider function

$$\begin{aligned} f(\pi'^s_{h,j}) =& \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{r}_{h,j}(s, a_j, \mathbf{a}_{-j}) + \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{P}_h(s, a_j, \mathbf{a}_{-j}) \cdot \overline{V}^{*,\pi_{-j}}_{h+1,j} \\ &+ b_h(s, \pi'^s_{h,j}, \pi^s_{h,-j}) + H \sum_{\mathbf{a}_{-j}:(a_j,\mathbf{a}_{-j}) \notin \mathcal{K}(s)} \pi^s_{h,-j}(\mathbf{a}_{-j}). \end{aligned}$$

Lemma B.2.5 shows that $b_h(s, \pi'^s_{h,j}, \pi^s_{h,-j})$ is convex with respect to $\pi'^s_{h,j}$, while all the other terms are linear with respect to $\pi'^s_{h,j}$. As a result, $f(\pi'^s_{h,j})$ is a convex function and thus we have

$$\max_{\pi'^s_{h,j} \in \Delta(\mathcal{A}_j)} f(\pi'^s_{h,j}) = \max_{\pi'^s_{h,j} \in D(\mathcal{A}_j)} f(\pi'^s_{h,j}).$$

Then we have

$$\begin{aligned} &\max_{a_j \in \mathcal{A}_j} \overline{V}_{h,j}(s, a_j) \\ =& \max_{\pi'^s_{h,j} \in D(\mathcal{A}_j)} f(\pi'^s_{h,j}) \\ =& \max_{\pi'^s_{h,j} \in \Delta(\mathcal{A}_j)} f(\pi'^s_{h,j}) \\ =& \max_{\pi'^s_{h,j} \in \Delta(\mathcal{A}_j)} \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{r}_{h,j}(s, a_j, \mathbf{a}_{-j}) + \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{P}_h(s, a_j, \mathbf{a}_{-j}) \cdot \overline{V}^{*,\pi_{-j}}_{h+1,j} \\ &+ b_h(s, \pi'^s_{h,j}, \pi^s_{h,-j}) + H \sum_{\mathbf{a}_{-j}:(a_j,\mathbf{a}_{-j}) \notin \mathcal{K}(s)} \pi^s_{h,-j}(\mathbf{a}_{-j}) \\ =& \max_{\pi'^s_{h,j} \in \Delta(\mathcal{A}_j)} \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{r}_{h,j}(s, a_j, \mathbf{a}_{-j}) + \max_{\pi_j} \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{P}_h(s, a_j, \mathbf{a}_{-j}) \cdot \overline{V}^{\pi}_{h+1,j} \\ &+ b_h(s, \pi'^s_{h,j}, \pi^s_{h,-j}) + H \sum_{\mathbf{a}_{-j}:(a_j,\mathbf{a}_{-j}) \notin \mathcal{K}(s)} \pi^s_{h,-j}(\mathbf{a}_{-j}) \qquad \text{(Induction hypothesis)} \\ =& \max_{\pi_j} \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{r}_{h,j}(s, a_j, \mathbf{a}_{-j}) + \mathbb{E}_{a_j \sim \pi'^s_{h,j}, \mathbf{a}_{-j} \sim \pi^s_{h,-j}} \widehat{P}_h(s, a_j, \mathbf{a}_{-j}) \cdot \overline{V}^{\pi}_{h+1,j} \\ &+ b_h(s, \pi'^s_{h,j}, \pi^s_{h,-j}) + H \sum_{\mathbf{a}_{-j}:(a_j,\mathbf{a}_{-j}) \notin \mathcal{K}(s)} \pi^s_{h,-j}(\mathbf{a}_{-j}). \end{aligned}$$

So we have $\overline{V}^{*,\pi_{-j}}_{h,j}(s_h) = \max_{\pi_j} \overline{V}^{\pi}_{h,j}(s_h)$. (See Algorithm 17 and Algorithm 18 for the definition of both quantities) □

Lemma B.4.2. *Fix $\pi' \in \Pi, j \in [m], h \in [H]$ and $s \in \mathcal{S}$, with probability $1-\delta$ we have*

$$\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \pi'_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}), \underline{V}^{\pi'}_{h+1,j}\right\rangle - \widehat{r}_{h,j}(s,\mathbf{a}) - \left\langle \widehat{P}_{h,j}(s,\mathbf{a}), \underline{V}^{\pi'}_{h+1,j}\right\rangle\right)\right| \leq H\sqrt{2\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(4/\delta)},$$

and

$$\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \pi'_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}), \overline{V}^{\pi'}_{h+1,j}\right\rangle - \widehat{r}_{h,j}(s,\mathbf{a}) - \left\langle \widehat{P}_h(s,\mathbf{a}), \overline{V}^{\pi'}_{h+1,j}\right\rangle\right)\right| \leq H\sqrt{2\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(4/\delta)}.$$

Proof. We use $k^i_h(s,a,b)$ to denote the index of (s,a,b) appears in the dataset at timestep h for ith time. With probability $1-\delta$, we have

$$\begin{aligned}
&\left|\sum_{(\mathbf{a})\in\mathcal{K}_h(s)} \pi'_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}), \underline{V}^{\pi'}_{h+1,j}\right\rangle - \widehat{r}_{h,j}(s,\mathbf{a}) - \left\langle \widehat{P}_h(s,\mathbf{a}), \underline{V}^{\pi'}_{h+1,j}\right\rangle\right)\right| \\
=&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \sum_{i=1}^{n_h(s,\mathbf{a})} \frac{\pi'_h(\mathbf{a}|s)}{n_h(s,\mathbf{a})}\left(r^{k^i_h(s,\mathbf{a})}_{h,j} - r_{h,j}(s,\mathbf{a})\right)\right. \\
&\left. + \sum_{(\mathbf{a})\in\mathcal{K}_h(s)} \sum_{i=1}^{n_h(s,\mathbf{a})} \frac{\pi'_h(\mathbf{a}|s)}{n_h(s,\mathbf{a})}\left(\underline{V}^{\pi'}_{h+1,j}(s^{k^i_h(s,\mathbf{a})}_{h+1}) - \left\langle P_h(s,\mathbf{a}), \underline{V}^{\pi'}_{h+1,j}\right\rangle\right)\right| \\
\leq&\sqrt{\frac{1}{2}\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(2/\delta)} + H\sqrt{\frac{1}{2}\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(2/\delta)} \\
\leq&H\sqrt{2\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(2/\delta)},
\end{aligned}$$

where the first inequality is from Hoeffding's inequality and the fact that $\underline{V}_{h+1,j}$ has no dependence on the dataset at timestep h. The second argument holds similarly. Rescaling δ to $\delta/2$ and with an union bound we can prove the lemma. □

Lemma B.4.3. *With probability $1-\delta$, for all $\pi \in \Pi, j \in [m], h \in [H], s \in \mathcal{S}$, we have*

$$\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)} \pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}), \underline{V}^{\pi}_{h+1,j}\right\rangle - \widehat{r}_{h,j}(s,\mathbf{a}) - \left\langle \widehat{P}_h(s,\mathbf{a}), \underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right| \leq b_h(s,\pi^s_h),$$

$$\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\overline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\overline{V}^{\pi}_{h+1,j}\right\rangle\right)\right|\leq b_h(s,\pi^s_h).$$

Denote this event as $\mathcal{G}_{\text{marl}}$.

Proof. We prove the argument for $\underline{V}^{\pi}_{h+1,j}$ and the argument for $\overline{V}^{\pi}_{h+1,j}$ holds similarly. Suppose $\mathcal{V}$ is a ϵ_{cover}-covering of $[0,H]^S$ with respect to L-∞ norm and $|\mathcal{V}|\leq(1+HS/\epsilon_{\text{cover}})^S$. First, using a union bound for all $j\in[m],h\in[H],s\in\mathcal{S},\pi'^s_{h,j}\in\mathcal{C}(\Pi^{\text{prior}}_{h,j}(s)),V_{h+1}\in\mathcal{V}$ on Lemma B.4.2, with probability $1-\delta$ we have

$$\begin{aligned}
&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi'_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a})+\langle P_h(s,\mathbf{a}),V_{h+1}\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),V_{h+1}\right\rangle\right)\right|\\
\leq&H\sqrt{4\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}\log(4m\sum_{s\in\mathcal{S},h\in[H]}\prod_{j\in[m]}|\mathcal{C}(\Pi_{h,j}(s))|(1+HS/\epsilon_{\text{cover}})^S/\delta)}\\
\leq&H\sqrt{8\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}S\log(8m\mathcal{N}(\Pi)SH/\epsilon_{\text{cover}}\delta)}.
\end{aligned}$$

Note that $r_{h,j}(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle$ is bounded in $[-H,H]$ as $r_{h,j}(s,\mathbf{a})\in[0,1]$ and $\underline{V}^{\pi}_{h+1,j}\in[0,H-h]$. There exists $V_{h+1}\in\mathcal{V}$ such that $\|\underline{V}^{\pi}_{h+1,j}-V_{h+1}\|_\infty\leq\epsilon_{\text{cover}}$, which implies

$$\begin{aligned}
&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi'_h(\mathbf{a}|s)\left(r_h(s,\mathbf{a})+\langle P_h(s,\mathbf{a}),V_{h+1}\rangle-\widehat{r}_h(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),V_{h+1}\right\rangle\right)\right|\\
-&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi'_h(\mathbf{a}|s)\left(r_h(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_h(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right|\\
\leq&2\epsilon_{\text{cover}}.
\end{aligned}$$

For any $\pi^s_{h,j}\in\Pi_{h,j}(s)$, there exists $\pi'^s_{h,j}\in\mathcal{C}(\Pi_{h,j}(s))$ such that $\|\pi_{h,j}(\cdot|s)-\pi'_{h,j}(\cdot|s)\|_1\leq\epsilon_{\text{cover}}$ for all $j\in[m]$ and $s\in\mathcal{S}$. So with Lemma B.5.2, we have

$$\begin{aligned}
&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi'_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right.\\
&\left.-\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_h(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right|\\
\leq&m\epsilon_{\text{cover}}H.
\end{aligned}$$

By Lemma B.5.3, we have

$$\left|\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi'_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}}-\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}}\right|\leq\sqrt{2m\epsilon_{\text{cover}}}.$$

Combining all these parts together and then with probability $1-\delta$, we have

$$\begin{aligned}&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right|\\ \leq&H\sqrt{8\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}S\log(8m\mathcal{N}(\Pi,\epsilon_{\text{cover}})SH\delta)}+2\epsilon_{\text{cover}}+m\epsilon_{\text{cover}}H\\ &+H\sqrt{8m\epsilon_{\text{cover}}\log(8m\mathcal{N}(\Pi,\epsilon_{\text{cover}})SH/\delta)}.\end{aligned}$$

By Lemma B.5.1, we have

$$\begin{aligned}\mathcal{N}(\Pi,\epsilon_{\text{cover}})=&\frac{1}{SH}\sum_{s\in\mathcal{S},h\in[H]}\prod_{j\in[m]}|\mathcal{C}(\Pi_{h,j}(s),\epsilon_{\text{cover}})|\\ \leq&\prod_{j\in[m]}(3A_j/\epsilon_{\text{cover}})^{A_j}\\ \leq&(3(\sum_{j\in[m]}A_j)/\epsilon_{\text{cover}})^{\sum_{j\in[m]}A_j}.\end{aligned}$$

Set $\epsilon_{\text{cover}}=\frac{1}{\sum_{j\in[m]}A_jmH^2n^2}$ and with some calculations we can get

$$\begin{aligned}&\left|\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_h(s,\mathbf{a})+\left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle-\widehat{r}_{h,j}(s,\mathbf{a})-\left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right|\\ \leq&H\sqrt{8\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi_h(\mathbf{a}|s)^2}{n_h(s,\mathbf{a})}S\log(8m\mathcal{N}(\Pi)SHn/\delta)}+\sqrt{32\log(16\prod_{j\in[m]}A_jmSHn/\delta)/n}\\ \leq&H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\frac{\pi_h(a|s)^2}{n_h(s,\mathbf{a})}S\log(\mathcal{N}(\Pi))\iota}+\sqrt{\iota}/n.\end{aligned}$$

□

Lemma B.4.4. *Under event $\mathcal{G}_{\text{marl}}$, for all $j\in[m]$, $h\in[H]$, $\pi\in\Pi$ and $s\in\mathcal{S}$, we have*

$$\underline{V}^{\pi}_{h,j}(s)\leq V^{\pi}_{h,j}(s)\leq\overline{V}^{\pi}_{h,j}(s).$$

Proof. We prove this argument by induction. It holds for $h = H+1$ as $\underline{V}^{\pi}_{H+1,j}(s) = V^{\pi}_{H+1,j}(s) = \overline{V}^{\pi}_{H+1,j}(s)$. Suppose the argument holds for $h+1$ and we consider h.

$$
\begin{aligned}
\underline{V}^{\pi}_{h,j}(s) =& \operatorname{proj}_{[0,H-h+1]}\left\{\mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\widehat{r}_{h,j}(s,\mathbf{a}) + \mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\widehat{P}_h(s,\mathbf{a})\cdot\underline{V}^{\pi}_{h+1,j} - b_h(s,\pi^s_h)\right\}\\
=& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(\widehat{r}_{h,j}(s,\mathbf{a}) + \left\langle\widehat{P}_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right) - b_h(s,\pi^s_h)\right\}\\
\le& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),\underline{V}^{\pi}_{h+1,j}\right\rangle\right)\right\}\\
& \qquad\qquad \text{(Lemma B.4.3)}\\
\le& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),V^{\pi}_{h+1,j}\right\rangle\right)\right\}\\
& \qquad\qquad \text{(Induction hypothesis)}\\
\le& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{A}}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),V^{\pi}_{h+1,j}\right\rangle\right)\right\}\\
\le& \operatorname{proj}_{[0,H-h+1]}\left\{V^{\pi}_{h,j}(s)\right\}\\
=& V^{\pi}_{h,j}(s).
\end{aligned}
$$

$$
\begin{aligned}
&\overline{V}^{\pi}_{h,j}(s)\\
=& \operatorname{proj}_{[0,H-h+1]}\left\{\mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\widehat{r}_{h,j}(s,\mathbf{a}) + \mathbb{E}_{\mathbf{a}\sim\pi_h(\cdot|s)}\widehat{P}_h(s,\mathbf{a})\cdot\overline{V}^{\pi}_{h+1,j} + b_h(s,\pi^s_h) + H\sum_{a\notin\mathcal{K}(s)}\pi_h(\mathbf{a}|s)\right\}\\
=& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(\widehat{r}_{h,j}(s,\mathbf{a}) + \left\langle\widehat{P}_h(s,\mathbf{a}),\overline{V}^{\pi}_{h+1,j}\right\rangle\right) + b_h(s,\pi^s_h) + H\sum_{a\notin\mathcal{K}(s)}\pi_h(\mathbf{a}|s)\right\}\\
\ge& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{K}_h(s)}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),\overline{V}^{\pi}_{h+1,j}\right\rangle\right) + H\sum_{a\notin\mathcal{K}(s)}\pi_h(\mathbf{a}|s)\right\}\\
& \qquad\qquad \text{(Lemma B.4.3)}\\
\ge& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{A}}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),\overline{V}^{\pi}_{h+1,j}\right\rangle\right)\right\}\\
& \qquad\qquad (\overline{V}^{\pi}_{h+1,j}(s) \le H-h \text{ for all } s\in\mathcal{S})\\
\ge& \operatorname{proj}_{[0,H-h+1]}\left\{\sum_{\mathbf{a}\in\mathcal{A}}\pi_h(\mathbf{a}|s)\left(r_{h,j}(s,\mathbf{a}) + \left\langle P_h(s,\mathbf{a}),V^{\pi}_{h+1,j}\right\rangle\right)\right\} \qquad \text{(Induction hypothesis)}\\
=& \operatorname{proj}_{[0,H-h+1]}\left\{V^{\pi}_{h,j}(s)\right\}
\end{aligned}
$$

$=V_{h,j}^{\pi}(s)$.

□

Lemma B.4.5. *Under event $\mathcal{G}_{\mathrm{marl}}$, for any policy $\pi \in \Pi$, we have*

$$\mathrm{Gap}(\pi) \le \sum_{j\in[m]} \overline{V}_{1,j}^{*,\pi_{-j}}(s) - \underline{V}_{1,j}^{\pi}(s).$$

In addition, we have

$$\mathrm{Gap}(\pi^{\mathrm{output}}) \le \min_{\pi\in\Pi} \sum_{j\in[m]} \left[\overline{V}_{1,j}^{*,\pi_{-j}}(s) - \underline{V}_{1,j}^{\pi}(s)\right].$$

Proof. By Lemma B.4.4, we have

$$\mathrm{Gap}(\pi) = \max_{\pi'} \sum_{j\in[m]} V_{1,j}^{\pi'_j,\pi_{-j}}(s) - V_{1,j}^{\pi}(s) \le \max_{\pi'} \sum_{j\in[m]} \overline{V}_{1,j}^{\pi'_j,\pi_{-j}}(s) - \underline{V}_{1,j}^{\pi}(s).$$

Combined with Lemma B.4.1 we can prove the first argument. For the second argument, note that π_{output} is the minimizer of the RHS, so we have

$$\mathrm{Gap}(\pi^{\mathrm{output}}) \le \min_{\pi\in\Pi} \sum_{j\in[m]} \overline{V}_{1,j}^{*,\pi_{-j}}(s) - \underline{V}_{1,j}^{\pi}(s).$$

□

Lemma B.4.6. *Under event $\mathcal{G}_{\mathrm{marl}}$, for any strategy $\pi \in \Pi$, we have*

$$\underline{V}_{1,j}^{\pi}(s_1) \ge V_{1,j}^{\pi}(s_1) - \mathbb{E}_{\pi} \sum_{h\in[H]} \widehat{b}_h(s_h, \pi_h^{s_h}),\ \overline{V}_{1,j}^{\pi}(s_1) \le V_{1,j}^{\pi}(s_1) + \mathbb{E}_{\pi} \sum_{h\in[H]} \widehat{b}_h(s_h, \pi_h^{s_h}).$$

Proof. We prove the first argument and the second argument holds similarly.

$$\begin{aligned}
&V_{1,j}^{\pi}(s_1) - \underline{V}_{1,j}^{\pi}(s_1)\\
=&\mathbb{E}_{\mathbf{a}\sim\pi_1(\cdot|s_1)}\left[r_{1,j}(s_1,\mathbf{a}) + P_1(s_1,\mathbf{a})\cdot V_{2,j}^{\pi}\right] - \mathbb{E}_{\mathbf{a}\sim\pi_1(\cdot|s_1)}\left[\widehat{r}_{1,j}(s_1,\mathbf{a}) + \widehat{P}_1(s_1,\mathbf{a})\cdot \underline{V}_{2,j}^{\pi}\right] + b_1(s_1,\pi_1^{s_1})\\
=&\mathbb{E}_{\pi_1}\left[V_{2,j}^{\pi}(s_2) - \underline{V}_{2,j}^{\pi}(s_2)\right] + \mathbb{E}_{\pi_1}\left[r_{1,j}(s_1,\mathbf{a}) + P_1(s_1,\mathbf{a})\cdot \underline{V}_{2,j}^{\pi} - \widehat{r}_{1,j}(s_1,\mathbf{a}) - \widehat{P}_1(s_1,\mathbf{a})\cdot \underline{V}_{2,j}^{\pi}\right] + b_1(s_1,\pi_1^{s_1})\\
\le&\mathbb{E}_{\pi_1}\left[V_{2,j}^{\pi}(s_2) - \underline{V}_{2,j}^{\pi}(s_2)\right] + \sum_{\mathbf{a}\in\mathcal{K}_h(s_1)} \pi_1(\mathbf{a}|s_1)\left(r_{1,j}(s_1,\mathbf{a}) + P_1(s_1,\mathbf{a})\cdot V_{2,j}^{\pi} - \widehat{r}_{1,j}(s_1,\mathbf{a}) - \widehat{P}_1(s_1,\mathbf{a})\cdot V_{2,j}^{\pi}\right)\\
&+ \sum_{\mathbf{a}\notin\mathcal{K}_h(s_1)} \pi(\mathbf{a}|s_1)H + b_1(s_1,\pi_1^{s_1})
\end{aligned}$$

$$\leq \mathbb{E}_{\pi_1}\left[V_{2,j}^{\pi}(s_2) - \underline{V}_{2,j}^{\pi}(s_2)\right] + \sum_{\mathbf{a}\notin\mathcal{K}_h(s_1)} \pi_1(\mathbf{a}|s_1)H + 2b_1(s_1,\pi_1^{s_1})$$
$$= \mathbb{E}_{\pi_1}\left[V_{2,j}^{\pi}(s_2) - \underline{V}_{2,j}^{\pi}(s_2)\right] + \widehat{b}_1(s_1,\pi_1^{s_1}).$$

By telescoping we can prove the first argument. □

Lemma B.4.7. *Under good event $\mathcal{G}_{\mathrm{marl}}$, for any strategy $\pi \in \Pi$, we have*

$$\sum_{j\in[m]} \overline{V}_{1,j}^{*,\pi_{-j}}(s_1) - \underline{V}_{1,j}^{\pi}(s_1) \leq \mathrm{Gap}(\pi) + \max_{\pi'\in\Pi^{\mathrm{det}}} \sum_{j\in[m]} \mathbb{E}_{\pi'_j,\pi_{-j}}\left[\sum_{h=1}^{H} \widehat{b}_h(s_h, \pi_{h,j}'^{s_h}, \pi_{h,-j}^{s_h})\right] + m\mathbb{E}_{\pi}\sum_{h=1}^{H}\left[\widehat{b}_h(s_h,\pi_h^{s_h})\right].$$

Proof. Set $\widetilde{\pi} = \mathrm{argmax}_{\pi'\in\Pi^{\mathrm{full}}} \sum_{j\in[m]} \overline{V}_{1,j}^{\pi'_j,\pi_{-j}}(s_1) - \underline{V}_{1,j}^{\pi}(s_1)$. Lemma B.4.1 shows that there always exists a deterministic strategy $\widetilde{\pi} \in \Pi^{\mathrm{det}}$, which is used by Algorithm 18.

$$\max_{\pi'\in\Pi^{\mathrm{full}}} \sum_{j\in[m]} \overline{V}_{1,j}^{\pi'_j,\pi_{-j}}(s_1) - \underline{V}_{1,j}^{\pi}(s_1)$$
$$= \sum_{j\in[m]} \overline{V}_{1,j}^{\widetilde{\pi}_j,\pi_{-j}}(s_1) - \underline{V}_{1,j}^{\pi}(s_1)$$
$$\leq \sum_{j\in[m]}\left[V_{1,j}^{\widetilde{\pi}_j,\pi_{-j}}(s_1) - V_{1,j}^{\pi}(s_1) + \mathbb{E}_{\widetilde{\pi}_j,\pi_{-j}}\sum_{h\in[H]} \widehat{b}_h(s_h,\widetilde{\pi}_{h,j}^{s_h},\pi_{h,-j}^{s_h}) + \mathbb{E}_{\pi}\sum_{h\in[H]}\widehat{b}_h(s_h,\pi_h^{s_h})\right]$$
(Lemma B.4.6)
$$\leq \max_{\pi'\in\Pi^{\mathrm{det}}} \sum_{j\in[m]}\left[V_{1,j}^{\pi'_j,\pi_{-j}}(s_1) - V_{1,j}^{\pi}(s_1)\right] + \sum_{j\in[m]} \mathbb{E}_{\widetilde{\pi}_j,\pi_{-j}}\left[\sum_{h=1}^{H}\widehat{b}_h(s_h,\widetilde{\pi}_{h,j}^{s_h},\pi_{h,-j}^{s_h})\right] + m\mathbb{E}_{\pi}\sum_{h=1}^{H}\left[\widehat{b}_h(s_h,\pi_h^{s_h})\right]$$
$$\leq \mathrm{Gap}(\pi) + \max_{\pi'\in\Pi^{\mathrm{det}}} \sum_{j\in[m]} \mathbb{E}_{\pi'_j,\pi_{-j}}\left[\sum_{h=1}^{H}\widehat{b}_h(s_h,\pi_{h,j}'^{s_h},\pi_{h,-j}^{s_h})\right] + m\mathbb{E}_{\pi}\sum_{h=1}^{H}\left[\widehat{b}_h(s_h,\pi_h^{s_h})\right].$$

□

B.4.1 Dataset-dependent Bound

Lemma B.4.8. *Suppose $\widehat{C}(\pi)$ is finite. For any strategy $\pi' \in \Pi$, $h \in [H]$ and $j \in [m]$, we have*

$$\mathbb{E}_{\pi'_j,\pi_{-j}} b_h(s_h,\pi_{h,j}'^{s_h},\pi_{h,-j}^{s_h}) \leq 2HS\sqrt{\widehat{C}(\pi)\log(\mathcal{N}(\Pi))\iota/n}.$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{\pi'_j,\pi_{-j}} b_h(s_h, \pi'^{s_h}_{h,j}, \pi^{s_h}_{h,-j}) \\
=&\mathbb{E}_{\pi'_j,\pi_{-j}} H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s_h)} \frac{(\pi'_{h,j},\pi_{h,-j})(\mathbf{a}|s_h)^2}{n_h(s,\mathbf{a})} S\log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s_h)} \frac{d_h^{\pi'_j,\pi_{-j}}(s_h)(\pi'_{h,j},\pi_{h,-j})(\mathbf{a}|s_h)^2}{n_h(s_h,\mathbf{a})} S\log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n \\
=&\sum_{s_h\in\mathcal{S}} H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s_h)} \frac{d_h^{\pi'_j,\pi_{-j}}(s_h,\mathbf{a})^2}{n\widehat{d}_h(s_h,\mathbf{a})} S\log(\mathcal{N}(\Pi))\iota} + \sqrt{\iota}/n \\
\leq&\sum_{s_h\in\mathcal{S}} H\sqrt{\sum_{\mathbf{a}\in\mathcal{K}_h(s_h)} \widehat{C}(\pi) d_h^{\pi'_j,\pi_{-j}}(s_h,\mathbf{a}) S\log(\mathcal{N}(\Pi))\iota/n} + \sqrt{\iota}/n \\
\leq&H\sqrt{S^2\widehat{C}(\pi)\log(\mathcal{N}(\Pi))\iota/n} + \sqrt{\iota}/n \qquad \text{(Cauchy-Schwarz inequality)} \\
\leq&2HS\sqrt{\widehat{C}(\pi)\log(\mathcal{N}(\Pi))\iota/n}.
\end{aligned}
$$

□

Lemma B.4.9. *Suppose $\widehat{C}(\pi)$ is finite. For any strategy $\pi' \in \Pi$, $h \in [H]$ and $j \in [m]$, we have*

$$
\mathbb{E}_{\pi'_j,\pi_{-j}} \sum_{\mathbf{a}_h\notin\mathcal{K}_h(s_h)} (\pi'_{h,j},\pi_{h,-j})(\mathbf{a}_h|s_h) = 0.
$$

Proof. Similar to Lemma B.3.8, we have

$$
\begin{aligned}
&\mathbb{E}_{\pi'_j,\pi_{-j}} \sum_{\mathbf{a}_h\notin\mathcal{K}_h(s_h)} (\pi'_{h,j},\pi_{h,-j})(\mathbf{a}_h|s_h) \\
=&\mathbb{E}_{\pi'_j,\pi_{-j}} \sum_{\mathbf{a}_h:\widehat{d}_h(s_h,\mathbf{a}_h)=0} (\pi'_{h,j},\pi_{h,-j})(\mathbf{a}_h|s_h) \\
=&\sum_{\mathbf{a}:\widehat{d}_h(s_h,\mathbf{a}_h)=0} d_h^{\pi'_j,\pi_{-j}}(s_h,\mathbf{a}_h) \\
\leq&\widehat{C}(\pi) \sum_{\mathbf{a}:\widehat{d}_h(s_h,\mathbf{a}_h)=0} \widehat{d}_h(s_h,\mathbf{a}_h) \\
=&0.
\end{aligned}
$$

□

Lemma B.4.10. *For any strategy $\pi \in \Pi$ and $j \in [m]$, we have*

$$\max_{\pi'} \mathbb{E}_{\pi'_j, \pi_{-j}} \left[\sum_{h=1}^{H} \widehat{b}_h(s_h, \pi'^{s_h}_{h,j}, \pi^{s_h}_{h,-j}) \right] \leq 2H^2 S \sqrt{\widehat{C}(\pi) \log(\mathcal{N}(\Pi))\iota / n}.$$

Proof. If $\widehat{C}(\pi)$ is infinite, the argument holds directly. Otherwise it can be derived from Lemma B.4.8 and Lemma B.4.9. □

Theorem B.4.11. *With probability $1 - \delta$, we have*

$$\mathrm{Gap}(\pi^{\mathrm{output}}) \leq \min_{\pi \in \Pi} \left[\mathrm{Gap}(\pi) + 4mH^2 S \sqrt{\widehat{C}(\pi) \log(\mathcal{N}(\Pi))\iota / n} \right].$$

Proof. This can be derived from Lemma B.4.10, Lemma B.4.5 and Lemma B.4.7. □

B.4.2 Dataset-independent Bound

Lemma B.4.12. *Suppose $p_{\min} = \min_{s,\mathbf{a},h}\{d^{\rho}_h(s, \mathbf{a}) : d^{\rho}_h(s, \mathbf{a}) > 0\}$. With probability $1 - \delta$, for all $h, s, \mathbf{a}$, we have*

$$n_h(s, \mathbf{a}) \geq \left(1 - \sqrt{\frac{2 \log(S\Pi_{j \in [m]} A_j H / \delta)}{n p_{\min}}} \right) n d_h(s, \mathbf{a}).$$

As a result, if $n \geq \frac{8 \log(S\Pi_{j \in [m]} A_j H / \delta)}{p_{\min}}$, for all strategy π, we have

$$2C(\pi) \geq \widehat{C}(\pi).$$

Proof. For a fixed $s, \mathbf{a}, h$, for any $\epsilon > 0$ we have

$$\mathbb{P}(n_h(s, \mathbf{a}) < (1 - \epsilon) n d_h(s, \mathbf{a})) \leq \exp\left(-\frac{\epsilon^2 n d_h(s, \mathbf{a})}{2} \right) \leq \exp\left(-\frac{\epsilon^2 n p_{\min}}{2} \right).$$

With a union bound, we have

$$\mathbb{P}(\exists h, s, a, b : \mathbb{P}(n_h(s, a, b) < (1 - \epsilon) n d_h(s, a, b))) \leq S\Pi_{j \in [m]} A_j H \exp\left(-\frac{\epsilon^2 n p_{\min}}{2} \right).$$

The RHS is smaller than δ if we set

$$\epsilon = \sqrt{\frac{2 \log(S\Pi_{j \in [m]} A_j H / \delta)}{n p_{\min}}}.$$

If $n \geq \frac{8 \log(S\Pi_{j \in [m]} A_j H / \delta)}{p_{\min}}$, we have $\widehat{d}_h(s, \mathbf{a}) = \frac{n_h(s, \mathbf{a})}{n} \geq \frac{d_h(s, \mathbf{a})}{2}$. By Definition 3.2.4 and Definition 3.2.3, we have

$$2C(\pi) \geq \widehat{C}(\pi).$$

□

Theorem B.4.13. *If* $n \geq \frac{8\log(S\Pi_{j\in[m]}A_jH/\delta)}{p_{\min}}$*, with probability* $1-\delta$*, we have*

$$\mathrm{Gap}(\pi^{\mathrm{output}}) \leq \min_{\pi\in\Pi}\left[\mathrm{Gap}(\pi) + 4mH^2S\sqrt{2C(\pi)\log(\mathcal{N}(\Pi)\iota/n}\right].$$

Proof. This can be derived by Lemma B.4.12 and Theorem B.4.11. □

B.5 Technical Lemmas

Lemma B.5.1. *(L-1 covering number of probability simplex) For probability simplex* $\Delta(\mathcal{A})$ *and* $A = |\mathcal{A}|$*, there exists a subset* $\Delta'(\mathcal{A}) \subset \Delta(\mathcal{A})$ *such that for any* $p \in \Delta(\mathcal{A})$*, there exists* $p' \in (\mathcal{A})$ *such that* $\|p-p'\|_1 \leq \epsilon$*. In addition,*

$$|\Delta'(\mathcal{A})| \leq \left(\frac{3A}{\epsilon}\right)^A.$$

Proof. We construct ϵ'-net for $\epsilon/2 < \epsilon' \leq \epsilon$ such that $1/\epsilon'$ is integer. Then this ϵ'-net is directly a ϵ-net as $\epsilon' \leq \epsilon$. Define $D(\mathcal{A}) = \{(n_1\epsilon', n_2\epsilon', \cdots, n_A\epsilon'), \sum_{i=1}^A = \frac{1}{\epsilon'}, n_i \in [0, 1/\epsilon']\} \subset \Delta(\mathcal{A})$. For $p = (p_1, p_2, \cdots, p_A) \in \Delta(\mathcal{A})$, suppose

$$k_i\epsilon' \leq p_i < (k_i+1)\epsilon',$$

for some non-negative integers $\{k_i\}$. Set $k = \sum_{i=1}^A k_i$ Then we have $1/\epsilon' - A < k \leq 1/\epsilon'$. Now we construct $p' = (n_1\epsilon', n_2\epsilon', \cdots, n_A\epsilon') \in D(\mathcal{A})$ such that

$$\begin{cases} n_i = k_i + 1, & i \in [1/\epsilon' - k] \\ n_i = k_i, & \text{otherwise.} \end{cases}$$

Then we have $|p_i - p_i'| \leq \epsilon'$ for all $i \in [A]$, which implies

$$\|p - p'\| \leq A\epsilon'.$$

So $|D(\mathcal{A})| \leq \left(\frac{1+\epsilon'}{\epsilon'}\right)^A \leq \left(\frac{3}{\epsilon}\right)^A$ is an $A\epsilon$-net of $\Delta(\mathcal{A})$. We can prove the lemma by rescaling ϵ. □

Lemma B.5.2. *Suppose* $\pi_j, \pi_j' \in \Delta(\mathcal{A}_j)$ *such that* $\left\|\pi_j - \pi_j'\right\|_1 \leq \epsilon$ *for all* $j \in [m]$*. For any function* $f(\mathbf{a}) \in [-H, H]$*, we have*

$$|\mathbb{E}_{\mathbf{a}\sim\pi}f(\mathbf{a}) - \mathbb{E}_{\mathbf{a}\sim\pi'}f(\mathbf{a})| \leq m\epsilon H.$$

Proof.

$$
\begin{aligned}
&|\mathbb{E}_{\mathbf{a}\sim\pi} f(\mathbf{a}) - \mathbb{E}_{\mathbf{a}\sim\pi'} f(\mathbf{a})| \\
=&\left|\sum_{\mathbf{a}} \Pi_{j=1}^m \pi_j(a_j) f(\mathbf{a}) - \sum_{\mathbf{a}} \Pi_{j=1}^m \pi'_j(a_j) f(\mathbf{a})\right| \\
=&\left|\sum_{j=1}^m \sum_{\mathbf{a}_{-j}\in\Pi_{i\neq j}\mathcal{A}_i} \Pi_{i=1}^{j-1}\pi_i(a_i)\Pi_{i=j+1}^m \pi'_i(a_i) \sum_{a_j\in\mathcal{A}_j} \left(\pi_j(a_j) - \pi'_j(a_j)\right) f(\mathbf{a})\right| \\
\leq&\left|\sum_{j=1}^m \sum_{\mathbf{a}_{-j}\in\Pi_{i\neq j}\mathcal{A}_i} \Pi_{i=1}^{j-1}\pi_i(a_i)\Pi_{i=j+1}^m \pi'_i(a_i)\epsilon H\right| \\
=&m\epsilon H.
\end{aligned}
$$

□

Lemma B.5.3. *Suppose $\pi_j, \pi'_j \in \Delta(\mathcal{A}_j)$ such that $\left\|\pi_j - \pi'_j\right\|_1 \leq \epsilon$ for all $j \in [m]$. For any set $\mathcal{K} \subset \Pi_{j\in[m]}\mathcal{A}_j$ and function $n(\mathbf{a}) \geq 1$ we have*

$$
\left|\sqrt{\sum_{\mathbf{a}\in\mathcal{K}} \frac{\pi(\mathbf{a})^2}{n(\mathbf{a})}} - \sqrt{\sum_{\mathbf{a}\in\mathcal{K}} \frac{\pi'(\mathbf{a})^2}{n(\mathbf{a})}}\right| \leq \sqrt{2m\epsilon}.
$$

Proof.

$$
\begin{aligned}
&\left|\sqrt{\sum_{\mathbf{a}\in\mathcal{K}} \frac{\pi(\mathbf{a})^2}{n(\mathbf{a})}} - \sqrt{\sum_{\mathbf{a}\in\mathcal{K}} \frac{\pi'(\mathbf{a})^2}{n(\mathbf{a})}}\right| \\
\leq&\sqrt{\left|\sum_{\mathbf{a}\in\mathcal{K}} \frac{\pi(\mathbf{a})^2 - \pi'(\mathbf{a})^2}{n(\mathbf{a})}\right|} \\
=&\sqrt{\left|\sum_{j=1}^m \sum_{\mathbf{a}_{-j}\in\prod_{i\neq j}\mathcal{A}_i} \prod_{i=1}^{j-1}\pi_i^2(a_i) \prod_{i=j+1}^m \pi_i'^2(a_i) \sum_{a_j\in\mathcal{A}_j} \left(\pi_j^2(a_j) - \pi_j'^2(a_j)\right) \mathbf{1}(\mathbf{a}\in\mathcal{K})/n(\mathbf{a})\right|} \\
\leq&\sqrt{\left|\sum_{j=1}^m \sum_{\mathbf{a}_{-j}\in\prod_{i\neq j}\mathcal{A}_i} \prod_{i=1}^{j-1}\pi_i^2(a_i) \prod_{i=j+1}^m \pi_i'^2(a_i) 2\epsilon\right|} \\
\leq&\sqrt{2m\epsilon}.
\end{aligned}
$$

□

Appendix C

DEFERRED CONTENTS FROM CHAPTER ??

C.1 Properties of Independent Linear Markov Games

Proposition 4.3.2. *ν-misspecified state abstraction Markov games (Example 2) are $H\nu$-misspecified independent linear Markov games with* $\Pi^{\text{abstraction}} = \{\pi \mid \pi_h(\cdot \mid s) = \pi_h(\cdot \mid s'), \psi(s) = \psi(s')\}$, *$d_i = |\mathcal{Z}|A_i$ for all $i \in [m]$ and feature $\phi_i(s, a_i) = e_{\psi(s),a_i}$ to be the canonical basis in $\mathbb{R}^{d_i}$.*

Proof. For all player i, we will let $d_i = |\mathcal{Z}|A_i$ and $\phi_i(s, a_i) = e_{(\psi(s),a_i)}$ be the canonical basis in $\mathbb{R}^{d_i}$. For any policy $\pi \in \Pi^{\text{estimate}}$, by the definition of $\theta_h^{\overline{\pi},\pi_{-i},V}$ (See Equation (4.1)), we have

$$\theta_h^{\overline{\pi},\pi_{-i},V}(z, a_i) = \frac{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s) Q_{h,i}^{\pi_{-i},V}(s, a_i)}{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s)} \in [0, H + 1 - h],$$

where $d_h^{\overline{\pi}}(\cdot)$ is the distribution over $\mathcal{S}$ induced by following policy $\overline{\pi}$ till step h. Thus we have

$$\begin{aligned}
&\text{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s_h, a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V}\right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \\
=&\,\text{proj}_{[0,H+1-h]}\left(\theta_h^{\overline{\pi},\pi_{-i},V}(\psi(s_h), a_{h,i})\right) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \\
=&\theta_h^{\overline{\pi},\pi_{-i},V}(\psi(s_h), a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) \\
=&\frac{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s)\left(Q_{h,i}^{\pi_{-i},V}(s, a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i})\right)}{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s)}.
\end{aligned}$$

On the other hand, for any $z = \psi(s_h) = \psi(s_h')$, $i \in [m]$, $h \in [H]$, $V \in \mathcal{V}$ and $\pi \in \Pi^{\text{estimate}}$, we have

$$\begin{aligned}
&\left|Q_{h,i}^{\pi_{-i},V}(s_h, a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h', a_{h,i})\right| \\
=&\left|\mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[r_{h,i}(s_h, a_{h,i}, a_{h,-i}) + V_{h+1}(s_{h+1})\right] - \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h')}\left[r_{h,i}(s_h', a_{h,i}, a_{h,-i}) + V_{h+1}(s_{h+1})\right]\right| \\
\leq&\mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[\left|r_{h,i}(s_h, \mathbf{a}_{h,i}) - r_{h,i}(s_h', \mathbf{a}_{h,i})\right| + \left|\mathbb{E}_{s_{h+1}\sim\mathbb{P}_h(\cdot|s_h,\mathbf{a}_{h,i})}\left[V_{h+1}(s_{h+1})\right] - \mathbb{E}_{s_{h+1}\sim\mathbb{P}_h(\cdot|s_h',\mathbf{a}_{h,i})}\left[V_{h+1}(s\right.\right.\right.
\end{aligned}$$

(For $\pi \in \Pi^{\text{estimate}}$, we have $\pi_{h,-i}(\cdot \mid s_h) = \pi_{h,-i}(\cdot \mid s_h')$)

$$\leq \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[\epsilon_h(z) + \left|\sum_{s_{h+1}\in\mathcal{S}}\left(\mathbb{P}_h(s_{h+1}\mid s_h, \mathbf{a}_{h,i}) - \mathbb{P}_h(s_{h+1}\mid s_h', \mathbf{a}_{h,i})\right)V_{h+1}(s_{h+1})\right|\right]$$

$$\leq \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}(\cdot|s_h)}\left[\epsilon_h(z) + (H-h)\epsilon_h(z)\right]$$

$$= (H-h+1)\epsilon_h(z).$$

Thus we have

$$\begin{aligned}
&\left|\sum_{h=1}^{H}\mathbb{E}_{\widetilde{\pi}}\left[\mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s_h,a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V}\right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i})\right]\right| \\
\leq &\sum_{h=1}^{H}\mathbb{E}_{\widetilde{\pi}}\left[\left|\mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s_h,a_{h,i}), \theta_h^{\overline{\pi},\pi_{-i},V}\right\rangle\right) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i})\right|\right] \\
= &\sum_{h=1}^{H}\mathbb{E}_{\widetilde{\pi}}\left[\left|\frac{\sum_{s:\psi(s)=\psi(s_h)} d_h^{\overline{\pi}}(s)\left(Q_{h,i}^{\pi_{-i},V}(s,a_{h,i}) - Q_{h,i}^{\pi_{-i},V}(s_h,a_{h,i})\right)}{\sum_{s:\psi(s)=z} d_h^{\overline{\pi}}(s)}\right|\right] \\
\leq &\sum_{h=1}^{H}\mathbb{E}_{\widetilde{\pi}}\left[(H-h+1)\epsilon_h(\psi(s_h))\right] \\
\leq &H\nu,
\end{aligned}$$

where the last inequality is by the definition of ν-misspecified state abstraction Markov games. □

Proposition 4.3.3. *Congestion games (Example 3) are independent linear Markov games with $S = 1$, $H = 1$ and $d_i = F$ for all $i \in [m]$ and misspecification error $\nu = 0$.*

Proof. As $S = 1$ and $H = 1$, we will ignore s and h in the notation. For all player i and action $a_i \in \mathcal{A}_i$, we set $\phi_i(a_i) \in \{0,1\}^F$ such that

$$[\phi_i(a_i)]_f = \begin{cases} 1, & \forall f \in a_i \\ 0, & \forall f \notin a_i. \end{cases}$$

We only need to construct $\theta_i^{\pi_{-i}}$ such that $\left\|\theta_i^{\pi_{-i}}\right\| \leq \sqrt{F}$ and $\left\langle\phi_i(a_i), \theta_i^{\pi_{-i}}\right\rangle = \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[R_i(\mathbf{a})\right] \in [0,1]$ for all policy π and then we will have

$$\mathbb{E}_{a_i\sim\widetilde{\pi}_i}\left[\mathrm{proj}_{[0,1]}\left\langle\phi_i(a_i), \theta_i^{\pi_{-i}}\right\rangle - \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[R_i(\mathbf{a})\right]\right] = 0$$

for all $\widetilde{\pi}$.

For any player i and product policy π_{-i}, we can set

$$\left[\theta_i^{\pi_{-i}}\right]_f = \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[R^f(n^f(a_{-i})+1)\right], \forall f \in \mathcal{F},$$

where we use $n^f(a_{-i})$ to denote the number of players except i using facility f. As each element in $\theta_i^{\pi_{-i}}$ is bounded between $[0,1]$, we have $\left\|\theta_i^{\pi_{-i}}\right\| \leq \sqrt{F}$. In addition, we have

$$\left\langle \phi_i(a_i), \theta_i^{\pi_{-i}} \right\rangle = \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[\sum_{f\in a_i}\left(R^f(n^f(a_{-i})+1)\right)\right] = \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[\sum_{f\in a_i} R^f(n^f(\mathbf{a}))\right] = \mathbb{E}_{a_{-i}\sim\pi_{-i}}\left[R_i(\mathbf{a})\right],$$

which concludes the proof. □

C.2 Proofs for Section 4.4

We will set the parameters for Algorithm 3 to be

- $\lambda = \frac{2\log(16 d_{\max} mNHT/\delta)}{\log(36/35)}$
- $W = H\sqrt{d_{\max}}$
- $\beta = 16(W+H)\sqrt{\lambda + d_{\max}\log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)}$
- $T_{\text{Trig}} = 64\log(8mHN^2/\delta)$
- $K_{\max} = \min\{\frac{2Hmd_{\max}\log(N+\lambda)}{\log(1+T_{\text{Trig}}/4)}, N\}$
- $T = \widetilde{O}(H^4\log(A_{\max})\epsilon^{-2})$ for Markov CCE and $T = \widetilde{O}(H^4 A_{\max}\log(A_{\max})\epsilon^{-2})$ for Markov CE
- $N = \widetilde{O}(m^2H^4d_{\max}^2\epsilon^{-2})$.

We will use subscript k,t to denote the variables in episode k and inner loop t, and subscript h,i to denote the variables at step h and for player i. We will use K to denote the episode that the Algorithm 3 ends ($n^{\text{tot}} = N$ or $K = K_{\max}$) . Immediately we have $K \leq K_{\max} \leq N$.

By the definition of the no-regret learning oracle (Assumption 4.4.1 and Assumption 4.4.2), we have the following two lemmas.

Lemma C.2.1. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. For all $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s,a_i) \geq \max_{a_i \in \mathcal{A}_i} \frac{1}{T}\sum_{t=1}^{T} \overline{Q}_{h,i}^{k,t}(s,a_i) - \frac{H}{T}\cdot \mathrm{Reg}(T).$$

Lemma C.2.2. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.2. For all $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s,a_i) \geq \max_{\psi_i \in \Psi_i} \frac{1}{T}\sum_{t=1}^{T}\sum_{a_i \in \mathcal{A}_i} \pi_{h,i}^{k,t}(a_i \mid s)\overline{Q}_{h,i}^{k,t}(s,\psi_h(a_i \mid s_i)) - \frac{H}{T}\cdot \mathrm{SwapReg}(T).$$

C.2.1 Concentration

The population covariance matrix for episode k, inner loop t, step h and player i is defined as

$$\Sigma_{h,i}^{k} := \mathbb{E}\left[\Sigma_{h,i}^{k,t}\right] = \lambda I + \sum_{l=1}^{k-1} n^l \Sigma_{h,i}^{\pi^l},$$

where $\Sigma_{h,i}^{\pi^k} = \mathbb{E}_{\pi^k}\left[\phi_i(s_h, a_{h,i})\phi_i(s_h, a_{h,i})^\top\right]$. Note that $s_h^l, a_{h,i}^l$ is sampled following the same policy for each inner loop t, so the expected covariance is the same for different t.

We define $\pi^{k,\mathrm{cov}}$ to be the mixture policy of the policy cover Π^k, where policy π^l is given weight/probability $\frac{n^l}{\sum_{j=1}^{k-1} n^j}$. Then we define the on-policy population fit to be

$$\widetilde{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\| \leq W} \mathbb{E}_{(s_h, a_{h,i}) \sim \pi^{k,\mathrm{cov}}} \left\{ \langle \phi_i(s_h, a_{h,i}), \theta \rangle - \mathbb{E}_{a_{h,-i} \sim \pi_{h,-i}^{k,t}(\cdot \mid s)} \left[r_{h,i}(s_h, \mathbf{a}_h) + \overline{V}_{h+1,i}^{k}(s') \right] \right\}^2,$$

$$\widehat{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\| \leq W} \mathbb{E}_{(s_h, a_{h,i}) \sim \pi^{k,\mathrm{cov}}} \left\{ \langle \phi_i(s_h, a_{h,i}), \theta \rangle - \mathbb{E}_{a_{h,-i} \sim \pi_{h,-i}^{k,t}(\cdot \mid s)} \left[r_{h,i}(s_h, \mathbf{a}_h) + \underline{V}_{h+1,i}^{k}(s') \right] \right\}^2.$$

Lemma C.2.3. *(Concentration) With probability at least $1 - \delta/2$, for all $k \in [K]$, $h \in [H]$, $t \in [T]$, $i \in [m]$, we have*

$$\left\| \overline{\theta}_{h,i}^{k,t} - \widetilde{\theta}_{h,i}^{k,t} \right\|_{\Sigma_{h,i}^{k}} \leq 8(W+H)\sqrt{\lambda + d_i \log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)} \leq \beta/2, \tag{C.1}$$

$$\left\| \underline{\theta}_{h,i}^{k,t} - \widehat{\theta}_{h,i}^{k,t} \right\|_{\Sigma_{h,i}^{k}} \leq 8(W+H)\sqrt{\lambda + d_i \log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)} \leq \beta/2, \tag{C.2}$$

$$\frac{1}{2}\Sigma_{h,i}^{k,t} \preceq \Sigma_{h,i}^{k} \preceq \frac{3}{2}\Sigma_{h,i}^{k,t}. \tag{C.3}$$

Proof. By applying Lemma C.7.8 with $Y_{\max} = H$ and union bound, (C.1) and (C.2) holds with probability at least $1 - \delta/4$. For (C.3), we can prove it holds with probability at least $1 - \delta/4$ by applying Lemma C.7.9 with $\lambda > \frac{2\log(16 d_i m K_{\max} HT/\delta)}{\log(36/35)}$ and union bound. □

Lemma C.2.4. *With probability at least $1 - \delta/2$, the following two events hold:*

- *Suppose at episode k, Line 42: $T_{h,i} \geq T_{\text{Trig}}$ is triggered, then we have*

$$\mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} \geq \frac{1}{2n^k} \sum_{j=1}^{n^k} \left\|\phi_i(s_h^{k,j}, a_{h,i}^{k,j})\right\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} \geq \frac{T_{\text{Trig}}}{2n^k},$$

  where j denotes the j-th trajectory collected in the policy cover update (Line 33).

- *For any $k \in [K_{\max}]$, $h \in [H]$, $i \in [m]$, we have*

$$\mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} \leq \frac{2T_{\text{Trig}}}{n^k}.$$

Proof. Note that if at episode k, $T_{h,i} \geq T_{\text{Trig}}$ is triggered, we will have $n^k \leq N$ as otherwise $n^{\text{tot}} = N$ will be triggered. By Lemma C.7.2 with $X_j = \left\|\phi_i(s_h^{k,j}, a_{h,i}^{k,j})\right\|_{\left[\Sigma^{1,k}_{h,i}\right]^{-1}}$, $n_{\max} = N$ and $T_{\text{Trig}} \geq 64\log(8mHK_{\max}N/\delta)$, we have that the argument holds with probability at least $1 - \delta/(2mK_{\max}H)$ for any fixed $k \in [K_{\max}]$, $h \in [H]$ and $i \in [m]$. Then we can prove the lemma by applying union bound. □

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma C.2.3 and Lemma C.2.4 hold, which is with probability at least $1 - \delta$ by Lemma C.2.3 and Lemma C.2.4.

We define the misspecification error to be

$$\overline{\Delta}^{k,t}_{h,i}(s, a_i) := \mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \overline{V}^k_{h+1,i}(s')\right] - \text{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s, a_i), \widetilde{\theta}^{k,t}_{h,i} \right\rangle\right),$$

$$\underline{\Delta}^{k,t}_{h,i}(s, a_i) := \mathbb{E}_{a_{-i} \sim \pi^{k,t}_{h,-i}(\cdot|s)} \left[r_{h,i}(s, \mathbf{a}) + \underline{V}^k_{h+1,i}(s')\right] - \text{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s, a_i), \widehat{\theta}^{k,t}_{h,i} \right\rangle\right).$$

Then by the definition of ν-misspecified linear Markov games, we have the following lemma.

Lemma C.2.5. *For any policy π, we have*

$$\left|\sum_{h=1}^{H} \mathbb{E}_\pi \left[\overline{\Delta}^{k,t}_{h,i}(s, a_i)\right]\right| \leq \nu, \qquad \left|\sum_{h=1}^{H} \mathbb{E}_\pi \left[\underline{\Delta}^{k,t}_{h,i}(s, a_i)\right]\right| \leq \nu.$$

C.2.2 Proofs for Markov CCE

Lemma C.2.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$ we have*

$$-\overline{\Delta}^{k,t}_{h,i}(s,a_i) \le \overline{Q}^{k,t}_{h,i}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}^{k}_{h+1,i}(s')\right]\right] \le 3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^k_{h,i}]^{-1}} - \overline{\Delta}^{k,t}_{h,i}(s,a_i),$$

$$-3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^k_{h,i}]^{-1}} - \underline{\Delta}^{k,t}_{h,i}(s,a_i) \le \underline{Q}^{k,t}_{h,i}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \underline{V}^{k}_{h+1,i}(s')\right]\right] \le -\underline{\Delta}^{k,t}_{h,i}(s,a_i).$$

Proof. We only prove the first argument and the second one holds similarly.

By Lemma C.2.3, for any $s \in \mathcal{S}$, $a_i \in \mathcal{A}_i$, $h \in [H]$, $i \in [m]$, $k \in [K]$, we have

$$\left|\left\langle \phi_i(s,a_i), \overline{\theta}^{k,t}_{h,i} - \widetilde{\theta}^{k,t}_{h,i}\right\rangle\right| \le \left\|\phi_i(s,a_i)\right\|_{[\Sigma^k_{h,i}]^{-1}} \left\|\overline{\theta}^{k,t}_{h,i} - \widetilde{\theta}^{k,t}_{h,i}\right\|_{\Sigma^k_{h,i}} \le \beta/2\left\|\phi_i(s,a_i)\right\|_{[\Sigma^k_{h,i}]^{-1}},$$

where the first inequality is from Cauchy-Schwarz inequality. As a result, we have

$$\begin{aligned}
\overline{Q}^{k,t}_{h,i}(s,a_i) =& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}^{k,t}_{h,i}\right\rangle + \beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right) \\
\ge& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}^{k,t}_{h,i}\right\rangle + \frac{1}{2}\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k}_{h,i}]^{-1}}\right) \qquad \text{(Lemma C.2.3)} \\
\ge& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}^{k,t}_{h,i}\right\rangle\right) \\
=& \mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}^{k}_{h+1,i}(s')\right] - \overline{\Delta}^{k,t}_{h,i}(s,a_i)
\end{aligned}$$

and

$$\begin{aligned}
\overline{Q}^{k,t}_{h,i}(s,a_i) =& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}^{k,t}_{h,i}\right\rangle + \beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right) \\
\le& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \overline{\theta}^{k,t}_{h,i}\right\rangle + 2\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k}_{h,i}]^{-1}}\right) \qquad \text{(Lemma C.2.3)} \\
\le& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}^{k,t}_{h,i}\right\rangle + 3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k}_{h,i}]^{-1}}\right) \\
\le& \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(s,a_i), \widetilde{\theta}^{k,t}_{h,i}\right\rangle\right) + 3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k}_{h,i}]^{-1}} \\
=& \mathbb{E}_{a_{-i}\sim\pi^{k,t}_{h,-i}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}^{k}_{h+1,i}(s')\right] - \overline{\Delta}^{k,t}_{h,i}(s,a_i) + 3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma^{k}_{h,i}]^{-1}},
\end{aligned}$$

which concludes the proof. □

Lemma C.2.7. *(Optimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$\overline{V}^{k}_{1,i}(s_1) \ge V^{\dagger,\pi^k_{-i}}_{1,i}(s_1) - \sum_{h=1}^{H}\mathbb{E}_{\dagger,\pi^k_{-i}}\left[\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}^{k,t}_{h,i}(s_h,a_{h,i})\right] \ge V^{\dagger,\pi^k_{-i}}_{1,i}(s_1) - \nu.$$

Proof. For any $k \in [K]$, $i \in [m]$, under the good event $\mathcal{G}$, we have

$$
\begin{aligned}
&\overline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
=&\operatorname{proj}_{[0,H]}\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) + \frac{H}{T}\cdot\mathrm{Reg}(T)\right) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
\geq&\operatorname{proj}_{[0,H]}\left(\max_{a_{1,i}\in\mathcal{A}_i}\frac{1}{T}\sum_{t=1}^{T}\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i})\right) - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) && \text{(Lemma C.2.1)} \\
\geq&\max_{a_{1,i}\in\mathcal{A}_i}\frac{1}{T}\sum_{t=1}^{T}\left\{\mathbb{E}_{a_{-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s,\mathbf{a}) + \overline{V}_{2,i}^{k}(s')\right] - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right\} - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
& && \text{(Lemma C.2.6)} \\
\geq&\mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[r_{1,i}(s_1,\mathbf{a}_1) + \overline{V}_{2,i}^{k}(s') - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] - V_{1,i}^{\dagger,\pi_{-i}^{k}}(s_1) \\
=&\mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[\overline{V}_{2,i}^{k}(s_2) - V_{2,i}^{\dagger,\pi_{-i}^{k}}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] \\
\geq&-\mathbb{E}_{\dagger,\pi_{-i}^{k}}\left[\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right] \\
\geq&-\nu, && \text{(Lemma C.2.5)}
\end{aligned}
$$

where we use $\mathbb{E}_{\dagger,\pi_{-i}^{k}}$ to denote $\mathbb{E}_{\pi_i',\pi_{-i}^{k}}$ such that π_i' is a best response of π_{-i}^{k}. □

Lemma C.2.8. *(Pessimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$
\underline{V}_{1,i}^{k}(s_1) \leq V_{1,i}^{\pi^k}(s_1) - \sum_{h=1}^{H}\mathbb{E}_{\pi^k}\frac{1}{T}\left[\sum_{t=1}^{T}\underline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right] \leq V_{1,i}^{\pi^k}(s_1) + \nu.
$$

Proof. For any $k \in [K]$, $i \in [m]$, under the good event $\mathcal{G}$, we have

$$
\begin{aligned}
&\underline{V}_{1,i}^{k}(s_1) - V_{1,i}^{\pi^k}(s_1) \\
=&\frac{1}{T}\sum_{t=1}^{T}\sum_{a\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\underline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) - V_{1,i}^{\pi^k}(s_1) \\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s_1,\mathbf{a}_1) + \underline{V}_{2,i}^{k}(s_2)\right] - \underline{\Delta}_{1,i}^{k,t}(s_1,a_i)\right] - V_{1,i}^{\pi^k}(s_1) \\
& && \text{(Lemma C.2.6)} \\
=&\mathbb{E}_{\mathbf{a}_1\sim\pi_1^k(\cdot|s_1)}\left[r_{1,i}(s_1,\mathbf{a}_1) + \underline{V}_{2,i}^{k}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1,a_i)\right] - V_{1,i}^{\pi^k}(s_1)
\end{aligned}
$$

$$
\begin{aligned}
=&\mathbb{E}_{\mathbf{a}_1\sim\pi_1^k(\cdot|s_1)}\left[\underline{V}_{2,i}^k(s_2)-V_{2,i}^{\pi^k}(s_2)-\frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1,a_i)\right]\\
\leq&-\sum_{h=1}^{H}\mathbb{E}_{\pi^k}\left[\frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right]\\
\leq&\nu, && \text{(Lemma C.2.5)}
\end{aligned}
$$

which concludes the proof. □

Lemma C.2.9. *Under the good event $\mathcal{G}$, for all $k\in[K]$ and $i\in[m]$, we have*

$$
V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)-V_{1,i}^{\pi^k}(s_1)-2\nu\leq\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^k(s_1)\leq 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}+\frac{H^2}{T}\cdot\mathrm{Reg}(T)+2\nu.
$$

Proof. The first inequality is from Lemma C.2.7 and Lemma C.2.8. Now we prove the second argument. Under the good event $\mathcal{G}$, for all $k\in[K]$ and $i\in[m]$, we have

$$
\begin{aligned}
&\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^k(s_1)\\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_i\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i})+\frac{H}{T}\cdot\mathrm{Reg}(T)-\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\underline{Q}_{1,i}^{k,t}(s_1,a_{1,i})\\
\leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left(\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s_1,\mathbf{a}_1)+\overline{V}_{2,i}^k(s_2)\right]\right]+3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\overline{\Delta}_{1,i}^{k,t}(s,a_{1,i})\right)\\
&-\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left(\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{h,i}(s_1,\mathbf{a}_1)+\underline{V}_{2,i}^k(s_2)\right]\right]-3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\underline{\Delta}_{1,i}^{k,t}(s,a_{1,i})\right)\\
&+\frac{H}{T}\cdot\mathrm{Reg}(T) && \text{(Lemma C.2.6)}\\
\leq&\frac{1}{T}\sum_{t=1}^{T}\left[\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t}(\cdot|s_1)}\left[\overline{V}_{2,i}^k(s_2)-\underline{V}_{2,i}^k(s_2)\right]\right]\\
&+\mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)}\left[6\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})-\frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]+\frac{H}{T}\cdot\mathrm{Reg}(T)\\
=&\mathbb{E}_{\pi_1^k}\left[\overline{V}_{2,i}^k(s_2)-\underline{V}_{2,i}^k(s_2)\right]+\mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)}\left[6\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})-\frac{1}{T}\sum_{t=1}^{T}\underline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]\\
&+\frac{H}{T}\cdot\mathrm{Reg}(T)\\
\leq&6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}-\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\left(\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})+\underline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right)+\frac{H^2}{T}\cdot\mathrm{Reg}(T)\\
\leq&6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}+2\nu+\frac{H^2}{T}\cdot\mathrm{Reg}(T),
\end{aligned}
$$

which completes the proof. □

Lemma C.2.10. *Under the good event $\mathcal{G}$, for all $i \in [m]$, we have*

$$\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma^k_{h,i}\right]^{-1}} \leq 4T_{\mathrm{Trig}} d_i \log\left(1 + \frac{N}{d_i\lambda}\right).$$

Proof. First, by the triggering condition, we have

$$\sum_{j=1}^{n^k} \left\|\phi_i(s^j_h, a^j_{h,i})\right\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} = \sum_{j=1}^{n^k-1} \left\|\phi_i(s^j_h, a^j_{h,i})\right\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} + \left\|\phi_i(s^{n^k}_h, a^{n^k}_{h,i})\right\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} \leq T_{\mathrm{Trig}} + 1,$$

where j denotes the j-th trajectory collected in the policy cover update (Line 33). By Lemma C.2.4, we have

$$n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma^k_{h,i}\right]^{-1}} \leq 2n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma^{k,1}_{h,i}\right]^{-1}} \leq 4T_{\mathrm{Trig}}.$$

Then by Lemma C.7.6, we have

$$n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma^k_{h,i}\right]^{-1}} \leq 4T_{\mathrm{Trig}} \log \frac{\det(\Sigma^{k+1}_{h,i})}{\det(\Sigma^k_{h,i})}.$$

Thus we have

$$\begin{aligned}
\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma^k_{h,i}\right]^{-1}} &\leq \sum_{k=1}^{K} 4T_{\mathrm{Trig}} \log \frac{\det(\Sigma^{k+1}_{h,i})}{\det(\Sigma^k_{h,i})} \\
&= 4T_{\mathrm{Trig}} \log \frac{\det(\Sigma^{K+1}_{h,i})}{\det(\Sigma^1_{h,i})} \\
&\leq 4T_{\mathrm{Trig}} \left[d_i \log\left(\frac{d_i\lambda + N}{d_i}\right) - d_i \log(\lambda)\right] \\
&= 4T_{\mathrm{Trig}} d_i \log\left(1 + \frac{N}{d_i\lambda}\right),
\end{aligned}$$

where we utilized the fact that

$$\log\det(\Sigma^{K+1}_{h,i}) \leq d_i \log\left(\frac{\mathrm{trace}(\Sigma^{K+1}_{h,i})}{d_i}\right) \leq d_i \log\left(\frac{d_i\lambda + N}{d_i}\right),$$

and complete the proof. □

Lemma C.2.11. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \leq 6mH\beta\sqrt{4N(T_{\mathrm{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)} + \frac{H^2N}{T}\cdot\mathrm{Reg}(T) + 2\nu N.$$

Proof. By Lemma C.2.12, under the good event $\mathcal{G}$, we have $\sum_{k=1}^{K} n^k = n^{\text{tot}} = N$. Thus we have

$$
\begin{aligned}
&\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) \\
\leq &\sum_{k=1}^{K} n^k \max_{i\in[m]} \left[6\beta \mathbb{E}_{\pi^k} \sum_{h=1}^{H} \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}\right] + \frac{H^2}{T}\sum_{k=1}^{K} n^k \text{Reg}(T) + 2\nu N \quad \text{(Lemma C.2.9)} \\
\leq &6\beta \sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \sqrt{\|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2 N}{T}\cdot \text{Reg}(T) + 2\nu N \\
\leq &6\beta \sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K} n^k \sqrt{\mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2 N}{T}\cdot \text{Reg}(T) + 2\nu N \\
&\qquad\qquad \text{(Concavity of } f(x) = \sqrt{x}\text{)} \\
\leq &6\beta \sum_{i\in[m]}\sum_{h=1}^{H} \sqrt{\sum_{k=1}^{K} n^k} \sqrt{\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}}} + \frac{H^2 N}{T}\cdot \text{Reg}(T) + 2\nu N \\
&\qquad\qquad \text{(Cauchy–Schwarz inequality)} \\
\leq &6\beta \sum_{i\in[m]}\sum_{h=1}^{H} \sqrt{N4(T_{\text{Trig}}+1)d_i \log\left(1 + \frac{N}{d_i\lambda}\right)} + \frac{H^2 N}{T}\cdot \text{Reg}(T) + 2\nu N \quad \text{(Lemma C.2.10)} \\
\leq &6\beta m H \sqrt{N4(T_{\text{Trig}}+1)d_{\max} \log\left(1 + \frac{N}{\lambda}\right)} + \frac{H^2 N}{T}\cdot \text{Reg}(T) + 2\nu N.
\end{aligned}
$$

□

Lemma C.2.12. *Under the good event $\mathcal{G}$, we have*

$$K \leq \frac{2Hmd_{\max}\log(N+\lambda)}{\log(1 + T_{\text{Trig}}/4)},$$

which means $K < K_{\max}$ and Algorithm 3 ends due to Line 42 ($n^{\text{tot}} = N_{\max}$).

Proof. By Lemma C.2.4, for any player i and $h \in [H]$, whenever $T_{h,i}^k \geq T_{\text{Trig}}$ is triggered, with probability at least $1-\delta$ we have

$$
\begin{aligned}
n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{\left[\Sigma_{h,i}^k\right]^{-1}} \geq &\frac{1}{2} n^k \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{\left[\Sigma_{h,i}^{k,1}\right]^{-1}} && \text{(Lemma C.2.3)} \\
\geq &\frac{1}{4}\sum_{j=1}^{n^k} \left\|\phi_i(s_h^j, a_{h,i}^j)\right\|^2_{\left[\Sigma_{h,i}^{k,1}\right]^{-1}} && \text{(Lemma C.2.4)} \\
\geq &\frac{T_{\text{Trig}}}{4}.
\end{aligned}
$$

Then by Lemma C.7.6, we have

$$\frac{\det(\Sigma_{h,i}^{k+1})}{\det(\Sigma_{h,i}^{k})} \geq 1 + n^k \mathbb{E}_{\pi^k} \left\|\phi_i(s_h, a_{h,i})\right\|^2_{\left[\Sigma_{h,i}^k\right]^{-1}} \geq 1 + \frac{T_{\text{Trig}}}{4}.$$

Suppose $s_{h,i}$ is the number of triggering $T_{h,i}^k \geq T_{\text{Trig}}$ at level h and player i, then we have

$$\frac{\det(\Sigma_{h,i}^{K+1})}{\det(\Sigma_{h,i}^{1})} \geq \left(1 + \frac{T_{\text{Trig}}}{4}\right)^{s_{h,i}}.$$

In addition, we have

$$\log(\det(\Sigma_{h,i}^1)) = d_i \log(\lambda), \log\det((\Sigma_{h,i}^{K+1})) \leq d_i \log\left(\frac{\text{trace}(\Sigma_{h,i}^{K+1})}{d_i}\right) \leq d_i \log\left(\frac{d_i\lambda + N}{d_i}\right),$$

which gives

$$s_{h,i} \leq \frac{d_i \log(N/d_i + \lambda)}{\log(1 + T_{\text{Trig}}/4)}.$$

Thus, the total number of triggering is bounded by

$$\sum_{i\in[m]} \sum_{h\in[H]} s_{h,i} + 1 \leq \frac{2mHd_{\max} \log(N + \lambda)}{\log(1 + T_{\text{Trig}}/4)},$$

where the additional 1 is from the event $n^{\text{tot}} = N$. □

Theorem 4.4.3. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon + 4\nu)$-approximate Markov CCE. The sample complexity is*

$$O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4 \log(A_{\max})\epsilon^{-4}),$$

where $d_{\max} = \max_{i\in[m]} d_i$ and $A_{\max} = \max_{i\in[m]} A_i$.

Proof. Under the good event $\mathcal{G}$, by Lemma C.2.12, the algorithm ends by $n^{\text{tot}} = N$. By Lemma C.2.11, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma C.2.3 and Lemma C.2.4), we have

$$\begin{aligned}
&\min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right) \\
\leq &\frac{1}{N}\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1)\right)
\end{aligned}$$

$$\leq 6mH\beta\sqrt{4(T_{\text{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)/N}+\frac{H^2}{T}\cdot\text{Reg}(T)+2\nu.$$

By setting $N=\widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T=\widetilde{O}(H^4\log(A_{\max})\epsilon^{-2})$, we can have

$$\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^k(s_1)\right)\leq\epsilon+2\nu.$$

Then by Lemma C.2.9 we have

$$\begin{aligned}\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^{\text{output}}}(s_1)-V_{1,i}^{\pi^{\text{output}}}(s_1)\right)&\leq\max_{i\in[m]}\left(\overline{V}_{1,i}^{k^{\text{output}}}(s_1)-\underline{V}_{1,i}^{k^{\text{output}}}(s_1)\right)+2\nu\\&=\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^k(s_1)\right)+2\nu\\&\leq\epsilon+4\nu,\end{aligned}$$

which completes the proof. □

C.2.3 Proofs for Markov CE

Lemma C.2.13. *(Optimism) Let $\psi_i^k=\text{argmax}_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)$ for all $k\in[K]$ and $i\in[m]$. Under the good event $\mathcal{G}$, for all $k\in[K]$ and $i\in[m]$, we have*

$$\overline{V}_{1,i}^k(s_1)\geq\max_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)-\sum_{h=1}^H\mathbb{E}_{\psi_i^k\diamond\pi^k}\left[\frac{1}{T}\sum_{t=1}^T\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right]\geq\max_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)-\nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k\in[K]$, $h\in[H]$, $i\in[m]$, $s_1\in\mathcal{S}$, we have

$$\begin{aligned}&\overline{V}_{1,i}^k(s_1)-\max_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)\\&=\text{proj}_{[0,H]}\left(\frac{1}{T}\sum_{t=1}^T\sum_{a_i\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i})+\frac{H}{T}\cdot\text{SwapReg}(T)\right)-V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)\\&\geq\text{proj}_{[0,H]}\left(\max_{\psi_{1,i}}\frac{1}{T}\sum_{t=1}^T\sum_{a_i\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\overline{Q}_{1,i}^{k,t}(s_1,\psi_1(a_{1,i}\mid s_1))\right)-V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)\end{aligned}$$

(Lemma C.2.1)

$$\geq\max_{\psi_{1,i}}\frac{1}{T}\sum_{t=1}^T\mathbb{E}_{\mathbf{a}_1\sim\psi_{1,i}\diamond\pi_1^{k,t}(\cdot|s_1)}\left[r_{1,i}(s_1,\mathbf{a}_1)+\underline{V}_{2,i}^k(s_2)-\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]-V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)$$

(Lemma C.2.6)

$$\geq\mathbb{E}_{\psi_{1,i}^k\diamond\pi_1^k}\left[r_{1,i}(s_1,\mathbf{a}_1)+\overline{V}_{2,i}^k(s')-\frac{1}{T}\sum_{t=1}^T\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]-V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)$$

$$
\begin{aligned}
&= \mathbb{E}_{\psi_{1,i}^k \diamond \pi_1^k}\left[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\dagger,\pi_{-i}^k}(s_2) - \frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] \\
&\geq -\mathbb{E}_{\psi_i^k \diamond \pi^k}\left[\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right] \\
&\geq -\nu, && \text{(Lemma C.2.5)}
\end{aligned}
$$

which concludes the proof. □

Lemma C.2.14. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$
\max_{\psi_i} V_{1,i}^{\psi_i \diamond \pi^k}(s_1) - V_{1,i}^{\pi^k}(s_1) - 2\nu \leq \overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \leq 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T}\cdot\mathrm{SwapReg}(T) + 2\nu.
$$

Proof. The first inequality is from Lemma C.2.13 and Lemma C.2.8. Now we prove the second inequality. Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have

$$
\begin{aligned}
&\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \\
\leq &\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s)\overline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) + \frac{H}{T}\cdot\mathrm{SwapReg}(T) - \frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\underline{Q}_{1,i}^{k,t}(s_1,a_{1,i}) \\
\leq &\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left(\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s)}\left[r_{1,i}(s_1,\mathbf{a}_1) + \overline{V}_{2,i}^k(s_2)\right]\right] + 3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1,a_1\right. \\
&- \frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left(\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s_1,\mathbf{a}_1) + \underline{V}_{2,i}^k(s_2)\right]\right] - 3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \underline{\Delta}_{1,i}^{k,t}(s\right. \\
&+ \frac{H}{T}\cdot\mathrm{SwapReg}(T) && \text{(Lemma C.2.6)} \\
= &\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t}(\cdot|s_1)}\left[\overline{V}_{2,i}^k(s_2) - \underline{V}_{2,i}^k(s_2)\right] + \mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)}\left[6\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i}) - \underline{\Delta}_{1,i}^{k,t}(s_1\right.\right. \\
&+ \frac{H}{T}\cdot\mathrm{SwapReg}(T) \\
= &\mathbb{E}_{\pi_1^k}\left[\overline{V}_{2,i}^k(s_2) - \underline{V}_{2,i}^k(s_2)\right] + \mathbb{E}_{a_{1,i}\sim\pi_{1,i}^k}\left[6\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i}) - \underline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right] + \frac{H}{T}\cdot\mathrm{SwapR}\right. \\
\leq &6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}} - \mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left(\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i}) + \underline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})\right) + \frac{H^2}{T}\cdot\mathrm{SwapReg}(T) \\
\leq &6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T}\cdot\mathrm{SwapReg}(T) + 2\nu.
\end{aligned}
$$

□

Lemma C.2.15. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) \le 6mH\beta\sqrt{4N(T_{\text{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)} + \frac{H^2N}{T}\cdot\text{SwapReg}(T) + 2\nu.$$

Proof. The proof is similar to the proof for Lemma C.2.11, where the only difference is that we replace Lemma C.2.11 with Lemma C.2.14 in the proof. □

Theorem 4.4.4. *Suppose Algorithm 3 is instantiated with no-regret learning oracles satisfying Assumption 4.4.2. Then for ν-misspecified independent linear Markov games with $\Pi^{\text{estimate}} = \{\pi^{k,t}\}_{k,t=1,1}^{K,T}$, with probability at least $1-\delta$, Algorithm 3 will output an $(\epsilon+4\nu)$-approximate Markov CE. The sample complexity is*

$$O(mHTK_{\max}N) = \widetilde{O}(m^4H^{10}d_{\max}^4A_{\max}\log(A_{\max})\epsilon^{-4}).$$

Proof. Under the good event $\mathcal{G}$, by Lemma C.2.12, the algorithm ends by $n^{\text{tot}} = N$. By Lemma C.2.15, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma C.2.3 and Lemma C.2.4), we have

$$\begin{aligned}
&\min_{k\in[K]}\max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) \\
\le&\frac{1}{N}\sum_{k=1}^{K} n^k \sum_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) \\
\le&6mH\beta\sqrt{4(T_{\text{Trig}}+1)d_{\max}\log\left(1+\frac{N}{\lambda}\right)/N} + \frac{H^2}{T}\cdot\text{SwapReg}(T) + 2\nu.
\end{aligned}$$

By setting $N = \widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T = \widetilde{O}(H^4A_{\max}\log(A_{\max})\epsilon^{-2})$, we can have

$$\min_{k\in[K]}\max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) \le \epsilon + 2\nu.$$

Then by Lemma C.2.14, we have

$$\begin{aligned}
\max_{i\in[m]}\left(\max_{\psi_i} V_{1,i}^{\psi_i\diamond\pi^{\text{output}}}(s_1) - V_{1,i}^{\pi^{\text{output}}}(s_1)\right) \le& \max_{i\in[m]}\left(\overline{V}_{1,i}^{k^{\text{output}}}(s_1) - \underline{V}_{1,i}^{k^{\text{output}}}(s_1)\right) + 2\nu \\
=& \min_{k\in[K]}\max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{k}(s_1)\right) + 2\nu \\
\le&\epsilon + 4\nu,
\end{aligned}$$

which thus completes the proof. □

C.3 Algorithms for Learning Markov CCE/CE without Communication

In this section, we present a communication-free algorithm for independent linear Markov games. The key difference is that we leverage an agile policy cover update scheme, i.e., the policy cover is updated whenever a new π^k is learned (Line 25), and the policy certification is replaced by a uniform sampling procedure (Line 27).

We will set the parameters for Algorithm 21 to be

- $\lambda = \frac{2\log(16 d_{\max} mKHT/\delta)}{\log(36/35)}$
- $W = H\sqrt{d_{\max}}$
- $\beta = 16(W+H)\sqrt{\lambda + d_{\max}\log(32WN(W+H)) + 4\log(8mK_{\max}HT/\delta)}$
- $T = \widetilde{O}(H^4 \log(A_{\max})\epsilon^{-2})$ for Markov CCE and $T = \widetilde{O}(H^4 A_{\max}\log(A_{\max})\epsilon^{-2})$ for Markov CE
- $K = \widetilde{O}(m^2 H^4 d_{\max}^2 \epsilon^{-2})$.

C.3.1 Concentration

The population covariance matrix for episode k, inner loop t, step h and player i is defined as

$$\Sigma_{h,i}^k := \mathbb{E}\left[\widehat{\Sigma}_{h,i}^{k,t}\right] = \lambda I + \sum_{l=1}^{k-1} \Sigma_{h,i}^{\pi^l},$$

where $\Sigma_{h,i}^{\pi^k} = \mathbb{E}_{\pi^k}\left[\phi_i(s_h, a_{h,i})\phi_i(s_h, a_{h,i})^\top\right]$. Note that $s_h^l, a_{h,i}^l$ is sampled following the same policy for each inner loop t, so the expected covariance is the same for different t.

We define $\pi^{k,\mathrm{cov}}$ to be the mixture policy in $\Pi^k = \{\pi^l\}_{l=1}^{k-1}$, where policy π^l is given weight/probability $\frac{1}{k-1}$, and also define

$$\widetilde{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\|\le W} \mathbb{E}_{(s_h,a_{h,i})\sim\pi^{k,\mathrm{cov}}}\left\{\langle\phi_i(s_h,a_{h,i}),\theta\rangle - \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s_h,\mathbf{a}_h) + \overline{V}_{h+1,i}^k(s')\right]\right\}^2,$$

$$\widehat{\theta}_{h,i}^{k,t} := \operatorname*{argmin}_{\|\theta\|\le W} \mathbb{E}_{(s_h,a_{h,i})\sim\pi^{k,\mathrm{cov}}}\left\{\langle\phi_i(s_h,a_{h,i}),\theta\rangle - \mathbb{E}_{a_{h,-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s_h,\mathbf{a}_h) + \underline{V}_{h+1,i}^k(s')\right]\right\}^2.$$

Algorithm 20 Communication-free **P**olicy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (Communication-free **PReFI**) (Part 1)

Input: λ, β, K, T

Initialization: Policy Cover $\Pi = \emptyset$.

for episode $k = 1, 2, \ldots, K$ **do**

  Set $\overline{V}^k_{H+1,i}(\cdot) = \underline{V}^k_{H+1,i}(\cdot) = 0$.

  for $h = H, H-1, \ldots, 1$ **do** ▷ Retrain policy with the current policy cover

    Initialize $\pi^{1,k}_{h,i}$ to be uniform policy for all player i. Initialize $\overline{V}^k_{h,i}(\cdot) = \underline{V}^k_{h,i}(\cdot) = 0$.

    Each player i initializes a no-regret learning instance (Protocol 1) at each state $s \in \mathcal{S}$ and step $h \in [H]$, for which we will use $\textsc{No\_Regret\_Update}_{h,i,s}(\cdot)$ to denote the update.

    for $t = 1, 2, \ldots, T$ **do**

      for $i \in [m]$ **do**

        Set Dataset $\mathcal{D}^{k,t}_{h,i} = \emptyset$

        for $l = 1, 2, \ldots, k-1$ **do**

          Draw a joint trajectory $(s^l_1, \mathbf{a}^l_1, r^l_{1,i}, \ldots, s^l_h, \mathbf{a}^l_h, r^l_{h,i}, s^l_{h+1})$ from $\pi^l_{1:h-1} \circ \left(\pi^l_{h,i}, \pi^{k,t}_{h,-i}\right)$, where π^l is the policy learned at episode l stored in policy cover Π.

          Add $(s^l_h, a^l_{h,i}, r^l_{h,i}, s^l_{h+1})$ to $\mathcal{D}^{k,t}_{h,i}$.

        end for

        Set $\Sigma^{k,t}_{h,i} = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \phi_i(s,a)\phi_i(s,a)^\top$.

        Set $\overline{\theta}^{k,t}_{h,i} = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}^{k,t}_{h,i}} \left(\langle \phi_i(s,a), \theta \rangle - r - \overline{V}^k_{h+1,i}(s')\right)^2$.

        Set $\overline{Q}^{k,t}_{h,i}(\cdot,\cdot) = \operatorname{proj}_{[0,H+1-h]}\left(\left\langle \phi_i(\cdot,\cdot), \overline{\theta}^{k,t}_{h,i} \right\rangle + \beta \left\|\phi_i(\cdot,\cdot)\right\|_{[\Sigma^{k,t}_{h,i}]^{-1}}\right)$.

        Update $\overline{V}^k_{h,i}(s) \leftarrow \frac{t-1}{t}\overline{V}_{h,i}(s) + \frac{1}{t}\sum_{a_i \in \mathcal{A}_i} \pi^{k,t}_{h,i}(a_i|s)\overline{Q}^{k,t}_{h,i}(s,a)$ for all $s \in \mathcal{S}$.

        Update the no-regret learning instance at step h and state s: $\pi^{k,t+1}_{h,i}(\cdot \mid s) \leftarrow \textsc{No\_Regret\_Update}_{h,i,s}(1 - \overline{Q}^{k,t}_{h,i}(s,\cdot)/H)$ for all $s \in \mathcal{S}$.

      end for

    end for

    Set $\overline{V}^k_{h,i}(s) \leftarrow \operatorname{proj}_{[0,H+1-h]}\left(\overline{V}^k_{h,i}(s) + \frac{H}{T} \cdot (\text{Swap})\text{Reg}(T)\right)$ for all $i \in [m]$ and $s \in \mathcal{S}$.

  end for

Algorithm 21 Communication-free **P**olicy **Re**ply with **F**ull **I**nformation Oracle in Independent Linear Markov Games (Communication-free **PReFI**) (Part 2)

Set π^k to be the Markov joint policy such that $\pi_h^k(\mathbf{a}|s) = \frac{1}{T}\sum_{t=1}^{T}\prod_{i\in[m]}\pi_{h,i}^{k,t}(a_i|s)$.

Update $\Pi \leftarrow \Pi \bigcup \{\pi^k\}$. ▷ Policy cover update

end for

Sample $k \sim \mathrm{Unif}(K)$ and output $\pi^{\mathrm{output}} = \pi^k$.

Lemma C.3.1. *(Concentration) With probability at least $1-\delta/2$, for all $k \in [K]$, $h \in [H]$, $t \in [T]$, $i \in [m]$, we have*

$$\left\|\overline{\theta}_{h,i}^{k,t} - \widetilde{\theta}_{h,i}^{k,t}\right\|_{\Sigma_{h,i}^k} \le 8(W+H)\sqrt{\lambda + d_i \log(32WK(W+H)) + 4\log(8mKHT/\delta)} \le \beta/2, \tag{C.4}$$

$$\left\|\underline{\theta}_{h,i}^{k,t} - \widehat{\theta}_{h,i}^{k,t}\right\|_{\Sigma_{h,i}^k} \le 8(W+H)\sqrt{\lambda + d_i \log(32WK(W+H)) + 4\log(8mKHT/\delta)} \le \beta/2, \tag{C.5}$$

$$\frac{1}{2}\Sigma_{h,i}^{k,t} \preceq \Sigma_{h,i}^k \preceq \frac{3}{2}\Sigma_{h,i}^{k,t}. \tag{C.6}$$

Proof. The proof is the same as the proof for Lemma C.2.3. □

With a slight abuse of the notation, we will still denote the high probability event in Lemma C.3.1 as $\mathcal{G}$ Now we define

$$\overline{\Delta}_{h,i}^{k,t}(s,a_i) = \mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s,a_i), \widetilde{\theta}_{h,i}^{k,t}\right\rangle\right),$$

$$\underline{\Delta}_{h,i}^{k,t}(s,a_i) = \mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \underline{V}_{h+1,i}^k(s')\right] - \mathrm{proj}_{[0,H+1-h]}\left(\left\langle\phi_i(s,a_i), \widehat{\theta}_{h,i}^{k,t}\right\rangle\right).$$

Lemma C.3.2. *Under good event $\mathcal{G}$, for all $k \in [K]$, $t \in [T]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$ and $a_i \in \mathcal{A}_i$ we have*

$$-\overline{\Delta}_{h,i}^{k,t}(s,a_i) \le \overline{Q}_{h,i}^{k,t}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')\right]\right] \le 3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^k]^{-1}} - \overline{\Delta}_{h,i}^{k,t}(s,a_i),$$

$$-3\beta\left\|\phi_i(s,a_i)\right\|_{[\Sigma_{h,i}^k]^{-1}} - \underline{\Delta}_{h,i}^{k,t}(s,a_i) \le \underline{Q}_{h,i}^{k,t}(s,a_i) - \left[\mathbb{E}_{a_{-i}\sim\pi_{h,-i}^{k,t}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \underline{V}_{h+1,i}^k(s')\right]\right] \le -\underline{\Delta}_{h,i}^{k,t}(s,a_i).$$

Proof. The proof is the same as the proof for Lemma C.2.6. □

C.3.2 Proofs for Learning Markov CCE with Algorithm 21

Lemma C.3.3. *Under the good event $\mathcal{G}$, for all $k \in [K]$ and $i \in [m]$, we have*

$$V_{1,i}^{\dagger,\pi^k_{-i}}(s_1) - V_{1,i}^{\pi^k}(s_1) - \nu \le \overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \le 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H}{T} \cdot \mathrm{Reg}(T) + \nu.$$

Proof. The first inequality is from Lemma C.2.7. Now we prove the second argument:

$$\begin{aligned}
&\overline{V}_{1,i}^k(s_1) - V_{1,i}^{\pi^k}(s_1) \\
\le& \frac{1}{T} \sum_{t=1}^T \sum_{a_{1,i} \in \mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1) \overline{Q}_{1,i}^{k,t}(s_1, a_{1,i}) + \frac{H}{T} \cdot \mathrm{Reg}(T) - V_{1,i}^{\pi^k}(s_1) \\
\le& \frac{1}{T} \sum_{t=1}^T \sum_{a_{1,i} \in \mathcal{A}_i} \pi_{1,i}^{k,t}(a_{1,i} \mid s_1) \left(\left[\mathbb{E}_{a_{1,-i} \sim \pi_{1,-i}^{k,t}(\cdot|s_1)} \left[r_{h,i}(s_1, \mathbf{a}_1) + \overline{V}_{2,i}^k(s_2) \right] \right] + 3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \right) \\
&+ \frac{H}{T} \cdot \mathrm{Reg}(T) - V_{1,i}^{\pi^k}(s_1) && \text{(Lemma C.3.2)} \\
\le& \frac{1}{T} \sum_{t=1}^T \left(\left[\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^{k,t}(\cdot|s_1)} \left[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2) \right] \right] + \mathbb{E}_{a_{1,i} \sim \pi_{1,i}^{k,t}(\cdot|s_1)} \left[3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} - \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \right] \right) \\
&+ \frac{H}{T} \cdot \mathrm{Reg}(T) \\
\le& \mathbb{E}_{\pi_1^k} \left[\overline{V}_{2,i}^k(s_2) - V_{2,i}^{\pi^k}(s_2) \right] + \mathbb{E}_{a_{1,i} \sim \pi_{1,i}^k(\cdot|s_1)} \left[3\beta \|\phi_i(s_1, a_{1,i})\|_{[\Sigma_{1,i}^k]^{-1}} - \frac{1}{T} \sum_{t=1}^T \overline{\Delta}_{1,i}^{k,t}(s_1, a_{1,i}) \right] + \frac{H}{T} \cdot \mathrm{Reg}(T) \\
\le& 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} - \mathbb{E}_{\pi^k} \sum_{h=1}^H \frac{1}{T} \sum_{t=1}^T \overline{\Delta}_{h,i}^{k,t}(s_h, a_{h,i}) + \frac{H^2}{T} \cdot \mathrm{Reg}(T) \\
\le& 3\beta \mathbb{E}_{\pi^k} \sum_{h=1}^H \|\phi_i(s_h, a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}} + \frac{H^2}{T} \cdot \mathrm{Reg}(T) + \nu. && \text{(Lemma C.2.5)}
\end{aligned}$$

□

Lemma C.3.4. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^K \mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}} \le d_i \log(1 + \frac{K}{d_i \lambda}).$$

Proof. As $\|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-1}} \le 1$, by Lemma C.7.6 we have

$$\mathbb{E}_{\pi^k} \|\phi_i(s_h, a_{h,i})\|^2_{[\Sigma_{h,i}^k]^{-2}} \le \log \frac{\det(\Sigma_{h,i}^{k+1})}{\det(\Sigma_{h,i}^k)}.$$

Thus we have

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}} \leq& \sum_{k=1}^{K}\log\frac{\det(\Sigma^{k+1}_{h,i})}{\det(\Sigma^k_{h,i})}\\ =&\log\frac{\det(\Sigma^{K+1}_{h,i})}{\det(\Sigma^1_{h,i})}\\ \leq& d_i\log(1+\frac{K}{d_i\lambda}),\end{aligned}$$

where we utilized the fact that

$$\log\det(\Sigma^{K+1}_{h,i}) \leq d_i\log\left(\frac{\mathrm{trace}(\Sigma^{K+1}_{h,i})}{d_i}\right) \leq d_i\log\left(\frac{d_i\lambda+K}{d_i}\right).$$

□

Lemma C.3.5. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1)-V^{\pi^k}_{1,i}(s_1)\right) \leq 3mH\beta\sqrt{Kd_{\max}\log\left(1+\frac{K}{\lambda}\right)}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K.$$

Proof.

$$\begin{aligned}&\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1)-V^{\pi^k}_{1,i}(s_1)\right)\\ \leq&3\beta\sum_{k=1}^{K}\max_{i\in[m]}\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma^k_{h,i}]^{-1}}+\frac{H^2}{T}\sum_{k=1}^{K}\mathrm{Reg}(T)+\nu K && \text{(Lemma C.3.3)}\\ =&3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K}\mathbb{E}_{\pi^k}\sqrt{\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}}}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K\\ \leq&3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sum_{k=1}^{K}\sqrt{\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}}}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K\\ & && \text{(Concavity of } f(x)=\sqrt{x})\\ \leq&3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sqrt{K\sum_{k=1}^{K}\mathbb{E}_{\pi^k}\left\|\phi_i(s_h,a_{h,i})\right\|^2_{[\Sigma^k_{h,i}]^{-1}}}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K\\ & && \text{(Cauchy–Schwarz inequality)}\\ \leq&3\beta\sum_{i\in[m]}\sum_{h=1}^{H}\sqrt{Kd_i\log\left(1+\frac{K}{d_i\lambda}\right)}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K && \text{(Lemma C.2.10)}\\ \leq&3mH\beta\sqrt{Kd_{\max}\log\left(1+\frac{K}{\lambda}\right)}+\frac{H^2K}{T}\cdot\mathrm{Reg}(T)+\nu K.\end{aligned}$$

□

Theorem C.3.6. *Suppose Algorithm 21 is instantiated with no-regret learning oracles satisfying Assumption 4.4.1. Then for ν-misspecified linear Markov games, with probability 0.9, Algorithm 21 will output an $(\epsilon + 2\nu)$-approximate Markov CCE. The sample complexity is $O(mHTK^2) = \widetilde{O}(m^5H^{13}d_{\max}^6\log(A_{\max})\epsilon^{-6})$, where $d_{\max} = \max_{i\in[m]} d_i$ and $A_{\max} = \max_{i\in[m]} A_i$.*

Proof. By Lemma C.3.5, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma C.2.3), we have

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^k}(s_1)-V_{1,i}^{\pi^k}(s_1)\right)\leq&\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}_{1,i}^{k}(s_1)-V_{1,i}^{\pi^k}(s_1)\right)+\nu\\ &\text{(Lemma C.2.7)}\\ \leq&3mH\beta\sqrt{d_{\max}\log\left(1+\frac{K}{\lambda}\right)/K}+\frac{H^2}{T}\cdot\mathrm{Reg}(T)+2\nu.\\ &\text{(Lemma C.3.5)}\end{aligned}$$

By Markov's inequality, we set $K = \widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T = \widetilde{O}(H^4\log(A_{\max})\epsilon^{-2})$, with probability 0.9 we have

$$\max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^{\text{output}}}(s_1)-V_{1,i}^{\pi^{\text{output}}}(s_1)\right)\leq\epsilon+2\nu.$$

□

C.3.3 Proofs for Learning Markov CE with Algorithm 21

Lemma C.3.7. *Under the good event $\mathcal{G}$, for all $k\in[K]$ and $i\in[m]$, we have*

$$\max_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)-V_{1,i}^{\pi^k}(s_1)-\nu\leq\overline{V}_{1,i}^{k}(s_1)-V_{1,i}^{\pi^k}(s_1)\leq3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\|\phi_i(s_h,a_{h,i})\|_{[\Sigma_{h,i}^k]^{-1}}+\frac{H}{T}\cdot\mathrm{SwapReg}(T)+\nu.$$

Proof. The first inequality is from Lemma C.2.13. Now we prove the second argument.

$$\begin{aligned}&\overline{V}_{1,i}^{k}(s_1)-V_{1,i}^{\pi^k}(s_1)\\ \leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s)\overline{Q}_{1,i}^{k,t}(s,a_{1,i})+\frac{H}{T}\cdot\mathrm{SwapReg}(T)-V_{1,i}^{\pi^k}(s_1)\\ \leq&\frac{1}{T}\sum_{t=1}^{T}\sum_{a_{1,i}\in\mathcal{A}_i}\pi_{1,i}^{k,t}(a_{1,i}\mid s_1)\left(\left[\mathbb{E}_{a_{1,-i}\sim\pi_{1,-i}^{k,t}(\cdot|s_1)}\left[r_{1,i}(s_1,\mathbf{a}_1)+\overline{V}_{2,i}^{k}(s_2)\right]\right]+3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right)\end{aligned}$$

$$
\begin{aligned}
&+\frac{H}{T}\cdot\mathrm{SwapReg}(T)-V_{1,i}^{\pi^k}(s_1) && \text{(Lemma C.3.2)}\\
\le&\frac{1}{T}\sum_{t=1}^{T}\left(\left[\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t}(\cdot|s_1)}\left[\overline{V}_{2,i}^{k}(s_2)-V_{2,i}^{\pi^k}(s_2)\right]\right]+3\beta\mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k,t}(\cdot|s_1)}\left\|\phi_i(s,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right)\\
&+\frac{H}{T}\cdot\mathrm{SwapReg}(T)\\
\le&\mathbb{E}_{\pi_1^k}\left[\overline{V}_{2,i}^{k}(s_2)-V_{2,i}^{\pi^k}(s_2)\right]+\mathbb{E}_{a_{1,i}\sim\pi_{1,i}^{k}(\cdot|s_1)}\left[3\beta\left\|\phi_i(s_1,a_{1,i})\right\|_{[\Sigma_{1,i}^k]^{-1}}-\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{1,i}^{k,t}(s_1,a_{1,i})\right]+\frac{H}{T}\cdot\mathrm{SwapReg}(T\\
\le&3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}}-\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T}\overline{\Delta}_{h,i}^{k,t}(s_h,a_{h,i})+\frac{H^2}{T}\cdot\mathrm{SwapReg}(T)\\
\le&3\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H}\left\|\phi_i(s_h,a_{h,i})\right\|_{[\Sigma_{h,i}^k]^{-1}}+\frac{H^2}{T}\cdot\mathrm{SwapReg}(T)+\nu, && \text{(Lemma C.2.5)}
\end{aligned}
$$

which completes the proof. □

Lemma C.3.8. *Under the good event $\mathcal{G}$, we have*

$$
\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}_{1,i}^{k}(s_1)-V_{1,i}^{\pi^k}(s_1)\right)\le 3mH\beta\sqrt{Kd_{\max}\log\left(1+\frac{K}{\lambda}\right)}+\frac{H^2K}{T}\cdot\mathrm{SwapReg}(T)+\nu.
$$

Proof. The proof is the same as the proof for Lemma C.3.5 where we replace Lemma C.3.3 with Lemma C.3.7 in the proof. □

Theorem C.3.9. *Suppose Algorithm 21 is instantiated with no-regret learning oracles satisfying Assumption 4.4.2. Then for ν-misspecified linear Markov games, with probability 0.9, Algorithm 21 will output an $(\epsilon+2\nu)$-approximate Markov CCE. The sample complexity is $O(mHTK^2)=\widetilde{O}(m^5H^{13}d_{\max}^6A_{\max}\log(A_{\max})\epsilon^{-6})$, where $d_{\max}=\max_{i\in[m]}d_i$ and $A_{\max}=\max_{i\in[m]}A_i$.*

Proof. By Lemma C.3.8, under the good event $\mathcal{G}$, which happens with probability at least $1-\delta$ (Lemma C.3.1), we have

$$
\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\max_{\psi_i}V_{1,i}^{\psi_i\diamond\pi^k}(s_1)-V_{1,i}^{\pi^k}(s_1)\right)\le&\frac{1}{K}\sum_{k=1}^{K}\max_{i\in[m]}\left(\overline{V}_{1,i}^{k}(s_1)-V_{1,i}^{\pi^k}(s_1)\right)+\nu\\
&\text{(Lemma C.2.13)}\\
\le&3mH\beta\sqrt{d_{\max}\log\left(1+\frac{K}{\lambda}\right)/K}+\frac{mH^2}{T}\cdot\mathrm{SwapReg}(T)+2\nu.\\
&\text{(Lemma C.3.8)}
\end{aligned}
$$

By Markov's inequality, we set $K = \widetilde{O}(m^2H^4d_{\max}^3\epsilon^{-2})$ and $T = \widetilde{O}(H^4A_{\max}\log(A_{\max})\epsilon^{-2})$, with probability 0.9, we have

$$\max_{i\in[m]}\max_{\psi_i}\left(V_{1,i}^{\psi_i\diamond\pi^{\text{output}}}(s_1) - V_{1,i}^{\pi^{\text{output}}}(s_1)\right) \le \epsilon,$$

which completes the proof. □

C.4 Algorithms for Learning Optimal Policies in Misspecified Linear MDP

In this section, we adapt Algorithm 3 to the linear MDP setting. As the single-agent degeneration of independent linear Markov games, we can remove the no-regret learning loop in Algorithm 3 and achieve better sample complexity. The analysis is almost the same as the analysis for Algorithm 3 in Appendix C.2 with $T = 1$ and $m = 1$.

We will set the parameters for Algorithm 3 to be

- $\lambda = \frac{2\log(16dNH/\delta)}{\log(36/35)}$
- $W = H\sqrt{d}$
- $\beta = 16(W+H)\sqrt{\lambda + d\log(32W(W+H)) + 4\log(8K_{\max}H/\delta)}$
- $T_{\text{Trig}} = 64\log(8HN^2/\delta)$
- $K_{\max} = \min\{\frac{2Hd\log(N+\lambda)}{\log(1+T_{\text{Trig}}/4)}, N\}$
- $N = \widetilde{O}(H^4d^2\epsilon^{-2})$.

We will use K to denote the episode that Algorithm 22 ends ($n^{\text{tot}} = N$ or $K = K_{\max}$). Immediately we have $K \le K_{\max} \le N$.

The population covariance matrix for episode k, step h is defined as

$$\Sigma_h^k := \mathbb{E}\left[\widehat{\Sigma}_h^k\right] = \lambda I + \sum_{l=1}^{k-1} n^l \Sigma_h^{\pi^l},$$

where $\Sigma_h^{\pi^k} = \mathbb{E}_{\pi^k}\left[\phi(s_h,a_h)\phi(s_h,a_h)^\top\right]$.

Algorithm 22 Policy Replay for Misspecified MDP with linear function approximation (Part 1)

Input: ϵ, δ, λ, β, T_{Trig}, $K_{\max}$, N

Initialization: Policy Cover $\Pi = \emptyset$. $n^{\text{tot}} = 0$.

for episode $k = 1, 2, \ldots, K_{\max}$ **do**

  Set $\overline{V}^k_{H+1}(\cdot) = \underline{V}^k_{H+1}(\cdot) = 0$, $n^k = 0$.

  for $h = H, H-1, \ldots, 1$ **do** ▷ Retrain policy with the current policy cover

    Initialize $\overline{V}^k_h(\cdot) = \underline{V}^k_h(\cdot) = 0$.

    Set Dataset $\mathcal{D}^k_h = \emptyset$.

    for $l = 1, 2, \ldots, \sum_{j=1}^{k-1} n^j$ **do**

      Sample π^l with probability $n^l / \sum_{j=1}^{k-1} n^j$.

      Draw a joint trajectory $(s^l_1, a^l_1, r^l_1, \ldots, s^l_H, a^l_H, r^l_H, s^l_{H+1})$ from π^l.

      Add $(s^l_h, a^l_{h,i}, r^l_{h,i}, s^l_{h+1})$ to $\mathcal{D}^k_h$.

    end for

    Set $\widehat{\Sigma}^k_h = \lambda I + \sum_{(s,a,r,s') \in \mathcal{D}^k_h} \phi(s,a)\phi(s,a)^\top$.

    Set $\overline{\theta}^k_h = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}^k_h} \left(\langle \phi(s,a), \theta \rangle - r - \overline{V}^k_{h+1}(s') \right)^2$.

    Set $\underline{\theta}^k_h = \operatorname{argmin}_{\|\theta\| \le H\sqrt{d}} \sum_{(s,a,r,s') \in \mathcal{D}^k_h} \left(\langle \phi(s,a), \theta \rangle - r - \underline{V}^k_{h+1}(s') \right)^2$.

    Set $\overline{Q}^k_h(\cdot,\cdot) = \text{proj}_{[0,H+1-h]} \left(\left\langle \phi(\cdot,\cdot), \overline{\theta}^k_h \right\rangle + \beta \left\| \phi(\cdot,\cdot) \right\|_{[\widehat{\Sigma}^k_h]^{-1}} \right)$.

    Set $\underline{Q}^k_h(\cdot,\cdot) = \text{proj}_{[0,H+1-h]} \left(\left\langle \phi(\cdot,\cdot), \underline{\theta}^k_h \right\rangle - \beta \left\| \phi(\cdot,\cdot) \right\|_{[\widehat{\Sigma}^k_h]^{-1}} \right)$.

    Set $\overline{V}^k_h(\cdot) = \max_{a \in \mathcal{A}} \overline{Q}^k_h(\cdot, a)$.

    Set $\overline{V}^k_h(\cdot) = \underline{Q}^k_h(\cdot, \operatorname{argmax}_{a \in \mathcal{A}} \overline{Q}^k_h(\cdot, a))$

  end for

  Set π^k to be the policy such that $\pi^k_h(s) = \operatorname{argmax}_{a \in \mathcal{A}} \overline{Q}^k_h(s,a)$ for all $(h,s) \in [H] \times \mathcal{S}$.

  if $n^{\text{tot}} = N$ **then**

    Set $k^{\text{output}} = \operatorname{argmin}_k \left(\overline{V}^k_1(s_1) - \underline{V}^k_1(s_1) \right)$.

    Output $\pi^{\text{output}} = \pi^{k^{\text{output}}}$.

  end if

  Set $T_{h,i} = 0$, for all $h \in [H], i \in [m]$.

  repeat ▷ Update policy cover

    Reset to $s = s_1$, $n^k = n^k + 1$, $n^{\text{tot}} = n^{\text{tot}} + 1$.

Algorithm 23 Policy Replay for Misspecified MDP with linear function approximation (Part 2)

for $h = 1, 2, \ldots, H$ **do**
  Play $a = \pi_h^k(\cdot|s)$.
  $T_h \to T_h + \|\phi(s,a)\|^2_{[\widehat{\Sigma}_h^k]^{-1}}$.
  Get next state s', $s \to s'$.
end for
until $\exists h \in [H]$ such that $T_h \geq T_{\text{Trig}}$ or $n^{\text{tot}} = N$.
Update $\Pi \leftarrow \Pi \bigcup \{(\pi^k, n^k)\}$.
end for

We define $\pi^{k,\text{cov}}$ to be the mixture policy in $\Pi^k = \{(\pi^l, n^l)\}_{l=1}^{k-1}$, where policy π^l is given weight/probability $\frac{n^l}{\sum_{j=1}^{k-1} n^j}$. Then we define the on-policy population fit to be

$$\widetilde{\theta}_h^k := \operatorname*{argmin}_{\|\theta\| \leq W} \mathbb{E}_{(s_h,a_h)\sim\pi^{k,\text{cov}}} \left\{ \langle \phi(s_h,a_h), \theta \rangle - \mathbb{E}\left[r_h(s_h,a_h) + \overline{V}_{h+1}^k(s') \right] \right\}^2,$$

$$\widehat{\theta}_h^k := \operatorname*{argmin}_{\|\theta\| \leq W} \mathbb{E}_{(s_h,a_h)\sim\pi^{k,\text{cov}}} \left\{ \langle \phi(s_h,a_h), \theta \rangle - \mathbb{E}\left[r_h(s_h,a_h) + \underline{V}_{h+1}^k(s') \right] \right\}^2.$$

We define the misspecification error to be

$$\overline{\Delta}_h^k(s,a) := \mathbb{E}\left[r_h(s,a) + \overline{V}_{h+1}^k(s') \right] - \text{proj}_{[0,H+1-h]}\left(\left\langle \phi(s,a), \widetilde{\theta}_h^k \right\rangle \right),$$

$$\underline{\Delta}_h^k(s,a) := \mathbb{E}\left[r_h(s,a) + \underline{V}_{h+1}^k(s') \right] - \text{proj}_{[0,H+1-h]}\left(\left\langle \phi(s,a), \widehat{\theta}_h^k \right\rangle \right).$$

Lemma C.4.1. *(Concentration) With probability at least $1-\delta/2$, for all $k \in [K]$, $h \in [H]$, we have*

$$\left\| \overline{\theta}_h^k - \widetilde{\theta}_h^k \right\|_{\Sigma_h^k} \leq 8(W+H)\sqrt{\lambda + d\log(32WN(W+H)) + 4\log(8K_{\max}H/\delta)} \leq \beta/2, \quad \text{(C.7)}$$

$$\left\| \underline{\theta}_h^{k,t} - \widehat{\theta}_h^k \right\|_{\Sigma_h^k} \leq 8(W+H)\sqrt{\lambda + d\log(32WN(W+H)) + 4\log(8K_{\max}H/\delta)} \leq \beta/2, \quad \text{(C.8)}$$

$$\frac{1}{2}\widehat{\Sigma}_h^k \preceq \Sigma_h^k \preceq \frac{3}{2}\widehat{\Sigma}_h^k. \quad \text{(C.9)}$$

Proof. The proof is the same as the proof for Lemma C.2.3. □

Lemma C.4.2. *With probability at least $1-\delta/2$, the following two events hold:*

- *Suppose at episode k, Line 34: $T_h \geq T_{\text{Trig}}$ is triggered, then we have*

$$\mathbb{E}_{\pi^k} \|\phi(s_h, a_h)\|^2_{\left[\widehat{\Sigma}_h^k\right]^{-1}} \geq \frac{1}{2n^k} \sum_{j=1}^{n^k} \left\|\phi(s_h^{k,j}, a_h^{k,j})\right\|^2_{\left[\widehat{\Sigma}_h^k\right]^{-1}} \geq \frac{T_{\text{Trig}}}{2n^k},$$

  where j denotes the j-th trajectory collected in the policy cover update (Line 27).

- *For any $k \in [K_{\max}]$, $h \in [H]$, we have*

$$\mathbb{E}_{\pi^k} \|\phi(s_h, a_h)\|^2_{\left[\widehat{\Sigma}_h^k\right]^{-1}} \leq \frac{2T_{\text{Trig}}}{n^k}.$$

Proof. The proof is the same as the proof for Lemma C.2.4. □

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma C.4.1 and Lemma C.4.2 hold, which holds with probability at least $1 - \delta$ by Lemma C.4.1 and Lemma C.4.2.

Lemma C.4.3. *Under good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we have*

$$-\overline{\Delta}_h^k(s,a) \leq \overline{Q}_h^k(s,a) - \left[\mathbb{E}\left[r_h(s,a) + \overline{V}_{h+1}^k(s')\right]\right] \leq 3\beta \|\phi(s,a)\|_{[\Sigma_h^k]^{-1}} - \overline{\Delta}_h^k(s,a),$$

$$-3\beta \|\phi(s,a)\|_{[\Sigma_h^k]^{-1}} - \underline{\Delta}_h^k(s,a) \leq \underline{Q}_h^k(s,a) - \left[\mathbb{E}\left[r_h(s,a) + \underline{V}_{h+1}^k(s')\right]\right] \leq -\underline{\Delta}_h^k(s,a).$$

Proof. The proof is the same as the proof for Lemma C.2.6. □

Lemma C.4.4. *(Optimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$\overline{V}_1^k(s_1) \geq V_1^*(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\pi^*}\left[\overline{\Delta}_h^k(s_h, a_h)\right] \geq V_1^*(s_1) - \nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, we have

$$\begin{aligned}
&\overline{V}_1^k(s_1) - V_1^*(s_1) \\
=&\max_{a_1 \in \mathcal{A}} \overline{Q}_1^k(s_1, a_1) - V_1^*(s_1) \\
\geq&\overline{Q}_1^k(s_1, \pi_1^*(s_1)) - Q_1^*(s_1, \pi_1^*(s_1)) \\
\geq&\mathbb{E}\left[r_1(s_1, \pi_1^*(s_1)) + \overline{V}_2^k(s_2)\right] - \overline{\Delta}_1^k(s_1, \pi_1^*(s_1)) - Q_1^*(s_1, \pi_1^*(s_1)) && \text{(Lemma C.4.3)} \\
=&\mathbb{E}_{\pi^*}\left[\overline{V}_2^k(s_2) - V_2^*(s_2)\right] - \overline{\Delta}_1^k(s_1, \pi_1^*(s_1)) \\
\geq& -\mathbb{E}_{\pi^*}\left[\sum_{h=1}^{H} \overline{\Delta}_h^k(s_h, a_h)\right]
\end{aligned}$$

$$\geq -\nu. \qquad \text{(Lemma C.2.5)}$$

□

Lemma C.4.5. *(Pessimism) Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$\underline{V}_1^k(s_1) \leq V_1^{\pi^k}(s_1) - \sum_{h=1}^{H} \mathbb{E}_{\pi^k}\left[\underline{\Delta}_h^k(s_h, a_h)\right] \leq V_1^{\pi^k}(s_1) + \nu.$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, we have

$$\begin{aligned}
&\underline{V}_1^k(s_1) - V_1^{\pi^k}(s_1) \\
=&\underline{Q}_1^k(s_1, \pi_1^k(s_1)) - V_1^{\pi^k}(s_1) \\
\leq&\mathbb{E}_{a_1=\pi_1^k(s_1)}\left[r_1(s_1, a_1) + \underline{V}_2^k(s_2) - \underline{\Delta}_1^k(s_1, a_1)\right] - V_1^{\pi^k}(s_1) && \text{(Lemma C.4.3)} \\
=&\mathbb{E}_{a_1=\pi_1^k(s_1)}\left[\underline{V}_2^k(s_2) - V_2^{\pi^k}(s_2) - \underline{\Delta}_1^k(s_1, a_1)\right] \\
\leq& -\mathbb{E}_{\pi^k}\left[\sum_{h=1}^{H} \underline{\Delta}_h^k(s_h, a_h)\right] \\
\leq&\nu. && \text{(Lemma C.2.5)}
\end{aligned}$$

□

Lemma C.4.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, we have*

$$V_1^*(s_1) - V_1^{\pi^k}(s_1) - 2\nu \leq \overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \leq 6\beta\mathbb{E}_{\pi^k}\sum_{h=1}^{H} \|\phi(s_h, a_h)\|_{[\Sigma_h^k]^{-1}} + 2\nu.$$

Proof. The proof is the same as the proof for Lemma C.2.9. □

Lemma C.4.7. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \|\phi(s_h, a_h)\|^2_{[\Sigma_h^k]^{-1}} \leq 4T_{\text{Trig}} d \log\left(1 + \frac{N}{d\lambda}\right).$$

Proof. The proof is the same as the proof for Lemma C.2.10. □

Lemma C.4.8. *Under the good event $\mathcal{G}$, we have*

$$\sum_{k=1}^{K} n^k \left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) \leq 6H\beta\sqrt{4N(T_{\text{Trig}} + 1)d \log\left(1 + \frac{N}{\lambda}\right)} + 2\nu N.$$

Proof. The proof is the same as the proof for Lemma C.2.11. □

Lemma C.4.9. *Under the good event $\mathcal{G}$, we have*

$$K \leq \frac{2Hd\log(N+\lambda)}{\log(1+T_{\text{Trig}}/4)},$$

which means $K < K_{\max}$ and Algorithm 22 ends due to $n^{\text{tot}} = N_{\max}$.

Proof. The proof is the same as the proof for Lemma C.2.12. □

Theorem C.4.10. *For ν-misspecified linear MDP, with probability at least $1-\delta$, Algorithm 22 will output an $(\epsilon + 4\nu)$-approximate optimal policy. The sample complexity is $O(HK_{\max}N) = \widetilde{O}(H^6d^4\epsilon^{-2})$.*

Proof. Under the good event $\mathcal{G}$, by Lemma C.4.9, the algorithm ends by $n^{\text{tot}} = N$. By Lemma C.4.8, we have

$$\min_{k\in[K]}\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) \leq \frac{1}{N}\sum_{k=1}^{K} n^k\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) \leq 6H\beta\sqrt{4(T_{\text{Trig}}+1)d\log\left(1+\frac{N}{\lambda}\right)/N} + 2\nu.$$

By setting $N = \widetilde{O}(H^4d^3\epsilon^{-2})$, we have

$$\min_{k\in[K]} \overline{V}_1^k(s_1) - \underline{V}_1^k(s_1) \leq \epsilon + 2\nu.$$

Then by Lemma C.4.6, we have

$$V_1^*(s_1) - V_1^{\pi^{\text{output}}}(s_1) \leq \overline{V}_1^{k^{\text{output}}}(s_1) - \underline{V}_1^{k^{\text{output}}}(s_1) + 2\nu = \min_{k\in[K]}\left(\overline{V}_1^k(s_1) - \underline{V}_1^k(s_1)\right) + 2\nu \leq \epsilon + 4\nu.$$

□

C.5 Proofs for Learning in Markov Potential Games

C.5.1 Proofs for Learning Markov NE with Algorithm 5

We will set the parameter for Algorithm 5 to be

- $K = 5mH\epsilon^{-1}$

Lemma C.5.1. *With probability at least $1-\delta/2$, for all $k \in [K]$ and $i \in [m]$, $\widehat{\pi}_i^{k+1}$ is an $(\epsilon/8 + O(\nu))$-approximate optimal policy in the ν-misspecified linear MDP induced by all the players except player i following policy π_{-i}^k.*

Proof. The argument follows from the property of LinearMDP_Solver (Assumption 4.5.2) and a union bound. □

Lemma C.5.2. *Suppose for all $k \in [K]$ and $i \in [m]$, we execute policy π^k and $(\widehat{\pi}_i^{k+1}, \pi_{-i}^k)$ for $\widetilde{O}(H^2\epsilon^{-2})$ episodes, With probability at least $1-\delta/2$, for all $k \in [K]$ and $i \in [m]$, we have*

$$\left|\widehat{V}_{1,i}^{\pi^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right| \le \frac{\epsilon}{8}, \qquad \left|\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^t}(s_1) - V_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1)\right| \le \frac{\epsilon}{8}.$$

Proof. The argument follows directly by Hoeffding's inequality and a union bound. □

We will denote the event in Lemma C.5.1 and Lemma C.5.2 to be the good event $\mathcal{G}$.

Lemma C.5.3. *Under the good event $\mathcal{G}$, for any $k \in [K]$, if $\max_{i\in[m]} \Delta_i^k > \epsilon/2$ and $j = \operatorname{argmax}_{i\in[m]} \Delta_i^k$, we have*

$$V_{1,j}^{\pi^{k+1}}(s_1) - V_{1,j}^{\pi^k}(s_1) \ge \epsilon/4.$$

And if $\max_{i\in[m]} \Delta_i^k \le \epsilon/2$, we have

$$\max_{i\in[m]} \left(V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1)\right) \le \epsilon.$$

Proof. Under the good event $\mathcal{G}$, if $\max_{i\in[m]} \Delta_i^k > \epsilon/2$ and $j = \operatorname{argmax}_{i\in[m]} \Delta_i^k$, we have

$$\begin{aligned} V_{1,j}^{\pi^{k+1}}(s_1) - V_{1,j}^{\pi^k}(s_1) \ge& \widehat{V}_{1,j}^{\widehat{\pi}_j^{k+1},\pi_{-i}^k}(s_1) - \epsilon/8 - \widehat{V}_{1,j}^{\pi^k}(s_1) - \epsilon/8 && \text{(Lemma C.5.2)} \\ \ge& \epsilon/4. \end{aligned}$$

On the other hand, if $\max_{i\in[m]} \Delta_i^k = \max_{i\in[m]} \left(\widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) - \widehat{V}_{1,i}^{\pi^k}(s_1)\right) \le \epsilon/2$, for all $i \in [m]$ we have

$$\begin{aligned} V_{1,i}^{\dagger,\pi_{-i}^k}(s_1) - V_{1,i}^{\pi^k}(s_1) \le& V_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^t}(s_1) + \frac{\epsilon}{8} + O(\nu) - V_{1,i}^{\pi^k}(s_1) \\ \le& \widehat{V}_{1,i}^{\widehat{\pi}_i^{k+1},\pi_{-i}^k}(s_1) + \frac{\epsilon}{8} + O(\nu) + \epsilon/8 - \widehat{V}_{1,i}^{\pi^k}(s_1) + \epsilon/8 \\ & && \text{(Lemma C.5.1 and Lemma C.5.2)} \\ \le& \epsilon + O(\nu), \end{aligned}$$

completing the proof. □

Theorem 4.5.3. *For ν-misspecified independent linear Markov potential games with $\Pi^{\text{estimate}} = \{\pi^k\}_{k=1}^K$, with probability at least $1-\delta$, Algorithm 5 will output an $(\epsilon + O(\nu))$-approximate pure Markov NE. The sample complexity is*

$$O(m^2 H \epsilon^{-1} \cdot \text{LinearMDP\_SC}(\epsilon/8, \delta/(10m^2 H\epsilon^{-1}), d_{\max})).$$

Proof. Suppose Algorithm 5 does not output a policy, then it ends due to $k = K$. Then under the good event $\mathcal{G}$, by the first argument of Lemma C.5.3, for all $k \in [K]$, and $j^k = \text{argmax}_{i\in[m]} \Delta_i^k$, we have

$$\Phi(\pi^{k+1}) - \Phi(\pi^k) = V_{1,j^t}^{\pi^{k+1}}(s_1) - V_{1,j^k}^{\pi^k}(s_1) \geq \epsilon/4.$$

As we set $K = 5mH/\epsilon$, we have $\Phi(\pi^{K+1}) > mH \geq \Phi_{\max}$, which is a contradiction. So Algorithm 5 will output a policy π^{output}. As the LinearMDP_Solver always outputs a deterministic policy, π^{output} is a deterministic policy. Then by the second argument of Lemma C.5.3, when Algorithm 5 terminates, it will output an ϵ-approximate pure NE π^{output}. □

C.6 Proofs for Section 4.6

We will set the parameters for Algorithm 6 to be

- $T_{\text{Trig}} = 12 \log(8K_{\max}HS/\delta)$
- $K_{\max} = 9HS\log(N_{\max})$
- $N_{\max} = \widetilde{O}(H^4 S A_{\max}\epsilon^{-2})$ for Markov CCE and $N_{\max} = \widetilde{O}(H^4 S A_{\max}^2 \epsilon^{-2})$ for Markov CE
- $\beta_n = \sqrt{\frac{8H^2 T_{\text{Trig}} \log(2mK_{\max}HS/\delta)}{n \vee T_{\text{Trig}}}}$.

We will use subscript k, t to denote the variables in episode k and inner loop t, and subscript h, i to denote the variables at step h and for player i. We will use K to denote the episode that the Algorithm 6 ends (Line 30 is triggered or $n^{\text{tot}} = N_{\max}$ or $K = K_{\max}$) and N to denote n^{tot} when Algorithm 6 ends. Immediately we have $K \leq K_{\max} \leq N_{\max}$.

By the definition of the adversarial multi-armed bandit oracles (Assumption 4.6.1 and Assumption 4.6.2), we have the following two lemmas.

Lemma C.6.1. *For all $k \in [K]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\begin{aligned}
&\frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \\
\geq &\max_{a_i \in \mathcal{A}_i} \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a}_{-i} \sim \pi_{h,-i}^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) - \frac{n_h^k(s)}{H} \cdot \mathrm{BReg}(n_h^k(s)).
\end{aligned}$$

Lemma C.6.2. *For all $k \in [K]$, $h \in [H]$, $i \in [m]$ and $s \in \mathcal{S}$ we have*

$$\begin{aligned}
&\frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \\
\geq &\max_{\psi_{h,i}} \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \psi_{h,i} \diamond \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) - \frac{n_h^k(s)}{H} \cdot \mathrm{BSwapReg}(n_h^k(s)).
\end{aligned}$$

C.6.1 Concentration

Lemma C.6.3. *With probability at least $1 - \delta/2$, for all $k \in [K]$, $h \in [H]$, $i \in [m], s \in \mathcal{S}$, we have*

$$\left| \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} (r_{h,i}^{k, t_h^k(j;s)} + \overline{V}_{h+1,i}^k(s_{h+1}^{k, t_h^k(j;s)})) - \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \overline{V}_{h+1,i}^k(s')) \right| \leq \beta_{n_h^k(s)},$$

$$\left| \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} (r_{h,i}^{k, t_h^k(j;s)} + \underline{V}_{h+1,i}^k(s_{h+1}^{k, t_h^k(j;s)})) - \frac{1}{n_h^k(s)} \sum_{j=1}^{n_h^k(s)} \mathbb{E}_{\mathbf{a} \sim \pi_h^{k, t_h^k(j;s)}(\cdot|s)} (r_{h,i}(s, \mathbf{a}) + \underline{V}_{h+1,i}^k(s')) \right| \leq \beta_{n_h^k(s)},$$

where

$$\beta_{n_h^k(s)} = \sqrt{\frac{8H^2 T_{\mathrm{Trig}} \log(2mK_{\max}HS/\delta)}{n_h^k(s) \vee T_{\mathrm{Trig}}}}.$$

Proof. If $n_h^k(s) \leq T_{\mathrm{Trig}}$, we have $\beta_{n_h^k(s)} \geq H$ and the arguments hold directly. If $n_h^k(s) \geq T_{\mathrm{Trig}}$, we have

$$\beta_{n_h^k(s)} = \sqrt{\frac{8H^2 T_{\mathrm{Trig}} \log(2mK_{\max}HS/\delta)}{n_h^k(s) \vee T_{\mathrm{Trig}}}} \geq \sqrt{\frac{8H^2 \log(2mK_{\max}HS/\delta)}{n_h^k(s)}},$$

and by Hoeffding's inequality and union bound, we can prove that the arguments hold with probability at least $1 - \delta/2$. □

Lemma C.6.4. *With probability at least $1-\delta/2$, for all $k \in [K_{\max}]$, $h \in [H]$, $i \in [m], s \in \mathcal{S}$, we have*

$$n_h^k(s) \vee T_{\text{Trig}} \geq \frac{1}{2}\left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s)\right) \vee T_{\text{Trig}}, n^k d_h^{\pi^k}(s) \leq 2\left(n_h^k(s) \vee T_{\text{Trig}}\right).$$

In addition, if $T_h^k(s) = n_h^k(s) \vee T_{\text{Trig}}$ is triggered, we have

$$n^k d_h^{\pi^k}(s) \geq \frac{1}{2}\left(n_h^k(s) \vee T_{\text{Trig}}\right).$$

Proof. $n_h^k(s)$ is the sum of $\sum_{l=1}^{k-1} n^l$ independent Bernoulli random variables such that there are n^l random variables with mean $d_h^{\pi^l}(s)$ for $l \in [k-1]$. By Lemma C.7.3 and union bound, with probability at least $1 - \delta/4$, for all $k \in [K_{\max}]$, $h \in [H]$, $s \in \mathcal{S}$, we have

$$n_h^k(s) \vee T_{\text{Trig}} \geq \frac{1}{2}\left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s)\right) \vee T_{\text{Trig}},$$

where $T_{\text{Trig}} \geq 12\log(8K_{\max}HS/\delta)$.

$T_h^k(s)$ is the sum of n^k i.i.d. Bernoulli random variables with mean n_h^k. For the second argument, by Lemma C.7.2 and union bound, with probability at least $1 - \delta/4$, for all $k \in [K_{\max}]$, $h \in [H]$, $s \in \mathcal{S}$, we have

$$n^k d_h^{\pi^k}(s) \leq 2(n_h^k(s) \vee T_{\text{Trig}}),$$

and if $T_h^k(s) = n_h^k(s) \vee T_{\text{Trig}}$ is triggered, we have

$$n^k d_h^{\pi^k}(s) \geq \frac{1}{2}T_h^k(s) = \frac{1}{2}\left(n_h^k(s) \vee T_{\text{Trig}}\right).$$

□

We denote $\mathcal{G}$ to be the good event where the arguments in Lemma C.6.3 and Lemma C.6.4 hold, which holds with probability at least $1 - \delta$.

C.6.2 Proofs for Learning Markov CCE with Algorithm 6

Lemma C.6.5. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\overline{V}_{h,i}^k(s) \geq V_{h,i}^{\dagger,\pi_{-i}^k}(s).$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have

$$
\begin{aligned}
\overline{V}^k_{h,i}(s) &= \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}(r_{h,i}^{k,t_h^k(j;s)} + \overline{V}^k_{h+1,i}(s_{h+1}^{k,t_h^k(j;s)})) + \frac{H}{n_h^k(s)}\cdot \mathrm{BReg}(n_h^k(s)) + \beta_{n_h^k(s)}\right)\\
&\geq \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s')) + \frac{H}{n_h^k(s)}\cdot \mathrm{BReg}(n_h^k(s))\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Lemma C.6.3)}\\
&\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi_{h,-i}^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + \overline{V}^k_{h+1,i}(s'))\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Lemma C.6.1)}\\
&\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi_{h,-i}^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + V^{\dagger,\pi^k_{-i}}_{h+1,i}(s'))\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Induction basis)}\\
&= \mathrm{proj}_{[0,H+1-h]}\left(\max_{a_i\in\mathcal{A}_i}\mathbb{E}_{\mathbf{a}_{-i}\sim\pi^k_{h,-i}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + V^{\dagger,\pi^k_{-i}}_{h+1,i}(s'))\right)\\
&\geq V^{\dagger,\pi^k_{-i}}_{h,i}(s).
\end{aligned}
$$

□

Lemma C.6.6. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\underline{V}^k_{h,i}(s) \leq V^{\pi^k}_{h,i}(s).$$

Proof. Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have

$$
\begin{aligned}
\underline{V}^k_{h,i}(s) &= \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}(r_{h,i}^{k,t_h^k(j;s)} + \underline{V}^k_{h+1,i}(s_{h+1}^{k,t_h^k(j;s)})) - \beta_{n_h^k(s)}\right)\\
&\leq \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + \underline{V}^k_{h+1,i}(s'))\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Lemma C.6.3)}\\
&\leq \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + V^{\pi^k}_{h+1,i}(s'))\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Induction basis)}
\end{aligned}
$$

$$
\begin{aligned}
&= \mathrm{proj}_{[0,H+1-h]}\left(\mathbb{E}_{\mathbf{a}\sim\pi^k_{h,-i}(\cdot|s)}(r_{h,i}(s,\mathbf{a}) + V^{\dagger,\pi^k_{-i}}_{h+1,i}(s'))\right) \\
&\le V^{\pi^k}_{h,i}(s).
\end{aligned}
$$

□

Lemma C.6.7. *Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$
\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le \widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^{H}\sqrt{\frac{H^2 A_i T_{\mathrm{Trig}}}{n^k_h(s_h)\vee T_{\mathrm{Trig}}}}\right]\right).
$$

Proof. We bound $\overline{V}^k_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1)$ and $V^{\pi^k}_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)$ separately.

$$
\begin{aligned}
&\overline{V}^k_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1) \\
&= \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n^k_1(s_1)}\sum_{j=1}^{n^k_1(s_1)}(r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s_2^{k,t^k_h(j;s_1)})) + \frac{H}{n^k_1(s_1)}\cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)}\right) - V^{\pi^k}_{1,i}(s_1) \\
&\le \frac{1}{n^k_1(s_1)}\sum_{t=1}^{n^k_1(s)}(r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s_2^{k,t^k_h(j;s_1)})) + \frac{H}{n^k_1(s_1)}\cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)} - V^{\pi^k}_{1,i}(s_1) \\
&\le \frac{1}{n^k_1(s_1)}\sum_{t=1}^{n^k_1(s_1)}\mathbb{E}_{\mathbf{a}_1\sim\pi_1^{k,t^k_h(j;s_1)}(\cdot|s)}(r_{1,i}(s_1,\mathbf{a}_1) + \overline{V}^k_{2,i}(s_2)) + \frac{H}{n^k_1(s_1)}\cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} - V^{\pi^k}_{1,i}(s_1) \\
&\hfill \text{(Lemma C.6.3)} \\
&= \mathbb{E}_{\mathbf{a}_1\sim\pi^k_1(\cdot|s)}(r_{1,i}(s_1,\mathbf{a}_1) + \overline{V}^k_{2,i}(s_2)) + \frac{H}{n^k_1(s_1)}\cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} - V^{\pi^k}_{1,i}(s_1) \\
&= \mathbb{E}_{\pi^k_1}\left[\overline{V}^k_{2,i}(s_2) - V^{\pi^k}_{2,i}(s_2)\right] + \frac{H}{n^k_1(s_1)}\cdot \mathrm{BReg}(n^k_1(s_1)) + 2\beta_{n^k_1(s_1)} \\
&= \mathbb{E}_{\pi^k}\left[\sum_{h=1}^{H}\frac{H}{n^k_h(s_h)}\cdot \mathrm{BReg}(n^k_h(s_h)) + 2\beta_{n^k_h(s_h)}\right],
\end{aligned}
$$

where the first inequality is from

$$
\frac{1}{n^k_1(s_1)}\sum_{t=1}^{n^k_1(s)}(r^{k,t^k_h(j;s_1)}_{1,i} + \overline{V}^k_{2,i}(s_2^{k,t^k_h(j;s_1)})) + \frac{H}{T}\cdot \mathrm{BReg}(n^k_1(s_1)) + \beta_{n^k_1(s_1)} \ge 0.
$$

In addition, we have

$$
\begin{aligned}
&V^{\pi^k}_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \\
&= V^{\pi^k}_{1,i}(s_1) - \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n^k_1(s)}\sum_{j=1}^{n^k_1(s_1)}(r^{k,t^k_1(j;s)}_{1,i} + \underline{V}^k_{2,i}(s_2^{k,t^k_1(j;s)})) - \beta_{n^k_1(s_1)}\right)
\end{aligned}
$$

$$
\begin{aligned}
\leq & V_{1,i}^{\pi^k}(s_1) - \frac{1}{n_1^k(s_1)} \sum_{j=1}^{n_1^k(s_1)} (r_{1,i}^{k,t_1^k(j;s)} + \underline{V}_{2,i}^k(s_2^{k,t_1^k(j;s)})) + \beta_{n_1^k(s_1)} \\
\leq & V_{1,i}^{\pi^k}(s_1) - \frac{1}{n_1^k(s_1)} \sum_{j=1}^{n_1^k(s_1)} \left(\mathbb{E}_{\mathbf{a}_1 \sim \pi_1^{k,t_h^k(j;s_1)(\cdot|s_1)}} (r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}_{2,i}^k(s_2)) \right) + 2\beta_{n_1^k(s_1)} \\
& \qquad \text{(Lemma C.6.3)} \\
= & V_{1,i}^{\pi^k}(s_1) - \mathbb{E}_{\mathbf{a}_1 \sim \pi_1^k(\cdot|s_1)} (r_{1,i}(s_1, \mathbf{a}_1) + \underline{V}_{2,i}^k(s_2)) + 2\beta_{n_1^k(s_1)} \\
= & \mathbb{E}_{\mathbf{a}_1 \sim \pi_1^k} (V_{2,i}^{\pi^k}(s_2) - \underline{V}_{2,i}^k(s_2)) + 2\beta_{n_1^k(s_1)} \\
\leq & \mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} 2\beta_{n_h^k(s_h)} \right],
\end{aligned}
$$

where the first inequality is from

$$
\frac{1}{n_1^k(s)} \sum_{j=1}^{n_1^k(s_1)} (r_{1,i}^{k,t_1^k(j;s)} + \underline{V}_{2,i}^k(s_2^{k,t_1^k(j;s)})) - \beta_{n_1^k(s_1)} \leq H + 1 - h.
$$

Then we have

$$
\begin{aligned}
\overline{V}_{1,i}^{\pi^k}(s_1) - \underline{V}_{1,i}^k(s_1) \leq & \mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} \frac{H}{n_h^k(s_h)} \cdot \mathrm{BReg}(n_h^k(s_h)) + 4\beta_{n_h^k(s_h)} \right] \\
\leq & \widetilde{O} \left(\mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} \sqrt{\frac{H^2 A_i}{n_h^k(s_h) \vee 1}} + \sqrt{\frac{H^2 T_{\mathrm{Trig}}}{n_h^k(s_h) \vee T_{\mathrm{Trig}}}} \right] \right) \\
\leq & \widetilde{O} \left(\mathbb{E}_{\pi^k} \left[\sum_{h=1}^{H} \sqrt{\frac{H^2 A_i T_{\mathrm{Trig}}}{n_h^k(s_h) \vee T_{\mathrm{Trig}}}} \right] \right).
\end{aligned}
$$

□

Lemma C.6.8. *Under the good event $\mathcal{G}$, for all $i \in [m]$, we have*

$$
\sum_{k=1}^{K} n^k \max_{i \in [m]} \left(\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^{\pi^k}(s_1) \right) \leq \widetilde{O} \left(H^2 \sqrt{S A_{\max} T_{\mathrm{Trig}} N} \right).
$$

Proof. Under the good event $\mathcal{G}$, for all $i \in [m]$, we have

$$
\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k} \sqrt{\frac{1}{n_h^k(s_h) \vee T_{\mathrm{Trig}}}}
$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} n^k \sum_{s\in\mathcal{S}} d_h^{\pi^k}(s) \sqrt{\frac{1}{n_h^k(s) \vee T_{\text{Trig}}}} \\
&\le \sum_{s\in\mathcal{S}} \sum_{k=1}^{K} n^k d_h^{\pi^k}(s) \sqrt{\frac{2}{(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s)) \vee T_{\text{Trig}}}} && \text{(Lemma C.6.4)} \\
&\le \sum_{s\in\mathcal{S}} \sqrt{32 \sum_{k=1}^{K} n^k d_h^{\pi^k}(s)} && \text{(Lemma C.6.4 and Lemma C.7.7)} \\
&\le \sqrt{32S \sum_{k=1}^{K} n^k}. && (\textstyle\sum_{s\in\mathcal{S}} \sum_{k=1}^{K} n^k d_h^{\pi^k}(s) = S \sum_{k=1}^{K} n^k)
\end{aligned}
$$

Plugging it into Lemma C.6.7, we can prove the lemma. □

Lemma C.6.9. *Under the good event $\mathcal{G}$, we have*

$$K \le 9HS \log(N_{\max}),$$

which means $K < K_{\max}$ and Algorithm 6 ends due to either Line 21 ($\max_{i\in[m]} \overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \le \epsilon$) or Line 34 ($n^{\text{tot}} = N_{\max}$).

Proof. By Lemma C.6.4, for any $h \in [H]$ and $s \in \mathcal{S}$, whenever $T_h^k(s) = n_h^k(s) \vee T_{\text{Trig}}$ is triggered, we have

$$n^k d_h^{\pi^k}(s) \ge \frac{1}{2}(n_h^k(s) \vee T_{\text{Trig}}) \ge \frac{1}{4} \left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s) \right).$$

Thus for any $h \in [H]$ and $s \in \mathcal{S}$, whenever $T_h^k(s) = n_h^k(s) \vee T_{\text{Trig}}$ is triggered, we have

$$\sum_{l=1}^{k} n^l d_h^{\pi^l}(s) \ge \frac{5}{4} \left(\sum_{l=1}^{k-1} n^l d_h^{\pi^l}(s) \right).$$

In addition, for any $h \in [H]$ and $s \in \mathcal{S}$, for the first time $T_h^k(s) = n_h^k(s) \vee T_{\text{Trig}}$ is triggered, we have

$$\sum_{l=1}^{k} n^l d_h^{\pi^l}(s) \ge n^k d_h^{\pi^k}(s) \ge \frac{1}{2}(n_h^k(s) \vee T_{\text{Trig}}) \ge T_{\text{Trig}}.$$

As $\sum_{l=1}^{k} n^l d_h^{\pi^l}(s)$ is non-decreasing and upper bounded by $N_{\max}$, the number of triggering for any $h \in [H]$ and $s \in \mathcal{S}$ is bounded by $\log(N_{\max}/T_{\text{Trig}})/\log(5/4) \le 8\log(N_{\max})$, and the total number of triggering is bounded by $8HS\log(N_{\max}) + 1$, where 1 is from the last triggering $n^{\text{tot}} = N_{\max}$. □

Theorem 4.6.3. *Suppose Algorithm 6 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 4.6.1. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 6 will output an ϵ-approximate Markov CCE. The sample complexity is $\widetilde{O}(HK_{\max}N_{\max}) = \widetilde{O}(H^6S^2A_{\max}\epsilon^{-2})$.*

Proof. Suppose under the good event $\mathcal{G}$, the algorithm does not end with Line 21 ($\max_{i\in[m]} \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le \epsilon$). Then by Lemma C.6.9, the algorithm ends by $N = N_{\max}$. By Lemma C.6.8, under the good event $\mathcal{G}$, we have

$$\begin{aligned}\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le& \frac{1}{N}\sum_{k=1}^{K} n^k \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right)\\ \le& \widetilde{O}\left(H^2\sqrt{SA_{\max}T_{\mathrm{Trig}}/N_{\max}}\right).\end{aligned}$$

Let $N_{\max} = \widetilde{O}(H^4SA_{\max}\epsilon^{-2})$ we can have

$$\min_{k\in[K]}\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon,$$

which contradicts with Line 21. Thus Algorithm 6 will end at episode k such that

$$\max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon.$$

By Lemma C.6.5 and Lemma C.6.6, we have

$$\max_{i\in[m]}\left(V^{\dagger,\pi^k_{-i}}_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1)\right) \le \max_{i\in[m]}\left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon,$$

completing the proof. □

C.6.3 Proofs for Learning Markov CE with Algorithm 6

Lemma C.6.10. *Under the good event $\mathcal{G}$, for all $k\in[K]$, $h\in[H]$, $i\in[m]$, $s\in\mathcal{S}$, we have*

$$\overline{V}^k_{h,i}(s) \ge \max_{\psi_i} V^{\psi_i\diamond\pi^k}_{h,i}(s).$$

Proof. We prove the lemma by mathematical induction on h. The argument holds for $h = H+1$ as both sides are 0. Suppose the argument holds for $h+1$. By the update rule of $\overline{V}^k_{h,i}(s)$, we have

$$\overline{V}^k_{h,i}(s) = \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n^k_h(s)}\sum_{j=1}^{n^k_h(s)}(r^{k,t^k_h(j;s)}_{h,i} + \overline{V}^k_{h+1,i}(s^{k,t^k_h(j;s)}_{h+1})) + \frac{H}{n^k_h(s)}\mathrm{BSwapReg}(n^k_h(s)) + \beta_{n^k_h(s)}\right)$$

$$\geq \mathrm{proj}_{[0,H+1-h]}\left(\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\overline{V}_{h+1,i}^k(s'))+\frac{H}{n_h^k(s)}\mathrm{BSwapReg}(n_h^k(s))\right)$$

(Lemma C.6.3)

$$\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}}\frac{1}{n_h^k(s)}\sum_{j=1}^{n_h^k(s)}\mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^{k,t_h^k(j;s)}(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\overline{V}_{h+1,i}^k(s'))\right)$$

(Lemma C.6.2)

$$= \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}}\mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^k(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\overline{V}_{h+1,i}^k(s'))\right)$$

$$\geq \mathrm{proj}_{[0,H+1-h]}\left(\max_{\psi_{h,i}}\mathbb{E}_{\mathbf{a}\sim\psi_{h,i}\diamond\pi_h^k(\cdot|s)}(r_{h,i}(s,\mathbf{a})+\max_{\psi_i}V_{h+1,i}^{\psi_i\diamond\pi^k}(s'))\right)$$

(Induction basis)

$$\geq \max_{\psi_i} V_{h,i}^{\psi_i\diamond\pi^k}(s).$$

□

Lemma C.6.11. *Under the good event $\mathcal{G}$, for all $k\in[K]$, $i\in[m]$, we have*

$$\overline{V}_{1,i}^k(s_1)-V_{1,i}^{\pi^k}(s_1)\leq\widetilde{O}\left(\mathbb{E}_{\pi^k}\left[\sum_{h=1}^H\sqrt{\frac{H^2A_i^2T_{\mathrm{Trig}}}{n_h^k(s_h)\vee T_{\mathrm{Trig}}}}\right]\right).$$

Proof. The proof is the same as the proof of Lemma C.6.7 and we replace BReg with BSwapReg. □

Lemma C.6.12. *Under the good event $\mathcal{G}$, for all $k\in[K]$, $h\in[H]$, $i\in[m]$, $s\in\mathcal{S}$, we have*

$$\sum_{k=1}^K n^k\max_{i\in[m]}\left(\overline{V}_{1,i}^k(s_1)-\underline{V}_{1,i}^{\pi^k}(s_1)\right)\leq\widetilde{O}\left(H^2\sqrt{SA_{\max}^2T_{\mathrm{Trig}}N}\right).$$

Proof. The proof is the same as the proof of Lemma C.6.8 and we replace Lemma C.6.7 with Lemma C.6.11 in the proof. □

Theorem 4.6.4. *Suppose Algorithm 6 is instantiated with adversarial multi-armed bandit oracles satisfying Assumption 4.6.2. Then for tabular Markov games, with probability at least $1-\delta$, Algorithm 6 will output an ϵ-approximate Markov CE. The sample complexity is $\widetilde{O}(HK_{\max}N_{\max})=\widetilde{O}(H^6S^2A_{\max}^2\epsilon^{-2})$.*

Proof. Suppose under the good event $\mathcal{G}$, the algorithm does not end with Line 21 ($\max_{i\in[m]} \overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1) \le \epsilon$). Then by Lemma C.6.9, the algorithm ends by $N = N_{\max}$. By Lemma C.6.12, under the good event $\mathcal{G}$, we have

$$\begin{aligned}\min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le& \frac{1}{N}\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \\ \le& \widetilde{O}\left(H^2\sqrt{SA^2_{\max}T_{\mathrm{Trig}}/N_{\max}}\right).\end{aligned}$$

Let $N_{\max} = \widetilde{O}(H^4SA^2_{\max}\epsilon^{-2})$ we can have

$$\min_{k\in[K]} \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon,$$

which contradicts with Line 21. Thus Algorithm 6 will end at episode k such that

$$\max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon.$$

By Lemma C.6.10 and Lemma C.6.6, we have

$$\max_{i\in[m]} \left(\max_{\psi_i} V^{\psi_i\diamond\pi^k}_{1,i}(s_1) - V^{\pi^k}_{1,i}(s_1)\right) \le \max_{i\in[m]} \left(\overline{V}^k_{1,i}(s_1) - \underline{V}^k_{1,i}(s_1)\right) \le \epsilon.$$

□

C.7 Technical Tools

Lemma C.7.1. *(Theorem 4 in Maurer and Pontil [2009]) For $n \ge 2$, let $X_1, \cdots, X_n$ be i.i.d. random variables with values in $[0,1]$ and let $\delta > 0$. Define $\widehat{X} = \frac{1}{n}\sum_{i=1}^n X_i$ and $\widehat{\sigma} = \frac{1}{n-1}\sum_{i=1}^n (X_i - \widehat{X})$. Then we have*

$$\mathbb{P}\left[\left|\widehat{X} - \mathbb{E}[X]\right| > \sqrt{\frac{2\widehat{\sigma}\log(4/\delta)}{n}} + \frac{7\log(4/\delta)}{3(n-1)}\right] \le \delta.$$

Lemma C.7.2. *Consider i.i.d. random variables $X_1, X_2, \ldots$ with support in $[0,1]$ and $\widehat{S}_n = \frac{1}{n}\sum_{i=1}^n X_i$. Suppose $\overline{n} = \min_n\{n : \sum_{i=1}^n X_i \ge T_{\mathrm{Trig}}\}$ with $T_{\mathrm{Trig}} \ge 64\log(4n_{\max}/\delta)$. Then if $\overline{n} \le n_{\max}$, with probability at least $1-\delta$, we have*

$$\frac{1}{2}\widehat{S}_{\overline{n}} \le \mathbb{E}[X] \le \frac{3}{2}\widehat{S}_{\overline{n}},$$

and in addition, for $n \le \min\{\overline{n}, n_{\max}\}$, we have

$$\mathbb{E}[X] \le \frac{2T_{\mathrm{Trig}}}{n}.$$

Proof. Define the empirical variance to be

$$\widehat{\sigma}_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \widehat{S}_n)^2.$$

By Lemma C.7.1, we have that for any fixed $n \geq 2$,

$$\mathbb{P}\left[\left|\widehat{S}_n - \mathbb{E}[X]\right| \leq \sqrt{\frac{2\log(4n_{\max}/\delta)\widehat{\sigma}_n}{n}} + \frac{7\log(4n_{\max}/\delta)}{3(n-1)}\right] \geq 1 - \frac{\delta}{n_{\max}}.$$

Thus we have

$$\mathbb{P}\left[\left|\widehat{S}_n - \mathbb{E}[X]\right| \leq \sqrt{\frac{2\log(4n_{\max}/\delta)\widehat{\sigma}_n}{n}} + \frac{7\log(4n_{\max}/\delta)}{3(n-1)}, \forall 2 \leq n \leq n_{\max}\right] \geq 1 - \sum_{n=2}^{n_{\max}} \frac{\delta}{n_{\max}} \geq 1 - \delta. \tag{C.10}$$

The empirical variance can be bounded by

$$\widehat{\sigma}_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \widehat{S}_n)^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^2 - n\widehat{X}^2\right) \leq \frac{1}{n-1}\sum_{i=1}^{n} X_i \leq 2\widehat{S}_n.$$

Thus for $T_{\text{Trig}} \geq 64\log(4n_{\max}/\delta)$, we have $\overline{n}\widehat{S}_{\overline{n}} \geq T_{\text{Trig}} \geq 64\log(4n_{\max}/\delta)$ and

$$\sqrt{\frac{2\log(4\overline{n}/\delta)\widehat{\sigma}_{\overline{n}}}{\overline{n}}} + \frac{7\log(4\overline{n}/\delta)}{3(\overline{n}-1)} \leq \sqrt{\frac{4\log(4\overline{n}/\delta)\widehat{S}_{\overline{n}}}{\overline{n}}} + \frac{7\log(4\overline{n}/\delta)}{3(\overline{n}-1)} \leq \frac{\widehat{S}_{\overline{n}}}{2}.$$

Plugging it into (C.10), we can prove the first argument.

For $n \leq \min\{\overline{n}, N\}$, we have $\sum_{i=1}^{n} X_i \leq T_{\text{Trig}} + 1 \leq 2T_{\text{Trig}}$, which means

$$\widehat{\sigma}_n \leq 2\widehat{S}_n \leq \frac{4T_{\text{Trig}}}{n}.$$

Plugging it into (C.10), and with $T_{\text{Trig}} \geq 64\log(4n_{\max}/\delta)$, we can prove the second argument. □

Lemma C.7.3. *Suppose $X_1, X_2, \cdots, X_n$ are i.i.d. Bernoulli random variables with $\mathbb{E}[X] = p$ and $N = \sum_{i=1}^{n} X_i$. For any $a \geq 12\log(2/\delta)$ with probability at least $1-\delta$, we have*

$$\frac{1}{2}(N \vee a) \leq np \vee a \leq 2(N \vee a).$$

Proof. By the multiplicative Chernoff bound, we have

$$\mathbb{P}\left[|N - np| \geq \frac{1}{2}np\right] \leq 2\exp\left(-\frac{np}{12}\right).$$

Thus if $np \geq 12\log(2/\delta)$, we have

$$\mathbb{P}\left[\frac{1}{2}np \leq N \leq 2np\right] \leq \delta.$$

If $np < 12\log(2/\delta)$, by Bernstein inequality, with probability $1-\delta$ we have

$$\mathbb{P}\left[N - np > t\right] \leq \exp\left(-\frac{t^2/2}{np + t/3}\right).$$

Let $t = a \geq np$ and we have

$$\mathbb{P}\left[N > 2a\right] \leq \exp\left(-\frac{a^2/2}{np + a/3}\right) \leq \exp(-3a/8) \leq \delta.$$

Note that if $N \leq 2a$, we directly have

$$\frac{1}{2}\left(N \vee a\right) \leq np \vee a \leq 2\left(N \vee a\right).$$

□

Lemma C.7.4. *(Lemma 20.1 in Lattimore and Szepesvári [2020]) The Euclidean sphere $S^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$. There exists a set $\mathcal{C}_\epsilon \subset \mathbb{R}^d$ with $|\mathcal{C}_\epsilon| \leq (3/\epsilon)^d$ such that for all $x \in S^{d-1}$ there exists $y \in C_\epsilon$ with $\|x - y\|_2 \leq \epsilon$.*

Lemma C.7.5. *Let $\Sigma \succeq \lambda I$ be a positive definite matrix and M be a positive semidefinite matrix with eigenvalue upper-bounded by 1. Let $\Sigma' = \Sigma + M$. Then we have*

$$\log\det(\Sigma') \geq \log\det(\Sigma) + \mathrm{Tr}(\Sigma^{-1}M).$$

Proof.

$$\begin{aligned}\det(\Sigma') &= \det(\Sigma + M)\\ &= \det(\Sigma)\det(I + \Sigma^{-1/2}M\Sigma^{-1/2}).\end{aligned}$$

Denote $\lambda_1, \ldots, \lambda_d$ as the eigenvalues of $\Sigma^{-1/2}M\Sigma^{-1/2}$. Then we have

$$x^\top\Sigma^{-1/2}M\Sigma^{-1/2}x \leq \left\|\Sigma^{-1/2}x\right\|_2^2 = x^\top\Sigma^{-1}x \leq \lambda^{-1},$$

which means $\lambda_i \in [0, \lambda^{-1}]$ for all $i \in [d]$. Thus, we have

$$\log\det(\Sigma') = \log\det(\Sigma) + \sum_{i=1}^{d}\log(1+\lambda_i) \geq \log\det(\Sigma) + \sum_{i=1}^{d}\frac{\lambda}{\lambda+1}\lambda_i = \log\det(\Sigma) + \frac{\lambda}{\lambda+1}\mathrm{Tr}(\Sigma^{-1}M),$$

completing the proof. □

Lemma C.7.6. *(Lemma 11 in Zanette and Wainwright [2022]) For any random vector $\phi \in \mathbb{R}^d$, scalar $\alpha > 0$ and positive definite matrix Σ, we have*

$$\frac{\alpha}{L}\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2 \leq \log\frac{\det(\Sigma+\alpha\mathbb{E}[\phi\phi^\top])}{\det(\Sigma)} \leq \alpha\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2,$$

whenever $\alpha\mathbb{E}\left\|\phi\right\|_{\Sigma^{-1}}^2 \leq L$ for some $L \geq e-1$.

Lemma C.7.7. *Let $b > 0$ and $a_1, a_2, \cdots, a_n > 0$ such that $a_{n+1} \leq c \cdot \left(\sum_{l=1}^{n-1} a_l \vee b\right)$ for all $n \geq 1$ and some constant c. Then we have*

$$\sum_{i=1}^{\infty} a_i \sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \leq 2\sqrt{(c+1)\sum_{l=1}^{n} a_l}.$$

Proof. Note that for any $i \geq 1$ we have

$$\sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \leq \sqrt{\frac{c+1}{(\sum_{l=1}^{i} a_l) \vee b}}.$$

Let $f(x) = \sqrt{\frac{c+1}{x \vee b}}$ for $x \geq 0$ and immediately we have $f(x)$ is non-increasing. Then we have

$$\begin{aligned}
\sum_{i=1}^{n} a_i \sqrt{\frac{1}{(\sum_{l=1}^{i-1} a_l) \vee b}} \leq & \sum_{i=1}^{\infty} a_i \sqrt{\frac{c+1}{(\sum_{l=1}^{i} a_l) \vee b}} \\
= & \sum_{i=1}^{n} a_i f(\sum_{l=1}^{i} a_l) \\
\leq & \int_0^{\sum_{l=1}^{n} a_l} f(x) \\
\leq & 2\sqrt{(c+1)\sum_{l=1}^{n} a_l}.
\end{aligned}$$

□

Lemma C.7.8. *(Lemma 4 in Zanette and Wainwright [2022]) Let $X \in \mathbb{R}^d$ be a random vector and Y be a random variable such that $\left\|X\right\|_2 \leq 1$, $|Y| \leq Y_{\max}$, $(X,Y) \sim \mathbb{P}$ for some distribution $\mathbb{P}$. Let $\{(x_i, y_i)\}_{i=1}^n$ be n i.i.d. samples from $\mathbb{P}$. Then we define*

$$\beta^* := \operatorname*{argmin}_{\left\|\beta\right\|_2 \leq W} \mathbb{E}_{(X,Y)\sim\mathbb{P}}(Y - \langle X, \beta\rangle)^2,$$

$$\widehat{\beta} := \operatorname*{argmin}_{\left\|\beta\right\|_2 \leq W} \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle x_i, \beta\rangle)^2.$$

Then with probability at least $1-\delta$*, we have*

$$\left\|\beta^* - \widehat{\beta}\right\|_{n\mathbb{E}[XX^\top]+\lambda I} \leq 8(W + Y_{\max})\sqrt{d\log(32Wn(W+Y_{\max})) + \log(1/\delta) + \lambda}.$$

Lemma C.7.9. *(Covariance Concentration) (Proposition 1 in Zanette and Wainwright [2022]) Suppose* $\{Z_k\}_{k=1^K}$ *is a sequence of independent, symmetric and positive definite random matrices of dimension* d *such that*

$$0 \leq \lambda_{\min}(Z_k) \leq \lambda_{\max}(Z_k) \leq 1, \forall k \in [K].$$

Let $\widehat{\Sigma} = \lambda I + \sum_{k=1}^K Z_k$ *and* $\Sigma = \mathbb{E}[\widehat{\Sigma}]$ *for some* $\lambda \geq 0$*. For any* $\delta \in (0,1)$ *and* $\lambda > 2\frac{\log(2d/\delta)}{\log(36/35)}$*, with probability at least* $1-\delta$ *we have*

$$\frac{1}{2}\widehat{\Sigma} \preceq \Sigma \preceq \frac{3}{2}\widehat{\Sigma}.$$

Appendix D

DEFERRED CONTENTS FROM CHAPTER ??

D.1 Challenges in Non-stationary Games

In this section, we discuss the challenges in non-stationary games in more detail.

D.1.1 Challenges in Test-based Algorithms

The idea of achieving optimal regret using consecutive testing in a parameter-free fashion was first proposed in Auer et al. [2019b]. Here we restate the idea as follows. Consider the multi-armed bandit setting. There are K arms, T episodes and L abrupt changes. The regret can be decomposed as

- Most of the time, we run the standard UCB algorithm. If we always restart the UCB algorithm right after each abrupt change, the accumulated regret is upper bounded by $O\left(\sqrt{K(T/L)L}\right) = O\left(\sqrt{KTL}\right)$.

- Intending to detect changes on one arm that make the optimal arm incur D regret, the algorithm starts a test at each step with probability $p_D = D\sqrt{l/KT}$ where l is the number of changes detected thus far. The test should last $n_D = O(1/D^2)$ steps to make the confidence bound no larger than D. In expectation, the test incurs $p_D T n_D \Delta = O\left(\frac{\Delta}{D}\sqrt{\frac{lT}{K}}\right)$ regret. Here Δ is the real gap of the detected arm. To cover all possible Δ, we may detect for gaps of size $D = D_0, 2D_0, 4D_0, \cdots$. D_0 is the smallest gap that is worth noticing*. This incurs $O\left(\sqrt{\frac{LT}{K}}K\right) = O\left(\sqrt{KTL}\right)$ regret.

- The expected number of episodes before we start to detect for a change of size D is $D/p_D = \sqrt{KT/l}$. Summing over all changes, this part incurs $O\left(KTL\right)$ regret

*We can take $D_0 = \sqrt{K/T}$ because even if each step we suffer an extra D_0 regret, the total regret will still remain.

In all, the scheme suffer $O\left(\sqrt{KLT}\right)$ regret, which is optimal. In the game setting, however, the second part can become $\frac{1}{D_0}\sqrt{\frac{lT}{K}}K$ and we will no longer have a no-regret algorithm.

D.1.2 Challenges in Bandit-over-RL Algorithms

The high-level idea of BORL is as follows [Cheung et al., 2020]. First partition the whole time horizon T into intervals with length H. Each interval is one step for an adversarial bandit algorithm A. Inside each interval, one instance of the base algorithm is run, with the tunable parameter selected by A. The arms for A are the possible parameters of the base algorithm and the reward is the total reward from one interval. Let the action at timestep t be a_t and $r(a_t)$ be its expected reward, a_t^* be the optimal action at timestep t and $R(w)$ be the expected return from one interval if we chooses parameter w. The regret can then be decomposed as

$$\sum_{t=1}^{T}\left[r\left(a_t^*\right)-r\left(a_t\right)\right]=\left[\sum_{t=1}^{T} r\left(a_t^*\right)-\sum_{h=1}^{T/H} R(w_h)\right]+\left[\sum_{h=1}^{T/H} R(w_h)-\sum_{t=1}^{T} r(a_t)\right]$$

where w_h is the best parameter in interval h. The first term is bounded by the base algorithm regret upper bound and the second term is bounded by the adversarial bandit regret guarantee. If we apply the same to minimize, for example, the Nash regret

$$\sum_{t=1}^{T}\max_{i\in[m]}\left(V_i^M(\dagger,\pi_{-i})-V_i^M(\pi)\right),$$

we easily find the max hinders the same decomposition. Even if we drop the max and focus on individual regret, the decomposition is

$$\sum_{t=1}^{T}\left[V_i^M(\dagger,\pi_{-i})-V_i^M(\pi)\right]=\left[\sum_{i=1}^{T} V_i^M(\dagger,\pi_{-i})-\sum_{h=1}^{T/H} R(w_h)\right]+\left[\sum_{h=1}^{T/H} R(w_h)-\sum_{t=1}^{T} V_i^M(\pi)\right]$$

where the first term loses meaning. The fundamental reason is that in MAB, at timestep t, any action is competing with a fixed action a_t^*, while in a game, a policy π is competing with $\arg\max_{\pi_i'} V_i^M(\pi_i',\pi_{-i})$, which depends on π itself. This difficulty can also be seen from Figure 5.1.

D.2 Omitted Proofs in Section 5.4

In this section, we analyze the performance of Algorithm 8. For convenience, we denote the intervals corresponding to each LEARN_EQ by $\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_K$ and the committing phases as $\mathcal{J}_1, \mathcal{J}_2, \cdots, \mathcal{J}_K$. The committed policy are $\pi^1, \pi^2, \cdots, \pi^K$ respectively. Here $K = \lceil T/(C_1(\epsilon) + T_1) \rceil$ and $\mathcal{J}_K$ can be empty.

Lemma D.2.1. *If $x > 1$, $x/2 < \lceil x \rceil < x + 1$.*

Remark D.2.2. It is a basic algebraic lemma that will be used very often to get over the roundings.

Lemma D.2.3. *If π is an ϵ-EQ of episode t, then it is also an $\left(\epsilon + 2H\Delta_{[t,t']}\right)$-equilibrium for any episode $t' > t$.*

Proof. To facilitate this proof, we define some more notations. The value function of player i at timestep h_0, episode t, state s is defined to be

$$V_{h_0,i}^{\pi,M}(s) = \mathbb{E}_\pi \left[\sum_{h=h_0}^{H} r_{h,i}(s_h, \mathbf{a}_h) \mid M, s_{h_0} = s \right]. \tag{D.1}$$

Here M is the model at episode t. We also denote the model at episode t' by M'. We have the recursion

$$V_{h_0,i}^{\pi,M}(s) = \sum_{\mathbf{a}} \pi(\mathbf{a} \mid s) \left[\sum_{s'} \mathbb{P}_{h_0}^{M}\left(s'|s,\mathbf{a}\right) V_{h_0+1,i}^{\pi,M}(s') + R_{h_0,i}^{M}(s,\mathbf{a}) \right].$$

Assume

$$\left| V_{h_0+1,i}^{\pi,M}(s) - V_{h_0+1,i}^{\pi,M'}(s) \right| \le H \sum_{h=h_0+1}^{H} \left(\left\| \mathbb{P}_h^M - \mathbb{P}_h^{M'} \right\|_1 + \left\| R_h^M - R_h^{M'} \right\|_1 \right),$$

then we have

$$\begin{aligned}
&\left| V_{h_0,i}^{\pi,M}(s) - V_{h_0,i}^{\pi,M'}(s) \right| \\
\le &\sum_{\mathbf{a}} \pi(\mathbf{a} \mid s) \left[\sum_{s'} \left(\mathbb{P}_{h_0}^{M}\left(s'|s,\mathbf{a}\right) - \mathbb{P}_{h_0}^{M'}\left(s'|s,\mathbf{a}\right) \right) V_{h_0+1,i}^{\pi,M'}(s') \right] \\
&+ \sum_{\mathbf{a}} \pi(\mathbf{a} \mid s) \left[\sum_{s'} \mathbb{P}_{h_0}^{M}\left(s'|s,\mathbf{a}\right) \left(V_{h_0+1,i}^{\pi,M'}(s') - V_{h_0+1,i}^{\pi,M}(s') \right) \right]
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{\mathbf{a}} \pi(\mathbf{a} \mid s)\left[R_{h_0,i}^{M}(s,\mathbf{a}) - R_{h_0,i}^{M'}(s,\mathbf{a})\right] \\
\leq & H\left|\mathbb{P}_{h_0}^{M}\left(s'|s,\mathbf{a}\right) - \mathbb{P}_{h_0}^{M'}\left(s'|s,\mathbf{a}\right)\right| + \left|V_{h_0+1,i}^{\pi,M}(s) - V_{h_0+1,i}^{\pi,M'}(s)\right| + \left|R_{h_0,i}^{M}(s,\mathbf{a}) - R_{h_0,i}^{M'}(s,\mathbf{a})\right| \\
\leq & H\sum_{h=h_0}^{H}\left(\left\|\mathbb{P}_{h}^{M} - \mathbb{P}_{h}^{M'}\right\|_1 + \left\|R_{h}^{M} - R_{h}^{M'}\right\|_1\right).
\end{aligned}
$$

Since the assumption holds trivially for $h_0 = H$, by induction we get

$$
\left|V_1^{\pi,M}(s) - V_1^{\pi,M'}(s)\right| \leq \Delta_{[t,t']}\,.
$$

Finally by definition of the equilibria, we get the conclusion. □

Lemma D.2.4. *With probability $1-T\delta$, π^k is $\left(\epsilon + c_1^{\Delta}\Delta_{\mathcal{I}_k}\right)$-approximate equilibrium in the last episode of $\mathcal{I}_k$ for all $k \in [K]$.*

Proof. This is by the union bound and $K \leq T$. □

The following theorem is conditioned on this high-probability event.

Proposition 5.4.1. *With probability $1 - T\delta$, the regret of Algorithm 8 satisfies*

$$
\mathrm{Regret}(T) \leq \frac{4TC_1(\epsilon)}{T_1} + T\epsilon + 2\max\left\{c_1^{\Delta}, H\right\}T_1\Delta.
$$

Proof. According to Assumption 5.2.6, π^k is an $\left(\epsilon + c_1^{\Delta}\Delta_{\mathcal{I}_k}\right)$-approximate equilibrium for the last episode of $\mathcal{I}_k$. Hence it is an $\left(\epsilon + 2\max\left\{c_1^{\Delta}, H\right\}\Delta_{\mathcal{I}_k\cup\mathcal{J}_k}\right)$-approximate equilibrium for any episode in $\mathcal{J}_k$ according to Lemma D.2.3. In the proof we omit the max with H and recover it in the conclusion.

$$
\begin{aligned}
\mathrm{Regret}(T) =& \sum_{k=1}^{K}\left(|\mathcal{I}_k| + |\mathcal{J}_k|\left(\epsilon + 2c_1^{\Delta}\Delta_{\mathcal{I}_k\cup\mathcal{J}_k}\right)\right) \\
\leq & K\lceil C_1(\epsilon)\rceil + T\epsilon + 2c_1^{\Delta}T_1\Delta \\
\leq & \frac{4TC_1(\epsilon)}{T_1} + T\epsilon + 2c_1^{\Delta}T_1\Delta.
\end{aligned}
$$

□

Corollary 5.4.3. *With probability $1 - T\delta$, the regret of Algorithm 8 satisfies*

$$
\mathrm{Regret}(T) \leq \begin{cases} 13\left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}\right)^{1/4} T^{3/4}, & \alpha = -2, \\ 13\left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}\right)^{1/5} T^{4/5}, & \alpha = -3, \end{cases}
$$

Protocol 5 Scheduling TEST_EQ in a block with length 2^n

Input: Joint Markov policy π, failure probability δ, tolerance ϵ.
for $\tau = 0, 1, \ldots, 2^n - 1$ **do**
  for $q = 0, 1, \cdots, Q$ **do**
    if τ is a multiple of 2^{c+q} **then**
      With probability $p(q)$, schedule a TEST_EQ for $\epsilon(q)$ starting from τ.
    end if
  end for
end for

by setting

$$T_1 = \left\lceil \sqrt{\frac{TC_1(\epsilon)}{\max\left\{c_1^{\Delta}, H\right\}\Delta}} \right\rceil, \quad \epsilon = \begin{cases} \left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}/T\right)^{1/4}, & \alpha = -2, \\ \left(\Delta c_1 \max\left\{c_1^{\Delta}, H\right\}/T\right)^{1/5}, & \alpha = -3. \end{cases}$$

Proof. As before, we omit the max with H in the proof.

$$\begin{aligned} \text{Regret}(T) \leq & 8TC_1(\epsilon)\sqrt{\frac{c_1^{\Delta}\Delta}{TC_1(\epsilon)}} + T\epsilon + 4c_1^{\Delta}\Delta\sqrt{\frac{TC_1(\epsilon)}{c_1^{\Delta}\Delta}} \\ = & 12\sqrt{c_1^{\Delta}T\Delta C_1(\epsilon)} + T\epsilon \\ = & 12\sqrt{c_1^{\Delta}T\Delta c_1}\epsilon^{\alpha/2} + T\epsilon \end{aligned}$$

Applying Lemma D.2.1, we get the desired conclusion. □

D.3 Omitted Proofs in Section 5.5

In Section D.3.1 we present the proof for Proposition 5.5.2 and Proposition 5.5.3. In Section D.3.2 and D.3.3, we analyze the performance of Algorithm 9. We first analyze the performance of single block in Section D.3.2 and then present the subsequent proof in Section D.3.3. For convenience, the episodes in Section D.3.2 refer to τ and the episodes in Section D.3.3 refer to t.

D.3.1 Proofs Regarding Construction of TEST_EQ

Proposition 5.5.2. *Suppose MDP M' is induced by MDP M and recommendation policy π. Then the optimal policy in MDP M' corresponds to a best strategy modification to recommendation policy π in MDP M.*

Proof. To facilitate the proof, we define some notations here. We define the value function of policy π in an MDP M at timestep h_0 and state s as

$$V_{h_0}^{\pi,M}(s) = \mathbb{E}_\pi \left[\sum_{h=h_0}^{H} r_h(s_h, a_h) \mid M, s_{h_0} = s \right].$$

The mean reward from $r_h(\cdot|s,a)$ is denoted as $R_h(s,a)$. Let π' be a policy in M, then

$$V_h^{\pi',M'}((s,b)) = \sum_a \pi'(a \mid (s,b)) \left[\sum_{(s',b')} P'_h((s',b')|(s,b),a) \, V_{h+1}^{\pi',M'}((s',b')) + R'_h((s,b),a) \right]$$

Additionally, the Q-function of a state-action pair (s,b) under policy π at timestep h_0 for agent i in Markov game M is defined as

$$Q_{h_0,i}^{\pi,M}(s,b) = \mathbb{E}_\pi \left[\sum_{h=h_0}^{H} r_{h,i}(s_h, \mathbf{a}_h) \mid M, s_{h_0} = s, a_{h_0,i} = b \right].$$

Assume π' is a deterministic policy and ψ_i is a strategy modification such that its choice is the same as the choice of π', then

$$\begin{aligned} Q_{h_0,i}^{\psi_i \diamond \pi, M}(s, \psi_i(b)) = \sum_{(s',b')} & \mathbb{P}_h(s'|s, \pi_h(s) = b, \psi_i(b)) \pi_{h+1}(b' \mid s') Q_{h_0+1,i}^{\psi_i \diamond \pi, M}(s', \psi_i(b)) \\ & + R_{h_0}(s, \psi_i(b) \mid \pi_h(s) = b) \end{aligned}$$

by definition of M' we can directly see that

$$V_h^{\pi',M'}((s,b)) = Q_{h,i}^{\psi_i \diamond \pi, M}(s, \psi_i(b)) \tag{D.2}$$

Hence the optimal policy of M' corresponds to a best strategy modification to recommendation policy. □

Proposition 5.5.3. *As long as LEARN_OP satisfies Assumption 5.5.1, Protocol 3 satisfies Assumption 5.2.7.*

Proof. We first consider the NE and CCE case. The main logic has been stated in the main text. We restate it here with environmental changes involved. Denote the intervals that run Line 2, 4, 5 by $\mathcal{I}, \mathcal{J}, \mathcal{K}$ respectively. Then with high probability, the estimation of $\widehat{V}_i(\pi)$ departs from the true value by at most $\epsilon/6 + \Delta_{\mathcal{I}}$ and that of $\widehat{V}_i(\pi_i', \pi_{-i})$ is at most $\epsilon/3 + c_3^{\Delta}\Delta_{\mathcal{J}} + \Delta_{\mathcal{K}}$. Combine all the error we get the conclusion. In terms of sample complexity

$$C_2(\epsilon) = \widetilde{O}\left(mC_3(\epsilon) + \epsilon^{-2}\right) = \widetilde{O}\left(mC_3(\epsilon)\right).$$

The last equality use the information-theoretic lower bound $C_3(\epsilon) = \Omega(\epsilon^{-2})$. Then we consider the CE case. By Equation D.2 we can prove the correctness of this algorithm using the same argument as before. In terms of sample complexity, it is the same as before except that we need to change the size of state space from $|\mathcal{S}|$ to $|\mathcal{S}|\,|\mathcal{A}|$. Finally, $c_2^{\Delta} = c_3^{\Delta}$ □

By Wei and Luo [2021], we know that we have $c_2^{\Delta} = c_3^{\Delta} = O(H)$.

D.3.2 Single Block Analysis

Divide $[C_1(\epsilon) + 1, 2^n]$ into $\mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_K$ such that $\mathcal{I}_k = [s_k, e_k], s_1 = C_1(\epsilon) + 1, e_K = 2^n, e_k + 1 = s_{k+1}$ and

$$\Delta_{\mathcal{I}_k} \leq \frac{1}{c_2^{\Delta}} \max\left\{\frac{1}{\sqrt{|\mathcal{I}_k|}}, 2^{-n/4-1}\right\}$$

Intervals with such property are called near-stationary. Let $E_n \in \mathcal{I}_l$ be the last episode (The block may be ended due to a failed TEST_EQ). Define $e_k' = \min\{E_n, e_k\}, \mathcal{I}_k' = [s_i, e_k']$. If $k > l$, $\mathcal{I}_k' = \varnothing$. For convenience, we denote $\tau_n = C_1(\epsilon) + 1$ in the following proof.

Definition D.3.1. For $k \in [K], q \in \{0, 1, \cdots, Q\}$, let

$$\tau_k(q) = \min\left\{\tau \in \mathcal{I}_k' |\ \pi \text{ is not a } 2\epsilon(q)\text{-EQ at } \tau\right\}, \xi_k(q) = \left[e_k' - \tau_k(q) + 1\right]_+.$$

First, we are going to show that with high probability no TEST_EQ is aborted.

Lemma D.3.2. *With probability $1 - 2QT\delta$, for any TEST_EQ instance testing gap $\epsilon(q)$ maintained from s to e, it returns fail if the policy is not $(2\epsilon(q) + c_2^{\Delta}\Delta_{[s,e]})$-NE/CCE for any $\tau \in [s, e]$. In equivalence, $e - s < 2^{c+q}$ and all TEST_EQ function as desired.*

Proof. By union bound, the probability all TEST_EQ function as desired is $1-QT\delta$. There are 2^{q-r} possible starting points for a test occupying 2^r episodes. For each of them, TEST_EQ exists with probability $1/(\epsilon(r)2^{n/2})$. By Bernstein's inequality, with probability $1-\delta$, the number of such tests is upper-bounded by

$$\begin{aligned}
&2^{q-r}\frac{1}{\epsilon(r)2^{n/2}}+\sqrt{2\cdot 2^{q-r}\frac{1}{\epsilon(r)2^{n/2}}\log\frac{1}{\delta}}+\log\frac{1}{\delta}\\
\leq&2\cdot 2^{q-r}\frac{1}{\epsilon(r)2^{n/2}}+2\log\frac{1}{\delta}\\
=&\frac{2^{q-r/2+1}}{\sqrt{c_2}2^{n/2}}+2\log\frac{1}{\delta}.
\end{aligned}$$

By union bound, this inequality holds for all TEST_EQ with probability $1-QT\delta$. So the total length of all shorter tests is upper bounded by

$$\begin{aligned}
&\sum_{r=0}^{q-1}\left(\frac{2^{q-r/2+1}}{\sqrt{c_2}2^{n/2}}+2\log\frac{1}{\delta}\right)2^r\\
\leq&2^{q+1}\frac{2^{\frac{q-1}{2}}-1}{\sqrt{2}-1}\frac{1}{\sqrt{c_2}2^{n/2}}+2\log\frac{1}{\delta}(2^q-1)\\
\leq&5\sqrt{c_2}\left(2^{\frac{q-1}{2}}-1\right)+\log\frac{1}{\delta}2^{q+1}\\
\leq&\max\left\{5\sqrt{c_2},2\log\frac{1}{\delta}\right\}2^q
\end{aligned}$$

Here we use $2^q<2^Q<c_2 2^{n/2}$. Using the union bound, we get the conclusion. □

In subsequent proofs, we condition on the high probability event described in this lemma.

Lemma D.3.3. *With probability $1-Q\delta$, for all $r\in[Q]$.*

$$\sum_{k=1}^{l}\left[\xi_k(r)-2^{c+r}\right]_+\leq 2^{c+r-1}\epsilon(r-2)\sqrt{2^n}\log\frac{1}{\delta}=2^c\sqrt{2^{r+n}c_2}$$

Proof. For each $r\in[Q]$.

$$\begin{aligned}
&2^{-c-r+2}\sum_{k=1}^{l}\left[\xi_k(r)-2^{c+r}\right]_+\\
=&2^{-c-r+2}\sum_{k=1}^{l}\left[e'_k-\tau_k(r)+1-2^{c+r}\right]_+\\
\leq&\sum_{k=1}^{K}\sum_{\tau\in\mathcal{I}_k}\mathbb{1}\left[\tau\in[\tau_k(r),e'_k-2^{c+r-1}],\tau\bmod 2^{c+r-2}\equiv 0\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\tau=\tau_n}^{2^n} \mathbb{1}\left[\tau \in [\tau_k(r), e'_k - 2^{c+r-1}], \tau \bmod 2^{c+r-2} \equiv 0\right] \\
&\le \sum_{\tau=\tau_n}^{2^n} \mathbb{1}\left[\tau \in [\tau_k(r), e'_k - 2^{c+r-1}], \tau \bmod 2^{c+r-2} \equiv 0 \text{ and there is no test for } \epsilon(r)/2 \text{ starting at any } t \in [\tau_n, \tau]\right] \\
&\quad + \sum_{\tau=\tau_n}^{2^n} \mathbb{1}\left[\tau \in [1, E_n - 2^{c+r-1}] \text{ and there is a test for } \epsilon(r)/2 \text{ starting at some } t \in [\tau_n, \tau]\right] \\
&\le \left[1 + \frac{\log(1/\delta)}{-\log(1 - 1/\left(\epsilon(r-2)\sqrt{2^n}\right))}\right] + 0 \le 2\epsilon(r-2)\sqrt{2^n}\log\frac{1}{\delta}
\end{aligned}
$$

The first inequality holds because in an interval of length w, there are at least $(w+2-2u)/u$ points whose indices are multiples of u. The third inequality holds with probability $1-\delta$. The first sum is bounded using the fact the test is started i.i.d. with constant probability $1/\left(\epsilon(r-2)\sqrt{2^n}\right)$. In the second sum, the condition implies that the ending time of the test is before $t + 2^{c+r-2} - 1 \le e_i - 2^{c+r-2} - 1 \le e_i$ so the test is within $\mathcal{I}_k$ and $t + 2^{c+r-2} - 1 \le \tau + 2^{c+r-2} - 1 < E_n$ so the test ends before the block ends. However, the test is for $\epsilon(r)$ and the variation during the test is bounded by $\Delta_{\mathcal{I}_k} < 2^{-n/4} = \epsilon < \epsilon(r)$, so such TEST_EQ must return Fail. □

In subsequent proofs, we further condition on the high probability event described in this lemma.

Lemma D.3.4. *The total number of near-stationary intervals*

$$
l \le 1 + 2\min\left\{2^{n/3}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{2/3}, 2^{n/4}c_2^{\Delta}\Delta_{[1,E_n]}\right\} \tag{D.3}
$$

Proof. We divide $[\tau_n, E_n] = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_l$ in such a way that $[s_k, e_k]$ is near-stationary but $[s_k, e_k + 1]$ is not near-stationary. Then

$$
\begin{aligned}
\Delta_{[\tau_n, E_n]} &\ge \sum_{k=1}^{l-1} \Delta_{[s_k, e_k+1]} \\
&\ge \frac{1}{c_2^{\Delta}} \sum_{k=1}^{l-1} \max\left\{\frac{1}{\sqrt{e_k - s_k + 2}}, 2^{-n/4-1}\right\} \\
&\ge \frac{1}{c_2^{\Delta}} \max\left\{\sum_{k=1}^{l-1} \frac{1}{2\sqrt{e_k - s_k + 1}}, (l-1)2^{-n/4-1}\right\}
\end{aligned}
$$

Hence by Hölder's inequality

$$\begin{aligned}
l \leq & 1 + \min\left\{ \left(\sum_{k=1}^{l-1} (e_k - s_k + 1)^{-1/2} \right)^{2/3} \left(\sum_{k=1}^{l-1} (e_k - s_k + 1) \right)^{1/3}, 2^{n/4+1} c_2^{\Delta} \Delta_{[\tau_n, E_n]} \right\} \\
\leq & 1 + 2 \min\left\{ \left(c_2^{\Delta} \Delta_{[\tau_n, E_n]} \right)^{2/3} |[\tau_n, E_n]|^{1/3}, 2^{n/4} c_2^{\Delta} \Delta_{[\tau_n, E_n]} \right\} \\
\leq & 1 + 2 \min\left\{ 2^{n/3} \left(c_2^{\Delta} \Delta_{[1, E_n]} \right)^{2/3}, 2^{n/4} c_2^{\Delta} \Delta_{[1, E_n]} \right\}
\end{aligned}$$

□

Lemma D.3.5. *With probability* $1 - 3QT\delta$

$$\text{Regret}([1, E_n]) \leq 2^{3n/4+4} + 4Q \left(2^{n/2+c} \sqrt{c_2 l} + 2^{c+n/2} c_2 \right) + c_2 \log \frac{1}{\delta} 2^{n/2+1} + c_1 2^{-\alpha n/4}$$

Proof. First we consider the regret generated by TEST_EQ. We need to count the number of steps all the tests go for. Similar to the calculation in Lemma D.3.3. The number of tests with length 2^q is upper bounded by

$$\frac{2^{n-r/2+1}}{c\sqrt{c_2} 2^{n/2}} + 2 \log \frac{1}{\delta}.$$

So the total length of all TEST_EQ is upper bounded by

$$\begin{aligned}
& \sum_{r=0}^{Q} \left(\frac{2^{n-r/2+1}}{2^c \sqrt{c_2} 2^{n/2}} + 2 \log \frac{1}{\delta} \right) 2^r \\
\leq & 2^{n+1} \frac{2^{Q/2} - 1}{\sqrt{2} - 1} \frac{1}{c\sqrt{c_2} 2^{n/2}} + 2 \log \frac{1}{\delta} \left(2^Q - 1 \right) \\
\leq & \frac{5}{2^c} 2^{3n/4} + c_2 \log \frac{1}{\delta} 2^{n/2+1}
\end{aligned}$$

Then we consider the regret generated by committing.

$$\begin{aligned}
& \sum_{\tau \in \mathcal{I}'_k} \text{Gap}^{M^t} \left(\pi^t \right) \\
\leq & \sum_{\tau \in \mathcal{I}'_k} \left(\mathbb{1}\left[\text{Gap}^{M^t} \left(\pi^t \right) \leq 2\epsilon(Q) \right] 2\epsilon(Q) \right. \\
& \quad \left. + \sum_{r=0}^{Q-1} \mathbb{1}\left[2\epsilon(r+1) \leq \text{Gap}^{M^t} \left(\pi^t \right) \leq 2\epsilon(r) \right] 2\epsilon(r) + \mathbb{1}\left[\text{Gap}^{M^t} \left(\pi^t \right) > \epsilon(0) \right] 1 \right) \\
\leq & 2|\mathcal{I}'_k| \epsilon(Q) + 2 \sum_{r=0}^{Q-1} \epsilon(r) \xi_i(r+1) + 2\epsilon(0) \xi_i(0)
\end{aligned}$$

$$\leq 2|\mathcal{I}'_k|\epsilon(Q) + 4\sum_{r=0}^{Q}\epsilon(r)\xi_i(r)$$

In the second inequaility we use $\epsilon(0) = \sqrt{c_2} > 1$ and in the third inequality we use $\epsilon(r) \leq 2\epsilon(r+1)$. Summing over all intervals we have

$$\text{Regret}([1, E_n]) \leq 2^{n+1}\epsilon(Q) + 4\sum_{r=0}^{Q}\sum_{k=1}^{l}\epsilon(r)\xi_k(r).$$

Furthermore

$$\begin{aligned}
\sum_{k=1}^{l}\epsilon(r)\xi_k(r) &= \sum_{k=1}^{l}\epsilon(r)\min\{\xi_k(r), 2^{c+r}\} + \sum_{k=1}^{l}\epsilon(r)\left[\xi_k(r) - 2^{c+r}\right]_+ \\
&\leq 2^c\sum_{k=1}^{l}\sqrt{c_2\min\{\xi_k(r), 2^{c+r}\}} + \sum_{k=1}^{l}\epsilon(r)\left[\xi_k(r) - 2^{c+r}\right]_+ \\
&\leq 2^c\sum_{k=1}^{l}\sqrt{c_2\left|\mathcal{I}'_k\right|} + 2^{c+n/2}c_2
\end{aligned}$$

The last inequality uses Lemma D.3.3. Hence

$$\begin{aligned}
\text{Regret}([1, E_n]) &\leq 2^{n+1}\epsilon(Q) + 4Q\left(2^c\sum_{k=1}^{l}\sqrt{c_2\left|\mathcal{I}'_k\right|} + 2^{c+n/2}c_2\right) + \frac{5}{2^c}2^{3n/4} + c_2\log\frac{1}{\delta}2^{n/2+1} + C_1(\epsilon) \\
&\leq 2^{3n/4+4} + 4Q\left(2^c\sqrt{c_2 l\sum_{k=1}^{l}\left|\mathcal{I}'_k\right|} + 2^{c+n/2}c_2\right) + c_2\log\frac{1}{\delta}2^{n/2+1} + c_1 2^{-\alpha n/4} \\
&\leq 2^{3n/4+4} + 4Q\left(2^{n/2+c}\sqrt{c_2 l} + 2^{c+n/2}c_2\right) + c_2\log\frac{1}{\delta}2^{n/2+1} + c_1 2^{-\alpha n/4}
\end{aligned}$$

□

To keep the notation clean, from now on we make frequent use of the big-O notation and hide the dependencies on logarithmic factors on relevant variables. We also assume Δ is always large enough so that we can drop the 1 in Inequality D.3.

Lemma 5.5.4. *With probability $1 - 3QT\delta$, the regret inside this block*

$$\text{Regret} = \widetilde{O}\left(2^{3n/4} + c_2\min\left\{2^{2n/3}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/3}, 2^{5n/8}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/2}\right\} + 2^{n/2}c_2^{3/2} + 2^{-\alpha n/4}c_1\right) \tag{5.1}$$

Proof. We may restate the bounds in Lemma D.3.4 and D.3.5 as

$$l = O\left(\min\left\{2^{n/3}\left(c_2^{\Delta}\Delta\right)_{[1,E_n]}^{2/3}, 2^{n/4}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)\right\}\right)$$

$$\text{Regret}([1, E_n]) = \widetilde{O}\left(2^{3n/4} + 2^{n/2}c_2\sqrt{l} + 2^{n/2}c_2^{3/2} + 2^{-\alpha n/4}c_1\right)$$

Combine them together we get

$$\text{Regret}([1, E_n]) = \widetilde{O}\left(2^{3n/4} + c_2 \min\left\{2^{2n/3}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/3}, 2^{5n/8}\left(c_2^{\Delta}\Delta_{[1,E_n]}\right)^{1/2}\right\} + 2^{n/2}c_2^{3/2} + 2^{-\alpha n/4}c_1\right) \tag{D.4}$$

□

D.3.3 Proof for Theorem 5.5.6

Due to the doubling structure inside each segment, from Formula D.4 we get

$$\text{Regret}\left(\mathcal{J}_j\right) = \widetilde{O}\left(\left|\mathcal{J}_j\right|^{3/4} + c_2 \min\left\{\left|\mathcal{J}_j\right|^{2/3}\left(c_2^{\Delta}\Delta_{\mathcal{J}_j}\right)^{1/3}, \left|\mathcal{J}_j\right|^{5/8}\left(c_2^{\Delta}\Delta_{\mathcal{J}_j}\right)^{1/2}\right\} + c_2^{3/2}\left|\mathcal{J}_j\right|^{1/2} + c_1\left|\mathcal{J}_j\right|^{-\alpha/4}\right)$$

Lemma D.3.6.

$$J = O\left(T^{1/5}\left(\max\left\{c_1^{\Delta}, c_2^{\Delta}\right\}\Delta\right)^{4/5}\right).$$

Proof. For any segment $\mathcal{J}_j$,

$$\max\left\{c_1^{\Delta}, c_2^{\Delta}\right\}\Delta_{\mathcal{J}_j} \geq \epsilon(Q) - \epsilon \geq \left(\sqrt{2} - 1\right)\left|\mathcal{J}_j\right|^{-1/4}$$

since the ending of a segment is caused by a False returned by TEST_EQ. Then by the same logic as in Lemma D.3.4 we get the conclusion □

Hence by Hölder inequality

$$\begin{aligned}\text{Regret}(T) =& \widetilde{O}\left(J^{1/4}T^{3/4} + c_2 \min\left\{T^{2/3}\widetilde{\Delta}^{1/3}, T^{5/8}\widetilde{\Delta}^{1/2}\right\} + c_2^{3/2}J^{1/2}T^{1/2} + c_1 J^{1+\alpha/4}T^{-\alpha/4}\right) \\ =& \begin{cases} \widetilde{O}\left(\check{\Delta}^{1/5}T^{4/5} + c_2 \min\left\{\widetilde{\Delta}^{1/3}T^{2/3}, \widetilde{\Delta}^{1/2}T^{5/8}\right\} + \left(c_1 + c_2^{3/2}\right)\check{\Delta}^{2/5}T^{3/5}\right) & \alpha = -2 \\ \widetilde{O}\left(c_1\check{\Delta}^{1/5}T^{4/5} + c_2 \min\left\{\widetilde{\Delta}^{1/3}T^{2/3}, \widetilde{\Delta}^{1/2}T^{5/8}\right\} + c_2^{3/2}\check{\Delta}^{2/5}T^{3/5}\right) & \alpha = -3 \end{cases}\end{aligned}$$

D.4 Base Algorithms Satisfying Assumption 5.2.6

In table D.1 we summarize the results of this section.

Table D.1: Parameters of the Base Algorithms. In this table we only show the magnitude of parameter, with $\widetilde{O}(\cdot)$ omitted except for the α column.

Types of Games	c_1	α	c_1^Δ
Zero-sum (NE)	$A+B$	-2	1
General-sum (CCE)	$A_{\max}$	-2	1
General-sum (CE)	$A_{\max}^2$	-2	1
Potential (NE)	$m^2 A_{\max}$	-3	1
Congestion (NE)	$m^2 F^3$	-2	mF
Zero-sum Markov (NE)	$H^5 S(A+B)$	-2	H^2
General-sum Markov (CCE)	$H^6 S^2 A_{\max}$	-2	HS
General-sum Markov (CE)	$H^6 S^2 A_{\max}^2$	-2	HS
Markov Potential (NE)	$m^2 H^4 S A_{\max}$	-3	H^2

D.4.1 Two-Player Zero-Sum Matrix Games (NE)

In this part we consider the following algorithm: each player independently runs an optimal adversarial multi-armed bandit algorithm (e.g. EXP.3) and finally output the product of respective average policies of the whole time horizon. We will prove that this algorithm satisfies Assumption 5.2.6 in terms of learning NE in two-player zero-sum matrix games.

Proof. We adopt some new notations in this proof. Let $R^t \in [0,1]^{A\times B}$ be the reward matrix at episode t. The policy of the max and min players are represented by $x^t \in [0,1]^A, y^t \in [0,1]^B$. Each entry represents the probability they choose the corresponding action. The reward received by the max and min players are respectively ${x^t}^\top R^t y^t$ and $-{x^t}^\top R^t y^t$. With probability $1-\delta$ the adversarial MAB algorithms satisfy

$$\frac{1}{T}\sum_{t=1}^{T} {x^t}^\top R^t y^t - \min_y \frac{1}{T}\sum_{t=1}^{T} {x^t}^\top R^t y \le c_{\text{adv}}\sqrt{AT}$$

$$\max_x \frac{1}{T}\sum_{t=1}^{T} x^\top R^t y^t - \frac{1}{T}\sum_{t=1}^{T} {x^t}^\top R^t y^t \le c_{\text{adv}}\sqrt{BT}$$

where $c_{\text{adv}} = \widetilde{O}(1)$. The output policy $\overline{x} = \sum_{t=1}^{T} x^t/T$ and $\overline{y} = \sum_{t=1}^{T} y^t/T$ satisfy

$$V_{\max}^{M^T}(\dagger, \overline{y}) + V_{\min}^{M^T}(\overline{x}, \dagger) = \max_x x^\top R^T \overline{y} + \min_y \overline{x}^\top R^T y \le c_{\text{adv}}\sqrt{BT} + \Delta + c_{\text{adv}}\sqrt{AT} + \Delta$$

By the definition of zero-sum game

$$\begin{aligned}\text{NEGAP}(\overline{x}, \overline{y}) \le & \frac{V_{\max}^{M^T}(\dagger, \overline{y}) - V_{\max}^{M^T}(\overline{x}, \overline{y}) + V_{\min}^{M^T}(\overline{x}, \dagger) - V_{\min}^{M^T}(\overline{x}, \overline{y})}{2} \\ = & \frac{V_{\max}^{M^T}(\dagger, \overline{y}) + V_{\min}^{M^T}(\overline{x}, \dagger)}{2} = \widetilde{O}\left(\sqrt{(A+B)T}\right) + 2\Delta.\end{aligned}$$

Hence this algorithm satisfies Assumption 5.2.6 with $C_1(\epsilon) = \widetilde{O}\left((A+B)\epsilon^{-2}\right), c_1^{\Delta} = 2$. □

D.4.2 Multi-Player General-Sum Matrix Games (CCE)

In this part we consider the following algorithm: each player independently runs an optimal adversarial multi-armed bandit algorithm (e.g. EXP.3) and finally output the average joint policy of the whole time horizon. We will prove that this algorithm satisfies Assumption 5.2.6 in terms of learning CCE in multi-player general-sum matrix games.

Proof. We define the loss of player i at episode t by playing a_i as

$$l_i^t(a_i) = 1 - \mathbb{E}_{a_{-i} \sim \pi_{-i}^t}\left[r_i(a_i, a_{-i}) \mid M^t\right]$$

then with probability $1-\delta$, the adversarial MAB algorithm satisfies

$$\sum_{t=1}^{T} \langle \pi^t(\cdot), l_i^t(\cdot)\rangle - \min_{a_i \in \mathcal{A}_i} \sum_{t=1}^{T} l_i^t(a_i) \le c_{\text{adv}}\sqrt{A_i T}, \quad c_{\text{adv}} = \widetilde{O}(1)$$

For convenience, we denote the reward function at timestep t by r^t. Let the output policy $\pi = \sum_{t=1}^{T} \pi^t/T$, we have

$$\begin{aligned}& V_i^{M^T}(\pi) \\ =& \mathbb{E}_{\mathbf{a}\sim\pi}\left[r_i^T(\mathbf{a})\right] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{a_i \sim \pi_i^t} \mathbb{E}_{a_{-i}\sim\pi_{-i}^t}\left[r_i^T(a_i, a_{-i})\right] \\ =& 1 - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{a_i\sim\pi_i^t}\left[l_i^t(a_i)\right] + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{a_i\sim\pi_i^t}\mathbb{E}_{a_{-i}\sim\pi_{-i}^t}\left[r_i^T(a_i, a_{-i}) - r_i^t(a_i, a_{-i})\right] \\ \ge & 1 - \frac{1}{T}\min_{a_i\in\mathcal{A}_i}\sum_{t=1}^{T} l_i^t(a_i) - c_{\text{adv}}\sqrt{A_i/T} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{a_i\sim\pi_i^t}\mathbb{E}_{a_{-i}\sim\pi_{-i}^t}\left[r_i^T(a_i, a_{-i}) - r_i^t(a_i, a_{-i})\right]\end{aligned}$$

$$
\begin{aligned}
\geq & \frac{1}{T} \max_{a_i \in \mathcal{A}_i} \sum_{t=1}^{T} \mathbb{E}_{a_{-i} \sim \pi_{-i}^t} \left[r_i^t(a_i, a_{-i}) \right] - c_{\text{adv}} \sqrt{A_i / T} - \Delta \\
\geq & \frac{1}{T} \max_{a_i \in \mathcal{A}_i} \sum_{t=1}^{T} \mathbb{E}_{a_{-i} \sim \pi_{-i}^t} \left[r_i^T(a_i, a_{-i}) \right] - \Delta - c_{\text{adv}} \sqrt{A_i / T} - \Delta \\
\geq & V_i^{M^T}(\dagger, \pi_{-i}) - c_{\text{adv}} \sqrt{A_i / T} - 2\Delta
\end{aligned}
$$

By definition of CCE we know this algorithm satisfies Assumption 5.2.6 with $C_1(\epsilon) = \widetilde{O}\left(A_{\max} \epsilon^{-2}\right), c_1^{\Delta} = 2$ □

D.4.3 Multi-Player General-Sum Matrix Games (CE)

This part is very similar to the last part. Instead of using standard adversarial bandit algorithms, we use no-swap-regret algorithm for adversarial bandits (for example, Ito [2020]) and the proof is almost the same. We can achieve with probability $1 - \delta$,

$$
\sum_{t=1}^{T} \langle \pi^t(\cdot), l_i^t(\cdot) \rangle - \min_{\psi_i} \sum_{t=1}^{T} \langle (\psi_i \diamond \pi^t)(\cdot), l_i^t(\cdot) \rangle \leq c_{\text{adv}} A_i \sqrt{T}, \quad c_{\text{adv}} = \widetilde{O}(1)
$$

where ψ_i is a strategy modification. By substituting all $\min, \max$ related terms correspondingly we get the proof for CE and $C_1(\epsilon) = \widetilde{O}\left(A_{\max}^2 \epsilon^{-2}\right)$

D.4.4 Congestion Games (NE)

In this part we will show the Nash-UCB algorithm proposed in Cui et al. [2022] satisfies Assumption 5.2.6. We carry out the proof by pointing out the modifications we need to make in their proof. In their proof, k stands for the episode index instead of t and K is the total episodes instead of T.

Lemma D.4.1. *(Modified Lemma 3 in Cui et al. [2022]) With high probability,*

$$
\left| \tilde{r}_i^k - r_i \right| (\mathbf{a}) \leq \max_{i \in [m]} \left\| A_i(\mathbf{a}) \right\|_{(V^k)^{-1}} \sqrt{\widetilde{\beta_k}}, \widetilde{\beta_k} = O(mF + Km\Delta^2)
$$

Proof. We denote the average reward vector by $\overline{\theta}$ and the reward vector of the last epsiode by θ^T, other notations are similar, then

$$
\left| \tilde{r}_i^k - r_i^T \right| (\mathbf{a})
$$

$$
\begin{aligned}
&\leq \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \left\|\widehat{\theta} - \theta^T\right\|_{V^k} \\
&\leq \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \left(\left\|\widehat{\theta} - \overline{\theta}\right\|_{V^k} + \left\|\overline{\theta} - \theta^T\right\|_{V^k}\right) \\
&\leq \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \left(\left\|\overline{\theta}\right\|_2 + \sqrt{\log\det \overline{V^k} + \tilde{\iota}} + \sqrt{Km}\Delta\right)
\end{aligned}
$$

□

The rest of the proof is carried out with the new $\widetilde{\beta}_k$ and finally the regret becomes

$$\text{Nash-Regret}(K) = \widetilde{O}\left(mF^{3/2}\sqrt{K} + mFK\Delta\right).$$

Finally this algorithm can be converted into a version with sample complexity guarantee and $C_1(\epsilon) = m^2F^3\epsilon^{-2}, c_1^{\Delta} = mF$ as stated in the original paper using the certified policy trick from Bai et al. [2020].

D.4.5 Multi-Player General-Sum Markov Games (CCE,CE)

In this part we will show how to adapt the proof in Cui et al. [2023] to the non-stationary game case. For simplicity, we will follow the proof in Cui et al. [2023] in general and only point out critical changes. Note that they use k as epoch index while we have been using k as episode index. For consistency, we will use κ as the episode index in this section. As a reminder, we will use r^κ, P^κ and M^κ to denote the reward function, the transition kernel and the game at episode κ.

We use the superscript κ in $\mathbb{E}^\kappa[\cdot]$ to denote that the underlying game is M^κ. We further use $\kappa_h^k(j;s)$ to denote the episode index when state s is visited for the jth time at step h and epoch k in the no-regret learning phase (Line 12 in Algorithm 3), and we use $\overline{\kappa}_h^k(j;s)$ to denote the episode index when state s is visited for the jth time at step h and epoch k in the no-regret learning phase (Line 12 in Algorithm 3). We will change the algorithm in Line 34 where we replace $n_h^k(s_h)$ with $\sum_{l=1}^{k-1} T_h^k(s_h)$. We will modify all the lemmas in the proof below. We use N^k to denote $\sum_{l=1}^k n^k$.

First, we will replace $\mathbb{E}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}[\cdot]$ with $\mathbb{E}^{\kappa_h^k(j;s)}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}[\cdot]$ in all the lemmas, which takes the expectation with the underlying game when $\pi_h^{k,t_h^k(j;s)}(\cdot \mid s)$ is used. It is easy to verify that Lemma 35, Lemma 36, Lemma 37 hold after the modification.

Second, we will replace $n_h^k(s)$ with $\sum_{l=1}^{k-1} T_h^l(s)$ and $n^k d_h^{\pi^k}(s)$ with $\sum_{J=1}^{n^k} d_h^{\pi^k}(s;k,J)$, where $d_h^{\pi^l}(s;k,J)$ is the visiting density for model at epoch k and Jth trajectory sampled in the policy cover update phase. In addition, we also add the following argument in the lemma:

$$n_h^k(s) \vee \mathrm{Trig} \geq \frac{1}{2}\left(\sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j)\right) \vee T_{\mathrm{Trig}},$$

where $d_h^{\pi^l}(s;k,j)$ is the visiting density for model at epoch k and jth trajectory sampled in the no-regret learning phase. It is easy to verify that Lemma 38 hold after the modification.

Third, we will consider a baseline model M^0, which can be the game at any episode, and use $V_{h,i}^{\pi}(s)$ to denote the corresponding value function. Now we show that Lemma 39, Lemma 40 and Lemma 41 holds with an addition tolerance Δ.

Lemma D.4.2. *(Modified Lemma 39 in Cui et al. [2023]) Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\overline{V}_{h,i}^k(s) \geq V_{h,i}^{\dagger,\pi_{-i}^k}(s) - \sum_{h'=h}^{H} \Delta_h.$$

Proof. Note that we have

$$\left|\mathbb{E}^{\kappa_h^k(j;s)}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')\right] - \mathbb{E}^{M^0}_{\mathbf{a}\sim\pi_h^{k,t_h^k(j;s)}(\cdot|s)}\left[r_{h,i}(s,\mathbf{a}) + \overline{V}_{h+1,i}^k(s')\right]\right| \leq \Delta_h.$$

The rest of the proof follows Cui et al. [2023]. □

Lemma D.4.3. *(Modified Lemma 40 in Cui et al. [2023]) Under the good event $\mathcal{G}$, for all $k \in [K]$, $h \in [H]$, $i \in [m]$, $s \in \mathcal{S}$, we have*

$$\underline{V}_{h,i}^k(s) \leq V_{h,i}^{\pi^k}(s) + \sum_{h'=h}^{H} \Delta_h.$$

Proof. The proof follows the proof for Lemma D.4.2. □

Lemma D.4.4. *(Modified Lemma 41 in Cui et al. [2023]) Under the good event $\mathcal{G}$, for all $k \in [K]$, $i \in [m]$, we have*

$$\overline{V}_{1,i}^k(s_1) - \underline{V}_{1,i}^k(s_1) \leq \widetilde{O}\left(\mathbb{E}_{\pi^k}^{M^0}\left[\sum_{h=1}^{H} \sqrt{\frac{H^2 A_i T_{\mathrm{Trig}}}{n_h^k(s_h) \vee T_{\mathrm{Trig}}}}\right]\right) + 2\Delta.$$

Proof. The proof follows the proof for Lemma D.4.2. □

Lemma D.4.5. *(Modified Lemma 42 in Cui et al. [2023]) Under the good event $\mathcal{G}$, for all $i \in [m]$, we have*

$$\sum_{k=1}^{K} n^k \max_{i\in[m]} \left(\overline{V}_{1,i}^{k}(s_1) - \underline{V}_{1,i}^{\pi^k}(s_1)\right) \leq \widetilde{O}\left(H^2\sqrt{SA_{\max}T_{\text{Trig}}N}\right).$$

Proof. By Lemma D.4.4 and the proof in Cui et al. [2023], we only need to bound $\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k}^{M^0} \sqrt{\frac{1}{n_h^k(s_h)\vee T_{\text{Trig}}}}$. By the definition of Δ, we can easily prove that

$$\sum_{s\in\mathcal{S}} \left| \frac{n^k}{N^k} \sum_{j=1}^{N^k} d_h^{\pi^k}(s;k+1,j) - \sum_{J=1}^{n^k} d_h^{\pi^k}(s;k,J) \right| \leq n^k\Delta,$$

$$\sum_{s\in\mathcal{S}} \left| n^k d_h^{\pi^k}(s) - \left(\sum_{l=1}^{k} \frac{n^l}{N^k} \sum_{j=1}^{N^k} d_h^{\pi^l}(s;k+1,j) - \sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j) \right) \right| \leq N^k\Delta.$$

and we have

$$\sum_{s\in\mathcal{S}} \left(\sum_{l=1}^{k} \frac{n^l}{N^k} \sum_{j=1}^{N^k} d_h^{\pi^l}(s;k+1,j) - \sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j) \right) - 2\sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j) \leq 4N^k\Delta.$$

Then we have

$$\begin{aligned}
&\sum_{k=1}^{K} n^k \mathbb{E}_{\pi^k}^{M^0} \sqrt{\frac{1}{n_h^k(s_h)\vee T_{\text{Trig}}}} \\
=&\sum_{k=1}^{K} n^k \sum_{s\in\mathcal{S}} d_h^{\pi^k}(s) \sqrt{\frac{1}{n_h^k(s)\vee T_{\text{Trig}}}} \\
\leq&\sum_{s\in\mathcal{S}} \sum_{k=1}^{K} n^k d_h^{\pi^k}(s) \sqrt{\frac{2}{(\sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j)) \vee T_{\text{Trig}}}} \\
&\qquad \text{(Lemma 38 in Cui et al. [2023])} \\
\leq&NK\Delta + \sum_{s\in\mathcal{S}} \sum_{k=1}^{K} \left(\sum_{l=1}^{k} \frac{n^l}{N^k} \sum_{j=1}^{N^k} d_h^{\pi^l}(s;k+1,j) - \sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j) \right) \\
&\sqrt{\frac{2}{(\sum_{l=1}^{k-1} \frac{n^l}{N^{k-1}} \sum_{j=1}^{N^{k-1}} d_h^{\pi^l}(s;k,j)) \vee T_{\text{Trig}}}} \\
\leq&2NK\Delta + \sum_{s\in\mathcal{S}} \sqrt{32 \sum_{l=1}^{K} \frac{n^l}{N^K} \sum_{j=1}^{N^K} d_h^{\pi^l}(s;K,j)} \\
&\qquad \text{(Lemma 38 and Lemma 53 in Cui et al. [2023])}
\end{aligned}$$

$$\leq 2NK\Delta + \sqrt{32SN}.$$

□

Lemma 43 in Cui et al. [2023] holds directly with the modified update rule. As a result, following Theorem 4 in Cui et al. [2023], the same sample complexity result holds for learning an $\epsilon + \widetilde{O}(HS\Delta)$-CCE. Hence $C_1(\epsilon) = H^6S^2A_{\max}\epsilon^{-2}, c_1^{\Delta} = HS$.

D.4.6 Markov Potential Games (NE)

This setting is rather straightforward. Algorithm 3 in Song et al. [2021a] serves as a base algorithm. By noticing that any weighted average of the samples of rewards shifts by no more than $O(\Delta)$ in the non-stationary environment and by the very similar argument we made in Lemma D.2.3 or proof of Theorem 1 in Mao et al. [2021b] we can see $C_1(\epsilon) = m^2H^4SA_{\max}\epsilon^{-3}, c_1^{\Delta} = O(H^2)$.

Appendix E

DEFERRED CONTENTS FROM CHAPTER ??

E.1 Additional Motivating Examples

In this section, we present two additional motivating examples of our proposed models.

Example 6 (**Web Advertisements**). Consider a set of websites as the facility set and companies who want to advertise their products as the players. Due to budget constraints, each company may only choose some of these websites to put its product ad. For each website, the probability that a user will click on a certain ad (and then buy the product) depends on how many ads are put on the website. If a website receives too many ads, the probability that a user can see a certain ad will decrease, thus making it congested.* The reward each company will receive is measured by the amount of products sold during certain period of time, which is bandit feedback.

Example 7 (**Server Usage**). Consider a set of servers in a company as the facility set and server users as the players. Each user needs to request several servers to finish her computation task and the cost triggered from each server depends on the number of users requesting that server. Each user will try to minimize the total cost incurred from the servers she requested. As each user can see the cost from all the servers she requested, this is semi-bandit feedback.

E.2 Compute ϵ-approximate Nash Equilibrium in Potential Games

In this section, we show that the ϵ-NASH$(\cdot)$ operation in Algorithm 10 can be computed efficiently by using Algorithm 11.

In particular, we first show that the matrix game with reward functions $\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot)$ used in Algorithm 10 is a potential game in Lemma E.2.1. Then, we show that Algorithm 11

*Although the website's intelligent recommendation system may more or less mitigate this effect, it can be considered as a part of the reward function's property.

can efficiently compute an ϵ-approximate Nash equilibrium for potential games and output a product policy as shown in Lemma E.2.2.

Lemma E.2.1. *In line 6 of Algorithm 10, the matrix game with reward functions* $\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot)$ *forms a potential game for both settings of semi-bandit feedback and bandit feedback.*

Proof. In the setting of semi-bandit feedback, since $\overline{Q}_i^k(\mathbf{a}) = \sum_{f \in a_i} (\hat{r}^{k,f} + b^{k,f,\mathrm{r}})(\mathbf{a})$, the reward functions $\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot)$ form a congestion game, which we know is a potential game [Monderer and Shapley, 1996].

In the setting of bandit feedback, notice that by defining $\tilde{r}^{k,f}(i) = \widehat{\theta}^k_{i+m(f-1)}$ for $(i,f) \in [m] \times \mathcal{F}$, we can have $\tilde{r}_i^k(\mathbf{a}) = \left\langle A_i(\mathbf{a}), \widehat{\theta}^k \right\rangle = \sum_{f \in a_i} \tilde{r}^{k,f}(n^f(\mathbf{a}))$. Therefore, we claim that the desired potential function is

$$\Phi^k(\mathbf{a}) = \widetilde{\Phi}^k(\mathbf{a}) + \tilde{b}^{k,\mathrm{r}}(\mathbf{a}), \quad \text{where} \quad \widetilde{\Phi}^k(\mathbf{a}) = \sum_{f \in \mathcal{F}} \sum_{i=1}^{n^f(\mathbf{a})} \tilde{r}^{k,f}(i).$$

To see this, by referring to the definition of potential function in congestion game [Monderer and Shapley, 1996], since $\tilde{r}_i^k(\mathbf{a}) = \sum_{f \in a_i} \tilde{r}^{k,f}(n^f(\mathbf{a}))$, we have that

$$\widetilde{\Phi}^k(a_i, a_{-i}) - \widetilde{\Phi}^k(a_i', a_{-i}) = \tilde{r}_i(a_i, a_{-i}) - \tilde{r}_i(a_i', a_{-i}).$$

As a result, we have

$$\begin{aligned}
&\Phi^k(a_i, a_{-i}) - \Phi^k(a_i', a_{-i}) \\
=&\left(\tilde{r}_i(a_i, a_{-i}) + \tilde{b}^{k,\mathrm{r}}(a_i, a_{-i})\right) - \left(\tilde{r}_i(a_i', a_{-i}) + \tilde{b}^{k,\mathrm{r}}(a_i', a_{-i})\right) \\
=&\overline{Q}_i^k(a_i, a_{-i}) - \overline{Q}_i^k(a_i', a_{-i}),
\end{aligned}$$

which means that $\overline{Q}_1^k(\cdot), \ldots, \overline{Q}_m^k(\cdot)$ form a potential game. □

Lemma E.2.2. *Algorithm 11 can output an ϵ-approximate Nash equilibrium.*

Proof. Note that if at round k, we have $\max_{i \in [m]} \Delta_i \leq \epsilon$, then π^k is an ϵ-approximate Nash equilibrium. So we only need to prove that $\max_{i \in [m]} \Delta_i \leq \epsilon$ is satisfied at some round $k \in \{1, \ldots, \lceil \frac{mr_{\max}}{\epsilon} \rceil\}$.

Suppose the potential game $(\{\mathcal{A}_i\}_{i=1}^m, \{r_i\}_{i=1}^m)$ is associated with potential function $\Phi \in [0, \Phi_{\max}]$. Set $\pi^* = \operatorname{argmax}_{\pi \in \prod_{i\in[m]} \Delta(\mathcal{A}_i)} \Phi(\pi)$. Then for any $\pi \in \prod_{i\in[m]} \Delta(\mathcal{A}_i)$, we have

$$\begin{aligned}\Phi(\pi^*) - \Phi(\pi) =& \sum_{i\in[m]} \left(\Phi(\pi^*_{1:i}, \pi_{i+1:m}) - \Phi(\pi^*_{1:i-1}, \pi_{i:m})\right) \\ =& \sum_{i\in[m]} \left(V_i^{\pi^*_{1:i}, \pi_{i+1:m}} - V_i^{\pi^*_{1:i-1}, \pi_{i:m}}\right) \\ \leq& m r_{\max}.\end{aligned}$$

As a result, we can set $\Phi_{\max} = m r_{\max}$. On the other hand, if $j = \operatorname{argmax}_{i\in[m]} \Delta_i$ for round k, we have

$$\begin{aligned}\Phi(\pi^{k+1}) - \Phi(\pi^k) =& \Phi(\pi_j^{k+1}, \pi_{-j}^k) - \Phi(\pi^k) \\ =& V_j^{\pi_j^{k+1}, \pi_{-j}^k} - V_j^{\pi^k} \\ =& r_j(a_j^{k+1}, \pi_{-j}^k) - r_j(\pi^k) && (\pi^k \text{ is deterministic}) \\ =& \Delta_j \\ =& \max_{i\in[m]} \Delta_i.\end{aligned}$$

So there must exist $k \in \{1, \ldots, \lceil \frac{m r_{\max}}{\epsilon} \rceil\}$ such that $\max_{i\in[m]} \Delta_i \leq \epsilon$, otherwise $\Phi(\pi^k)$ increase at least ϵ at each round, which contradicts $\Phi \in [0, m r_{\max}]$. □

E.3 Analysis for Algorithm 10

Recall that the update rule in Algorithm 10 is $\overline{Q}_i^k(\mathbf{a}) = \hat{r}_i^k(\mathbf{a}) + b_i^{k,\mathrm{r}}(\mathbf{a})$, where we have

$$b_i^{k,\mathrm{r}}(\mathbf{a}) = \sum_{f\in a_i} b^{k,f,\mathrm{r}}(\mathbf{a}), \quad \text{and} \quad b^{k,f,\mathrm{r}}(\mathbf{a}) = \sqrt{\frac{\tilde{\iota}}{N^{k,f}(n^f(\mathbf{a})) \vee 1}}.$$

For proof convenience, we define auxiliary value functions

$$\begin{aligned}&\underline{Q}_i^k(\boldsymbol{a}) = \hat{r}_i^k(\mathbf{a}) - b_i^{k,\mathrm{r}}(\boldsymbol{a}), \\ &\overline{V}_i^k = \mathbb{E}_{\mathbf{a}\sim\pi^k}[\overline{Q}_i^k(\mathbf{a})] \quad \text{and} \quad \underline{V}_i^k = \mathbb{E}_{\mathbf{a}\sim\pi^k}[\underline{Q}_i^k(\mathbf{a})].\end{aligned}$$

With these definitions, we now begin to prove Theorem 6.4.1.

Proof of Theorem 6.4.1. **Semi-bandit Feedback.** By the update rules in Algorithm 10, in the setting of semi-bandit feedback, with probability at least $1-\delta$, simultaneously for all $(k,i,\mathbf{a}) \in [K] \times [m] \times \mathcal{A}$, we have

$$\overline{Q}_i^k(\mathbf{a}) - r_i(\mathbf{a}) = \sum_{f \in a_i} \left[(\hat{r}^{k,f} - r^f)(\mathbf{a}) + b^{k,f,\mathrm{r}}(\mathbf{a}) \right] \geq 0.$$

The second inequality above is obtained by using standard Hoeffding's inequality and union bound, Therefore, we have $\overline{Q}_i^k(\mathbf{a}) \geq r_i(\mathbf{a})$.

Then, since π^k is the ϵ-approximate Nash equilibrium policy of $\overline{Q}_1^k, \ldots, \overline{Q}_m^k$, we have

$$\begin{aligned}
\overline{V}_i^k = \mathbb{E}_{\mathbf{a}\sim\pi^k}[\overline{Q}_i^k(\mathbf{a})] &= \max_{\nu \in \Delta(\mathcal{A}_i)} \mathbb{E}_{\mathbf{a}\sim(\nu,\pi_{-i}^k)}[\overline{Q}_i^k(\mathbf{a})] - \epsilon \\
&\geq \max_{\nu \in \Delta(\mathcal{A}_i)} \mathbb{E}_{\mathbf{a}\sim(\nu,\pi_{-i}^k)}[r_i(\mathbf{a})] - \epsilon = V_i^{\dagger,\pi_{-i}^k} - \epsilon.
\end{aligned}$$

Meanwhile, by definition of $\underline{Q}_i^k(\mathbf{a})$ and $\underline{V}_i^k$, we can similarly show that $\underline{Q}_i^k(\mathbf{a}) \leq r_i(\mathbf{a})$ and $\underline{V}_i^k \leq V_i^{\pi^k}$. Therefore, we can have $V_i^{\dagger,\pi_{-i}^k} - V_i^{\pi^k} \leq \overline{V}_i^k - \underline{V}_i^k + \epsilon$.

Now, we define $\widetilde{Q}^k(\mathbf{a}) = \max_{i\in[m]} 2b_i^{k,\mathrm{r}}(\mathbf{a})$ and $\widetilde{V}^k = \mathbb{E}_{\mathbf{a}\sim\pi^k}[\widetilde{Q}^k(\mathbf{a})]$. Then, we can notice that

$$\max_{i\in[m]}(\overline{Q}_i^k - \underline{Q}_i^k)(\mathbf{a}) \leq \max_{i\in[m]} 2b_i^{k,\mathrm{r}}(\mathbf{a}) = \widetilde{Q}^k(\mathbf{a}),$$

$$\max_{i\in[m]}(\overline{V}_i^k - \underline{V}_i^k) \leq \mathbb{E}_{\mathbf{a}\sim\pi^k}\left[\max_{i\in[m]}(\overline{Q}_i^k - \underline{Q}_i^k)(\mathbf{a})\right] \leq \mathbb{E}_{\mathbf{a}\sim\pi^k}[\widetilde{Q}^k(\mathbf{a})] = \widetilde{V}^k.$$

We further define $\mathcal{M}^k = \mathbb{E}_{\mathbf{a}\sim\pi^k}\left[\widetilde{Q}^k(\mathbf{a})\right] - \widetilde{Q}^k(\mathbf{a}^k) = \widetilde{V}^k - \widetilde{Q}^k(\mathbf{a}^k)$. It is not hard to verify that $\mathcal{M}^k$ is a martingale difference sequence with respect to the history from episode 1 to $k-1$. Meanwhile, since $\left|b^{k,\mathrm{r}}(\mathbf{a})\right| = \sum_{f\in\mathcal{F}} \sqrt{\frac{\tilde{\iota}}{N^{k,f}(n^f(\mathbf{a}))\vee 1}} \leq F\sqrt{\tilde{\iota}}$. Thus, by Azuma-Hoeffding inequality, we have $\sum_{k=1}^K \mathcal{M}^k = \widetilde{\mathcal{O}}\left(F\sqrt{K}\right)$. Therefore, we have

$$\begin{aligned}
\text{Nash-Regret}(K) &= \sum_{k=1}^K \max_{i\in[m]} \left(V_i^{\dagger,\pi_{-i}^k} - V_i^{\pi^k}\right) \\
&= \sum_{k=1}^K \min\left\{\max_{i\in[m]} \left(V_i^{\dagger,\pi_{-i}^k} - V_i^{\pi^k}\right), F\right\} \\
&\qquad\qquad \text{(Since the value is always bounded by } F.\text{)} \\
&\leq \sum_{k=1}^K \min\left\{\max_{i\in[m]} \left(\overline{V}_i^k - \underline{V}_i^k\right), F\right\} + K\epsilon
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{k=1}^{K} \min\left\{\widetilde{V}^k, F\right\} + K\epsilon \\
&= \sum_{k=1}^{K} \left(\min\left\{\widetilde{Q}^k(\mathbf{a}^k), F\right\} + \mathcal{M}^k\right) + K\epsilon \\
&\leq \widetilde{\mathcal{O}}\left(F\sqrt{K}\right) + 2\sum_{k=1}^{K}\left\{\max_{i\in[m]} b_i^{k,\mathrm{r}}(\mathbf{a}^k), F\right\} && \text{(By taking } \epsilon = 1/K.\text{)} \\
&\leq \widetilde{\mathcal{O}}\left(F\sqrt{K}\right) + 2\sum_{f\in\mathcal{F}}\sum_{k=1}^{K}\sqrt{\frac{\tilde{\iota}}{N^{k,f}(n^f(\mathbf{a}^k))\vee 1}} \\
&\leq \widetilde{\mathcal{O}}\left(F\sqrt{mK}\right) && \text{(By Lemma E.3.4.)}
\end{aligned}
$$

Bandit Feedback. By using Lemma E.3.1, which guarantees optimistic estimation, we can similarly show that

$$
\text{Nash-Regret}(K) \leq \sum_{k=1}^{K}\mathcal{M}^k + \sum_{k=1}^{K}\min\left\{2\tilde{b}^{k,\mathrm{r}}(\mathbf{a}^k), F\right\} + K\epsilon.
$$

To have an upper bound on $\mathcal{M}^k$ here, recall that $\tilde{b}^{k,\mathrm{r}}(\mathbf{a}) = \max_{i\in[m]}\|A_i(\mathbf{a})\|_{(V^k)^{-1}}\sqrt{\tilde{\beta}_k}$ and $\sqrt{\tilde{\beta}_K} = \widetilde{\mathcal{O}}\left(\sqrt{F\tilde{d}}\right) = \widetilde{\mathcal{O}}\left(F\sqrt{m}\right)$. Meanwhile, we have $\|A_i(\mathbf{a})\|_{(V^k)^{-1}} \leq \|A_i(\mathbf{a})\|_I = \|A_i(\mathbf{a})\|_2 \leq \sqrt{F}$. Thus, we have $\left|\mathcal{M}^k\right| \leq \widetilde{\mathcal{O}}\left(\sqrt{mF^3}\right)$, which by Azuma-Hoeffding inequality implies $\sum_{k=1}^{K}\mathcal{M}^k = \widetilde{\mathcal{O}}\left(\sqrt{mF^3K}\right)$.

Then the sum of the bonus terms can be bounded by using Lemma E.3.2. In particular, with $\epsilon = 1/K$, we have

$$
\begin{aligned}
\text{Nash-Regret}(K) \leq& \widetilde{\mathcal{O}}\left(\sqrt{mF^3K}\right) + 2\sum_{k=1}^{K}\min\left\{\max_{i\in[m]}\left\|A_i(\mathbf{a}^k)\right\|_{(V^k)^{-1}}\sqrt{\tilde{\beta}_k}, F\right\} \\
\leq& \widetilde{\mathcal{O}}\left(\sqrt{mF^3K}\right) + 2\sqrt{K\sum_{k=1}^{K}\min\left\{\max_{i\in[m]}\|A_i(\mathbf{a}^k)\|^2_{(V^k)^{-1}}\tilde{\beta}_k, F^2\right\}} \\
\leq& \widetilde{\mathcal{O}}\left(\sqrt{mF^3K}\right) + \sqrt{\widetilde{\mathcal{O}}\left(mF^2K\right)\sum_{k=1}^{K}\min\left\{\max_{i\in[m]}\|A_i(\mathbf{a}^k)\|^2_{(V^k)^{-1}}, 1\right\}} \\
& && \text{(Since } \tilde{\beta}_k = \widetilde{\mathcal{O}}\left(mF^2\right).\text{)} \\
\leq& \widetilde{\mathcal{O}}\left(\sqrt{mF^3K}\right) + \widetilde{\mathcal{O}}\left(\sqrt{mF^2K\cdot mF}\right) && \text{(By Lemma E.3.2.)} \\
\leq& \widetilde{\mathcal{O}}\left(mF^{3/2}\sqrt{K}\right).
\end{aligned}
$$

□

E.3.1 Lemmas for Bandit Feedback

The following lemma, as a direct corollary of the confidence bound for least square estimators, shows that the reward estimation error can be bounded by the reward bonus term.

Lemma E.3.1. *With probability at least $1-\delta$, simultaneously for all $(i,k,\mathbf{a})$, it holds that $|(\tilde{r}_i^k - r_i)(\mathbf{a})| \le \tilde{b}^{k,\mathrm{r}}(\mathbf{a})$, where $\tilde{r}_i^k$ and $\tilde{b}^{k,\mathrm{r}}$ are defined in (6.2).*

Proof. By construction, we have

$$\begin{aligned}
|(\tilde{r}_i^k - r_i)(\mathbf{a})| &= \left|\left\langle A_i(\mathbf{a}), \widehat{\theta} - \theta \right\rangle\right| \\
&\le \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \left\|\widehat{\theta} - \theta\right\|_{V^k} \\
&\overset{\text{(i)}}{\le} \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \left(\|\theta\|_2 + \sqrt{F\log\left(\det(V^k)\right) + F\tilde{\iota}}\right),
\end{aligned}$$

where the inequality (i) above holds because of Theorem 20.5 in Lattimore and Szepesvári [2020] and the fact that the reward noise is $\sqrt{F}$-subGaussian. Since each element in θ is bounded in $[0,1]$ by construction, we have $\|\theta\|_2 \le \sqrt{\tilde{d}}$.

Then, by Lemma E.3.2, we have $\det\left(V^k\right) \le \left(1 + \frac{mkF}{\tilde{d}}\right)^{\tilde{d}}$ since by construction $\|A_i(\mathbf{a})\|_2^2 \le F$.

Finally, to make this bound valid for all player $i \in [m]$, we only need to take maximization over $i \in [m]$. Therefore, with probability at least $1-\delta$, we have

$$|(\tilde{r}_i^k - r_i)(\mathbf{a})| \le \max_{i\in[m]} \|A_i(\mathbf{a})\|_{(V^k)^{-1}} \sqrt{\tilde{\beta}_k} = \tilde{b}^{k,\mathrm{r}}(\mathbf{a}),$$

where $\sqrt{\tilde{\beta}_k} = \sqrt{\tilde{d}} + \sqrt{F\tilde{d}\log\left(1 + \frac{mkF}{\tilde{d}}\right) + F\tilde{\iota}}$. □

The following is a variant of the famous elliptical potential lemma, which helps bound the sum of reward bonus under bandit feedback. Here, we apply some techniques from the proof of Lemma 19.4 in Lattimore and Szepesvári [2020].

Lemma E.3.2. *Let $K, m \ge 1$ be integers. Suppose $V^k = I + \sum_{k'=1}^{k-1}\sum_{i=1}^{m} A_i^{k'}\left(A_i^{k'}\right)^\top$, where $A_i^{k'} \in \mathbb{R}^d$ and $\left\|A_i^{k'}\right\|_2^2 \le F$. Then, it holds that*

$$\det\left(V^k\right) \le \left(1 + \frac{mkF}{d}\right)^d, \quad \text{and} \quad \sum_{k=1}^{K} \min\left\{\max_{i\in[m]} \left\|A_i^k\right\|^2_{(V^k)^{-1}}, 1\right\} \le 2d\log\left(1 + \frac{mKF}{d}\right).$$

Proof. For the first upper bound about $\det\left(V^k\right)$, we have

$$\begin{aligned}
\det\left(V^k\right) &= \prod_{j=1}^{d} \lambda_j && (\lambda_1, \ldots, \lambda_d \text{ are eigenvalues of } V^k) \\
&\leq \left(\frac{\operatorname{tr}\left(V^k\right)}{d}\right)^d && \text{(By AM-GM inequality)} \\
&= \left(\frac{\operatorname{tr}\left(I\right) + \sum_{k'=1}^{k-1}\sum_{i=1}^{m}\left\|A_i^{k'}\right\|_2^2}{d}\right)^d \\
&\leq \left(1 + \frac{mkF}{d}\right)^d. && (\text{Since } \left\|A_i^{k'}\right\|_2^2 \leq F.)
\end{aligned}$$

For the second upper bound. First, we notice that $\min\{1, x\} \leq 2\log(1+x)$ for any $x \geq 0$. Thus, we have

$$\sum_{k=1}^{K} \min\left\{1, \max_{i\in[m]} \left\|A_i^k\right\|_{\left(V^k\right)^{-1}}^2\right\} \leq 2\sum_{k=1}^{K} \log\left(1 + \max_{i\in[m]} \left\|A_i^k\right\|_{\left(V^k\right)^{-1}}^2\right).$$

Then, for $k \geq 2$, we can notice that

$$\begin{aligned}
V^k =& V^{k-1} + \sum_{i=1}^{m} A_i^{k-1}\left(A_i^{k-1}\right)^\top \\
=& \left(V^{k-1}\right)^{1/2}\left(I + \left(V^{k-1}\right)^{-1/2}\left(\sum_{i=1}^{m} A_i^{k-1}\left(A_i^{k-1}\right)^\top\right)\left(V^{k-1}\right)^{-1/2}\right)\left(V^{k-1}\right)^{1/2} \\
=& \left(V^{k-1}\right)^{1/2}\left(I + \sum_{i=1}^{m}\left(\left(V^{k-1}\right)^{-1/2} A_i^{k-1}\right)\left(\left(V^{k-1}\right)^{-1/2} A_i^{k-1}\right)^\top\right)\left(V^{k-1}\right)^{1/2}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\det\left(V^k\right) =& \det\left(V^{k-1}\right)\det\left(I + \sum_{i=1}^{m}\left(\left(V^{k-1}\right)^{-1/2} A_i^{k-1}\right)\left(\left(V^{k-1}\right)^{-1/2} A_i^{k-1}\right)^\top\right) \\
\geq& \det\left(V^{k-1}\right)\left(1 + \max_{i\in[m]} \left\|A_i^{k-1}\right\|_{\left(V^{k-1}\right)^{-1}}^2\right) && \text{(By Lemma E.3.3.)} \\
\geq& \prod_{k'=1}^{k-1}\left(1 + \max_{i\in[m]} \left\|A_i^{k'}\right\|_{\left(V^{k'}\right)^{-1}}^2\right). && (\text{Since by definition, } V^1 = I.)
\end{aligned}$$

As a result, we have

$$\sum_{k=1}^{K} \min\left\{\max_{i\in[m]} \left\|A_i^k\right\|_{\left(V^k\right)^{-1}}^2, 1\right\} \leq 2\sum_{k=1}^{K} \log\left(1 + \max_{i\in[m]} \left\|A_i^k\right\|_{\left(V^k\right)^{-1}}^2\right)$$

$$\leq 2\log\left(\det\left(V^{K+1}\right)\right)$$
$$\leq 2d\log\left(1+\frac{mKF}{d}\right).$$

□

E.3.2 Technical Lemmas

Lemma E.3.3. *Let $y_1, \ldots, y_m \in \mathbb{R}^d$ be a set of vectors. Then, it holds that*

$$\det\left(I+\sum_{i=1}^{m} y_i y_i^\top\right) \geq 1+\max_{i\in[m]}\|y_i\|_2^2.$$

Proof. Since $I+\sum_{i=1}^{m} y_i y_i^\top \succeq I+y_i y_i^\top$ for any $i\in[m]$, we have $\det\left(I+\sum_{i=1}^{m} y_i y_i^\top\right) \geq \det\left(I+y_i y_i^\top\right)$ for any $i\in[m]$. That is, we have

$$\det\left(I+\sum_{i=1}^{m} y_i y_i^\top\right) \geq \max_{i\in[m]}\det\left(I+y_i y_i^\top\right) = 1+\max_{i\in[m]}\|y_i\|_2^2.$$

The last line above holds because the matrix $I+y_i y_i^\top$ has eigenvalues $1+\|y_i\|_2^2$ and 1. □

Lemma E.3.4. *For any $f\in\mathcal{F}$, it holds that*

$$\sum_{k=1}^{K}\sqrt{\frac{1}{\mathcal{N}^{k,f}(n^f(\mathbf{a}^k))\vee 1}} \leq \widetilde{\mathcal{O}}\left(\sqrt{mK}\right).$$

Proof. Here, we have

$$\begin{aligned}
\sum_{k=1}^{K}\sqrt{\frac{1}{N^{k,f}(n^f(\mathbf{a}^k))\vee 1}} =& \sum_{n=0}^{m}\sum_{\ell=1}^{N^{K,f}(n)}\sqrt{\frac{1}{\ell}} \\
\leq& 2\sum_{n=0}^{m}\sqrt{N^{K,f}(n)} && \text{(By standard technique)} \\
\leq& 2\sqrt{(m+1)\sum_{n=0}^{m}N^{K,f}(n)} \\
=& \widetilde{\mathcal{O}}\left(\sqrt{mK}\right).
\end{aligned}$$

The last equality above is based on a pigeon-hold principle argument similar to Lemma E.6.5. □

E.4 Analysis for Algorithm 12

E.4.1 Exploration Distribution and Smoothness

We choose the exploration distribution to be the G-optimal design and we have the following properties.

Lemma E.4.1. *(Unbiasedness) For any episode $k \in [K]$, $i \in [m]$ and $a \in \mathcal{A}_i$, we have*

$$\mathbb{E}_k\left[\widehat{\nabla}_i^k \Phi(a)\right] = \nabla_i^k \Phi(a),$$

where $\mathbb{E}_k[\cdot]$ is taken over all the randomness before episode k.

Proof. By the definition of $\widehat{\nabla}_i^k \Phi(a)$, we have

$$\begin{aligned}
\mathbb{E}_k\left[\widehat{\nabla}_i^k \Phi(a)\right] =& \mathbb{E}_k\left\langle \phi_i(a), \widehat{\theta}_i^k(\pi^k)\right\rangle \\
=& \mathbb{E}_k\left[\frac{1}{\tau}\sum_{t=1}^{\tau} \phi_i(a)^\top [\Sigma_i^k]^{-1} \phi_i(a_i^{k,t}) r_i^{k,t}\right] \\
=& \mathbb{E}_k\left[\phi_i(a)^\top [\Sigma_i^k]^{-1} \phi_i(a_i^{k,1}) r_i^{k,1}\right] \\
=& \mathbb{E}_k\left[\phi_i(a)^\top [\Sigma_i^k]^{-1} \phi_i(a_i^{k,1}) \phi_i(a_i^{k,1})^\top \theta_i^{k,1}(\pi^k)\right] \\
=& \sum_{a_i^k \in \mathcal{A}_i} \pi_i^k(a_i^{k,1}) \phi_i^\top(a) [\Sigma_i^k]^{-1} \phi_i(a_i^{k,1}) \phi_i(a_i^{k,1})^\top \theta_i(\pi^k) \\
& \qquad (a_i^{k,1} \text{ only depends on } \pi_i^k \text{ and } \theta_i^{k,1}(\pi^k) \text{ only depends on } \pi_{-i}^k) \\
=& \phi_i^\top(a) [\Sigma_i^k]^{-1} \left[\sum_{a_i^k \in \mathcal{A}_i} \pi_i^k(a_i^{k,1}) \phi_i(a_i^{k,1}) \phi_i(a_i^{k,1})^\top\right] \theta_i(\pi^k) \\
=& \phi_i^\top(a_i) \theta_i(\pi^k) \\
=& \nabla_i^k \Phi(a).
\end{aligned}$$

□

Lemma E.4.2. *For any episode $k \in [K]$, $i \in [m]$ and $a \in \mathcal{A}_i$, we have*

$$\left|\phi_i(a)^\top [\Sigma_i^k]^{-1} \phi_i(a_i^{k,t}) r_i^{k,t}\right| \le \frac{F^2}{\gamma}.$$

Proof. As $\pi_i^k = (1-\gamma)(\nu\widetilde{\pi}_i^k + (1-\gamma)\pi_i^{k-1}) + \gamma\rho_i$, we have

$$\Sigma_i^k = \mathbb{E}_{a_i\sim\pi_i^k}\phi_i(a_i)\phi_i(a_i)^\top \succeq \gamma\mathbb{E}_{a_i\sim\rho_i}\phi_i(a_i)\phi_i(a_i)^\top,$$

and ρ_i is the G-optimal design with respect to $\phi_i(\cdot)$, for any action $a \in \mathcal{A}_i$ we have

$$\|\phi_i(a)\|^2_{[\Sigma_i^k]^{-1}} \le \frac{1}{\gamma}\|\phi_i(a)\|^2_{[\mathbb{E}_{a_i\sim\rho_i}\phi_i(a_i)\phi_i(a_i)^\top]^{-1}} \le \frac{F}{\gamma}.$$

Then for any $t \in [\tau]$, since $|r_i^{k,t}| \le F$, we have

$$\left|r_i^{k,t}\phi_i^\top(a)[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})\right| \le \left|r_i^{k,t}\right|\|\phi_i(a)\|_{[\Sigma_i^k]^{-1}}\left\|\phi_i(a_i^{k,t})\right\|_{[\Sigma_i^k]^{-1}} \le \frac{F^2}{\gamma}.$$

As a result, we have

$$\left|\widehat{\nabla}_i^k\Phi(a)\right| = \left|\frac{1}{\tau}\sum_{t=1}^{\tau}\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})r_i^{k,t}\right| \le \frac{F^2}{\gamma}$$

□

Lemma E.4.3. *For any episode $k \in [K]$, $i \in [m]$ and $a \in \mathcal{A}_i$, we have*

$$\mathbb{E}_k\left[\left(\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})r_i^{k,t}\right)^2\right] \le \frac{F^3}{\gamma}.$$

Proof. We first show that for any $t \in [\tau]$, we have

$$\begin{aligned}
&\mathbb{E}_k\left[\left(\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})r_i^{k,t}\right)^2\right]\\
\le& F^2\mathbb{E}_k\left[\left(\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})\right)^2\right]\\
\le& F^2\mathbb{E}_k\left[\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a_i^{k,t})\phi_i(a_i^{k,t})^\top[\Sigma_i^k]^{-1}\phi_i(a)^\top\right]\\
=& F^2\phi_i(a)^\top[\Sigma_i^k]^{-1}\phi_i(a)\\
\le& \frac{F^3}{\gamma}.
\end{aligned}$$

□

Lemma E.4.4. *With probability $1-\delta$, for all $k \in [K]$, $i \in [m]$ and $a \in \mathcal{A}_i$, we have*

$$\left|\widehat{\nabla}_i^k\Phi(a) - \nabla_i^k\Phi(a)\right| \le c\sqrt{\frac{F^4\log(mK/\delta)}{\gamma\tau}} + \frac{cF^3\log(mK/\delta)}{\gamma\tau}$$

Proof. Recall that

$$\widehat{\nabla}_i^k \Phi(a_i) = \frac{1}{\tau} \sum_{t=1}^{\tau} \phi_i^\top(a_i)[\Sigma_i^k]^{-1} r_i^{k,t} \phi_i(a_i^{k,t}),$$

and $(a_i^{k,t}, r_i^{k,t})$ are drawn independently at each $t \in [\tau]$. Lemma E.4.1 shows that $\widehat{\nabla}_i^k \Phi(a_i)$ is an unbiased estimate of $\nabla_i^k \Phi(a_i)$ In addition, Lemma E.4.2 shows that $\phi_i^\top(a_i)[\Sigma_i^k]^{-1} r_i^{k,t} \phi_i(a_i^{k,t})$ is bounded by F^2/γ and Lemma E.4.3 shows that its second moment is bounded by F^3/γ. Then by Bernstein's inequality, for a fixed $k \in [K]$, $i \in [m]$ and $a \in \mathcal{A}_i$, with probability $1 - \delta$, we have

$$\left| \widehat{\nabla}_i^k \Phi(a) - \nabla_i^k \Phi(a) \right| \leq \sqrt{\frac{2F^3 \log(2/\delta)}{\gamma\tau}} + \frac{3F^2 \log(2/\delta)}{2\gamma\tau}.$$

The argument holds by applying the union bound and the fact that $|\mathcal{A}_i| \leq 2^F$.

□

Lemma E.4.5. *$\Phi(\cdot)$ is mF-Lipschitz and mF-smooth with respect to the L1 norm $\|\cdot\|_1$.*

Proof. Recall that $\Phi(\pi) = \mathbb{E}_{\mathbf{a}\sim\pi} \Phi(\mathbf{a})$ and $\Phi(\mathbf{a}) \in [0, mF]$.

$$\begin{aligned}
\Phi(\pi) - \Phi(\pi') =& \mathbb{E}_{\mathbf{a}\sim\pi} \Phi(\mathbf{a}) - \mathbb{E}_{\mathbf{a}\sim\pi'} \Phi(\mathbf{a}) \\
=& \sum_{i\in[m]} \mathbb{E}_{a_{1:i-1}\sim\pi'_{1:i-1}, a_{i:m}\sim\pi_{i:m}} \Phi(\mathbf{a}) - \mathbb{E}_{a_{1:i}\sim\pi'_{1:i}, a_{i+1:m}\sim\pi_{i+1:m}} \Phi(\mathbf{a}) \\
\leq& \sum_{i\in[m]} \left\|\pi_i - \pi'_i\right\|_1 \cdot \|\Phi\|_\infty \\
\leq& mF \left\|\pi - \pi'\right\|_1.
\end{aligned}$$

Similarly we have $\nabla_\pi \Phi(a_i) = \mathbb{E}_{a_{-i}\sim\pi_{-i}} \Phi(a_i, a_{-i})$. As a result, we have

$$\|\nabla_\pi \Phi - \nabla_{\pi'} \Phi\|_\infty \leq mF \left\|\pi - \pi'\right\|_1.$$

□

Definition E.4.6. (Frank Wolfe Gap) The Frank Wolfe gap of a joint strategy π for $\Phi(\cdot)$ is defined as

$$G(\pi) = \max_{\pi'} \left\langle \pi' - \pi, \nabla_\pi \Phi \right\rangle.$$

Lemma E.4.7. *Suppose the Frank Wolfe gap of π is ϵ. Then π is an ϵ-Nash policy.*

Proof. For a fixed player i, suppose player i change her strategy to π_i'.

$$\begin{aligned} V_i^{\pi_i', \pi_{-i}} - V_i^{\pi} &= \Phi(\pi_i', \pi_{-i}) - \Phi(\pi) \\ &= \left\langle \pi_i' - \pi_i, \nabla_{\pi_i} \Phi \right\rangle \\ &\leq \max_{\pi'} \left\langle \pi' - \pi, \nabla_{\pi} \Phi \right\rangle \\ &\leq \epsilon. \end{aligned}$$

□

E.4.2 Analysis for Frank Wolfe in Bandit Feedback

Theorem E.4.8. *Let $T = K\tau$. For the congestion game with bandit feedback, by running Algorithm 12 with gradient estimator $\widehat{\nabla}_i^k \Phi$ in (6.4) and exploration distribution ρ_i in (6.5), setting parameters $\nu = \frac{F}{m\sqrt{K}}$, $\gamma = \frac{F}{mK}$ and $\tau = K^2$, if $K \geq \frac{2F}{m}$, then with probability $1-\delta$, we have*

$$\textit{Nash-Regret}(T) = \tau \sum_{k=1}^{K} G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2 F^2 T^{5/6} + m^3 F^3 T^{2/3}\right).$$

Proof. Set $\nabla^k \Phi = \nabla \Phi(\Pi^k) \in \mathbb{R}^A$ and $\nabla_i^k \Phi = \nabla^k \Phi(\pi_i) \in \mathbb{R}^{A_i}$. As we have $\Phi(\cdot)$ is mF-smooth w.r.t. $\|\cdot\|_1$, we have

$$\begin{aligned} \Phi(\pi^{k+1}) \geq & \Phi(\pi^k) + \left\langle \nabla \Phi(\pi^k), \pi^{k+1} - \pi^k \right\rangle - \frac{mF}{2} \|\pi^{k+1} - \pi^k\|_1^2 \\ = & \Phi(\pi^k) + (1-\gamma)\nu \left\langle \nabla \Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle + \gamma \left\langle \nabla^k \Phi, \rho - \pi^k \right\rangle \\ & - \frac{mF}{2} (2\nu^2 \left\| \widetilde{\pi}^k - \pi^k \right\|_1^2 + 2\gamma^2 \left\| \rho - \pi^k \right\|_1^2) \\ \geq & \Phi(\pi^k) + (1-\gamma)\nu \left\langle \nabla \Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle - \gamma \left\| \nabla^k \Phi \right\|_\infty \left\| \rho - \pi^k \right\|_1 \\ & - \frac{mF}{2} (2\nu^2 \left\| \widetilde{\pi}^k - \pi^k \right\|_1^2 + 2\gamma^2 \left\| \rho - \pi^k \right\|_1^2) \\ \geq & \Phi(\pi^k) + (1-\gamma)\nu \left\langle \nabla \Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle - 2\gamma m^2 F - 4m^3 F(\nu^2 + \gamma^2). \end{aligned}$$

(By Lemma E.4.5.)

Define the true target policy at episode k

$$\widehat{\pi}_i^{k+1} = \operatorname*{argmax}_{\pi_i} \left\langle \pi_i, \nabla_i \Phi(\pi_i^k) \right\rangle,$$

and the Frank Wolfe gap of joint strategy π

$$G(\pi) = \max_{\pi'} \left\langle \pi' - \pi, \nabla\Phi(\pi) \right\rangle.$$

Then we have

$$\begin{aligned}
\left\langle \nabla\Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle =& \left\langle \widehat{\nabla}^k\Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle + \left\langle \nabla\Phi(\pi^k) - \widehat{\nabla}^k\Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle \\
\geq& \left\langle \widehat{\nabla}^k\Phi(\pi^k), \widehat{\pi}^{k+1} - \pi^k \right\rangle + \left\langle \nabla\Phi(\pi^k) - \widehat{\nabla}^k\Phi(\pi^k), \widetilde{\pi}^{k+1} - \pi^k \right\rangle \\
=& \left\langle \nabla\Phi(\pi^k), \widehat{\pi}^{k+1} - \pi^k \right\rangle + \left\langle \nabla\Phi(\pi^k) - \widehat{\nabla}^k\Phi(\pi^k), \widetilde{\pi}^{k+1} - \widehat{\pi}^{k+1} \right\rangle \\
\geq& G(\pi^k) - 2m \left\| \nabla\Phi(\pi^k) - \widehat{\nabla}^k\Phi(\pi^k) \right\|_\infty \\
\geq& G(\pi^k) - c\sqrt{\frac{m^2F^4\log(mK/\delta)}{\gamma\tau}} - \frac{cmF^3\log(mK/\delta)}{\gamma\tau}
\end{aligned}$$

Apply it to the previous bound and we have

$$\begin{aligned}
\Phi(\pi^{k+1}) \geq& \Phi(\pi^k) + (1-\gamma)\nu G(\pi^k) - c\frac{(1-\gamma)\nu}{\sqrt{\gamma\tau}}\sqrt{m^2F^4\log(mK/\delta)} \\
& - c\frac{(1-\gamma)\nu}{\gamma\tau}mF^3\log(mK/\delta) - \gamma 2m^2F - 4m^3F(\nu^2+\gamma^2).
\end{aligned}$$

Summing over $k \in [K]$ and we get

$$\begin{aligned}
\sum_{k=1}^K G(\pi^k) \leq& \frac{\Phi(\pi^{K+1}) - \Phi(\pi^1)}{(1-\gamma)\nu} + c\frac{K}{\sqrt{\gamma\tau}}\sqrt{m^2F^4\log(mK/\delta)} + c\frac{K}{\gamma\tau}mF^3\log(mK/\delta) \\
& + \frac{2m^2FK\gamma}{(1-\gamma)\nu} + \frac{4(\nu^2+\gamma^2)m^3FK}{(1-\gamma)\nu}.
\end{aligned}$$

Set $\nu = \frac{F}{m\sqrt{K}}$, $\gamma = \frac{F}{mK}$, $\tau = K^2$ and notice that when $K \geq \frac{2F}{m}$, we have $1-\gamma \geq \frac{1}{2}$. Since $\Phi(\cdot)$ is bounded in $[0, mF]$, we can have

$$\sum_{k=1}^K G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2F^2K^{1/2} + m^3F^3\right).$$

Then by Lemma E.4.7, for $T = K\tau$, we have

$$\text{Nash-Regret}(T) = \tau\sum_{k=1}^K G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2F^2T^{5/6} + m^3F^3T^{2/3}\right).$$

□

E.4.3 Algorithm and Analysis for Semi-bandit Feedback

In the setting of semi-bandit feedback, we will need a different gradient estimator $\widetilde{\nabla}_i^k \Phi(a_i)$ and a different exploration distribution $\widetilde{\rho}_i$ to utilize the extra reward information from each chosen facility.

Based on the analysis in Section 6.5, using (6.3), we have $\nabla_i^k \Phi(a_i) = \sum_{f \in a_i} [\theta_i(\pi^k)]_f$, where $[\theta_i(\pi^k)]_f = \mathbb{E}_{a_{-i} \sim \pi_{-i}^k} \left[r^f(n^f(a_{-i}) + 1) \right]$. Meanwhile, in semi-bandit feedback, the mean of t-th reward player i received for facility f at episode k is $r^f(n^f(a_i^{k,t}, a_{-i}^{k,t}))$. Therefore, we can use inverse propensity score (IPS) estimator to estimate $[\theta_i(\pi^k)]_f$. In particular, we have

$$[\widetilde{\theta}_i^k(\pi^k)]_f = \frac{1}{\tau} \sum_{t=1}^{\tau} [\widetilde{\theta}_i^{k,t}(\pi^k)]_f, \quad \text{where} \quad [\widetilde{\theta}_i^{k,t}(\pi^k)]_f = \frac{r^{k,t,f} \mathbb{1}\left\{ f \in a_i^{k,t} \right\}}{\mathbb{P}_{a_i \sim \pi_i^k}(f \in a_i)}.$$

Then, we can naturally have

$$\widetilde{\nabla}_i^k \Phi(a_i) = \sum_{f \in a_i} [\widetilde{\theta}_i^k(\pi^k)]_f. \tag{E.1}$$

Furthermore, by Lemma E.4.11, we can see that by using $\tilde{\rho}_i$ computed by Algorithm 24, for all players, we have $\mathbb{P}_{a_i \sim \pi_i^k}(f \in a_i) \geq \frac{\gamma}{2F}$ for all $f \in \bigcup_{a_i \in \mathcal{A}_i} a_i$.

Properties of the IPS estimator are summarized in Lemma E.4.12. By using these properties, we can have the following lemma.

Lemma E.4.9. *With probability $1 - \delta$, for all $k \in [K]$, $i \in [m]$ and $a_i \in \mathcal{A}_i$, we have*

$$\left| \widetilde{\nabla}_i^k \Phi(a_i) - \nabla_i^k \Phi(a_i) \right| \leq \sqrt{\frac{4F^3 \log(2mFK/\delta)}{\gamma\tau}} + \frac{2F^2 \log(2mFK/\delta)}{\gamma\tau}.$$

Proof. By Lemma E.4.12 and Bernstein's inequality, simultaneously for all $(i, k, f) \in [m] \times [K] \times \mathcal{F}$, with probability at least $1 - \delta$, we have

$$\left| [\widetilde{\theta}_i^k(\pi^k)]_f - [\theta_i(\pi^k)]_f \right| \leq \sqrt{\frac{4F \log(2mFK/\delta)}{\gamma\tau}} + \frac{2F \log(2mFK/\delta)}{\gamma\tau}.$$

Since $\widetilde{\nabla}_i^k \Phi(a_i) = \sum_{f \in a_i} [\widetilde{\theta}_i^k(\pi^k)]_f$, by triangle inequality, we have

$$\left| \widetilde{\nabla}_i^k \Phi(a_i) - \nabla_i^k \Phi(a_i) \right| \leq \sqrt{\frac{4F^3 \log(2mFK/\delta)}{\gamma\tau}} + \frac{2F^2 \log(2mFK/\delta)}{\gamma\tau}.$$

□

With this more refined gradient estimator, we can now have the following theorem.

Theorem E.4.10. *Let $T = K\tau$. For the congestion game with semi-bandit feedback, by running Algorithm 12 with gradient estimator $\widetilde{\nabla}_i^k \Phi$ in (E.1) and exploration distribution $\tilde{\rho}_i$ in Algorithm 24, setting parameters $\nu = \frac{\sqrt{F}}{m\sqrt{K}}$, $\gamma = \frac{\sqrt{F}}{mK}$ and $\tau = K^2$, if $K \geq \frac{2\sqrt{F}}{m}$, then with probability $1-\delta$, we have*

$$\textit{Nash-Regret}(T) = \tau \sum_{k=1}^{K} G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2 F^{3/2} T^{5/6} + m^3 F^2 T^{2/3}\right).$$

Proof. By following the proof of Theorem E.4.8 and applying the concentration inequality in Lemma E.4.9, we can have

$$\begin{aligned}\Phi(\pi^{k+1}) \geq & \Phi(\pi^k) + (1-\gamma)\nu G(\pi^k) - \frac{(1-\gamma)\nu}{\sqrt{\gamma\tau}}\sqrt{4m^2F^3\log(2mK/\delta)} \\ & - \frac{2(1-\gamma)\nu}{\gamma\tau} mF^2 \log(mK/\delta) - \gamma 2m^2 F - 4m^3 F(\nu^2+\gamma^2).\end{aligned}$$

Summing over $k \in [K]$ and we get

$$\begin{aligned}\sum_{k=1}^{K} G(\pi^k) \leq & \frac{\Phi(\pi^{K+1}) - \Phi(\pi^1)}{(1-\gamma)\nu} + \frac{K}{\sqrt{\gamma\tau}}\sqrt{4m^2F^3\log(mK/\delta)} + \frac{2K}{\gamma\tau} mF^2\log(mK/\delta) \\ & + \frac{2m^2FK\gamma}{(1-\gamma)\nu} + \frac{4(\nu^2+\gamma^2)m^3FK}{(1-\gamma)\nu}.\end{aligned}$$

Set $\nu = \frac{\sqrt{F}}{m\sqrt{K}}$, $\gamma = \frac{\sqrt{F}}{mK}$, $\tau = K^2$ and notice that when $K \geq \frac{2\sqrt{F}}{m}$, we have $1-\gamma \geq \frac{1}{2}$. Thus, we can have

$$\sum_{k=1}^{K} G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2 F^{3/2} K^{1/2} + m^3 F^2\right).$$

Then by Lemma E.4.7, for $T = K\tau$, we have

$$\text{Nash-Regret}(T) = \tau \sum_{k=1}^{K} G(\pi^k) = \widetilde{\mathcal{O}}\left(m^2 F^{3/2} T^{5/6} + m^3 F^2 T^{2/3}\right).$$

□

E.4.4 Lemmas for Semi-bandit Feedback

Lemma E.4.11. *Let $\mathcal{F}_i = \bigcup_{a_i \in \mathcal{A}_i} a_i$. For any player i, if $\tilde{\rho}_i$ is the output of Algorithm 24 and π_i^k contains a mixture of $\tilde{\rho}_i$ with weight γ, then we have $\mathbb{P}_{a_i \sim \pi_i^k}(f \in a_i) \geq \frac{\gamma}{2F}$ for any $f \in \mathcal{F}_i$.*

Algorithm 24 Compute Exploration Distribution $\tilde{\rho}_i$

Input: $\mathcal{A}_i$, player i's action set
Initialize $\widetilde{\mathcal{A}}_i \leftarrow \emptyset$
for $a_i \in \mathcal{A}_i$ **do**
    if $\exists f \in a_i$ such that $f \notin \bigcup_{a_i' \in \widetilde{\mathcal{A}}_i} a_i'$ **then**
        $\widetilde{\mathcal{A}}_i \leftarrow \widetilde{\mathcal{A}}_i \cup \{a_i\}$
    end if
    if $\mathcal{F}_i = \bigcup_{a_i' \in \widetilde{\mathcal{A}}_i} a_i'$ **then**
        break
    end if
end for
Assign $\tilde{\rho}_i(a_i) \leftarrow \frac{1}{2F}$ for each $a_i \in \widetilde{\mathcal{A}}_i$
Assign remaining probability mass arbitrarily to actions in $\mathcal{A}_i \setminus \widetilde{\mathcal{A}}_i$
return $\tilde{\rho}_i$

Proof. By Algorithm 24, whenever a new action is added into $\widetilde{\mathcal{A}}_i$, it contains facility not appeared in current $\widetilde{\mathcal{A}}_i$. Then, since there are at most $|\mathcal{F}_i| \leq F$ distinct facilities in the action set $\mathcal{A}_i$, the final $\widetilde{\mathcal{A}}_i$ must satisfy $|\widetilde{\mathcal{A}}_i| \leq F$. Therefore, $\tilde{\rho}_i$ is a valid distribution over $\mathcal{A}_i$.

Since π_i^k contains a mixture of $\tilde{\rho}_i$ with weight γ, for any $a_i \in \mathcal{A}_i$, we have $\pi_i^k(a_i) \geq \gamma \tilde{\rho}_i(a_i)$. Thus, we have

$$\begin{aligned}
\mathbb{P}_{a_i \sim \pi_i^k}(f \in a_i) = &\sum_{a_i \in \mathcal{A}_i} \pi_i^k(a_i) \mathbb{1}\{f \in a_i\} \\
\geq &\gamma \sum_{a_i \in \mathcal{A}_i} \tilde{\rho}_i(a_i) \mathbb{1}\{f \in a_i\} \\
\geq &\gamma \sum_{a_i \in \widetilde{\mathcal{A}}_i} \tilde{\rho}_i(a_i) \mathbb{1}\{f \in a_i\} \\
= &\frac{\gamma}{2F} \sum_{a_i \in \widetilde{\mathcal{A}}_i} \mathbb{1}\{f \in a_i\} \geq \frac{\gamma}{2F}.
\end{aligned}$$

The last inequality above holds since by construction, $\widetilde{\mathcal{A}}_i$ contains all facilities contained in $\mathcal{A}_i$.

□

Lemma E.4.12. *If π_i^k contains a mixture of $\tilde{\rho}_i$ given in Algorithm 24 with weight γ. Then, the IPS estimator $[\widetilde{\theta}_i^k(\pi^k)]_f$ satisfies*

$$\mathbb{E}_k\left[[\widetilde{\theta}_i^{k,t}(\pi^k)]_f\right] = [\theta_i(\pi^k)]_f, \quad |[\widetilde{\theta}_i^{k,t}(\pi^k)]_f| \le \frac{2F}{\gamma}, \quad \textit{and} \quad \mathbb{E}_k\left[[\widetilde{\theta}_i^{k,t}(\pi^k)]_f^2\right] \le \frac{2F}{\gamma}.$$

Proof. For the first property, since $\mathbb{E}_k\left[r^{k,t,f} \mid \mathbf{a}^{k,t}\right] = r^f(n^f(a_i^{k,t}, a_{-i}^{k,t}))$ and $\mathbf{a}^{k,t} \sim \pi^k$, We have

$$\begin{aligned}
&\mathbb{E}_k\left[[\widetilde{\theta}_i^{k,t}(\pi^k)]_f\right] \\
=&\mathbb{E}_{\mathbf{a}\sim\pi^k}\left[\frac{r^f(n^f(a_i,a_{-i}))\mathbb{1}\{f\in a_i\}}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')}\right] \\
=&\frac{1}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')}\cdot\mathbb{E}_{a_{-i}\sim\pi_{-i}^k}\left[\mathbb{E}_{a_i\sim\pi_i^k}\left[r^f(n^f(a_i,a_{-i}))\mathbb{1}\{f\in a_i\}\mid a_{-i}\right]\right] \\
=&\frac{1}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')}\cdot\mathbb{E}_{a_{-i}\sim\pi_{-i}^k}\left[\mathbb{E}_{a_i\sim\pi_i^k}\left[r^f(n^f(a_i,a_{-i}))\mid a_{-i}, f\in a_i\right]\mathbb{P}_{a_i\sim\pi_i^k}(f\in a_i\mid a_{-i})\right] \\
\overset{\text{(i)}}{=}&\frac{\mathbb{P}_{a_i\sim\pi_i^k}(f\in a_i)}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')}\cdot\mathbb{E}_{a_{-i}\sim\pi_{-i}^k}\left[r^f(n^f(a_{-i})+1)\right] \\
=&[\theta_i(\pi^k)]_f.
\end{aligned}$$

The equality (i) above holds because $\mathbb{E}_{a_i\sim\pi_i^k}\left[r^f(n^f(a_i,a_{-i}))\mid a_{-i}, f\in a_i\right] = r^f(n^f(a_{-i})+1)$ and $f\in a_i$ does not depend on a_{-i}.

For the second property, since $\mathbb{P}_{a_i\sim\pi_i^k}(f\in a_i) \ge \frac{\gamma}{2F}$ by Lemma E.4.11 and $r^{k,t,f}\in[0,1]$, we can immediately have $|[\widetilde{\theta}_i^{k,t}(\pi^k)]_f| \le \frac{2F}{\gamma}$.

For the third property, we have

$$\begin{aligned}
\mathbb{E}_k\left[[\widetilde{\theta}_i^{k,t}(\pi^k)]_f^2\right] =&\frac{\mathbb{E}_{\mathbf{a}\sim\pi^k}\left[r^f(n^f(a_i,a_{-i}))^2\mathbb{1}\{f\in a_i\}\right]}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')^2} \\
\le&\frac{\mathbb{E}_{\mathbf{a}\sim\pi^k}\left[\mathbb{1}\{f\in a_i\}\right]}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')^2} \\
=&\frac{\mathbb{P}_{a_i\sim\pi_i^k}(f\in a_i)}{\mathbb{P}_{a_i'\sim\pi_i^k}(f\in a_i')^2} \\
\le&\frac{2F}{\gamma}.
\end{aligned}$$

□

E.5 Algorithms for Independent Markov Congestion Games

In this section, present missing details of our centralized algorithm for independent Markov congestion games, which is summarized in Algorithm 25. The proof of its theoretical guarantee is given in Appendix E.6.

E.5.1 Algorithm for Semi-bandit Feedback

Under the semi-bandit feedback, the players can receive reward information from all facilities they choose. Therefore, we can similarly define

$$\begin{aligned} N_h^{k,f}(s^f, n) &= \sum_{k'=1}^{k} \mathbb{1}\left\{(s_h^{k',f}, n^f(\boldsymbol{a}_h^{k'})) = (s^f, n)\right\}, \\ \hat{r}_h^{k,f}(s^f, n) &= \frac{\sum_{k'=1}^{k} r_h^{k',f} \mathbb{1}\left\{(s_h^{k',f}, n^f(\boldsymbol{a}_h^{k'})) = (s^f, n)\right\}}{N_h^{k,f}(s^f, n) \vee 1}, \\ \widehat{P}_h^{k,f}(s'^f \mid s^f, n) &= \frac{\sum_{k'=1}^{k} \mathbb{1}\left\{(s_{h+1}^{k',f}, s_h^{k',f}, n^f(\boldsymbol{a}_h^{k'})) = (s'^f, s^f, n)\right\}}{N_h^{k,f}(s^f, n) \vee 1}. \end{aligned}$$

Then, the estimators for the reward function and transition kernel can be defined as

$$\hat{r}_{h,i}^k(s, \mathbf{a}) = \sum_{f \in a_i} \hat{r}_h^{k,f}(s^f, n^f(\mathbf{a})), \quad \widehat{P}_h^k(s' \mid s, \boldsymbol{a}) = \prod_{f \in \mathcal{F}} \widehat{P}_h^{k,f}(s'^f \mid s^f, n^f(\boldsymbol{a})) \tag{E.2}$$

Then, with $\iota = 2\log(4(m+1)(\sum_{f\in\mathcal{F}} S^f)T/\delta)$, we define the bonus term to be $b_h^k(s, \boldsymbol{a}) = b_h^{k,\mathrm{pv}}(s, \boldsymbol{a}) + b_h^{k,\mathrm{r}}(s, \boldsymbol{a})$, which is a sum of transition bonus and reward bonus. In particular, we have

$$b_h^{k,\mathrm{pv}}(s, \mathbf{a}) = \sum_{f \in \mathcal{F}} \sqrt{\frac{4H^2F^2S^f\iota}{N_h^{k,f}(s^f, n^f(\mathbf{a})) \vee 1}} + \sum_{f \neq f'} \sqrt{\frac{4H^2F^2\left(S^fS^{f'}\iota\right)^2}{N_h^{k,f}(s^f, n^f(\mathbf{a}))N_h^{k,f'}(s^{f'}, n^{f'}(\mathbf{a})) \vee 1}}, \tag{E.3}$$

$$b_h^{k,\mathrm{r}}(s, \mathbf{a}) = \sum_{f \in \mathcal{F}} \sqrt{\frac{\iota}{N_h^{k,f}(s^f, n^f(\mathbf{a})) \vee 1}}. \tag{E.4}$$

For convenience, we define $(\widehat{\mathbb{P}}_h^k V)(s, \mathbf{a}) = \mathbb{E}_{s' \sim \widehat{P}_h^k(\cdot|s,\mathbf{a})}[V(s')]$ with value function $V : \mathcal{S} \mapsto \mathbb{R}$.

Remark E.5.1. Unlike Algorithm 10 for congestion game, here, $\overline{Q}_{h,1}^k(s, \cdot), \ldots, \overline{Q}_{h,m}^k(s, \cdot)$ in line 6 of Algorithm 25 in general does not form a potential game. Therefore, we cannot use Algorithm 11 and ϵ-Nash is not always computationally efficient.

Algorithm 25 Nash-VI for IMCGs

Input: ϵ, accuracy parameter for Nash equilibrium computation
Initialize: $\overline{V}^k_{H+1,i}(s) \leftarrow 0$ for all $(i,k,s) \in [m] \times [K] \times \mathcal{S}$
for episode $k = 1, \ldots, K$ **do**
  for step $h = H, H-1, \ldots, 1$ **do**
    for player $i = 1, \ldots, m$ **do**
      Compute

$$\overline{Q}^k_{h,i}(s, \boldsymbol{a}) \leftarrow \min\left((\hat{r}^k_{h,i} + \widehat{\mathbb{P}}^k_h \overline{V}^k_{h+1,i} + b^k_h)(s, \boldsymbol{a}), HF\right)$$

for all $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$
    end for
    for $s \in \mathcal{S}$ **do**
      Compute $\pi^k_h(\cdot \mid s) \leftarrow \epsilon$-NASH$(\overline{Q}^k_{h,1}(s,\cdot), \ldots, \overline{Q}^k_{h,m}(s,\cdot))$
      for player $i = 1, \ldots, m$ **do**
        Update

$$\overline{V}^k_{h,i}(s) \leftarrow \mathbb{E}_{\boldsymbol{a} \sim \pi^k_h}[\overline{Q}^k_{h,i}(s, \boldsymbol{a})]$$

      end for
    end for
  end for
  for step $h = 1, \ldots, H$ **do**
    Take action $\boldsymbol{a}^k_h \sim \pi^k_h(\cdot \mid s^k_h)$, observe reward $r^{k,f}_h$ and next state s^k_{h+1}
    Update reward estimator $\hat{r}^k_{h,i}$, transition estimator $\widehat{P}^k_h$, and bonus term b^k_h
  end for
end for

E.5.2 Algorithm for Bandit Feedback

In bandit feedback scenario, since players' observation about state transitions remains unaffected, we only need to modify the reward estimator $\hat{r}^k_{h,i}$ defined in (E.2) and reward bonus term $b^{k,\mathrm{r}}_h(s, \mathbf{a})$ defined in (E.4).

Similar to the congestion game with bandit feedback introduced in Section 6.4.2, for

IMCGs, we can also write its reward function as $r_{h,i}(s,\mathbf{a}) = \langle A_i(s,\mathbf{a}), \theta_h \rangle$, where θ_h is unknown and $A_i(s,\mathbf{a})$ is a 0-1 vector.

In particular, define $\theta_h \in [0,1]^d$ with $d = m\sum_{f\in\mathcal{F}} S^f$ to be the vector such that $\theta_{h,i} = r_h^f(s^f, n)$ for some $f \in \mathcal{F}$ and $(s^f, n) \in \mathcal{S}^f \times [m]$. Then, we can similarly build estimator $\hat{r}_{h,i}^k$ through ridge regression as the following.[†]

$$\text{design matrix:} \quad V_h^k = I + \sum_{k'=1}^{k-1}\sum_{i=1}^{m} A_i(s_h^{k'}, \mathbf{a}_h^{k'}) A_i(s_h^{k'}, \mathbf{a}_h^{k'})^\top, \tag{E.5}$$

$$\theta_h \text{ estimator:} \quad \widehat{\theta}_h^k = \left(V_h^k\right)^{-1} \sum_{k'=1}^{k-1}\sum_{i=1}^{m} A_i(s_h^{k'}, \mathbf{a}_h^{k'}) r_{h,i}^{k'}, \tag{E.6}$$

$$\text{reward estimator:} \quad \tilde{r}_{h,i}^k(s,\mathbf{a}) = \left\langle A_i(s,\mathbf{a}), \widehat{\theta}_h^k \right\rangle, \tag{E.7}$$

$$\text{reward bonus:} \quad \tilde{b}_h^{k,\mathrm{r}}(s,\mathbf{a}) = \max_{i\in[m]} \|A_i(s,\mathbf{a})\|_{(V_h^k)^{-1}} \sqrt{\beta_k}, \tag{E.8}$$

where $\sqrt{\beta_k} = \sqrt{d} + \sqrt{Fd\log\left(1+\frac{mkF}{d}\right) + F\iota}$.

E.6 Analysis for Algorithm 25

E.6.1 Bellman Equations for Genera-sum Markov Games

Before analyzing Algorithm 25, we first give a brief review of the Bellman equations for general-sum Markov games. These equations are well-known among the literature Bai and Jin [2020a], Liu et al. [2021a], Jin et al. [2021c].

Fixed policies. Given a fixed policy π, for any $(h,i,s,\boldsymbol{a}) \in [H]\times[m]\times\mathcal{S}\times\mathcal{A}$, it holds that

$$Q_{h,i}^\pi(s,\boldsymbol{a}) = (r_{h,i} + \mathbb{P}_h V_{h+1,i}^\pi)(s,\boldsymbol{a}), \quad V_{h,i}^\pi = \mathbb{E}_{\boldsymbol{a}'\sim\pi_h(\cdot|s)}\left[Q_{h,i}^\pi(s,\boldsymbol{a}')\right], \tag{E.9}$$

where $V_{H+1,i}^\pi(s) = 0$ for any $(i,s) \in [m]\times\mathcal{S}$.

Best responses. Given a fixed policy π, define the best response value functions for player i as $Q_{h,i}^{\dagger,\pi_{-i}}(s,\boldsymbol{a}) = \max_{\pi_i\in\Delta(\mathcal{A}_i)} Q_{h,i}^{\pi_i,\pi_{-i}}(s,\boldsymbol{a})$ and $V_{h,i}^{\dagger,\pi_{-i}}(s) = \max_{\pi_i\in\Delta(\mathcal{A}_i)} V_{h,i}^{\pi_i,\pi_{-i}}(s)$.

[†]For the same reason, we take the regularization parameter in ridge regression to be 1.

Then, for any $(h, i, s, \boldsymbol{a}) \in [H] \times [m] \times \mathcal{S} \times \mathcal{A}$, it holds that

$$\begin{aligned} Q_{h,i}^{\dagger,\pi_{-i}}(s,\boldsymbol{a}) &= (r_{h,i} + \mathbb{P}_h V_{h+1,i}^{\dagger,\pi_{-i}})(s,\boldsymbol{a}), \\ V_{h,i}^{\dagger,\pi_{-i}}(s) &= \max_{\nu \in \Delta(\mathcal{A}_i)} \mathbb{E}_{\boldsymbol{a}' \sim (\nu, \pi_{h,-i})(\cdot|s)} \left[Q_{h,i}^{\dagger,\pi_{-i}}(s,\boldsymbol{a}') \right], \end{aligned} \tag{E.10}$$

where $V_{H+1,i}^{\dagger,\pi_{-i}}(s) = 0$ for any $(i, s) \in [m] \times \mathcal{S}$.

E.6.2 Proof of Theorem 6.6.2

Recall that the update rule in Algorithm 25 is

$$\overline{Q}_{h,i}^k(s,\boldsymbol{a}) \leftarrow \min\left\{ (\hat{r}_{h,i}^k + \widehat{\mathbb{P}}_h^k \overline{V}_{h+1,i}^k + b_h^k)(s,\boldsymbol{a}), HF \right\}, \quad \overline{V}_{h,i}^k(s) \leftarrow \mathbb{E}_{\boldsymbol{a} \sim \pi_h^k}[\overline{Q}_{h,i}^k(s,\boldsymbol{a})].$$

Similar to the proof of Theorem 6.4.1, we define auxiliary value functions

$$\underline{Q}_{h,i}^k(s,\boldsymbol{a}) \leftarrow \max\left\{ (\hat{r}_{h,i}^k + \widehat{\mathbb{P}}_h^k \underline{V}_{h+1,i}^k - b_h^k)(s,\boldsymbol{a}), 0 \right\}, \quad \underline{V}_{h,i}^k(s) \leftarrow \mathbb{E}_{\boldsymbol{a} \sim \pi_h^k}[\underline{Q}_{h,i}^k(s,\boldsymbol{a})]. \tag{E.11}$$

We now begin to prove the first part of Theorem 6.6.2.

Proof of Theorem 6.6.2. **Step 1.** We first consider the setting of semi-bandit feedback. Assume the result in Lemma E.6.2 holds since it is a high-probability event. Then, for any $(k, s) \in [K] \times \mathcal{S}$, it holds that

$$\max_{i \in [m]} \left(V_{1,i}^{\dagger,\pi_{-i}^k} - V_{1,i}^{\pi^k} \right)(s) \le \max_{i \in [m]} \left(\overline{V}_{1,i}^k - \underline{V}_{1,i}^k \right)(s) + H\epsilon.$$

By the update rules in Algorithm 25, we can notice the following recursive relations

$$\begin{aligned} (\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,\mathbf{a}) &\le \min\left\{ \widehat{\mathbb{P}}_h^k (\overline{V}_{h+1,i}^k - \underline{V}_{h+1,i}^k)(s,\mathbf{a}) + 2b_h^k(s,\mathbf{a}), HF \right\}, \\ (\overline{V}_{h,i}^k - \underline{V}_{h,i}^k)(s) &= \mathbb{E}_{\mathbf{a}' \sim \pi_h^k(\cdot|s)} \left[(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,\mathbf{a}') \right]. \end{aligned}$$

Thus, we define $\widetilde{V}_{H+1}^k(s) = 0$ for any $s \in \mathcal{S}$ and $\widetilde{Q}_h^k$, $\widetilde{V}_h^k$ recursively as

$$\widetilde{Q}_h^k(s,\mathbf{a}) = \min\left\{ (\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s,\mathbf{a}) + 2b_h^k(s,\mathbf{a}), HF \right\}, \quad \widetilde{V}_h^k(s) = \mathbb{E}_{\mathbf{a}' \sim \pi_h^k(\cdot|s)} \left[\widetilde{Q}_h^k(s,\mathbf{a}') \right]. \tag{E.12}$$

Obviously, we have $\max_{i \in [m]} (\overline{V}_{h,i}^k - \underline{V}_{h,i}^k)(s) \le \widetilde{V}_{H+1}^k$. Then, by inductively assuming the same relation holds for $h+1$, we can have

$$\max_{i \in [m]} (\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,\mathbf{a}) = \min\left\{ \max_{i \in [m]} \widehat{\mathbb{P}}_h^k (\overline{V}_{h+1,i}^k - \underline{V}_{h+1,i}^k)(s,\mathbf{a}) + 2b_h^k(s,\mathbf{a}), HF \right\}$$

$$
\begin{aligned}
&\leq \min\left\{(\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s,\mathbf{a}) + 2b_h^k(s,\mathbf{a}), HF\right\} \\
&= \widetilde{Q}_h^k(s,\mathbf{a}), \\
\max_{i\in[m]}(\overline{V}_{h,i}^k - \underline{V}_{h,i}^k)(s) &\leq \mathbb{E}_{\mathbf{a}'\sim\pi_h^k(\cdot|s)}\left[\max_{i\in[m]}(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,\mathbf{a}')\right] \\
&\leq \mathbb{E}_{\mathbf{a}'\sim\pi_h^k(\cdot|s)}\left[\widetilde{Q}_h^k(s,\mathbf{a}')\right] \\
&= \widetilde{V}_h^k(s).
\end{aligned}
$$

Therefore, by induction, for any $h \in [H]$, we have

$$
\max_{i\in[m]}(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,\mathbf{a}) \leq \widetilde{Q}_h^k(s,\mathbf{a}), \quad \max_{i\in[m]}(\overline{V}_{h,i}^k - \underline{V}_{h,i}^k)(s) \leq \widetilde{V}_h^k(s).
$$

As a result, we have

$$
\text{Nash-Regret}(K) = \sum_{k=1}^K \max_{i\in[m]}\left(V_{1,i}^{\dagger,\pi_{-i}^k} - V_{1,i}^{\pi^k}\right)(s) \leq \sum_{k=1}^K \widetilde{V}_1^k(s_1) + HK\epsilon.
$$

Step 2, Semi-bandit Feedback. We define the martingale difference sequences

$$
\begin{aligned}
\mathcal{M}_h^k(\widetilde{Q}) &= \mathbb{E}_{\mathbf{a}'\sim\pi_h^k(\cdot|s_h^k)}\left[\widetilde{Q}_h^k(s_h^k,\mathbf{a}')\right] - \widetilde{Q}_h^k(s_h^k,\boldsymbol{a}_h^k), \\
\mathcal{M}_h^k(\widetilde{V}) &= (\mathbb{P}_h \widetilde{V}_{h+1}^k)(s_h^k,\boldsymbol{a}_h^k) - \widetilde{V}_{h+1}^k(s_{h+1}^k).
\end{aligned}
$$

It is not hard to check that $\mathcal{M}_h^k(\widetilde{Q})$ and $\mathcal{M}_h^k(\widetilde{V})$ are both indeed martingale difference sequences with respect to the history till episode k and time step h.

With these definitions, we can now decompose the regret bound as

$$
\begin{aligned}
\widetilde{V}_h^k(s_h^k) &= \mathbb{E}_{\mathbf{a}'\sim\pi_h^k(\cdot|s_h^k)}\left[\widetilde{Q}_h^k(s_h^k,\mathbf{a}')\right] && \text{(By (E.12))} \\
&= \mathcal{M}_h^k(\widetilde{Q}) + \widetilde{Q}_h^k(s_h^k,\boldsymbol{a}_h^k) \\
&\leq \mathcal{M}_h^k(\widetilde{Q}) + 2b_h^k(s_h^k,\boldsymbol{a}_h^k) + (\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k)(s_h^k,\boldsymbol{a}_h^k) && \text{(By (E.12))} \\
&\overset{\text{(i)}}{\leq} \mathcal{M}_h^k(\widetilde{Q}) + 3b_h^k(s_h^k,\boldsymbol{a}_h^k) + (\mathbb{P}_h \widetilde{V}_{h+1}^k)(s_h^k,\boldsymbol{a}_h^k) \\
&= \mathcal{M}_h^k(\widetilde{Q}) + \mathcal{M}_h^k(\widetilde{V}) + 3b_h^k(s_h^k,\boldsymbol{a}_h^k) + \widetilde{V}_{h+1}^k(s_{h+1}^k)
\end{aligned}
$$

The above inequality (i) holds by applying Lemma E.6.2 and the fact $\widetilde{V}_h^k(s) \leq HF$, which comes from the definition in (E.12). Then, by unrolling this relation from $h = 1$ to $h = H$

and noticing $\widetilde{V}_{H+1}^k = \mathbf{0}$, we can have

$$\begin{aligned}
&\text{Nash-Regret}(K) \leq \sum_{k=1}^{K} \widetilde{V}_1^k(s_1) + HK\epsilon \\
\leq &\sum_{k=1}^{K}\sum_{h=1}^{H}\left(\mathcal{M}_h^k(\widetilde{Q}) + \mathcal{M}_h^k(\widetilde{V}) + 3b_h^k(s_h^k, \boldsymbol{a}_h^k)\right) + HK\epsilon \qquad \text{(E.13)} \\
\leq &\widetilde{\mathcal{O}}\left(HF\sqrt{T}\right) + 3\sum_{k=1}^{K}\sum_{h=1}^{H} b_h^k(s_h^k, \boldsymbol{a}_h^k) \\
&\qquad\qquad \text{(By Azuma-Hoeffding inequality and taking } \epsilon = 1/T.\text{)} \\
\leq &\widetilde{\mathcal{O}}\left(HF\sqrt{T}\right) + 6HF\sum_{f\in\mathcal{F}}\sum_{k=1}^{K}\sum_{h=1}^{H}\left(\sqrt{\frac{S^f\iota}{N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k))\vee 1}} + \sqrt{\frac{\iota}{N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k))\vee 1}}\right) \\
&\quad + 6HF\sum_{f\neq f'} S^f S^{f'} \sum_{k=1}^{K}\sum_{h=1}^{H}\sqrt{\frac{\iota^2}{\left(N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k))N_h^{k,f'}(s_h^{k,f'}, n^{f'}(\mathbf{a}_h^{k,f'}))\right)\vee 1}} \\
\leq &\widetilde{\mathcal{O}}\left(HF\sqrt{T}\right) + \widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} HFS^f\sqrt{mHT}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right) \\
&\qquad\qquad \text{(By Lemma E.6.5 and E.6.6)} \\
\leq &\widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} FS^f\sqrt{mH^3T}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right).
\end{aligned}$$

Step 3, Bandit Feedback. In the setting of bandit feedback, we only modify the reward estimator $\tilde{r}_{h,i}^k$ and its corresponding bonus term $\tilde{b}_h^{k,\mathrm{r}}$. Thus, by going through the proof of Lemma E.6.2, we can notice that to have the same result for bandit feedback, it suffice to use Lemma E.6.3 to show that the reward estimation error is bounded by the reward bonus term.

Then, by the inequality (E.13), we can notice that to achieve the final Nash-regret bound, we only need to bound the summation $\sum_{k=1}^{K}\sum_{h=1}^{H}\tilde{b}_h^{k,\mathrm{r}}(s_h^k, \boldsymbol{a}_h^k)$, which is

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H}\tilde{b}_h^{k,\mathrm{r}}(s_h^k, \boldsymbol{a}_h^k) \leq &\sqrt{\beta_K}\sum_{k=1}^{K}\sum_{h=1}^{H}\max_{i\in[m]}\left\|A_i(s_h^k, \boldsymbol{a}_h^k)\right\|_{(V_h^k)^{-1}} \quad \text{(By definition of } \tilde{b}_h^{k,\mathrm{r}} \text{ in (E.8).)} \\
&\leq \left(\sqrt{d} + \sqrt{Fd\log\left(1 + \frac{mKF}{d}\right) + F\iota}\right)\widetilde{\mathcal{O}}\left(H\sqrt{dFK}\right) \\
&\qquad\qquad \text{(By definition of } \beta_k \text{ and Lemma E.6.4.)} \\
&\leq \widetilde{\mathcal{O}}\left(d\sqrt{HF^2T}\right)
\end{aligned}$$

$$= \widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} mS^f\sqrt{HF^2T}\right). \qquad \text{(Since } d = m\textstyle\sum_{f\in\mathcal{F}} S^f\text{.)}$$

Therefore, by (E.13), with $\epsilon = 1/T$, under bandit feedback, we have

$$\begin{aligned}
&\text{Nash-Regret}(K)\\
\le&\sum_{k=1}^{K}\sum_{h=1}^{H}\left(\mathcal{M}_h^k(\widetilde{Q}) + \mathcal{M}_h^k(\widetilde{V}) + 3b_h^k(s_h^k, \boldsymbol{a}_h^k)\right)\\
\le&\widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}} FS^f\sqrt{mH^3T}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right) + \sum_{k=1}^{K}\sum_{h=1}^{H}\tilde{b}_h^{k,\mathrm{r}}(s_h^k, \boldsymbol{a}_h^k)\\
\le&\widetilde{\mathcal{O}}\left(\sum_{f\in\mathcal{F}}\left(\sqrt{mH^3}F + m\sqrt{HF^2}\right)S^f\sqrt{T}\right) + \widetilde{\mathcal{O}}\left(m^2H^2F\sum_{f\neq f'}\left(S^fS^{f'}\right)^2\right).
\end{aligned}$$

□

E.6.3 Lemmas for Semi-bandit Feedback

The following two lemmas shows that our value function estimations are indeed optimistic.

Lemma E.6.1. *With probability at least $1-\delta$, simultaneously for arbitrary value function $V \in [0, HF]^{\mathcal{S}}$ and any tuple $(k, h, s, \mathbf{a})$, it holds that $|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V(s, \boldsymbol{a})| \le b_h^{k,\mathrm{pv}}(s, \mathbf{a})$, where $b_h^{k,\mathrm{pv}}(s, \mathbf{a})$ is defined in (E.3).*

Proof. We define $\mathbb{P}_h^f$ to be the operator such that for some value function $V^f : \mathcal{S}^f \mapsto \mathbb{R}$, we have $(\mathbb{P}_h^f V^f)(s, \mathbf{a}) = \mathbb{E}_{s'^f\sim P_h^f(\cdot|s^f, n^f(\mathbf{a}))}\left[V^f(s'^f)\right]$. We also define $\widehat{\mathbb{P}}_h^{k,f}$ similarly. Then, by definition of our transition kernel, for operators $\mathbb{P}_h$ and $\widehat{\mathbb{P}}_h^k$, it holds that

$$\mathbb{P}_h = \prod_{f\in\mathcal{F}}\mathbb{P}_h^f \quad \text{and} \quad \widehat{\mathbb{P}}_h^k = \prod_{f\in\mathcal{F}}\widehat{\mathbb{P}}_h^{k,f}.$$

Therefore, by Lemma E.1 in Chen et al. [2020], since $\|V\|_\infty \le HF$, we have

$$\begin{aligned}
|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V(s, \boldsymbol{a})| \le \sum_{f\in\mathcal{F}}&\left|(\widehat{\mathbb{P}}_h^{k,f} - \mathbb{P}_h^f)\left(\prod_{f'\neq f}\mathbb{P}_h^{f'}\right)V(s, \mathbf{a})\right|\\
&+ 2HF\sum_{f\neq f'}\mathrm{errp}_h^{k,f}(s, \mathbf{a})\cdot\mathrm{errp}_h^{k,f'}(s, \mathbf{a}),
\end{aligned} \tag{E.14}$$

where $\mathrm{errp}_h^{k,f}(s, \mathbf{a}) = \|\widehat{P}_h^{k,f}(\cdot \mid s^f, n^f(\mathbf{a})) - P_h^f(\cdot \mid s^f, n^f(\mathbf{a}))\|_1$.

Now, notice that $\left(\prod_{f'\neq f}\mathbb{P}_h^{f'}\right)V(s,\mathbf{a})$ can be seen as some value function from $\mathcal{S}^f$ to $[0,HF]$. Therefore, by Lemma 12 in Bai and Jin [2020a], with probability at least $1-\frac{\delta}{2}$, simultaneously for any V and $(k,h,s,\mathbf{a})$, it holds that

$$\left|(\widehat{\mathbb{P}}_h^{k,f}-\mathbb{P}_h^f)\left(\prod_{f'\neq f}\mathbb{P}_h^{f'}\right)V(s,\mathbf{a})\right|\leq 2HF\sqrt{\frac{S^f\iota}{N_h^{k,f}(s^f,n^f(\mathbf{a}))\vee 1}},$$

where $\iota=2\log(4(m+1)(\sum_{f\in\mathcal{F}}S^f)T/\delta)$. Meanwhile, by standard Hoeffding's inequality and union bound, with probability at least $1-\frac{\delta}{2}$, simultaneously for any $(k,h,s,\mathbf{a})$, it holds that

$$\mathrm{errp}_h^{k,f}\leq S^f\sqrt{\frac{\iota}{N_h^{k,f}(s^f,n^f(\mathbf{a}))\vee 1}}.$$

Finally, by plugging above two concentration inequalities back into (E.14), we can have

$$|(\widehat{\mathbb{P}}_h^k-\mathbb{P}_h)V(s,\boldsymbol{a})|\leq b_h^{k,\mathrm{pv}}(s,\mathbf{a}).$$

□

Lemma E.6.2. *With probability at least $1-\delta$, for any $(k,h,i,s,\boldsymbol{a})\in[K]\times[H]\times[m]\times\mathcal{S}\times\mathcal{A}$, it holds that*

$$\overline{Q}_{h,i}^k(s,\boldsymbol{a})\geq Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\boldsymbol{a})-(H-h)\epsilon,\quad \underline{Q}_{h,i}^k(s,\boldsymbol{a})\leq Q_{h,i}^{\pi^k}(s,\boldsymbol{a}),\tag{E.15}$$

$$\overline{V}_{h,i}^k(s)\geq V_{h,i}^{\dagger,\pi_{-i}^k}(s)-(H-h+1)\epsilon,\quad \underline{V}_{h,i}^k(s)\leq V_{h,i}^{\pi^k}(s.),\tag{E.16}$$

where $\underline{Q}_{h,k}^k$ and $\underline{V}_{h,i}^k$ are defined in (E.11).

Proof. The proof is adapted from Liu et al. [2021a] and goes by induction from $h=H+1$ to $h=1$. We can see that inequalities (E.16) obviously hold when $h=H+1$ since by definition we have $\overline{V}_{H+1,i}^k(s)=\underline{V}_{H+1,i}^k(s)=0$ for any (k,i,s). Now, suppose inequalities (E.16) hold for $h+1$. Then, if we have $\overline{Q}_{h,i}^k(s,\mathbf{a})=HF$, it holds trivially that $\overline{Q}_{h,i}^k(s,\mathbf{a})\geq Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\mathbf{a})$. Otherwise, by Bellman equations (E.10) and update rule in Algorithm 25, we have

$$\begin{aligned}
&\overline{Q}_{h,i}^k(s,\mathbf{a})-Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\mathbf{a})\\
=&(\hat{r}_{h,i}^k-r_{h,i})(s,\mathbf{a})+(\widehat{\mathbb{P}}_h^k\overline{V}_{h+1,i}^k)(s,\mathbf{a})-(\mathbb{P}_hV_{h+1,i}^{\dagger,\pi_{-i}^k})(s,\mathbf{a})+b_h^k(s,\mathbf{a})\\
=&\underbrace{(\hat{r}_{h,i}^k-r_{h,i})(s,\mathbf{a})}_{\text{(A)}}+\underbrace{\widehat{\mathbb{P}}_h^k(\overline{V}_{h+1,i}^k-V_{h+1,i}^{\dagger,\pi_{-i}^k})(s,\mathbf{a})}_{\text{(B)}}+\underbrace{((\widehat{\mathbb{P}}_h^k-\mathbb{P}_h)V_{h+1,i}^{\dagger,\pi_{-i}^k})(s,\mathbf{a})}_{\text{(C)}}+b_h^k(s,\mathbf{a}).
\end{aligned}$$

Now, recall that $b_h^k(s,\boldsymbol{a}) = b_h^{k,\mathrm{pv}}(s,\boldsymbol{a}) + b_h^{k,\mathrm{r}}(s,\boldsymbol{a})$. By reward definition in congestion game, we have

$$(\hat{r}_{h,i}^k - r_{h,i})(s,\mathbf{a}) = \sum_{f\in a_i} (\hat{r}_{h,i}^{k,f}(s^f, n^f(\mathbf{a})) - r_{h,i}^f(s^f, n^f(\mathbf{a}))).$$

Thus, by using standard Hoefding's inequality and union bound, we can immediately have $|(\mathrm{A})| \le b_h^{k,\mathrm{r}}(s,\mathbf{a})$. Then, since $V_{h,i}^{\dagger,\pi_{-i}^k} \in [0,HF]^{\mathcal{S}}$, by Lemma E.6.1, we have $|(\mathrm{C})| \le b_h^{k,\mathrm{pv}}(s,\mathbf{a})$. That is, we have $(\mathrm{A})+(\mathrm{C})+b_h^k(s,\mathbf{a}) \ge 0$.

Then, by inductive hypothesis, we know that $\overline{V}_{h+1,i}^k \ge V_{h+1,i}^{\dagger,\pi_{-i}^k} - (H-h)\epsilon$, which implies $(\mathrm{B}) \ge 0$. Therefore, we have $\overline{Q}_{h,i}^k(s,\mathbf{a}) - Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\mathbf{a}) \ge -(H-h)\epsilon$.

For $\overline{V}_{h,i}^k$ and $V_{h,i}^{\dagger,\pi_{-i}^k}$, we notice that in Algorithm 25, π^k is computed as the ϵ-approximate Nash equilibrium of $(\overline{Q}_{h,1}^k, \ldots, \overline{Q}_{h,m}^k)$. Therefore, it holds that

$$\overline{V}_{h,i}^k(s) = \mathbb{E}_{\mathbf{a}\sim\pi_h^k(\cdot|s)}\left[\overline{Q}_{h,i}^k(s,\mathbf{a})\right] \ge \max_{\nu\in\Delta(\mathcal{A}_i)} \mathbb{E}_{\boldsymbol{a}'\sim(\nu,\pi_{h,-i}^k)(\cdot|s)}\left[\overline{Q}_{h,i}^k(s,\mathbf{a}')\right] - \epsilon.$$

By Bellman equations (E.10), we also have

$$V_{h,i}^{\dagger,\pi_{-i}^k}(s) = \max_{\nu\in\Delta(\mathcal{A}_i)} \mathbb{E}_{\boldsymbol{a}'\sim(\nu,\pi_{h,-i}^k)(\cdot|s)}\left[Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\mathbf{a}')\right].$$

Since $\overline{Q}_{h,i}^k(s,\mathbf{a}) - Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\mathbf{a}) \ge -(H-h)\epsilon$, we immediately have $\overline{V}_{h,i}^k(s) - V_{h,i}^{\dagger,\pi_{-i}^k}(s) \ge -(H-h+1)\epsilon$. Thus, by induction, we have that $\overline{Q}_{h,i}^k(s,\boldsymbol{a}) \ge Q_{h,i}^{\dagger,\pi_{-i}^k}(s,\boldsymbol{a}) - (H-h)\epsilon$ and $\overline{V}_{h,i}^k(s) \ge V_{h,i}^{\dagger,\pi_{-i}^k}(s) - (H-h+1)\epsilon$ for all $h\in[H]$.

The inequalities for $\underline{V}_{h,i}^k$ and $\underline{Q}_{h,i}^k$ can be proved similarly. □

E.6.4 Additional Lemmas for Bandit Feedback

The following lemma shows that the reward estimation error can be bounded by the reward bonus term.

Lemma E.6.3. *With probability at least $1-\delta$, simultaneously for all $(i,k,h,s,\mathbf{a})$, it holds that $|(\tilde{r}_{h,i}^k - r_{h,i})(s,\mathbf{a})| \le \tilde{b}_h^{k,\mathrm{r}}(s,\mathbf{a})$, where $\tilde{r}_{h,i}^k$ and $\tilde{b}_h^{k,\mathrm{r}}$ are defined in (E.7) and (E.8).*

Proof. The proof is extremely similar to Lemma E.3.1. By construction, we have

$$|(\tilde{r}_{h,i}^k - r_{h,i})(s,\mathbf{a})| = \left|\left\langle A_i(s,\mathbf{a}), \widehat{\theta}_h - \theta_h \right\rangle\right|$$

$$\leq \|A_i(s,\mathbf{a})\|_{(V_h^k)^{-1}} \left\|\widehat{\theta}_h - \theta_h\right\|_{V_h^k}$$
$$\leq \|A_i(s,\mathbf{a})\|_{(V_h^k)^{-1}} \left(\|\theta_h\|_2 + \sqrt{F\log\left(\det(V_h^k)\right) + F\iota}\right).$$

(By Theorem 20.5 in Lattimore and Szepesvári [2020].)

Since each element in θ_h is bounded in $[0,1]$ by construction, we have $\|\theta_h\|_2 \leq \sqrt{d}$.

Then, by Lemma E.3.2, we have $\det\left(V_h^k\right) \leq \left(1 + \frac{mkF}{d}\right)^d$ since by construction $\|A_i(s,\mathbf{a})\|_2^2 \leq F$.

Finally, to make this bound valid for all player $i \in [m]$, we only need to take maximization over $i \in [m]$. Therefore, with probability at least $1-\delta$, we have

$$|(\tilde{r}_{h,i}^k - r_{h,i})(s,\mathbf{a})| \leq \max_{i\in[m]} \|A_i(s,\mathbf{a})\|_{(V_h^k)^{-1}} \sqrt{\beta_k} = \tilde{b}_h^{k,\mathrm{r}}(s,\mathbf{a}),$$

where $\sqrt{\beta_k} = \sqrt{d} + \sqrt{Fd\log\left(1 + \frac{mkF}{d}\right) + F\iota}$. □

The follow lemma bound the sum of reward bonus under bandit feedback.

Lemma E.6.4. *For any $h \in [H]$, it holds that*

$$\sum_{k=1}^K \max_{i\in[m]} \left\|A_i(s_h^k, \boldsymbol{a}_h^k)\right\|_{(V_h^k)^{-1}} \leq \widetilde{\mathcal{O}}\left(\sqrt{dFK}\right),$$

where $d = m\sum_{f\in\mathcal{F}} S^f$.

Proof. First, since $V_h^k = I + \sum_{k'=1}^{k-1}\sum_{i=1}^m A_i(s_h^{k'}, \mathbf{a}_h^{k'}) A_i(s_h^{k'}, \mathbf{a}_h^{k'})^\top$, we have $V_h^k \succeq I$ and thus $\left(V_h^k\right)^{-1} \preceq I$. Therefore, we have

$$\left\|A_i(s_h^k, \boldsymbol{a}_h^k)\right\|_{(V_h^k)^{-1}} \leq \left\|A_i(s_h^k, \boldsymbol{a}_h^k)\right\|_I = \left\|A_i(s_h^k, \boldsymbol{a}_h^k)\right\|_2 \leq \sqrt{F}.$$

For simplicity, let $A_{h,i}^k = A_i(s_h^k, \boldsymbol{a}_h^k)$. Then, as a result, we have

$$\begin{aligned}
\sum_{k=1}^K \max_{i\in[m]} \left\|A_{h,i}^k\right\|_{(V_h^k)^{-1}} &= \sum_{k=1}^K \min\left\{\max_{i\in[m]} \left\|A_{h,i}^k\right\|_{(V_h^k)^{-1}}, \sqrt{F}\right\} \\
&\leq \sqrt{K\sum_{k=1}^K \min\left\{\max_{i\in[m]} \left\|A_{h,i}^k\right\|_{(V_h^k)^{-1}}^2, F\right\}} \\
&\leq \sqrt{FK\sum_{k=1}^K \min\left\{\max_{i\in[m]} \left\|A_{h,i}^k\right\|_{(V_h^k)^{-1}}^2, 1\right\}}
\end{aligned}$$

$$\leq \sqrt{2FKd \log\left(1 + \frac{mKF}{d}\right)} \qquad \text{(By Lemma E.3.2.)}$$
$$= \widetilde{\mathcal{O}}\left(\sqrt{dFK}\right).$$

□

E.6.5 Technical Lemmas

Lemma E.6.5. *For any $f \in \mathcal{F}$, it holds that*

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \sqrt{\frac{1}{N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k)) \vee 1}} \leq \widetilde{\mathcal{O}}\left(\sqrt{mHS^fT}\right).$$

Proof. Here, we have

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H} \sqrt{\frac{1}{N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k)) \vee 1}} &= \sum_{h=1}^{H}\sum_{s^f\in\mathcal{S}^f}\sum_{n=0}^{m}\sum_{\ell=1}^{N_h^{K,f}(s^f,n)} \sqrt{\frac{1}{\ell}} \\
&\leq 2\sum_{h=1}^{H}\sum_{s^f\in\mathcal{S}^f}\sum_{n=0}^{m} \sqrt{N_h^{K,f}(s^f,n)} \quad \text{(By standard technique)} \\
&\leq 2\sqrt{(m+1)HS^f \sum_{h=1}^{H}\sum_{s^f\in\mathcal{S}^f}\sum_{n=0}^{m} N_h^{K,f}(s^f,n)} \\
&= \widetilde{\mathcal{O}}\left(\sqrt{mHS^fT}\right).
\end{aligned}$$

The last line above holds because $\sum_{h=1}^{H}\sum_{s^f\in\mathcal{S}^f}\sum_{n=0}^{m} N_h^{K,f}(s^f,n) = T$. This is based on a pigeon-hole principle argument. In particular, whenever the players take one more action, for any $f \in \mathcal{F}$, the count for some tuple (h, s^f, n) will increase exactly by 1. □

Lemma E.6.6 (Chen et al. [2020])**.** *For any $f, f' \in \mathcal{F}$ and $f \neq f'$, it holds that*

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \sqrt{\frac{1}{\left(N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k))N_h^{k,f'}(s_h^{k,f'}, n^{f'}(\mathbf{a}_h^{k,f'}))\right) \vee 1}} \leq \widetilde{\mathcal{O}}\left(m^2HS^fS^{f'}\right).$$

Proof. We define the joint empirical counter

$$N_h^{k,f,f'}(s^f, s^{f'}, n, n') = \sum_{k'=1}^{k} \mathbb{1}\left\{(s_h^{k',f}, s_h^{k',f'}, n^f(\mathbf{a}_h^{k'}), n^{f'}(\mathbf{a}_n^{k'})) = (s^f, s^{f'}, n, n')\right\}.$$

Obviously, we have $N_h^{f,f'}(s^f, s^{f'}, n, n') \leq \min\left\{N_h^{k,f}(s^f, n), N_h^{k,f'}(s^{f'}, n')\right\}$, which implies

$$N_h^{k,f,f'}(s, s^{f'}, n, n') \leq \sqrt{N_h^{k,f}(s^f, n) N_h^{k,f'}(s^{f'}, n')}.$$

Therefore, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H}\sqrt{\frac{1}{\left(N_h^{k,f}(s_h^{k,f}, n^f(\boldsymbol{a}_h^k))N_h^{k,f'}(s_h^{k,f'}, n^{f'}(\mathbf{a}_h^{k,f'}))\right)\vee 1}} \\
\leq &\sum_{k=1}^{K}\sum_{h=1}^{H}\frac{1}{N_h^{k,f,f'}(s_h^{k,f}, s_h^{k,f'}, n^f(\boldsymbol{a}_h^k), n^{f'}(\boldsymbol{a}_h^k))\vee 1} \\
= &\sum_{h=1}^{H}\sum_{s^f\in\mathcal{S}^f}\sum_{s^{f'}\in\mathcal{S}^{f'}}\sum_{n=0}^{m}\sum_{n'=0}^{m}\sum_{\ell=1}^{N_h^{K,f,f'}(s^f,s^{f'},n,n')}\frac{1}{\ell} \\
= &\widetilde{\mathcal{O}}\left(m^2 H S^f S^{f'}\right).
\end{aligned}$$

□

Appendix F

DEFERRED CONTENTS FROM CHAPTER ??

F.1 Basics about Congestion Games

Lemma F.1.1. *If strategy x is an ϵ-NE in a congestion game, then x is an ϵ-minimizer of the corresponding potential function $\Phi(\cdot)$.*

Proof. Let $x^* = \operatorname{argmin}_{x\in\mathcal{X}} \Phi(x)$ and $y = \phi(x)$. First, we show that

$$\begin{aligned}
\nabla_{i,a}\Phi(x) =& \nabla_{i,a}\sum_f \int_0^{y_f} c_f(u)du \\
=& \nabla_{i,a}\sum_{f\in a} \int_0^{y_f} c_f(u)du \\
=& \sum_{f\in a} c_f(y_f)\nabla_{i,a} y_f \\
=& \sum_{f\in a} c_f(y_f)\nabla_{i,a} \sum_{i',a':f\in a'} x_{i',a'} \\
=& \sum_{f\in a} c_f(y_f).
\end{aligned}$$

Then we have

$$\begin{aligned}
\Phi(x) - \Phi(x^*) &\le \langle x - x^*, \nabla\Phi(x)\rangle && \text{(Convexity)} \\
&\le \sum_{i\in[m]} \langle x_i - x_i^*, \nabla_i\Phi(x)\rangle \\
&\le \sum_{i\in[m]} \left[\sum_{a\in\mathcal{A}_i} x_{i,a}\sum_{f\in a} c_f(y_f) - \min_{a\in\mathcal{A}_i} w_i \sum_{f\in a} c_f(y_f)\right] \\
&\le \sum_{i\in[m]}\sum_{a\in\mathcal{A}_i} x_{i,a}\epsilon \\
&= \sum_{i\in[m]} w_i\epsilon \\
&= \epsilon.
\end{aligned}$$

□

Lemma 7.4.1. Φ^{repa} *is convex under Assumption 7.3.1. If* $y^* = \operatorname{argmin}_y \Phi^{\text{repa}}(y)$*, then for any* $x \in \phi^{-1}(y^*)$*,* x *is a Nash equilibrium.*

Proof. For $y^1, y^2 \in \mathcal{Y}$, we have

$$\Phi^{\text{repa}}(y^1) + \Phi^{\text{repa}}(y^2) - 2\Phi^{\text{repa}}(\frac{y^1+y^2}{2}) = \sum_f \left[\int_0^{y_f^1} c_f(u)du + \int_0^{y_f^2} c_f(u)du - 2\int_0^{\frac{y_f^1+y_f^2}{2}} c_f(u)du \right].$$

Now we show that $\int_0^{y_f^1} c_f(u)du + \int_0^{y_f^2} c_f(u)du - 2\int_0^{\frac{y_f^1+y_f^2}{2}} c_f(u)du$ is nonnegative for all $f \in \mathcal{F}$. W.l.o.g., we assume $y_f^1 \le y_f^2$ and we have

$$\begin{aligned}
\int_0^{y_f^1} c_f(u)du + \int_0^{y_f^2} c_f(u)du - 2\int_0^{\frac{y_f^1+y_f^2}{2}} c_f(u)du &= \int_{\frac{y_f^1+y_f^2}{2}}^{y_f^2} c_f(u)du - \int_{y_f^1}^{\frac{y_f^1+y_f^2}{2}} c_f(u)du \\
&= \int_{y_f^1}^{\frac{y_f^1+y_f^2}{2}} \left[c_f(u + \frac{y_f^2 - y_f^1}{2}) - c_f(u) \right] du \\
&\ge 0,
\end{aligned}$$

where the last step is from Assumption 7.3.1 (monotonicity). As a result, Φ^{repa} is convex.

Let $y^* = \operatorname{argmin}_y \Phi^{\text{repa}}(y)$ and $x \in \phi^{-1}(y^*)$. If there exists $x' \in \mathcal{X}$ such that $\Phi(x') < \Phi(x)$, then we have $\Phi^{\text{repa}}(\phi(x')) < \Phi^{\text{repa}}(y^*)$, which contradicts the definition of y^*. As a result, x is the minimizer of $\Phi(\cdot)$, which means x is a Nash equilibrium. □

Lemma F.1.2. *The Nash load under tax* τ*:* $y(\tau) = \operatorname{argmin}_{y \in \mathcal{Y}} \Phi(y; \tau)$ *is not continuous w.r.t.* τ*. In addition, the social welfare* $\Psi(y(\tau))$ *is not convex w.r.t.* τ*.*

Proof. For the first part, we construct a congestion game with two facilities f_1, f_2, one commodity with action set $\{f_1, f_2\}$, and constant cost $c_1 = 1, c_2 = 1 - \epsilon$ with $\epsilon > 0$. Then for tax $\tau = 0$, we have $y(\tau) = [0, 1]$. For constant tax $\tau_1 = 0, \tau_2 = 2\epsilon$, we have $y(\tau) = [1, 0]$. As ϵ can be arbitrarily small, $y(\tau)$ is not continuous w.r.t. τ.

For the second part, we construct a congestion game with two facilities f_1, f_2, one commodity with action set $\{f_1, f_2\}$, and cost function $c_1 = 1, c_2(u) = \sqrt{u}$ for $u \in [0, 1]$. We apply constant tax $\tau : \tau_1 = t, \tau_2 = 0$ for $t \in [-1, 0]$. The Nash equilibrium under tax τ is $y(\tau) = [1 - (1+t)^2, (1+t)^2]$. Then the social cost is $\Psi(y(\tau)) = 1 - t(1+t)^2$, which is not convex on $[-1, 0]$. □

F.2 Missing Proofs in Section 7.5

Lemma 7.5.7. *If the subgradient of the cost function c_f is lower bounded by $\epsilon > 0$ for all $f \in \mathcal{F}$, then the potential function $\Phi^{\text{repa}}(y)$ is ϵ-strongly convex. However, $\Phi(x)$ is not necessarily strongly convex.*

Proof. First, by the definition of the potential function Φ, it is easy to show that $\nabla\Phi(y) = [c_f(y_f)]_{f\in\mathcal{F}}$. For $y^1, y^2 \in \mathcal{Y}$, we have

$$(\nabla\Phi(y^1)-\nabla\Phi(y^2))^\top(y^1-y^2) = \sum_{f\in\mathcal{F}}(c_f(y_f^1)-c_f(y_f^2))(y_f^1-y_f^2) \geq \sum_{f\in\mathcal{F}}\epsilon(y_f^1-y_f^2)^2 = \epsilon\left\|y^1-y^2\right\|_2^2,$$

which implies $\Phi(\cdot)$ is a ϵ-strongly convex function.

For the second argument, we only need to construct a congestion game such that there exists two strategy $x^1, x^2 \in \mathcal{X}$ such that for $\phi(tx^1 + (1-t)x^2)$ is a constant for $t \in [0,1]$, which implies the potential function $\Phi(tx^1 + (1-t)x^2) = \Phi^{\text{repa}}(\phi(tx^1 + (1-t)x^2))$ is a constant w.r.t. t. However, a strongly convex function cannot be a constant on a line, which implies Φ is not strongly convex.

We construct a congestion game with three facilities f_1, f_2, f_3, three actions $a_1 = \{f_1\}, a_2 = \{f_2\}, a_3 = \{f_3\}$ and three commodities with action set $\{a_1, a_2\}, \{a_2, a_3\}, \{a_3, a_1\}$. Strategy $x^1 : x_1^1 = [1,0,0], x_2^1 = [0,1,0], x_3^1 = [0,0,1]$ and $x^2 : x_1^2 = [0,1,0], x_2^2 = [0,0,1], x_3^2 = [1,0,0]$. Then $tx^1 + (1-t)x^2$ is a feasible strategy and we have $\phi(tx^1 + (1-t)x^2) = [1,1,1]$ for all $t \in [0,1]$. □

Lemma 7.5.8. *If two taxes τ and $\dot{\tau}$ only differ in facility f and the Nash loads y and $\dot{y}$ are different, then $y_f \neq \dot{y}_f$.*

Proof. For simplicity, we consider the equivalent tax-free case that we have two costs $c, \dot{c}$ with subgradient lower bounded by ϵ and they only differ in facility $\dot{f}$. The potential functions are

$$\Phi(Y) = \sum_f \int_0^{Y_f} c_f(u)du, \dot{\Phi}(Y) = \sum_f \int_0^{Y_f} \dot{c}_f(u)du.$$

By Lemma 7.5.7, Φ and $\dot{\Phi}$ are strongly convex and thus the Nash equilibrium load y and $\dot{y}$

are unique. Suppose $y_{\dot{f}} = \dot{y}_{\dot{f}}$. Consider any $Y \in \mathcal{Y}$ such that $Y_{\dot{f}} = y_{\dot{f}}$, we have

$$\begin{aligned}\dot{\Phi}(Y) - \dot{\Phi}(y) =& \sum_f \int_{y_f}^{Y_f} \dot{c}_f(u)du = \sum_{f \neq \dot{f}} \int_{y_f}^{Y_f} \dot{c}_f(u)du + \int_{y_{\dot{f}}}^{Y_{\dot{f}}} \dot{c}_{\dot{f}}(u)du \\ =& \sum_{f \neq \dot{f}} \int_{y_f}^{Y_f} c_f(u)du = \Phi(Y) - \Phi(y) \geq 0.\end{aligned}$$

As a result, we have $\dot{\Phi}(y) \leq \dot{\Phi}(\dot{y})$. By the optimality of $\dot{y}$, we have $y = \dot{y}$. By contradiction, if $y \neq \dot{y}$, we have $y_{\dot{f}} = \dot{y}_{\dot{f}}$. □

Lemma F.2.1. *If $|u_1 - u_2| \leq \Delta$, then for any $|u_3 - u_1| \leq \Delta$, we have*

$$\left| \frac{c_f(u_1) - c_f(u_2)}{u_1 - u_2} - c'_f(u_3) \right| \leq \epsilon.$$

Proof. This is a direct corollary of the β-smoothness. By mean value theorem, we have $\frac{c_f(u_1) - c_f(u_2)}{u_1 - u_2} = c'_f(u)$ for some $u \in [u_1, u_2]$. As $|u - u_3| \leq |u - u_1| + |u_1 - u_3| \leq 2\Delta \leq \frac{\epsilon}{2\beta}$, we have

$$\left| \frac{c_f(u_1) - c_f(u_2)}{u_1 - u_2} - c'_f(u_3) \right| \leq \epsilon.$$

□

Lemma F.2.2. *For round t and facility f, if $u \in \mathcal{K}_f^t$, then we have $\left| \tau_f^t(u) - \tau_f^*(u) \right| \leq 2\epsilon$.*

Proof. By the algorithm design, for each $u \in \mathcal{K}_f^t$, $\tau_f^t(u)$ will not change after u is added to $\mathcal{K}_f$. We will use induction on t to prove $\left| \widehat{\tau}_f^t(u) - \tau_f(u) \right| \leq \epsilon$ for $u \in \mathcal{K}_f^t$. At round $t = 1$, $\mathcal{K}_f^1 = \{0\}$ and $\widehat{\tau}_f^1(0) = \tau_f^*(0) = 0$ holds.

Suppose at round t, we have $\mathcal{K}_{\widetilde{f}^t}^{t+1} = \mathcal{K}_{\widetilde{f}^t}^t \bigcup \{u\}$ with $u \in \{[y_{\widetilde{f}^t}^t]_{\mathrm{L}}^+, [y_{\widetilde{f}^t}^t]_{\mathrm{L}}^-\} \backslash \mathcal{K}_{\widetilde{f}^t}^t$, and $\mathcal{K}_f^{t+1} = \mathcal{K}_f^t$ for $f \neq \widetilde{f}^t$. By the induction hypothesis, we only need to prove $\left| \widehat{\tau}_{\widetilde{f}^t}^{t+1}(u) - \tau_{\widetilde{f}^t}^*(u) \right| \leq 2\epsilon$. Recall that

$$\widehat{\tau}_{\widetilde{f}^t}^{t+1}(u) = \mathrm{clip}(u \cdot \frac{c_{\widetilde{f}^t}^t - \dot{c}_{\widetilde{f}^t}^t}{y_{\widetilde{f}^t}^t - \dot{y}_{\widetilde{f}^t}^t}, \widehat{\tau}_{\widetilde{f}^t}^t([y_{\widetilde{f}^t}^t]_{\mathcal{K}_{\widetilde{f}^t}^t}^-), \widehat{\tau}_{\widetilde{f}^t}^t([y_{\widetilde{f}^t}^t]_{\mathcal{K}_{\widetilde{f}^t}^t \bigcup \{1\}}^+)).$$

Then we have the following three cases. For simplicity we replace $\widetilde{f}^t$ with f.

(1) $\widehat{\tau}_f^{t+1}(u) = u \cdot \frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t}$. By Lemma F.2.6 and Lemma F.2.1, we have

$$\left| \widehat{\tau}_f^{t+1}(u) - \tau_f^*(u) \right| = \left| u \frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t} - u c'_f(u) \right| \leq \epsilon.$$

(2) $\widehat{\tau}_f^{t+1}(u) = \widehat{\tau}_f^t([y_f^t]^-_{\mathcal{K}_f^t})$ and $u \cdot \frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t} \leq \widehat{\tau}_f^t([y_f^t]^-_{\mathcal{K}_f^t})$. Then we have

$$\widehat{\tau}_f^{t+1}(u) = \widehat{\tau}_f^t([y_f^t]^-_{\mathcal{K}_f^t}) \leq \tau_f^*([y_f^t]^-_{\mathcal{K}_f^t}) + \epsilon \leq \tau_f^*(u) + \epsilon,$$

where the first inequality is from the induction hypothesis as $[y_f^t]^-_{\mathcal{K}_f^t} \in \mathcal{K}_f^t$ and the second inequality is from Assumption 7.4.3. In addition, we have

$$\widehat{\tau}_f^{t+1}(u) \geq u \cdot \frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t} \geq uc_f'(u) - \epsilon = \tau_f^*(u) - \epsilon.$$

(3) $\widehat{\tau}_f^{t+1}(u) = \widehat{\tau}_f^t([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}})$ and $u \cdot \frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t} \geq \widehat{\tau}_f^t([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}})$. Then we have

$$\widehat{\tau}_f^{t+1}(u) \leq u\frac{c_f^t - \dot{c}_f^t}{y_f^t - \dot{y}_f^t} \leq uc_f'(u) + \epsilon \leq \tau_f^*(u) + \epsilon.$$

If $[y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}} \in \mathcal{K}_f^t$, then we have

$$\widehat{\tau}_f^t(u) = \widehat{\tau}_f^t([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) \geq \tau_f^*([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) - \epsilon \geq \tau_f^*(u) - \epsilon.$$

If $[y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}} = 1$ and $1 \notin \mathcal{K}_f^t$, we still have

$$\widehat{\tau}_f^t(u) = \beta \geq uc_f'(u) = \tau_f^*(u).$$

For each of these three cases, the induction holds.

As $\tau_f^t(u) = \widehat{\tau}_f^t(u) + \epsilon u$ for all $f \in \mathcal{F}$ and $u \in [0,1]$, we have $\left|\tau_f^t(u) - \tau_f^*(u)\right| \leq 2\epsilon$. □

Lemma F.2.3. *For round t, if facility f is known, then we have $\left|\tau_f^t(y_f^t) - \tau_f^*(y_f^t)\right| \leq 3\epsilon$.*

Proof. If $y_f^t \in \mathcal{K}_f^t$, we can directly apply Lemma F.2.2. Otherwise, we set $u_1 = [y_f^t]^-_{\mathbf{L}}$ and $u_2 = [y_f^t]^+_{\mathbf{L}}$. Then we have $u_1 < y_f^t < u_2$ and $u_1, u_2 \in \mathcal{K}_f^t$. There exists $\lambda_1 \in [0,1], \lambda_1 + \lambda_2 = 1$ such that $y_f^t = \lambda_1 u_1 + \lambda_2 u_2$. By Lemma F.2.1, we have $\left|\tau_f^t(u_i) - \tau_f^*(u_i)\right| \leq 2\epsilon$ for $i \in \{1,2\}$. Then we have

$$\begin{aligned}
&\left|\tau_f^t(y_f^t) - \tau_f(y_f^t)\right| \\
=&\left|\lambda_1\tau_f^t(u_1) + \lambda_2\tau_f^t(u_2) - (\lambda_1 u_1 + \lambda_2 u_2)c_f'(u)\right| \\
\leq&\left|\lambda_1\tau_f^t(u_1) - \lambda_1 u_1 c_f'(u)\right| + \left|\lambda_2\tau_f^t(u_2) - \lambda_2 u_2 c_f'(u)\right| \\
\leq&\lambda_1\left|\tau_f^t(u_1) - \tau_f^*(u_1)\right| + \lambda_1 u_1\left|c_f'(u_1) - c_f'(u)\right| + \lambda_2\left|\tau_f^t(u_2) - \tau_f^*(u_2)\right| + \lambda_2 u_2\left|c_f'(u_2) - c_f'(u)\right|
\end{aligned}$$

$\leq 2\lambda_1\epsilon + \lambda_1\epsilon + 2\lambda_2\epsilon + \lambda_2\epsilon$ (Lemma F.2.1, β-smoothness and $|u - u_i| \leq \epsilon/\beta$.)

$\leq 3\epsilon$.

□

Lemma F.2.4. *If Algorithm 14 return* False *at round t, then for any $\widetilde{\tau} \in \mathbb{R}^F$ such that $\widetilde{\tau}_f = \tau_f^t(y_f)$ for $f \in \mathcal{F}\backslash\bar{\mathcal{F}}^t$ and $\widetilde{\tau}_f \in [l_f^t, r_f^t]$ for $f \in \bar{\mathcal{F}}^t$, we have $\mathrm{Gap}_i(x^t, c^t + \widetilde{\tau}) \geq 0$ for all $i \in [m]$. In addition, x^t is a Nash equilibrium for tax $\widetilde{\tau}$.*

Proof. For simplicity, we will omit t when there is no confusion. Algorithm 14 return False if and only if for all $i \in [m]$ and tax $\bar{\tau}_i : \bar{\tau}_{\bar{F}_i} = r_{\bar{F}_i}, \bar{\tau}_{\bar{F}_i'} = l_{\bar{F}_i'}, \bar{\tau}_{F\backslash(\bar{F}_i \bigcup \bar{F}_i')} = \tau_{F\backslash(\bar{F}_i \bigcup \bar{F}_i')}$, we have

$$\mathrm{Gap}_i(x, c + \bar{\tau}) = \min_{a: x_{i,a}=0} \sum_{f: f\in a} (c_f + \bar{\tau}_f) - \max_{a: x_{i,a}\neq 0} \sum_{f: f\in a} (c_f + \bar{\tau}_f) \geq 0.$$

By the definition of $\bar{F}_i$, for any $f \in \bar{F}_i$ and $a : x_{i,a} \neq 0$, we have $f \in a$. Similarly, for any $f \in \bar{F}_i'$ and $a : x_{i,a} \neq 0$, we have $f \notin a$. Thus for any $a : x_{i,a} \neq 0$, we have

$$\sum_{f: f\in a} (c_f + \bar{\tau}_f) - \sum_{f: f\in a} (c_f + \widetilde{\tau}_f) = \sum_{f\in \bar{F}_i} (r_f - \widetilde{\tau}_f) \geq 0.$$

For any $a : x_{i,a} = 0$, we have

$$\sum_{f: f\in a} (c_f + \bar{\tau}_f) - \sum_{f: f\in a} (c_f + \widetilde{\tau}_f) = \sum_{f\in \bar{F}_i' \bigcap a} (l_f - \widetilde{\tau}_f) \leq 0.$$

As a result, we have

$$\begin{aligned}\mathrm{Gap}_i(x^t, c^t + \widetilde{\tau}) &= \min_{a: x_{i,a}=0} \sum_{f: f\in a} (c_f + \widetilde{\tau}_f) - \max_{a: x_{i,a}\neq 0} \sum_{f: f\in a} (c_f + \widetilde{\tau}_f) \\ &\geq \min_{a: x_{i,a}=0} \sum_{f: f\in a} (c_f + \bar{\tau}_f) - \max_{a: x_{i,a}\neq 0} \sum_{f: f\in a} (c_f + \bar{\tau}_f) \geq 0.\end{aligned}$$

To prove that x^t is Nash equilibrium for tax $\widetilde{\tau}$, we only need to show that for in-support actions $a : x_{i,a}^t \neq 0$, the action costs $\sum_{f: f\in a}(c_f^t + \widetilde{\tau}_f)$ are the same. This can be derived by

$$\sum_{f: f\in a} (c_f^t + \tau_f^t) - \sum_{f: f\in a} (c_f^t + \widetilde{\tau}_f) = \sum_{f\in \bar{F}_i} (\tau_f^t - \widetilde{\tau}_f), \forall a : x_{i,a}^t \neq 0,$$

which is independent of a. As x^t is Nash equilibrium for tax τ^t, $\sum_{f: f\in a}(c_f^t + \tau_f^t)$ is also independent of a. □

Lemma F.2.5. *If Algorithm 14 return* False *at round t, then tax τ^t is an $6F\epsilon$-optimal tax.*

Proof. For known facility f, by Lemma F.2.3, we have $\left|\tau_f^*(y_f^t) - \tau_f^t(y_f^t)\right| \le 3\epsilon$. By Lemma F.2.2, for any $u \in \mathcal{K}_f^t$, we have $\left|\tau_f^*(u) - \tau_f^t(u)\right| \le 2\epsilon$. Thus for unknown facility f, we have

$$l_f^t = \tau_f^t([y_f^t]^-_{\mathcal{K}_f^t}) + \epsilon \cdot (y_f^t - [y_f^t]^-_{\mathcal{K}_f^t}) \le \tau_f^*([y_f^t]^-_{\mathcal{K}_f^t}) + 2\epsilon + \epsilon = \tau_f^*([y_f^t]^-_{\mathcal{K}_f^t}) + 3\epsilon,$$

$$r_f^t = \tau_f^t([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) + \epsilon \cdot (y_f^t - [y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) \ge \tau_f^*([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) - 2\epsilon - \epsilon = \tau_f^*([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) - 3\epsilon,$$

As τ_f^* is nondecreasing (Assumption 7.4.3), we have

$$l_f^t - 3\epsilon \le \tau_f^*([y_f^t]^-_{\mathcal{K}_f^t}) \le \tau_f^*(y_f^t) \le \tau_f^*([y_f^t]^+_{\mathcal{K}_f^t \bigcup \{1\}}) \le r_f^t + 3\epsilon.$$

Thus there exists tax $\widetilde{\tau}^t$ such that $\widetilde{\tau}_f^t(y_f^t)$ satisfies the condition of Lemma F.2.4 and $\left|\tau_f^*(y_f^t) - \widetilde{\tau}_f^t(y_f^t)\right| \le 3\epsilon$ for all $f \in \mathcal{F}$. x^t is the Nash equilibrium for tax $\widetilde{\tau}^t$, we have

$$\forall i \in [m], a, a' \in \mathcal{A}_i, \sum_{f \in a} c_f(y_f^t) + \widetilde{\tau}_f^t(y_f^t) \le \sum_{f \in a'} c_f(y_f^t) + \widetilde{\tau}_f^t(y_f^t), \text{ if } x_{i,a}^t > 0.$$

Thus we have

$$\forall i \in [m], a, a' \in \mathcal{A}_i, \sum_{f \in a} c_f(y_f^t) + \tau_f^*(y_f^t) \le \sum_{f \in a'} c_f(y_f^t) + \tau_f^*(y_f^t) + 6F\epsilon, \text{ if } x_{i,a}^t > 0.$$

By Lemma F.1.1, $\Psi(y_f^t) - \min_{y \in \mathcal{Y}} \Psi(y) \le 6F\epsilon$. □

Lemma F.2.6. *If Algorithm 14 output $\widetilde{\tau}^t, \widetilde{f}^t, \text{sign}^t$ at round t, then we have*

$$0 < \left|y_{\widetilde{f}^t}^t - \dot{y}_{\widetilde{f}^t}^t\right| \le \Delta.$$

Proof. First, we prove $\left|y_{\widetilde{f}^t}^t - \dot{y}_{\widetilde{f}^t}^t\right| > 0$. We consider the following two cases.

(1) Algorithm 14 return at Line 5. As we have $0 < y_i(\widetilde{f}^t) < w_i$, there exists $a, a' \in \mathcal{A}_i$ such that $x_{i,a} > 0, x_{i,a'} > 0$ and $\widetilde{f}^t \in a, \widetilde{f}^t \notin a'$. Suppose $y_{\widetilde{f}^t}^t = \dot{y}_{\widetilde{f}^t}^t$. Then by Lemma 7.5.8, we have $y^t = \dot{y}^t$ as τ^t and $\dot{\tau}^t$ only differ in facility $\widetilde{f}^t$. As a result, x^t is Nash equilibrium for tax $\dot{\tau}^t$. However, x^t is the Nash equilibrium for tax τ_f^t implies

$$\sum_{f \in a} c_f(y_f^t) + \tau_f^t(y_f^t) = \sum_{f \in a'} c_f(y_f^t) + \tau_f^t(y_f^t).$$

As τ^t and $\dot{\tau}^t$ only differ in facility $\widetilde{f}^t$ and $\widetilde{f}^t \in a, \widetilde{f}^t \notin a'$, we have

$$\sum_{f\in a} c_f(y_f^t) + \dot{\tau}_f^t(y_f^t) \neq \sum_{f\in a'} c_f(y_f^t) + \dot{\tau}_f^t(y_f^t),$$

which means x^t is not the Nash equilibrium for tax $\dot{\tau}$. By contradiction, we have $y^t_{\widetilde{f}^t} = \dot{y}^t_{\widetilde{f}^t}$.

(2) Algorithm 14 return $\widetilde{\tau}^u, \widetilde{f}$, sign at Line 23 or Line 31. As there exists $j \in [m]$ such that $\mathrm{Gap}_j(x, c+\widetilde{\tau}^{u+\mathrm{sign}\cdot\epsilon}) < 0$, x is not a Nash equilibrium under tax $\dot{\tau}^t$. Let $\ddot{\tau}^t : \ddot{\tau}_f^t = \tau_f^t \bigcup (y_f^t, \widetilde{\tau}_f^u)$ for $f \in \mathcal{F}$. Then $\dot{\tau}^t$ and $\ddot{\tau}^t$ only differs in $\widetilde{f}$ and x is the Nash equilibrium under tax $\ddot{\tau}^t$. By applying Lemma 7.5.8 with $\dot{\tau}^t$ and $\ddot{\tau}^u$, we have $y^t_{\widetilde{f}^t} = \dot{y}^t_{\widetilde{f}^t}$.

Second, we prove $\left|y^t_{\widetilde{f}^t} - \dot{y}^t_{\widetilde{f}^t}\right| \leq \Delta$. (1) Algorithm 14 return at Line 5. Suppose we have $\left|y_f^t - \dot{y}_f^t\right| > \Delta$. By the tax design, the (sub)gradient of the tax $(\tau_f^t)'(u) \geq \epsilon$ for $u \in [0,1]$. As a result, $\Phi(y, c+\tau^t)$ is ϵ-strongly convex by Lemma 7.5.7. As $y^t = \mathrm{argmin}_{y\in\mathcal{Y}} \Phi(y; c+\tau^t)$, we have

$$\Phi(\dot{y}^t; c+\tau^t) - \Phi(y^t; c+\tau^t) > \epsilon\Delta^2/2.$$

However, we have $\left|\Phi(y; c+\tau^t) - \Phi(y; c+\dot{\tau}^t)\right| \leq \delta$ for all $y \in \mathcal{Y}$. Thus we have

$$\Phi(\dot{y}^t; c+\dot{\tau}^t) - \delta \leq \Phi(y^t; c+\dot{\tau}^t) - \delta/2 \leq \Phi(y^t; c+\tau^t) \leq \Phi(\dot{y}^t; c+\tau^t) \leq \Phi(\dot{y}^t; c+\dot{\tau}^t) + \delta.$$

Comparing to the inequality above, we have $2\delta > \epsilon\Delta^2/2$, which is incorrect by the definition of δ. By contradiction, we have $\left|y_f^t - \dot{y}_f^t\right| \leq \Delta$.

(2) Algorithm 14 return $\widetilde{\tau}^u, \widetilde{f}$, sign at Line 23 or Line 31. Let $\ddot{\tau}^t : \ddot{\tau}_f^t = \tau_f^t \bigcup (y_f^t, \widetilde{\tau}_f^u)$ for $f \in \mathcal{F}$. Then x is the Nash equilibrium under tax $\ddot{\tau}^t$. Let $\ddot{\tau} : \ddot{\tau}_f^t = \tau_f^t \bigcup (y_f^t, \widetilde{\tau}_f)$ for all $f \in \mathcal{F}$. By the definition of $\ddot{\tau}^t$ and the feasible range $\widetilde{\tau}_f \in [l_f, r_f]$, the subgradient of $\ddot{\tau}_f^t$ is lower bounded by ϵ. As a result, $\Phi(\cdot; c+\ddot{\tau}^t)$ is ϵ-strongly convex on $[0,1]$. We can prove $\left|y_f^t - \dot{y}_f^t\right| \leq \Delta$ by following the analysis for case (1) and replacing τ^t with $\ddot{\tau}^t$.

□

Proposition 7.5.3. *If Algorithm 14 return* False *at round t, then tax τ^t is an $6\epsilon F$-optimal tax. If Algorithm 14 output $\widetilde{\tau}^t, \widetilde{f}^t, \mathrm{sign}^t$ at round t, then we have*

$$0 < \left|y^t_{\widetilde{f}^t} - \dot{y}^t_{\widetilde{f}^t}\right| \leq \Delta.$$

Proof. This is directly from Lemma F.2.5 and Lemma F.2.6. □

Lemma F.2.7. *Algorithm 13 return* False *in at most KF rounds.*

Proof. By Lemma F.2.6 and the update rule (7.1), if Algorithm 14 return $\widetilde{\tau}^t, f, \text{sign}$ at round t, then we will have one more known point, i.e., $\sum_{f\in\mathcal{F}} \mathcal{K}_f^{t+1} = \sum_{f\in\mathcal{F}} \mathcal{K}_f^t + 1$. As $\mathcal{K}_f^t \subseteq Ł$ for all $f \in \mathcal{F}$ and $|Ł| = K + 1$, we proved the lemma.

□

Theorem 7.5.4. *Under Assumption 7.3.1 and Assumption 7.4.3, Algorithm 13 will output a $6\epsilon F$ tax within $T \leq 2F\beta/\epsilon$ rounds. In addition, each round has at most two tax realizations.*

Proof. The proof is directly from Proposition 7.5.3 and Lemma F.2.7. □

F.3 Computation Complexity

In this section, we discuss the computation complexity of Algorithm 13 and Algorithm 14. We will show that these two algorithms can be implemented with $\widetilde{O}(\text{poly}(A, F, m))$ complexity for each round. For network congestion games, the computation complexity can be sharpened to $\widetilde{O}(\text{poly}(V, E, m))$, avoiding the dependence on A that can be exponential in V and E.

F.3.1 General Congestion Games

For Algorithm 13, we compute/update the value of the cost/tax function for each facility. As we use the dictionary data structure, computing value and updating value only have $O(\log K) = O(\log \beta/\epsilon)$ complexity. As a result, the complexity of one round in Algorithm 13 is $\widetilde{O}(F)$.

For Algorithm 14, $x \in \phi^{-1}(y)$ is a Caratheodory decomposition problem and can be formulated as a linear program with A variables, $F+m$ equation constraints and A inequality constraints (Proposition F.4.1), which can be solved in polynomial time [Cohen et al., 2021b].

The bottleneck is in computing $u = \text{argmax}_u\{u : \text{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j \in [m]\}$ for $\widetilde{\tau}^u : \widetilde{\tau}_f^u = u, \widetilde{\tau}_{\mathcal{F}\setminus\{f\}}^u = \tau'_{\mathcal{F}\setminus\{f\}}$, $f \in \bar{F}_i$. For simplicity, we use the notation: $\widetilde{c}^u = c + \widetilde{\tau}^u$ as the cost with tax $\widetilde{\tau}^u$. By Definition 7.5.9 and the definition of action cost, we have

$$\text{Gap}_j(x, c + \widetilde{\tau}^u) = \min_{a:x_{j,a}=0} \widetilde{c}_a^u - \max_{a:x_{j,a}\neq 0} \widetilde{c}_a^u. \tag{F.1}$$

For action cost $\widetilde{c}_a^u$, if $f \in a$, it is a linear function w.r.t. u in the form of $u + C$ for some constant C. Otherwise, it is a constant w.r.t. u. As a result, we can determine the function $\widetilde{c}_a^u$ with $O(F)$ computation as we only need to compute $\widetilde{c}_a^{\tau'_f}$ to decide the constant. Then we can compute $\mathrm{Gap}_j(x, c + \widetilde{\tau}^u)$ in closed form and compute $u_j = \mathrm{argmax}_u\{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0\}$ with $O(AF)$ complexity. Finally, $u = \min_{j\in[m]} u_j$ can be computed with $\emptyset(mAF)$ complexity. Similarly, $u = \mathrm{argmin}_u\{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j \in [m]\}$ has $\widetilde{O}(mAF)$ computation complexity.

F.3.2 Network Congestion Games

For network congestion games, Algorithm 14 can be implemented by applying shortest path algorithms on a modified network, thus avoiding the dependence on A. We will apply Dijkstra's algorithm with $\widetilde{O}(V + E)$ complexity while other shortest path algorithms can be used as well.

First, the Caratheodory decomposition $x \in \phi^{-1}(y)$ can be done efficiently with $O(VE + E^2)$ steps similar to the decomposition algorithm in [Panageas et al., 2023]. While their algorithm is for the flow polytopes with one commodity, it can be directly generalized to the multi-commodity case. We defer the algorithm and analysis to Appendix F.4.

For $u = \mathrm{argmax}_u\{u : \mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0, \forall j \in [m]\}$, the computation complexity can be boosted to $\widetilde{O}(m(E + V))$. To achieve this, we consider how (F.1) changes as u increases from τ'_f to r_f. By Algorithm 14, we have $\mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0$ when $u = \tau'_f$ as otherwise the algorithm ends at the previous iteration. In addition, facility f either has none of the Nash load or has all of the Nash load for facility j according to the algorithm design. For the first case, the in-support action costs will not change as u increases. $\mathrm{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0$ always holds as the off-support action costs are nondecreasing w.r.t. u.

For the second case (all in-support actions use f), the in-support action costs take the form of $u + C$ and C can be determined by applying shortest path algorithm with edge weight $c + \widetilde{\tau}^{\tau'_f}$. For off-support action cost, we observe that

$$\min_{a:x_{j,a}=0} \widetilde{c}_a^u = \min\Big\{\min_{a:x_{j,a}=0, f\in a} \widetilde{c}_a^u, \min_{a:x_{j,a}=0, f\notin a} \widetilde{c}_a^u\Big\} = \min\Big\{\min_{a:x_{j,a}=0, f\in a} \widetilde{c}_a^u, \min_{a: f\notin a} \widetilde{c}_a^{\tau'_f}\Big\}, \tag{F.2}$$

where the second equation is from that the action cost does not depend on u and $x_{j,a} = 0$

if $f \notin a$. The first term in (F.2) grows linearly w.r.t. u as $\widetilde{f} \in a$, so it is always larger than the in-support action cost. The second term in (F.2) is the shortest path length for commodity j that does not use facility f, which can be computed as the shortest path in the network after removing edge f. As a result, $u_j = \text{argmax}_u\{u : \text{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0\}$ can be computed with $O(E + V)$ complexity. Then the complexity for computing $u = \min_{j\in[m]} u_j$ is $\widetilde{O}(m(E + V))$.

Similarly, $u_j = \text{argmin}_u\{u : \text{Gap}_j(x, c + \widetilde{\tau}^u) \geq 0\}$ can be reduced to solving the shortest path that must use edge f in the network. We consider how (F.1) changes as u decreases from τ_f to l_f. Initially, the gap is nonnegative. If f has all of the Nash load, then the in-support action cost is a linear function $u + C$ and it decreases at least as fast as the first term. As a result, the gap is always nonnegative. Otherwise, f has none of the Nash load and in-support action costs remain constant.

We notice the following equation:

$$\min_{a:x_{j,a}=0} \widetilde{c}_a^u = \min\{\min_{a:x_{j,a}=0, f\in a} \widetilde{c}_a^u, \min_{a:x_{j,a}=0, f\notin a} \widetilde{c}_a^u\} = \min\{\min_{a:f\in a} \widetilde{c}_a^u, \min_{a:x_{j,a}, f\notin a} \widetilde{c}_a^{\tau'_f}\}, \tag{F.3}$$

where the second equation is from that $f \in a$ implies a is off-support ($x_{j,a} = 0$) and $f \notin a$ implies $\widetilde{c}_a^u$ is independent of u. The second term in (F.3) is a constant and is always greater than the in-support action cost. The first term in (F.3) is a linear function u and it can be determined by computing the shortest path that always uses $\widetilde{f}$ and with edge weights $c+\widetilde{\tau}^{\tau_f}$. This subproblem can be solved by applying the shortest path algorithm twice: the first one is to connect the source node and the starting node of $\widetilde{f}$, and the second one is to connect the end node of $\widetilde{f}$ and the target node. As a result, the complexity for $u = \max_{j\in[m]} u_j$ is $\widetilde{O}(m(E + V))$ as well. Thus the computation complexity for Algorithm 14 in network congestion games is $O(VE + E^2 + mV + mE)$.

F.4 Missing Proofs in Section F.3

Proposition F.4.1. *Finding $x \in \phi^{-1}(y)$ can be formulated as the following linear program.*

$$\min_{x\in\mathbb{R}^A} 1$$
$$s.t.\ y = \sum_{i\in[m]} \sum_{a_i\in\mathcal{A}_i} x_{i,a_i} a_i$$

$$w_i = \sum_{a_i \in \mathcal{A}_i} x_{i,a_i}, \forall i \in [m]$$
$$x_{i,a_i} \geq 0, \forall i \in [m], a_i \in \mathcal{A}_i$$

Proof. The second and third constraints guarantees x is a feasible strategy. The first constraint indicates $y = \phi(x)$. As a result, any feasible point of the program is a solution of $\phi^{-1}(y)$. □

Algorithm 26 Efficient Computation of Flow Decomposition (Modified from [Panageas et al., 2023])

1: **Input**: A load $y \in \mathcal{Y}$.
2: $x_{i,a} = 0$ for all $i \in [m]$ and $a \in \mathcal{A}_i$.
3: **while** $\exists f : y_f > 0$ **do**
4: Let $A = \{f : y_f > 0\}$.
5: Let $f_{\min} = \text{argmin}_{f \in A}\, y_f$ and $y_{\min} = \min_{f \in A} y_f$.
6: Let a be a (s_i, t_i) path of network $G(V, A)$ with $f_{\min} \in p$.
7: Let $x_{i,a} = y_{\min}$, $y_f = y_f - y_{\min}$ if $f \in a$.
8: **end while**

Proposition F.4.2. *Algorithm 26 can output a Caratheodory decomposition of y within E steps.*

Proof. During the algorithm, load y will always be nonnegative: $y_f \geq 0, \forall f \in \mathcal{F}$. For each round, we will have $y_{f_{min}}$ reduced to 0. As a result, the algorithm will end within at most E rounds.

We only need to prove that path a always exists in Line (6) for each round. First, y always remains a multi-commodity flow as Line (7) will not affect the law of conservation in the network. By flow decomposition theorem, there exists simple paths $a_1, a_2, \cdots, a_p$ such that

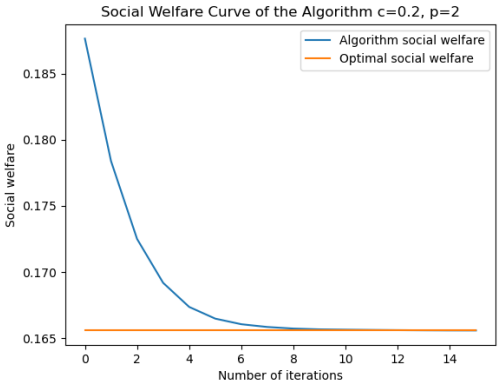
$$y = \sum_{i \in [p]} w_i a_i,$$

where $w_i > 0$ are positive flow weights. As $y_{f_{\min}} > 0$, there exists a_i such that $f_{\min} \in a_i$. Then for any $f \in a_i$, $y_f \geq w_i > 0$. As a result, path a exists for Line (6). □

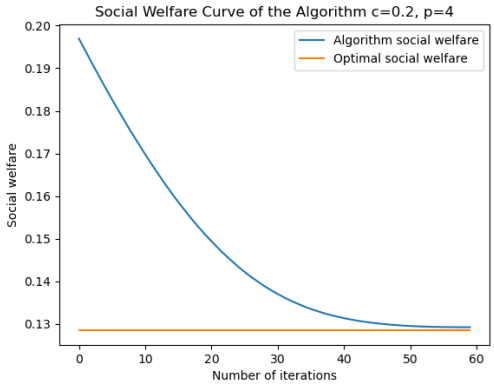
F.5 Experiments

We implemented our algorithm and conducted experiments on a classic example known as the nonlinear variant of Pigou's example [Nisan et al., 2007a]. Concretely, nonlinear variant of Pigou's example is a routing game with one source node s and one target node t. There are two edges connecting s and t. One edge has constant cost $c_0(x) = c, \forall x \in [0, 1]$ for some $c \in [0, 1]$, and the other edge has polynomial cost $c_1(x) = x^p$. One important property of such games is the price of anarchy grows without bound as $p \to \infty$, which urges proper tax to induce socially optimal behavior.

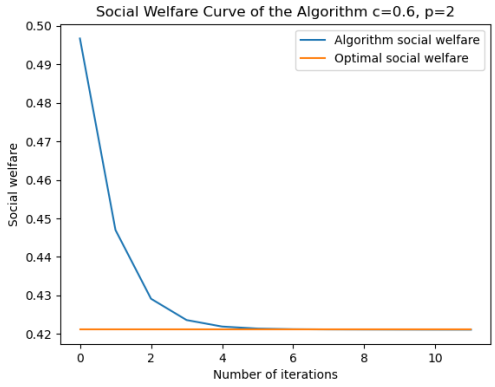
We apply our algorithm to learn the optimal tax with different c_0 and p. As we can see, the social welfare quickly converges to the optimal one. Another important observation is the learned tax function does not uniformly converge to the marginal cost tax, which is reasonable as accurate estimate is only necessary around the Nash equilibrium induced by the tax.



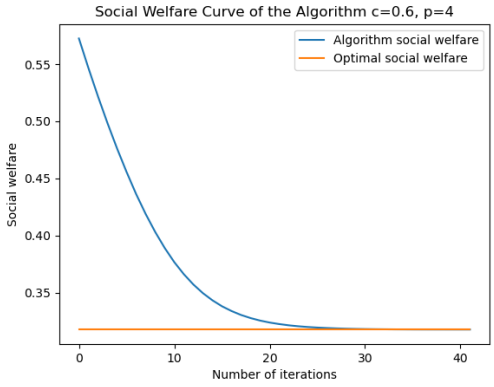
(a) c=0.2, p=2



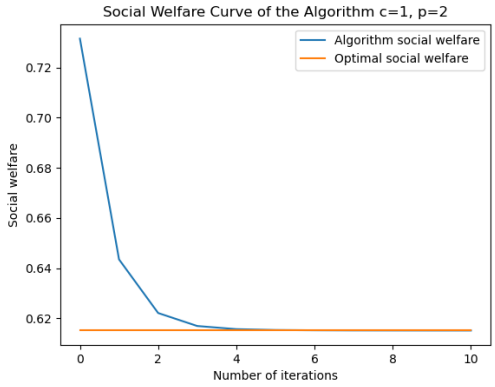
(b) c=0.2, p=4



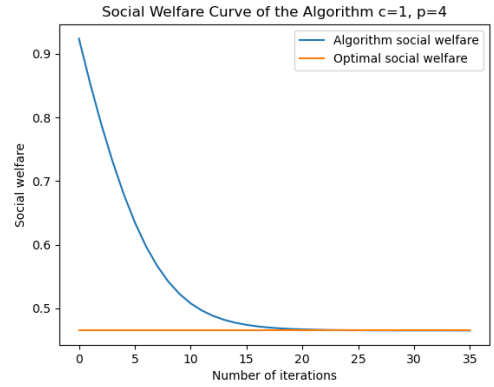
(c) c=0.6, p=2



(d) c=0.6, p=4

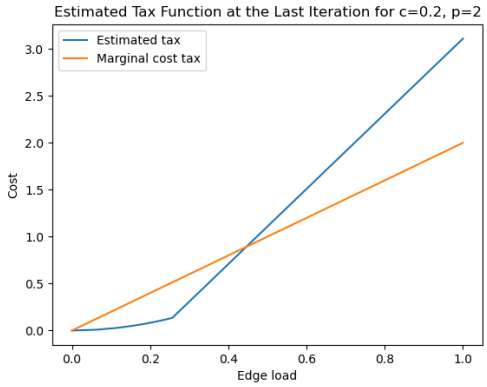


(e) c=1, p=2

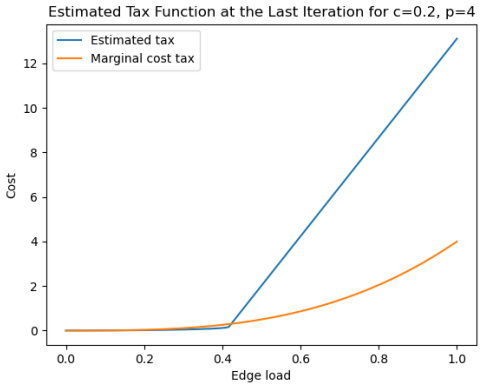


(f) c=1, p=4

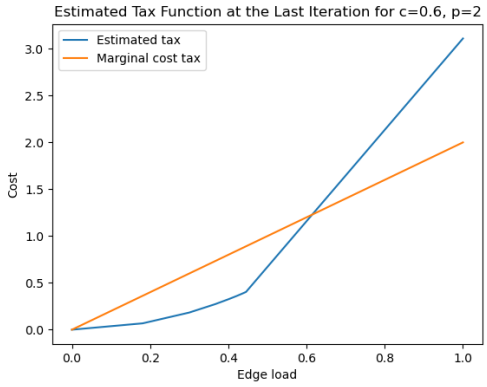
Figure F.1: Social Welfare Curves of the Algorithm for various values of c and p. We can observe that the social welfare converges to the optimal one quickly.



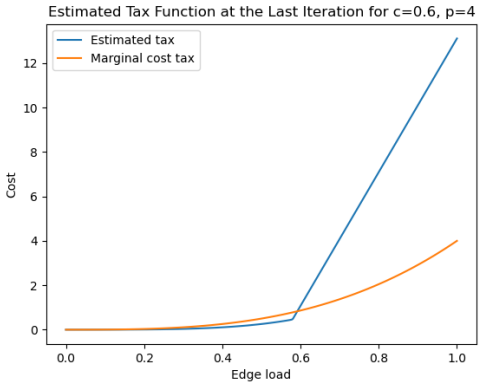
(a) c=0.2, p=2



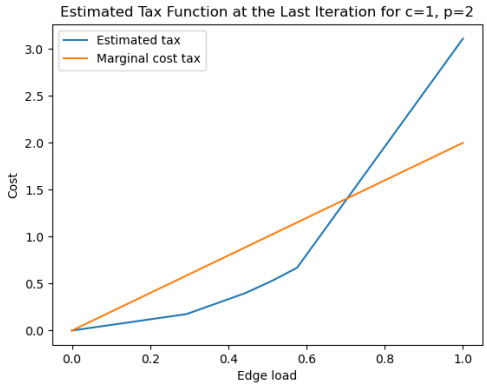
(b) c=0.2, p=4



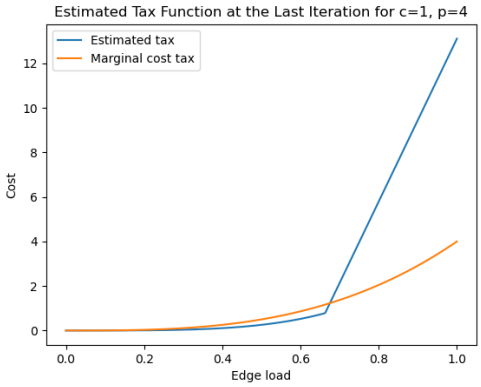
(c) c=0.6, p=2



(d) c=0.6, p=4



(e) c=1, p=2



(f) c=1, p=4

Figure F.2: Estimated Tax Functions at the Last Iteration for various values of c and p. The estimation is not uniformly accurate but they are accurate at the induced Nash equilibrium.