Toward Robust, Reliable, and Generalizable Models for Tabular Data

Josh Gardner

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

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Ludwig Schmidt, Chair

Tim Althoff

Jamie Morgenstern

Program Authorized to Offer Degree:
The Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

Toward Robust, Reliable, and Generalizable Models for Tabular Data

Josh Gardner

Chair of the Supervisory Committee:
Assistant Professor Ludwig Schmidt
Computer Science and Engineering

Tabular data – spreadsheet-type data with rows and columns – is widely used across many domains and real-world applications, from finance to healthcare to natural and social sciences. However, tabular data modeling has received far less attention, and seen far less rapid progress, than modern AI for other forms of data (such as images, natural language, and audio). In this dissertation, I conduct a series of studies to empirically assess the conditions under which modern tabular methods succeed and fail in prediction tasks, and leverage the results of these studies to develop a new approach to tabular modeling. Specifically, I first study the properties of tabular models under (1) subpopulation shift, and (2) distribution shift. The results show (among other findings) that simple, standard models with low in-distribution test error achieve best-in-class robustness under both forms of shift. Then, I leverage the findings of these studies to build a first-of-its-kind, state of the art foundation model for tabular data prediction, TabuLa. TabuLa exhibits the ability to perform zero-shot generalization to new tables, a capability not possible with existing state-of-the-art tabular models such as XGBoost and TabPFN. TabuLa also outperforms these state-of-the-art models in few-shot generalization, with sample efficiency up to 16x better than existing methods. Collectively, these works point toward new and exciting future directions for tabular data and open up opportunities for high-quality, general-purpose tabular AI models.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

I am grateful to my parents, whose support and encouragement at each step of my educational journey also gave me the courage to take the leap from middle school teacher to graduate student. My entire family provided support throughout my PhD and believed in me even when I was unsure of myself. Thank you.

I had the opportunity to learn from many incredible researchers across academia and industry during my PhD. Collaborating with each of you made me stretch higher, reach farther, and think deeper. This includes colleagues from across academia (Christopher Brooks, Ryan Baker, Juan Carlos Perdomo, Jamie Morgenstern, Sewoong Oh, Zoran Popovic, Rene Kizilcec, Renzhe Yu), the Google Magenta team (Ian Simon, Curtis Hawthorne, Jesse Engel, Ethan Manilow, Adam Roberts), and Spotify (Rachel Bittner, Simon Durand, Daniel Stoller). The skills I learned from all of these collaborators were the secret ingredients in this scholarly soup, even if not all are written in the final dissertation recipe.

To the PhD students who were there to share ideas, coffee, or to commiserate about Reviewer xD5F's brutal reviews: my imposter syndrome thanks you. This includes Anant Mittal, Matt Whitehill, Alex Fang, Matt Jordan, Mitchell Wortsman, Thao Nguyen, Jon Hayase, Anas Awadalla, Mike Merrill, and Gabriel Ilharco.

Ellen, you were the tailwind in the final two-year stretch of this marathon, supporting me when I needed it and always pushing me to consider the broader impact of my work.

Kevin Todd, thank you for being my PhD support system before, during, and "after" COVID. You were a constant reminder of the light at the end of the PhD tunnel.

Finally, I would like to thank my advisor, Ludwig Schmidt, for taking a chance on a course project that snowballed into this dissertation. Ludwig deploys a rare combination of positive encouragement and relentlessly high standards. I feel truly grateful to have worked with him. Ludwig's mentorship shaped me as a thinker, writer, and scientist – even if we may never see eye to eye on the superiority of Cervelo and Castelli cycling gear.

Chapter 1

INTRODUCTION

While models for images, text, and audio have advanced rapidly over the past seven years since the introduction of the Transformer architecture, tabular data modeling has seen considerably less impact.

This lack of advancement is despite the ubiquity of tabular data in many high-impact domains. For example: electronic health records (EHR) are commonly stored in tabular format (and their adoption and use was mandated in 2009 by the American Recovery and Reinvestment Act); in 2024, the Office of Management and Budget (OMB) directed federal agencies to transition to fully electronic records, leading to an expected future spike in already-abundant government tabular data records; and widespread adoption of automated tooling in the natural sciences has led to large-scale tabular data collection as a byproduct of many experiments. In these and many more application areas, tabular data represents a set of critical applications where major advances in the science and application of machine learning systems for tabular data could lead to significant benefits.

This dissertation presents a series of studies designed to empirically assess the current progress of tabular machine learning methods with respect to (1) their *subgroup robustness* and (2) their generalization, or *distribution shift*, capabilities. Then, this dissertation uses the insights from these empirical investigations to (3) propose a novel end-to-end approach to *transfer learning* for tabular data. Collectively, the works in this study introduce novel datasets, evaluation benchmarks, and trained models which expand both our understanding of effective methods for tabular prediction, and our ability to apply tabular prediction models in new scenarios. This work represents three small steps forward in the tabular domain, and I hope that its contents enable future steps for the machine learning and artificial intelligence research communities.

Chapter 2

SUBGROUP ROBUSTNESS GROWS ON TREES: AN EMPIRICAL BASELINE INVESTIGATION

2.1 Introduction

Over the past decade, the field of machine learning (ML) has seen a dramatic expansion along two related lines. On one hand, concerns about the fairness of ML models, and more broadly their performance on data outside the training distribution, have grown [129]. Both theoretical and empirical works have raised these concerns, demonstrating the vulnerability of models to learn biases from data or suffer performance drops under distribution shifts [108, 177, 86]. On the other hand, an abundance of methods have been proposed to address these limitations. These include *fairness* methods used to equalize metrics across groups, as well as *distributional robustness* methods which optimize for a worst-case distribution within a bounded distance of the training distribution.

Our work begins from the observation that, while "fairness" and "robustness" are distinct concepts, methods in both areas often have similar goals. In particular, they share two (sometimes implicit) goals: First, models should have low *disparity* (variation in performance over all subgroups should be minimized; cf. [208, 40, 55, 106]). Second, models should have good *worst-group performance* (the lowest accuracy over all subgroups should be maximized; cf. [204, 160]). As a consequence of these two criteria, both fairness and robustness typically entail improving *subgroup robustness*.

Our work focuses on subgroup robustness in the context of *tabular* data, for several reasons. First, tabular data is widely used in practice, being the most common format in areas with important fairness impacts such as medicine, finance, and recommender systems [98, 170]. Second, tabular data often directly encodes sensitive attributes with respect to which subgroup robustness is desired (race, gender, income level), and these attributes are often declared by the individuals represented in the dataset. This is an important difference compared to crowdsourced group annotations (e.g., gender or race for image datasets), which can be unreliable [51] or fail to reflect the lived experiences of the individuals represented [47]. Finally, tabular

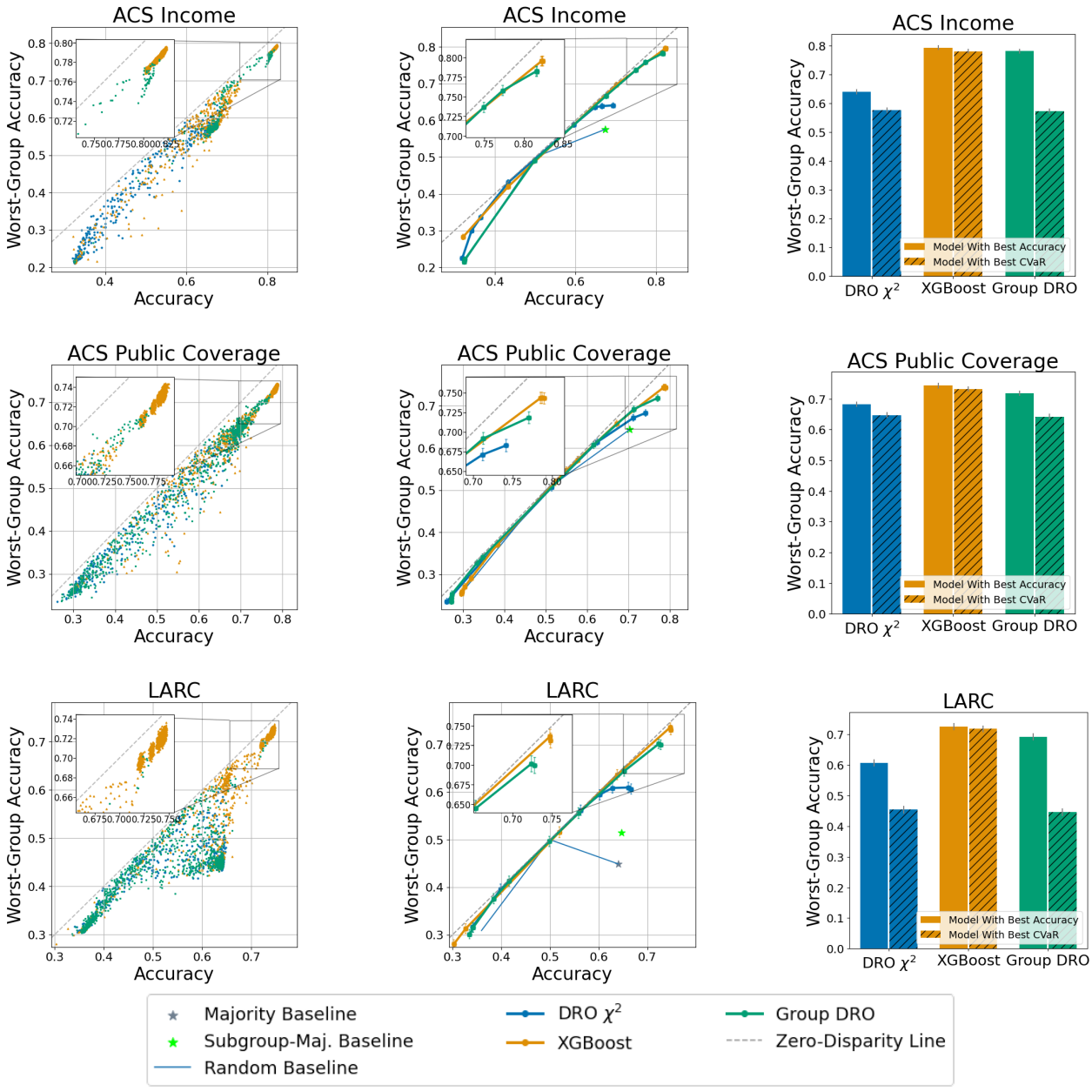


Figure 2.1: Results from three datasets in our study. (a) Left: Tree-based methods such as XGBoost show similar subgroup robustness, with sometimes better overall performance, as robustness-enhancing or disparity-mitigation methods. (b) Center: Model performance frontiers corresponding to (a) show similar accuracy-robustness frontiers for XGBoost and DRO/Group DRO. (c) Right: Tree-based methods' worst-group accuracy is robust to model selection metrics.

data is challenging to model. It is often heterogeneous, containing mixed data types; it lacks the spatial, linguistic, or temporal structures common to other data (image, text, audio) where machine learning has seen dramatic progress over the past decade; and it can be relatively small, containing personal information which cannot be scraped at Internet scale.

In this work, we conduct a series of experiments to jointly compare methods from the fairness and robustness literature. Despite the fact that many works often use similar metrics and datasets, there is a lack of large-scale empirical studies comparing relevant methods to each other, and to strong baselines. This lack of effective baselines is particularly concerning in light of recent works across several other areas of machine learning which have shown that simple baselines can perform surprisingly well against state-of-the-art techniques [116].

Our results show that modern tabular-data baselines can outperform even well-tuned state-of-the-art robustness- or fairness-enhancing methods with a fraction of the computational cost. Our analysis includes a set of tree-based models which achieve strong performance on tabular data but are absent from prior works on robustness, which largely only compare to deep learning-based approaches despite their tendency to perform poorly on tabular data [98]. In our experiments, we endeavor to tune both the "fair" or "robust" models and a representative suite of baseline methods for tabular data to obtain estimates of the empirical accuracy-subgroup robustness Pareto frontier for each model. To this end, we explore a wide space of architectures, regularization schemes, and training hyperparameters, across eight tabular datasets and 17 model classes. Our work is an empirical contribution concerned not with proposing new methods, but with conducting a novel and rigorous evaluation of existing methods for learning subgroup-robust models from tabular data. Our contributions are as follows:

- **Trees are subgroup-robust learners:** While previous works have shown that modern tree-based ensembles have strong average-group performance (see Section 2.2.3), we show that these models are also surprisingly robust to subgroup shift. Tree-based models achieve this robustness despite lacking any explicit robustness intervention and using only average-case classification losses, achieving competitive or better performance against state-of-the-art methods for fairness and robustness (Figure 2.1a,2.1b) over a large hyperparameter and model search space. To support the above analysis, we apply several techniques for the analysis of hyperparameter sweeps such as model performance frontier curves (Figure 2.1b,2.3) and hyperparameter sensitivity analysis (Figures 2.7, A.1, A.2).

- **Trees show show consistent performance across accuracy and robustness metrics:** We empirically investigate the relationship between three types of metrics (accuracy, robustness, fairness). We show that metrics within these groups often agree, but the relationships across metric groups (i.e. robust CVaR risk and accuracy) are inconsistent. One consequence of this finding is that model selection techniques affect tree-based models and "robust" or fairness-enhancing methods differently: tree-based methods show consistent subgroup robustness across metrics (Figure 2.1c), while robust and fairness-enhancing methods are "metric-brittle" and tend to display poor subgroup robustness according to other metrics (e.g. worst-group accuracy).
- **Trees are less sensitive to hyperparameters and less costly to train:** We show that, in addition to their improved robustness when fully tuned, trees also require less tuning and are less costly to train. We demonstrate that trees have decreased sensitivity to hyperparameters – even when accounting for hyperparameter settings which were part of our initial grid but performed poorly across all datasets – and that these models require considerably less compute and financial cost to train when compared to deep learning-based robust learning techniques.

In particular, our results highlight the importance of the underlying model class for subgroup robustness in tabular data. Our results suggest that subgroup robustness interventions based on the loss-based perspective alone may be limited, particularly because existing loss-based methods for robustness are incompatible with nondifferentiable functions such as tree-based ensembles. We provide further discussion in Section 2.7.

We provide code to reproduce our experiments, along with an interactive tool to explore the best-performing hyperparameter configurations, at `https://github.com/jpgard/subgroup-robustness-grows-on-trees`.

2.2 Related Work

2.2.1 Fairness-Enhancing Methods for Supervised Learning

A wide variety of works have addressed fairness in the context of machine learning, where "fairness" is often measured by the equalization of some metric over groups [16, 44]. Most methods can be characterized as performing either pre- in-, or post-processing, which attempt to ensure fairness considerations are met in different stages of the modeling lifecycle. *Preprocessing* [203, 34] attempts to modify the input data to meet fairness criteria, while preserving the structure of the inputs and their relationship to the prediction

target. *Inprocessing* uses a modified training procedure to explicitly optimize for a fairness-aware objective during model training. This includes using constrained or reduction-based optimization [1, 202], or including explicit regularizers designed to encourage fairness [18, 23, 106, 147]. *Postprocessing* [81, 139] operates on the predictions of a model, modifying them to achieve fairness criteria.

The impact of fairness under subpopulation shift is analyzed in [118], which shows theoretically that enforcing fairness during training can harm or improve the model, under certain conditions, and [171] conducts a causal analysis of fairness under covariate shift. Perhaps the work most closely related to the current study is Friedler et al. [63], which evaluates a set of pre-, in-, and postprocessing algorithms, demonstrating that many of these techniques show instability (variations in performance over different train-test splits) and that several fairness metrics are empirically correlated. Friedler et al. [63] do not, however, evaluate robust learning techniques, modern tree-based techniques (besides CART), or neural methods. Empirical evaluation of recent methods is needed.

2.2.2 Distributionally Robust Learning

While efforts in robust optimization date back several decades [21], recent works have adapted and extended robust learning approaches to deep learning. For example, several variants of distributionally robust optimization (DRO) have been proposed for the training of neural networks with robustness guarantees [91, 113, 160, 158, 204, 55]. While these approaches are not always explicitly oriented toward fairness, they are frequently evaluated in terms of their performance benefits for minimizing performance gaps between sensitive subgroups in real-world data [204, 160, 83].

A particular form of shift (and robustness) evaluated by these works is *subgroup shift* (also called subpopulation shift) [129], where the balance of subgroups in the data shifts between training and testing. Evaluating performance on individual subgroups is an extreme version of subgroup shift. In the context of fairness, these subgroups are often defined by one or more discrete sensitive attributes, such as race, gender, or income level, and intersections between these sensitive groups can often identify groups most susceptible to performance disparities [32].

2.2.3 Models for Tabular Data

Deep learning-based approaches largely have not achieved the transformative performance gains on tabular data that they have with other data modalities such as images, text, and audio [98, 28, 170]. The existing state-of-the-art for learning with tabular data is widely acknowledged to be tree-based methods, and in particular gradient-boosted trees [80, 140, 170, 28]. This includes GBM [64, 65], LightGBM [102], XGBoost [38], and CatBoost [52, 142], which often show only small differences in performance between them. Several deep learning-based approaches have recently attempted to close the gap of deep learning-based solutions for tabular data [98, 8, 92, 140, 101]. However, empirical analyses have shown that the reported performance of these methods does not generalize well to other datasets, and gradient boosting methods still achieve better performance with less tuning and a considerably smaller computational budget. For example, [170] and [28] both show, in separate analyses, that gradient boosting consistently outperforms these state-of-the-art tabular deep learning methods across several datasets, and tree-based methods achieve competitive performance with the deep learning approaches proposed in [73, 92].

The overall predictive performance of these tabular methods is evaluated in e.g. [28, 73, 98, 140, 170]. However, the existing literature does not investigate the fairness properties, subgroup performance, or robustness to subgroup shift of tabular data models; nor does it compare to subgroup-robust learning methods. Additionally, despite the widely-known strong performance of these models on tabular data, we are aware of no prior work which compares tree-based methods to robust neural network-based learners, despite the fact that the latter are commonly evaluated on tabular data (e.g. [91, 106, 204, 160]). Notably, [1] evaluates GBM in conjunction with their proposed inprocessing method, but does not evaluate subpopulation robustness; [49] evaluates GBM with fairness methods but does not compare to robust methods and does not tune the GBM with fairness methods.

2.3 Setup

Our work is primarily concerned with empirically evaluating the sensitivity of various supervised learning methods to subpopulation shift. Below, we introduce our formal model, and then describe the main axes of variation in our experiments. This includes (*i*) a large set of models, many of which have not been directly compared in previous works; (*ii*) the hyperparameters used for each model; (*iii*) a suite of eight real-world fairness datasets; and (*iv*) a set of evaluation metrics used across the disparate literature on robustness,

subpopulation shift, and fairness in machine learning.

2.3.1 Preliminaries

Our work evaluates the task of learning a model $f_\theta \in \mathcal{F}$, a function from model class $\mathcal{F}$ parameterized by θ. The parameters are learned from a dataset $D \coloneqq (x_i, y_i)_{i=1}^n \sim \mathcal{P}$ where $\mathcal{P}$ is the data-generating distribution. This matches the case, most common in practice, where a single model is learned by estimating $min_\theta \mathbb{E}(\mathcal{L}(y, f_\theta(x)))$ and deployed for all users. We evaluate the binary classification context, where $y \in \{0, 1\}$ and $f_\theta(x) = \hat{y} \in [0, 1]$ is the score assigned by f to the outcome $y_i = 1$. Each $x_i \in \mathbb{R}^d$ can be partitioned into $x_i = (x_{i,1}, \ldots, x_{i,d-1}, a)$ where a is a sensitive attribute. For notational convenience, we represent a as a single feature here, but in our data, a is typically defined as the concatenation of multiple binary sensitive attributes (e.g. gender = female, race = white). Let $D_a \coloneqq \{(x_i, y_i) \sim \mathcal{P} | a_i = a\}$ denote the subgroup of the dataset with a given sensitive identity.

Following many of the previous works in both fairness [44] and robustness [204, 106], our analysis focuses on the performance of f_θ on each D_a for all $a \in \mathcal{A}$. We are particularly interested in the worst-group performance and the loss disparity, respectively defined as

$$\mathcal{L}_{\text{WorstGroup}} \coloneqq \max_{a \in \mathcal{A}} \mathbb{E}_{D \sim \mathcal{P}}\big[\mathcal{L}_a(f_\theta)\big] \qquad \text{and} \qquad \mathcal{L}_{\text{DISP}} \coloneqq \max_{a, a' \in \mathcal{A}} \mathbb{E}_{D \sim \mathcal{P}}\big[|\mathcal{L}_a(f_\theta) - \mathcal{L}_{a'}(f_\theta)|\big] \tag{2.1}$$

. These metrics can be thought of as assessing the sensitivity of f_θ to subgroup shift, evaluating the worst-group shift from D to D_a (note that subgroup shift is itself a form of covariate shift [129]).

2.3.2 Models

Our goal is to conduct a thorough empirical comparison of the subgroup robustness of a suite of methods for tabular data. We provide a more detailed description of all models used in Section A.2.

- **Fairness-enhancing models**: We evaluate the LFR preprocessing method of [203]; the inprocessing method of[1], and the postprocessing method of [81]. As in [1, 49], we use GBM as the base learner for each model; however, unlike [49], we also tune the base learner parameters.
- **Robust models:** We evaluate both the CVaR and χ^2 constraint forms of DRO via [113]; the CVaR and χ^2 constraint forms of DORO [204]; Group DRO [160]; Marginal DRO [54]; and Maximum Weighted

Loss Discrepancy [106]. Each robust optimization method is used with a multilayer perceptron (MLP), as in most previous works when using tabular data, e.g. [106, 113, 160, 204]. We note that the use of these losses requires performing optimization over a continuous function (such as a neural network), which makes these loss-based robustness interventions incompatible with existing training procedures for e.g. tree-based models.

- **Tree-based models:** We evaluate GBM, LightGBM [102], XGBoost [38], and Random Forest models.
- **Baseline models:** As baselines, we include L_2-regularized logistic regression, Support Vector Machines, and fully-connected neural networks (MLP) with standard (non-DRO) ERM optimization.

2.3.3 Hyperparameter Sweeps

For each model, we conduct a grid search over a large set of hyperparameters. We give the complete set of hyperparameters tuned for each model in Section A.6.

Our hyperparameter search for each model is extensive by design, in order to ensure a reliable comparison of the best-performing models from each class. We expand the initial grid for continuous hyperparameters when a model on the Pareto frontier is at the edge of the grid. When one method includes another as a "base" learner (e.g. LFR preprocessing with GBM, or DRO with MLP), we include the full tuning grid for the base model (i.e. we explore the cross-product of all MLP hyperparameters with all DRO hyperparameters).

We note that previous works tend to either compare against a fixed baseline architecture and training hyperparameters (e.g. [204, 98]), perform manual tuning ([113]), or do not tune hyperparameters of the base model when using fairness-enhancing techniques ([49]).

2.3.4 Metrics

One goal of our work is to assess the empirical relationship between the model evaluation *metrics* used in the robustness, fairness, and classification literatures. Differences in the *training* objectives used across the various fair and robust models in prior work make such a comparison particularly useful. The relationship between these diverse evaluation metrics is not explored in existing work, where several different metrics have been used to compare the performance of models, evaluate their fairness with respect to subgroups, and measure their robustness to various shifts or outliers. We draw several metrics from prior works and

report them for our experiments. We also explore the empirical relationships between different metrics.

For each metric $\mathcal{L}$ (e.g. loss, accuracy, Equalized Odds), we can not only measure the overall empirical performance of a model, but we can measure the *worst-group* loss and *disparity* over subgroups of the data, as defined in Equation (2.1). Worst-group and disparity measures, for various formulations of the loss functions above, are a widely-used measure of fairness and robustness [24, 204, 160, 158, 83, 108]. In particular, we use the $\mathcal{L}_{\text{DISP}}$ and $\mathcal{L}_{\text{WorstGroup}}$ with accuracy as the loss function. Accuracy is widely used in practice, directly interpretable, and invariant to rescaling of the model's predictions.

We also use the **CVaR risk** metric from the robustness literature. CVaR risk is measured over a set of inputs $D = (x_i, y_i)_{i=1}^N \sim \mathcal{P}$ and measures the worst-case weighted loss, according to some loss function ℓ, at level α over the inputs in D:

$$\mathcal{L}_{\text{CVaR}}(D, \mathcal{P}) \coloneqq \sup_{q \in \Delta^N} \sum_{i=1}^{N} q_i \ell(f_\theta; x) \quad \text{s.t. } ||q||_\infty \leq \frac{1}{\alpha N} \tag{2.2}$$

(2.2) is the risk function optimized by the DRO CVaR model [113], but it is also used more widely as a measure of the tail risk of a classifier (cf. [204]).

Additional fairness and robustness metrics are reported in Section A.7 and defined in Section A.3.

2.3.5 Datasets

We evaluate the 17 models over eight datasets covering a variety of prediction tasks and domains. We use two binary sensitive attributes from each dataset, for a total of four nonoverlapping subgroups in each dataset. A summary of the datasets used in this work is given in Table 2.1. We provide additional details on each dataset, along with critical framing and context regarding these prediction tasks and their representations of individuals, in Section A.1.

2.4 Results: Tree Models are Subgroup-Robust Learners

Our main finding is that tree-based models are subgroup-robust learners. Tree-based models match or exceed the performance of distributionally robust and fairness-enhancing models in terms of best overall accuracy, worst-group accuracy, and accuracy disparity on all datasets evaluated.

Dataset	n	**Features**	**Target**	**Sensitive Attributes**
ACS Income	499,350	20	Income $\geq$ 56k	Race, Sex
ACS Public Coverage	341,487	19	Public Ins. Coverage	Race, Sex
Adult	48,845	14	Income $\geq$ 56k	Race, Sex
Behavioral Risk Factors Surveillance System (BRFSS)	175,745	28	Diabetes	Race, Sex
Communities & Crime	1994	113	High Crime	Income Level, Race
COMPAS	7,215	10	Recidivism	Race, Sex
German Credit	1,000	22	Credit Risk	Age, Sex
Learning Analytics Architecture (LARC)	169,032	26	At-Risk (Grade)	URM Status, Sex

Table 2.1: Overview of datasets used.

A summary of our results is shown in Figures 2.1, 2.2, 2.3, and 2.4. Following previous works, our main evaluation metrics are accuracy-based: overall accuracy, worst-group accuracy, and accuracy disparity. Over the wide sweep of datasets and model configurations evaluated, modern tree-based models – GBM, LightGBM, Random Forest and XGBoost – all achieve subgroup performance characteristics on par with, or better than, the set of robust or fairness-enhancing models evaluated. For example, on three of the four largest datasets in our study (ACS Income, ACS Public Coverage, LARC), XGBoost achieves significantly *better* $\mathcal{L}$ and $\mathcal{L}_w g$ than DRO-based methods (Group DRO, χ^2 DRO), as indicated by nonoverlapping Clopper-Pearson CIs ($\alpha = 0.05$).

The only cases where another algorithm achieves *better* maximum overall robustness (as measured by worst-group accuracy) than tree-based methods are DORO CVaR on Adult, and Inprocessing on Communities and Crime (see Figures 2.4, 2.4). We note that in both cases, (i) these differences are not statistically significant, based on the shown Clopper-Pearson confidence intervals at $\alpha = 0.05$, and (ii) these differences occur at points *below* the maximum overall accuracy for each model. In all cases, at the point of maximum overall accuracy, tree-based models achieve equivalent or better worst-group accuracy than all other models

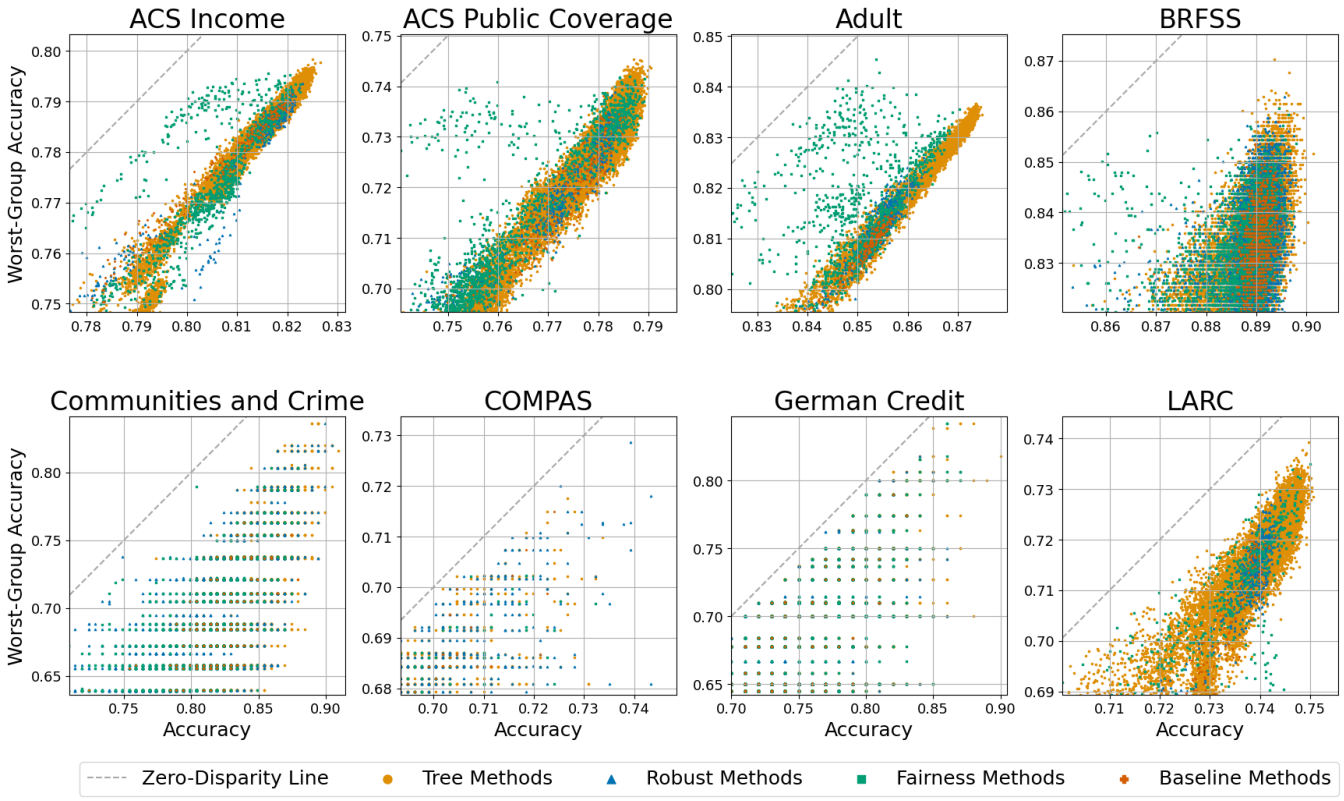


Figure 2.2: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and baseline models over eight tabular datasets. Dashed lines indicate $y = x$, when worst-group accuracy equal to overall accuracy (zero accuracy disparity). See Figures A.5-A.8 for more detailed results by algorithm. Note: "discretization" artifacts are due to small test/dataset sizes.

evaluated, based on Clopper-Pearson confidence intervals at $\alpha = 0.05$.

We note that this is particularly surprising because none of the tree-based models explicitly optimize for robustness, fairness, or subgroup performance in any way; in contrast, the distributionally robust learners and fairness-enhancing techniques explicitly optimize for such criteria and in some cases (DORO, DRO, Group DRO) provide explicit guarantees of various forms of robustness. A similar analysis showing accuracy *disparity* is shown in Figures A.9, A.10, and a complete set of model performance observations from each algorithm and dataset, are in Supplementary Section A.7.

To summarize our results over the large hyperparameter sweeps, we use *model performance frontiers*

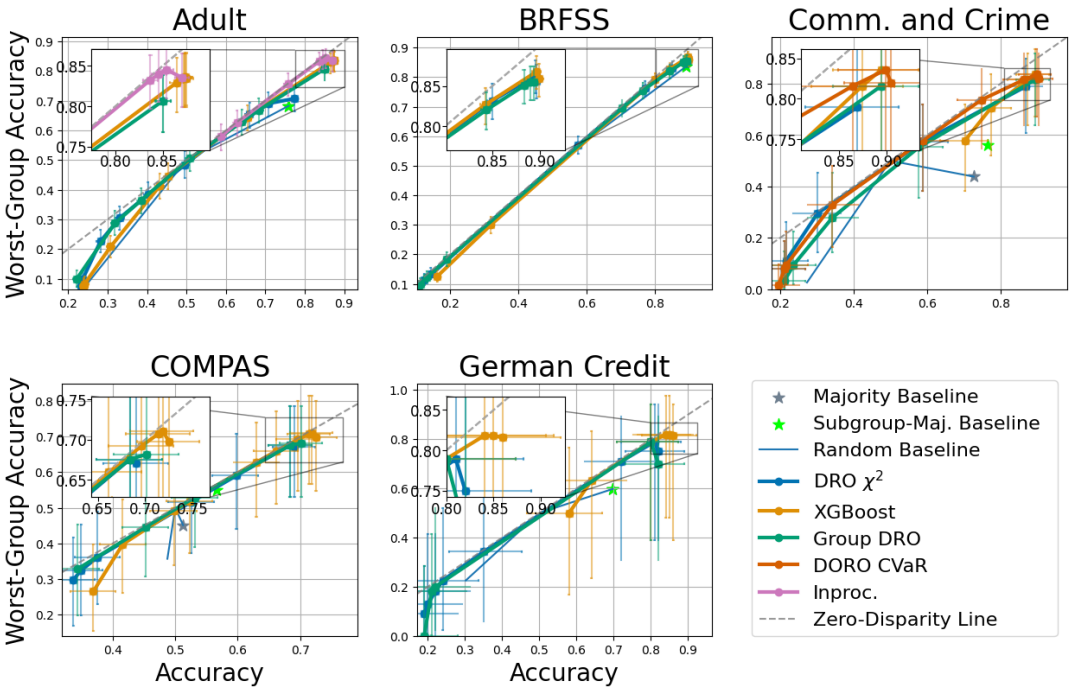


Figure 2.3: Model performance frontiers, formed by tracing the convex envelope of model performance. Tree-based models achieve comparable and sometimes improved frontiers with the highest-performing robustness methods. (See also Figure 2.1 for the remaining three datasets.)

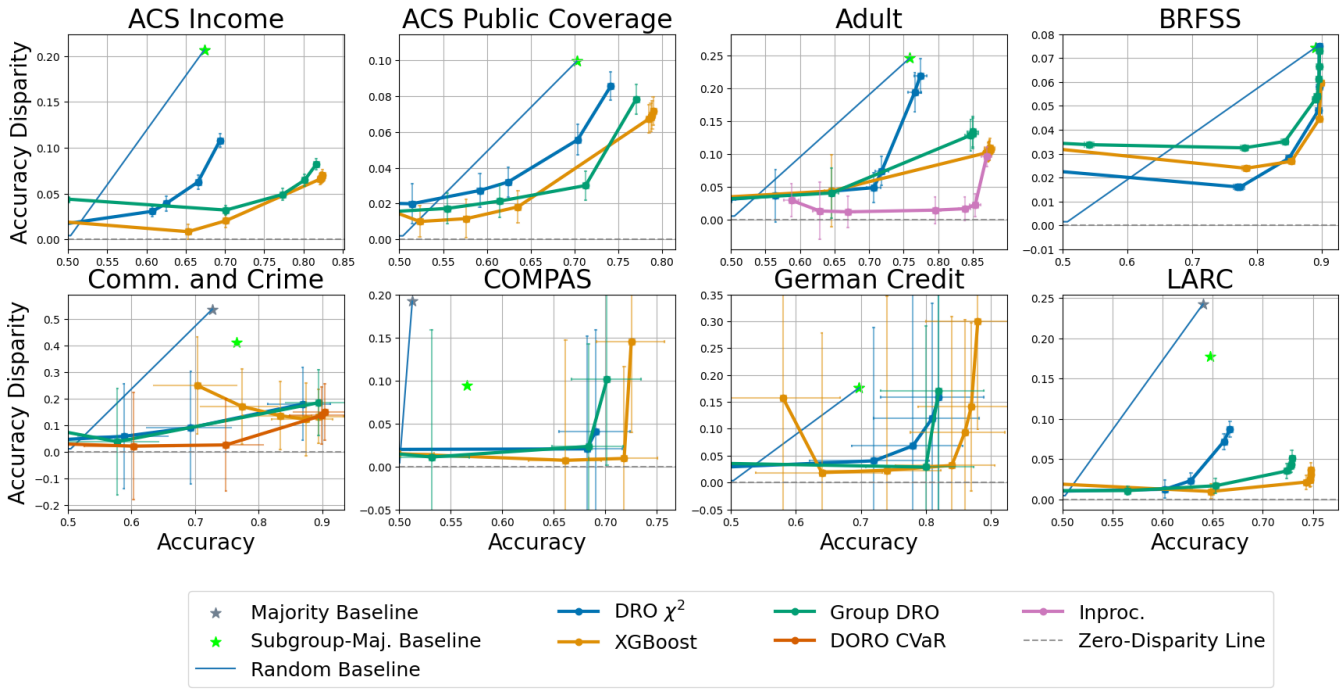


Figure 2.4: Model disparity frontiers, formed by tracing the convex envelope of model disparity. Tree-based models achieve comparable and sometimes improved frontiers compared to the highest-performing robustness methods, particularly in high-accuracy regions.

to measure the best possible set of tradeoffs between ($\mathcal{L}$ and $\mathcal{L}_{\text{WorstGroup}}$) and ($\mathcal{L}$ and $\mathcal{L}_{\text{DISP}}$). Model performance frontiers represent the envelope of the convex hull of all observations of $\mathcal{L}(f_\theta) \in \mathcal{F}$. An example is shown in Figure 2.1b and 2.1d, with the remaining datasets in Figures 2.3 and 2.4. The frontiers allow us to compare the best-possible tradeoff between accuracy and worst-group accuracy (Fig 2.3; higher is better) or between accuracy and accuracy disparity (Fig 2.4; lower is better).

2.5 Results: Robust Models Can Be Metric-Brittle

Our results show that, over all models and datasets, metrics which we refer to as "complementary" – those measuring the same event on different subsets, or with different conditioning – are strongly correlated with each other: accuracy and worst-group accuracy; CVaR and CVaR DORO; and Demographic Parity Difference and Equalized Odds Difference all show strong correlation with each model class $\mathcal{F}$; we show

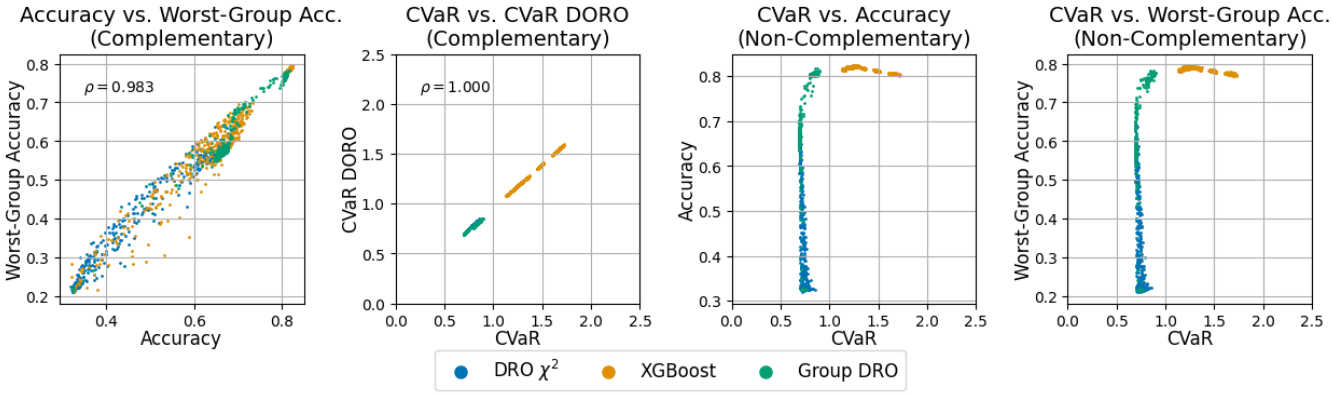


Figure 2.5: While "complementary" metrics (those measuring the same event with different conditioning) are closely correlated for all models, non-complementary metrics behave differently by model. Accuracy can vary widely for "robust" models with a fixed robust (CVaR) risk, while for tree models, the best (lowest) CVaR risk is typically associated with the best (highest) accuracy. ACS Income dataset shown; see Supplementary Figure A.11 and Section A.7 for additional datasets and metrics.

these correlations on Adult in Figure 2.5. The median ρ values over all datasets and algorithms are 0.87, 0.99, 0.79 for the three pairs of metrics, respectively (we show the complete set of correlations for each model and metric in Figure A.11). These results show that model selection based on one metric from each pair (Overall Accuracy) is likely to lead to a model with strong performance for the other metric in the pair (e.g. Worst-Group Accuracy).

In contrast, our experiments show that there is a severe lack of correlation across almost all of the non-complementary pairs, shown in the bottom row of Figure 2.5. In particular, our results show that models which achieve a strong "robust" risk – for example, a low CVaR DORO risk, one model selection rule used in [204] – can achieve very low accuracies. Indeed, the "robust" learning methods are the most susceptible to this discrepancy over metrics in our experiments, as shown in Figure 2.6. While these results confirm the finding in previous works that fairness metrics are correlated [63], our broader comparison shows a brittleness to model selection metrics outside these sets of complementary metrics, and reveal the very strict sense in which robustness guarantees apply only to a limited range of metrics. This discrepancy is of practical significance given that robust models are often selected using robust metrics (i.e. DORO CVaR, cf.

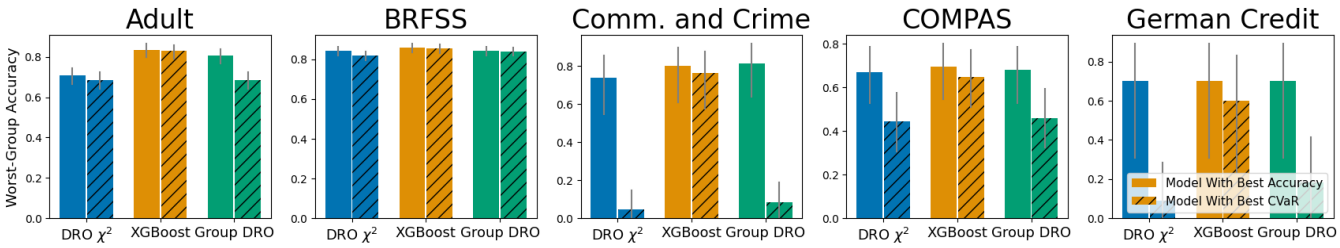


Figure 2.6: Worst-group accuracy for models with best overall accuracy (solid), and best CVaR (shaded). Tree-based models show better performance across non-complementary metrics (as indicated by similar heights of orange bars within each plot), whereas robustness- and fairness-based models do not (statistically-significant gaps between blue, green bars): worst-group accuracy drops significantly between best-accuracy and best-CVaR DRO χ^2 and Group DRO models (See also Figure 2.1 for the remaining three datasets.).

[204]), while models in practice are frequently evaluated using other metrics (i.e. accuracy, fairness metrics).

To illustrate the practical implications of this metric brittleness, we also explore the impact of various strategies for choosing a single $f_\theta \in \mathcal{F}$ over a set of models in each hyperparameter grid. While this problem is implicitly addressed in almost all previous works, the decision is often handled differently. For example, models are tuned and selected by hand [113], chosen based on worst-group performance [204], or based on robust risk metrics [160]. The empirical implications of model selection methods for fairness and robustness are not well-understood.

To address this question, we show the worst-group performance of models over our sweeps, selected according to either *best accuracy* or *best CVaR* in Figure 2.6. Figure 2.6 demonstrates the downstream impact of the lack of correlation between robust risk metrics and worst-group accuracy: distributionally-robust models can suffer significant drops in worst-group accuracy, when selected based on robust risk. In contrast, tree-based models show "metric robustness" – that is, tree-based models which perform best according to the robust risk measure (CVaR) still achieve worst-group accuracy near the highest-accuracy model of the same class (here, XGBoost).

2.6 Results: Trees Show Lower Hyperparameter Sensitivity and are Less Costly to Train

For many practical applications, both the time and financial costs of training models are of prime importance. These also affect the amount of expertise required to train a model (with sensitive models requiring greater expertise) and directly impact the carbon footprint of training [85]. We conduct a hyperparameter sensitivity analysis by pruning the hyperparameter search space to eliminate configurations that performed poorly across all datasets, ranking the remaining configurations by performance (i.e. accuracy), and plotting the decline in performance over the ranked models. These results, shown in Figures 2.7 Section A.6.2, demonstrate that tree-based models also show less sensitivity to hyperparameters than the other methods evaluated for both overall performance metrics (e.g., accuracy) and subgroup robustness metrics (e.g., worst-group accuracy).

These results are particularly important in light of the large differences in resources required to achieve similar levels of performance between robust models and the tree-based models: for example, the full hyperparameter grid sweep of size $12k$ XGBoost on the largest dataset in our study, ACS Income, completed in 1 CPU-day; DRO χ^2 sweep of 3250 training runs completed in 58 GPU-days. Due to the differences in hardware required to train various models, we conduct an estimated comparison of the *cost* of training an individual model, and of the full sweep. These results, shown in Figure A.12, show that tree-based models are also considerably less expensive to train and tune.

2.7 Conclusion

Machine learning has made significant progress in the past several years in identifying and addressing various challenges which can contribute to real-world performance disparities. However, our results suggest that another form of progress not widely acknowledged as having implications for fairness or robustness – advances in tabular data modeling with tree-based algorithms – have similar, if not greater, impact in practice than the state of the art in robust neural network learning or fairness-enhancing methods. Our results suggest that tree-based algorithms are a surprisingly strong baseline for subgroup robustness in tabular data, and that future work should compare against, and improve upon, the subgroup robustness of tree-based models. Our work suggests that, in practice, tree-based ensembles are an effective default for tasks where subgroup robustness is desired.

Our results also suggest that subgroup robustness on par with existing state of the art can be achieved with

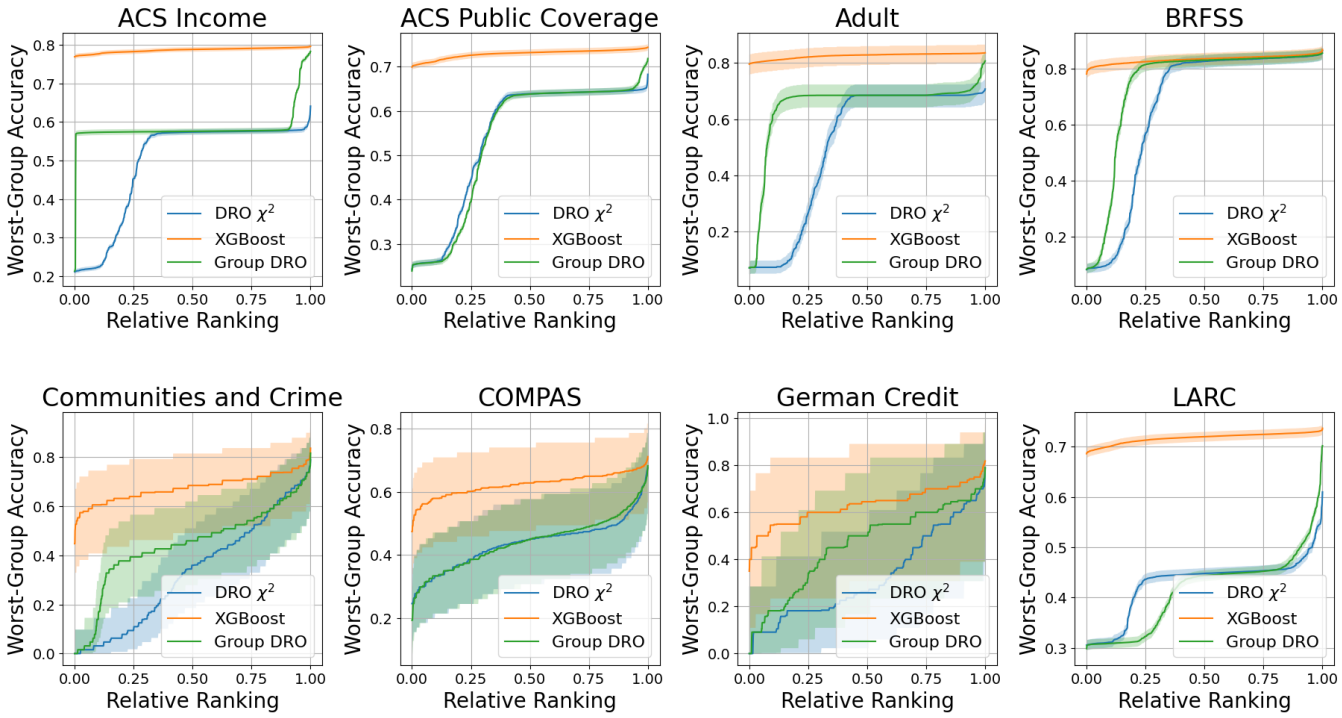


Figure 2.7: Performance over (truncated) hyperparameter grids on all datasets. Tree-based models (orange) show significantly less sensitivity to hyperparameter settings, for both subgroup and overall performance metrics (worst-group accuracy shown here), as indicated by nonoverlapping Clopper-Pearson CIs ($\alpha = 0.05$). For additional results and methodology for constructing these plots, see Section A.6.2.

tree-based classifiers that are easier and computationally cheaper to train and tune than either robust or fairness-enhancing models, which tend to scale poorly with dataset size (in both n and d). Our work thus contributes a critical baseline, similar to how other recent works have contributed much-needed baselines in areas of machine learning affected by the rapid proliferation of new methods [152, 181, 116], and provides a strong benchmark for future robust and fair learning methods.

Our findings are limited to the set of hyperparameters and models explored in our experiments – a superset of those from the works discussed above (e.g. [160, 204]). Our findings do not demonstrate that robust learning methods *cannot* achieve subgroup robustness on par with e.g. XGBoost, but merely that this is not achieved with the configurations widely used in the literature. We note that the loss-based interventions largely favored by existing robust and fair learning techniques require optimization over a continuous function, typically implemented as a feedforward neural network; this also makes it difficult to disentangle whether the functional form, or the training procedure and objective, lead to the improved subgroup robustness of tree-based models observed in this study. Future work should investigate the subgroup robustness of non-MLP models trained using robust techniques.

While our experiments do not identify the *cause* of trees' subgroup robustness, it is likely that this is a consequence of their strong overall performance on tabular data. These findings can be also viewed as an analogue of the empirical relationship between in-distribution and out-of-distribution accuracy in computer vision documented in [125] but now demonstrated for the tabular domain. It is possible that these improvements are due to (i) an inductive bias in tree-based models beter suited to modeling differences in subpopulations in tabular data, or (ii) the *ensembling* used by the tree-based models in this work. We leave an identification of such causal factors to future work.

2.8 Acknowledgements

This work is in part supported by the NSF AI Institute for Foundations of Machine Learning (IFML, CCF-2019844), Google, Microsoft, Open Philanthropy, and the Allen Institute for AI.

Moreover, our research utilized computational resources and services provided by the Hyak computing cluster at the University of Washington, and by Advanced Research Computing at the University of Michigan, Ann Arbor.

We are also grateful to Hongseok Namkoong, Tatsunori Hashimoto, John Miller, Michael Kim, Shafi Goldwasser, and attendees of the 2022 Institute for Foundations of Data Science Workshop on Distributional Robustness for useful feedback and discussions about the work, and to Christopher Brooks for assistance with the Learning Analytics Architecture (LARC) dataset.

Chapter 3

BENCHMARKING DISTRIBUTION SHIFT IN TABULAR DATA WITH TABLESHIFT

3.1 Introduction

Modern machine learning models have achieved near- or even super-human performance on many tasks. This has contributed to deployments of models across critical domains, including finance, public policy, and healthcare. However, in tandem with the growing deployment of machine learning models, researchers have also demonstrated concerning model performance drops under *distribution shift* – when the test/deployment data are not drawn from the same distribution as the training data. Analyses of these performance drops have primarily been confined to the domains of vision and language modeling (e.g. [76, 108], where effective benchmarks for distribution shift exist. Despite the widespread use of tabular data, the impact of distribution shift on *tabular* data has not been thoroughly investigated. While there are existing benchmarks for IID tabular classification, none of these focus on distribution shifts [73, 68].

This is particularly concerning in light of the known differences between tabular data and the modalities mentioned above (images, text, audio). First, in contrast to these modalities, where large neural models are the undisputed state-of-the-art, there is considerable debate about whether deep learning models improve performance over non-neural baselines (e.g. XGBoost, LightGBM) for tabular data, even without the presence of distribution shift [98, 28, 170]. Second, tabular data tends to contain structured features extracted from raw data (e.g. counts of activities, coded responses to questions), as opposed to the raw signals (e.g. activity event streams, pixel values, audio of responses) where modern machine learning methods perform well and where previous studies of distribution shift have focused. Third, tabular data requires fundamentally different preprocessing procedures, and the impact of these decisions is not widely understood, despite being known to have empirical impact [63]. Finally, high-quality tabular datasets can be difficult to access [72]; for example, due to the personal nature of many tabular datasets, tabular data cannot simply be scraped at Internet scale as many text and image datasets are. This makes finding high-quality tabular

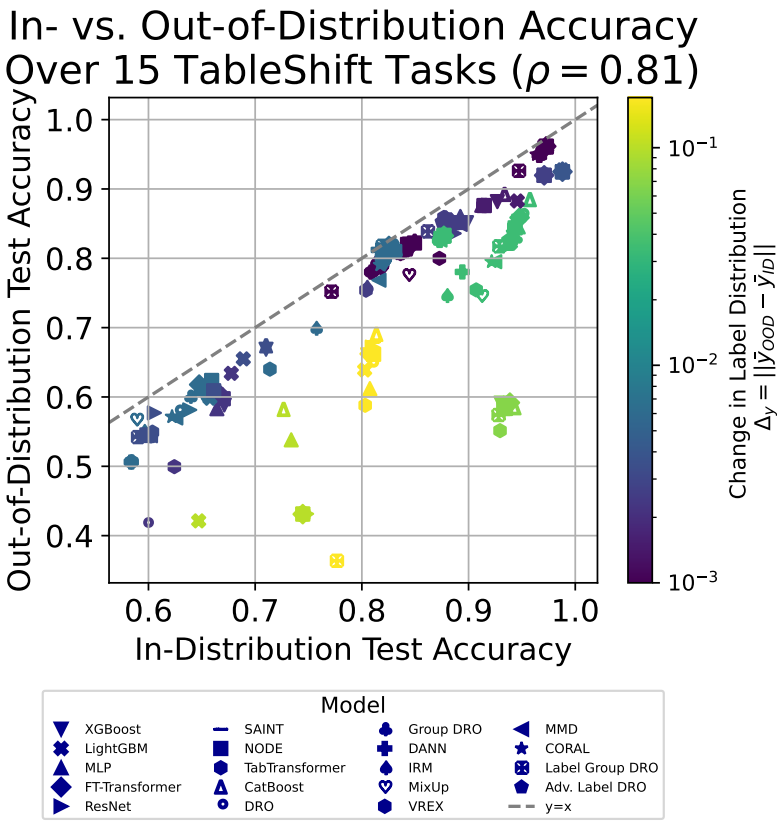


Figure 3.1: In-domain (ID) and out-of-domain (OOD) accuracy show a linear trend across 15 TableShift tasks and 19 model types ($\rho = 0.81$). ID accuracy (x-axis values) and change in the label distribution Δ_y (color) together explain 99% of the variance in OOD accuracy ($R^2 = 0.993$). For exact results see Section B.5.3.

distribution shift datasets particularly challenging.

Thus, the machine learning research community currently lacks not only (1) an empirical understanding of the impact of distribution shift on tabular data models, but also (2) a shared set of accessible and high-quality benchmarks to enable such investigations. We address both gaps in this work. Our main contributions are:

TableShift Benchmark: we introduce a curated, high-quality set of publicly-accessible benchmark tasks for (binary) tabular data classification under distribution shift. We describe the tasks in §3.3.1 and the API

in §3.3.2. TableShift includes a set of real-world tabular datasets from domains including finance [61], public policy [49], healthcare [37, 99, 36, 153, 191], and civic participation [7]. We select these datasets to ensure a diversity of tasks, distribution shifts, and dataset sizes.

Large-scale empirical study of distribution shift in tabular data: We conduct a large-scale study in §3.4, including state-of-the-art tree-based tabular models, tabular neural networks, distributional robustness methods, domain generalization methods, and label shift robustness methods. Our findings show (1) a strong linear trend between in-distribution (ID) and out-of-distribution (OOD) accuracy across benchmark tasks and models that was not previously identified for tabular data; (2) that no model consistently outperforms baseline methods, and (3) a correlation between the shift gap and the shift in label distribution, which is not ameliorated by label shift robustness methods included in our study.

Accessible TableShift API and baselines: We release a Python API for constructing rich datasets directly from their raw public forms. The API provides built-in documentation of data types and feature codings, alongside standardized preprocessing and transformation pipelines, making the datasets accessible in multiple formats suitable for training tabular models (e.g. in scikit-learn and PyTorch). We also release the set of baseline model implementations (including both state of the art tabular data models, robust learning models, and domain generalization methods) and end-to-end training code in order to facilitate future research on distribution shift in tabular data.

3.2 Setup, Task, and Notation

3.2.1 Task and Setting

Consider a dataset composed of examples $(x, y, d) \sim P_d$ where x is the input, y is the prediction target, and d the domain from which that example is drawn. All examples drawn from P_d have domain label d. We can view the overall data distribution as a mixture of domains $D = \{d_1, \ldots, d_D\}$, where $D \geq 2$. Training examples are drawn from the training distribution $P^{\text{train}} = \sum_{d \in D} q_d^{\text{train}} P_d$, and testing examples from $P^{\text{test}} = \sum_{d \in D} q_d^{\text{test}} P_d$, with domain weights $q_d \in [0, 1]$. We can define the training and testing domains as $D^{\text{train}} = \{d \in D : q_d^{\text{train}} > 0\}$ and $D^{\text{test}} = d \in D : q_d^{\text{test}} > 0$, respectively. We refer to cases where $|D^{\text{train}}| \geq 2$ as "domain generalization" tasks, because domain generalization models require at least two subdomains in the training data.

In a standard (IID) setting, our goal is to learn a classifier f_θ that accurately predicts y using examples from D^{train}. A *distribution shift* (or *domain shift*) occurs due to the fact that $P^{\text{train}} \neq P^{\text{test}}$. As a consequence of this shift, the joint distributions $p^{\text{train}}(x, y) \neq p^{\text{test}}(x, y)$ differ in training and testing. This difference can be composed of one or more changes to the underlying data generating process. This includes covariate shift, where $p(x)$ changes; label shift, where $p(y)$ changes, and concept shift, where $p(y|x)$ changes. In almost all real-world scenarios, distribution shifts are composed of an unknown mixture of all three forms of shift[1]. For a fixed classifier f_θ, we refer to

$$\Delta_{\text{Acc}} = \text{Acc}(f_\theta, D^{\text{test}}) - \text{Acc}(f_\theta, D^{\text{train}}) \tag{3.1}$$

as the "shift gap" (where both metrics are computed on examples not seen at training time). Note that the shift gap can be affected by changes in $p(y)$, $p(y|x)$, and $p(x)$. While disentangling the effects of these forms of shift is not a focus of the current work, we provide initial exploratory results on the impact of changes in $p(y)$, $p(y|x)$, and $p(x)$ over the benchmark tasks in Sections 3.5 and B.5.

In our setting, we assume that no information about the target D^{test} is available – i.e., there is no knowledge of the change in $p(y)$, $p(y|x)$, and $p(x)$, and no unlabeled data from the target domain.

3.2.2 Related Work

Here we provide a brief overview of related work necessary to contextualize our benchmark and main results. For a detailed overview of related work, see Section B.4.

Our work is closely related to the literature on distribution shift in machine learning. A series of recent works have demonstrated that even state-of-the-art models experience significant performance drops under distribution shift in tasks including vision, language modeling, and question answering [124, 125, 76, 108, 12]. This has led to the development of methods to mitigate susceptibility to such shifts [159, 113, 2, 9, 199, 198, 114, 98]. High-quality benchmarks, specifically tailored to distribution shift, have been essential in both measuring these gaps and assessing progress toward closing them [76, 108]. The use of tabular data is widespread in practice [28, 98, 170], including the use of sensitive personal data (race, gender, age) and for important tasks (credit scoring, medical diagnosis). However, the impact of distribution shift in the

[1]We note that this is a slight abuse of the terminology, as e.g. "label shift" typically refers to the case where *only* $p(y)$ changes.

tabular domain has received little attention in the research literature. In particular, benchmarks containing *tabular* distribution shifts are lacking (one notable exception is Shifts [119] and Shifts 2.0 [119], a multimodal benchmark of five tasks, two of which are tabular; for a more detailed overview of domain shift benchmarks and a comparison to TableShift, see Section B.7).

3.3 Tableshift: A Distribution Shift Benchmark for Tabular Data

This work introduces the TableShift benchmark. TableShift contains a set of 15 curated tasks designed to be a rigorous, challenging, diverse, and reliable benchmarking suite for tabular data under distribution shift, and we encapsulate them within a Python API.

3.3.1 TableShift Benchmark Tasks

To select tasks for TableShift, we identified datasets meeting the following formal criteria:

Open source: datasets must be publicly available, including data dictionaries documenting the source of the data (i.e. conditions of its collection), definitions of variables, and any preprocessing applied.

Real-world: does not contain simulated data.

Sufficient dimensionality and size: contains at least three features (in all cases, our benchmark datasets contain many more than three features) and at least 1000 observations. In particular, having large test sets is critical for making reliable statistical comparisons between models.

Heterogeneous: contains features of mixed types.

Binary Classification: supports a meaningful binary classification task (regression tasks are not included).

Shift Gap: We explicitly select datasets where strong hyperparameter-tuned tabular baselines display a statistically significant shift gap ($\Delta_{\text{Acc}} \neq 0$, see Eqn. (3.1)).

In addition to these criteria, we selected benchmark tasks and data sources that were *diverse*. TableShift includes tasks from many domains (finance, policy, civic participation, medical diagnosis) and from a variety of raw data sources (electronic health records, surveys/questionnaires, etc.) and with a diversity of shift gap (Δ_{Acc}) magnitudes.

A summary of the benchmark tasks is shown in Table 3.1. We give a detailed overview of each task, including

background and motivation, information on the data source, and distribution shifts, in Section B.2. Datasets and each individual feature of each task are also documented in the Python package. One important aspect of TableShift's diversity, shown in Table 3.1, is that not all real-world tasks support domain generalization (i.e. not all tasks have multiple training subdomains, $|D^{\text{train}} \geq 2|$). To reflect this, we include both types of tasks in the TableShift benchmark.

While the intended use of TableShift is for distribution shift, the package is also likely to be of high utility to all researchers studying tabular data modeling due to the data quality, detailed documentation, flexible preprocessing, and ease of use of the datasets in TableShift.

3.3.2 TableShift API

Successful existing benchmarks for distribution/domain shift in machine learning (e.g. WILDS, DomainBed) not only include high-quality datasets, but also make the data *accessible* by providing a high-quality API as an interface to the otherwise-disparate sources. This section describes the TableShift API. Providing this API for tabular data is particularly important, for several reasons.

First, the input and output of tabular data pipelines differ from other modalities: tabular datasets are stored in different formats from image and text datasets, and are used with a greater variety of machine learning tools (e.g. `scikit-learn`). Second, the preprocessing operations used in tabular data differ significantly from other data modalities. These preprocessing operations also require unique feature-level metadata such as data types (i.e. categorical vs. numeric; numeric values for categorical features are a common encoding scheme in practice) and codings for categorical variables. Finally, raw sources used to build tabular datasets can be difficult to access. Datasets are often scattered across hundreds or even thousands of files (e.g., the Sepsis task dataset is constructed from over $40k$ data files; the Childhood Lead dataset is joined from nearly 100 files containing disjoint feature sets provided by the National Health and Nutrition Examination Survey (NHANES)).

The TableShift API addresses each of these issues. It defines a set of primitives which allow for the construction of data pipelines which go from raw data sources to preprocessed data of any TableShift benchmark task in a few lines of Python code[2]. The resulting data is documented – each feature in the

[2]See https://tableshift.org and https://github.com/mlfoundations/tableshift

benchmark includes metadata which describes the feature and any encodings. The API natively supports a set of common data transformations, including one-hot and label encoding for categorical data; scaling and binning of numeric data; and handling of missing values. TableShift provides native output in a variety of data formats, including PyTorch DataLoaders, Pandas DataFrames, and Ray Datasets. Finally, *any* dataset in the TableShift benchmark can be loaded with default preprocessing parameters with an identical call to the API, providing a unified interface.

We provide a a detailed comparison between TableShift and related existing benchmarks in Section B.7. However, we emphasize that there is *no* existing benchmark suite for distribution shift in tabular data, and existing distribution shift benchmarks are incompatible with the unique constraints of tabular data discussed above.

3.4 Experiment Setup

We conduct a set of experiments to demonstrate the potential insights to be gained from using TableShift. As previously mentioned, there has been considerable debate about whether tree-based models (XGBoost, LightGBM, etc.) or specialized deep learning-based models (i.e. ResNet- and Transformer-based architectures) are more effective for tabular data modeling. However, previous investigations have not explored how these models perform under *distribution shift* in tabular data. Additionally, many methods have been proposed for robust learning and domain generalization but also not rigorously evaluated on tabular data. We present a series of experiments to evaluate 19 distinct methods using the TableShift benchmark.

3.4.1 Tabular Data Classification Techniques in our Comparison

We train and evaluate a set of tabular data classifiers from several families. For each, we give additional details and description in Section B.6, and the full hyperparameter grids in Table B.18. Implementations of these classifiers, including the hyperparameter tuning framework used to tune them, are available in the TableShift API. The classifiers compared in our experiments are:

Baseline Models: These models do not include any intervention for robustness to domain shift, but are generally effective for tabular data in the IID setting. We evaluate multilayer perceptrons (MLP), XGBoost [38], LightGBM [102], and CatBoost [52] as baseline methods. While we refer to these as "baselines" for convenience, we note that the methods based on gradient-boosted trees (XGBoost, LightGBM, CatBoost)

are still considered state-of-the-art on many tasks [74].

Tabular Neural Networks: We also include a set of state-of-the-art methods for modeling tabular data. The models we use are SAINT [173], TabTransformer [92], NODE [140], FT-Transformer, and tabular ResNet (the latter two via [73]).

Domain Robustness Models: These models attempt to ensure good performance on distributions close to the training data. These models attempt to optimize an objective over a worst-case distribution with bounded distance from the training data. We evaluate distributionally robust optimization (DRO) with both χ^2 and CVaR geometry [113], and group DRO (where the groups are domains) [159]. Both the DRO and group DRO models are parameterized over MLPs, as in both original works.

Label Shift Robustness Models: These models attempt to ensure good performance when the label distribution $P(y)$ changes. We evaluate Group DRO (where the groups are class labels) and the adversarial label robustness method of [207].

Domain Generalization Models: These are models designed with the goal of achieving low error rates on unseen test domains. In practice, this is achieved by achieving low error *disparity* across the subdomains in D^{train}. These methods require domain labels at training time, and training data drawn from multiple different domains ($|D^{\text{train}}| \geq 2$). Domain generalization models in our study are: Domain-Adversarial Neural Networks (DANN) [2], Invariant Risk Minimization (IRM) [9], Domain MixUp [199, 198], Risk Extrapolation (VReX) [112], DeepCORAL [176] and MMD [114].

We note that our goal of the current study is not to propose novel methods for distributionally robust learning; it is to conduct a comprehensive comparison of a large set of existing methods, many of which have not been previously compared to each other, on a high-quality benchmark. For example, while domain generalization models have been applied to image and text classification tasks (e.g. [108, 76]), to our knowledge these methods have not been previously investigated for mitigating distribution shift in *tabular* data in a large-scale benchmarking study. Indeed, we are aware of no prior applications of many of these domain generalization methods to tabular data. As a result, it is not clear *a priori* how these methods might compare to existing robustness or baseline methods due to the aforementioned differences between tabular data and these other data modalities.

The experiments described above cover both model architectures (different functional forms for the predictor

f_θ) and loss functions (different objective functions $\mathcal{L}$ used to train the model by attempting to find $\min_\theta \mathcal{L}(f_\theta(D^{\text{train}}))$). In order to train a classifier with gradient-based training, both are required. Except where noted otherwise, any method requiring gradient-based training (MLP, Tabular Neural Networks, Domain Generalization Models) is trained with standard empirical risk minimization and cross-entropy loss. Similarly, any method which itself is a loss function (i.e. all variants of DRO) is trained with f parameterized as an MLP, as is standard in prior works implementing and comparing these methods (e.g. [113, 159, 68]).

3.4.2 Methods

For each task, we conduct the following procedure.

First, we split the full dataset into D^{train} and D^{test}. We summarize the domain splits in Tables 3.1,3.1 and describe the splitting for each task in detail, along with background and motivation for each task domain split, in Section B.2. Within each domain, we have both a validation and a test set. We use the same domain splits, data preprocessing, and train/validation/test splits for all models and training runs, except where explicitly noted.

Second, we then conduct a hyperparameter sweep for each model described in Section 3.4.1. We use HyperOpt [22] to sample from the model hyperparameter space, in accordance with previous works (e.g. [73, 98]) which largely use adaptive hyperparameter optimization due to the variability in effective hyperparameter settings between datasets. We only train on the training set, and use the in-domain validation accuracy for hyperparameter tuning. We give the complete grid for each model in §B.9. Each model is tuned for 100 trials.

Finally, we evaluate the trained models on the test splits of each dataset. As recommended in [108], we use in-domain and out-of-domain *test* accuracy (not in-domain train accuracy) to evaluate the models. For all results shown, we use the best model selected according to (in-domain) validation accuracy; this follows the selection procedure used to study domain generalization in the image domain in [76].

3.5 Empirical Results

ID and OOD Accuracy are Correlated. Our results show that, across all models and tasks, in-distribution (ID) and out-of-distribution (OOD) accuracy are correlated: as ID performance improves, OOD performance also tends to improve (see Figure 3.1; $\rho = 0.81$). This linear trend holds *across datasets and*

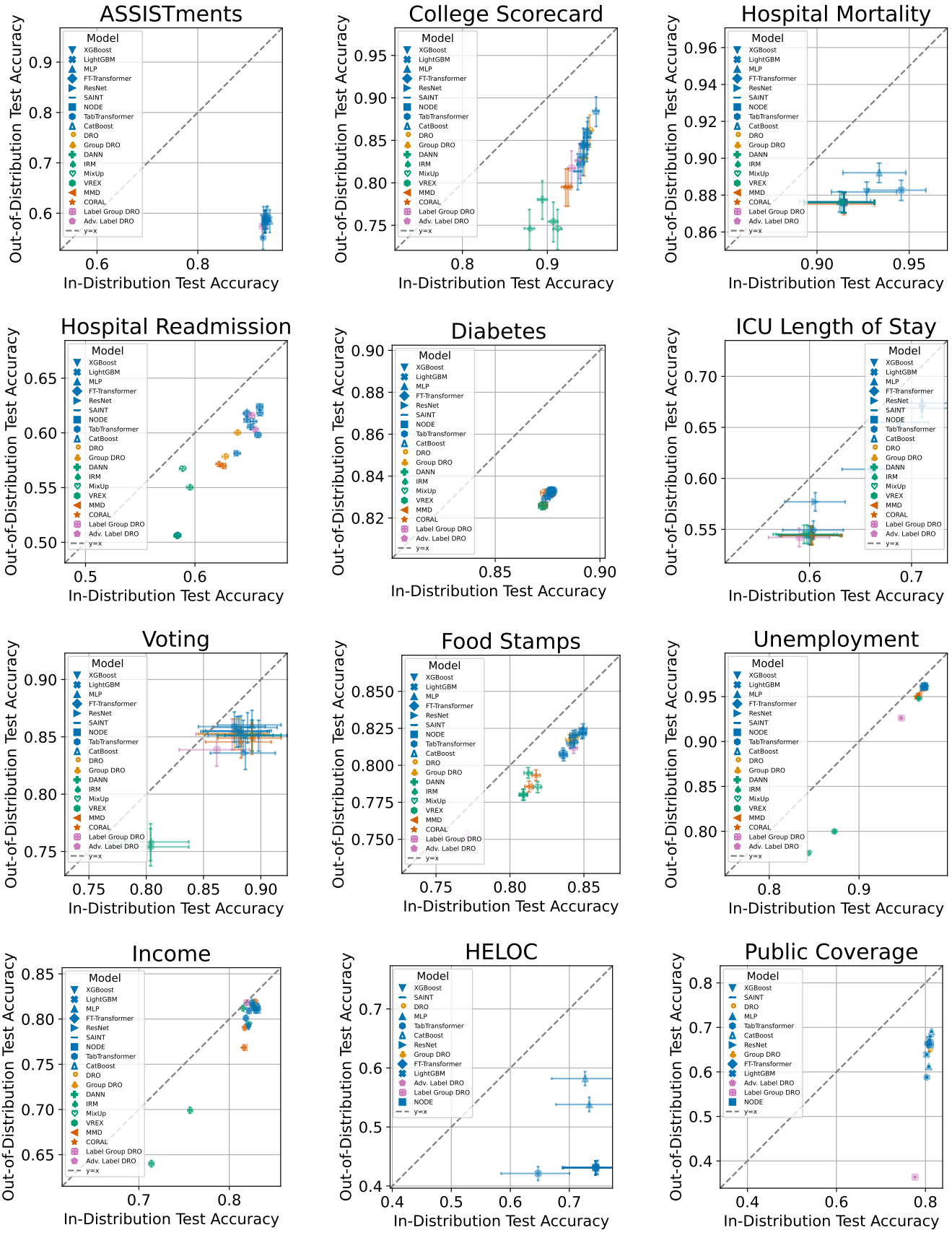


Figure 3.2: Results for baselines, robust learning, and domain generalization models across the 15 TableShift

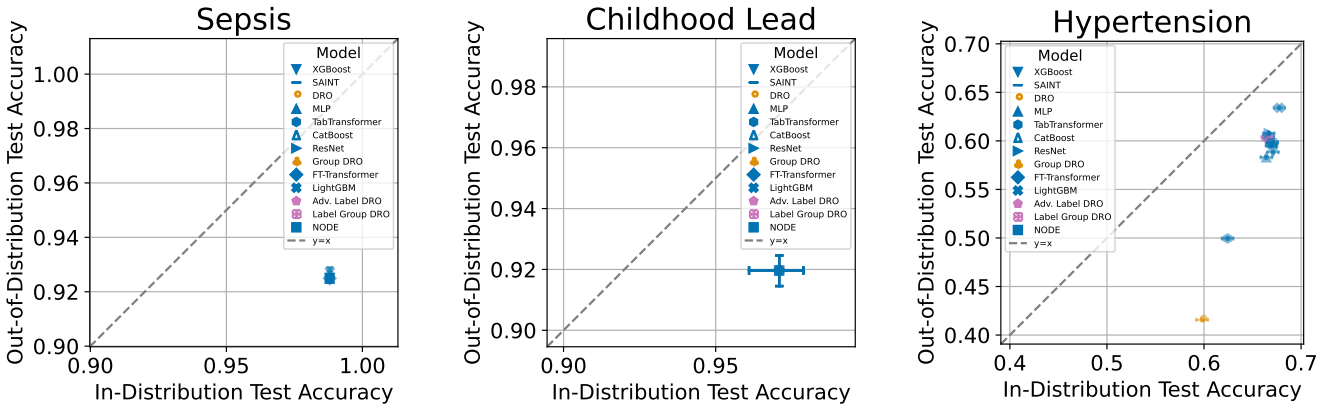


Figure 3.3: Additional results (cf. Figure 3.2). For exact results see Section B.5.3.

model classes. We note that, while this is consistent with findings for image [125] and question answering [124] models, the relationship between ID accuracy and OOD accuracy on tabular data was previously unknown. This result suggests that, for a wide variety of tabular data tasks, improving models' ID performance is likely to improve their OOD performance.

No Model Consistently Outperforms Baselines. While many models have been proposed for both (a) improving general performance on tabular data tasks over established baselines such as XGBoost and LightGBM, and (b) improving robustness to distribution shift, our results show that no model consistently outperforms the standard tabular baselines of XGBoost, LightGBM, or CatBoost in either respect. Figure 3.4a shows that, on average across all datasets, no model consistently achieves better performance (as measured as a fraction of the maximum OOD accuracy achieved by any model) compared to baseline methods. This finding has not been previously demonstrated in tabular data due to the lack of an existing benchmark.

No Method Eliminates Gaps. We investigate the empirical performance of several methods designed to improve robustness to distribution shift (described in Section 3.4.1). Our results shows that, on the datasets where multiple training subdomains are available (and thus where domain generalization is viable), there is weak evidence that several techniques reduce gaps due to distribution shift, but no technique eliminates these gaps. However, it is important to note that this gap reduction comes at the cost of in-distribution

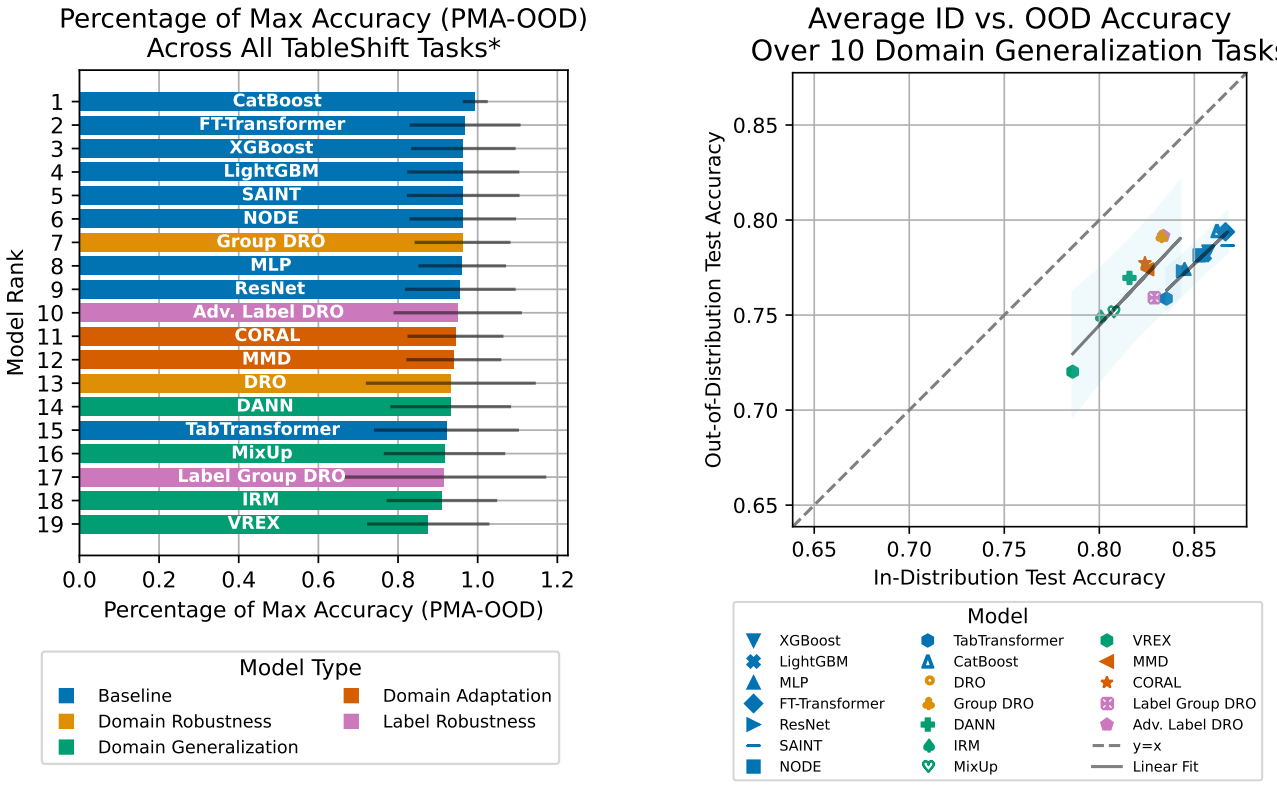
performance: as Figure 3.4b shows, all robustness-enhancing models tend to shrink gaps by *reducing average ID performance, not by improving OOD performance*. This is shown in Figure 3.4b by the two parallel lines: one set of blue points representing baselines + tabular NNs, and another, shifted left, representing robustness-engancing and domain generalization models. Furthermore, we note that all domain generalization and domain robustness methods evaluated (excluding DRO) require additional information that is only present for some datasets – namely, the discrete variable over which a shift will occur (e.g. "race" for diabetes task) and data from at least 2 categories of this variable.

Change in label distribution is correlated with shift gap. We investigate the degree to which the three factors mentioned previously ($p(x)$, $p(y|x)$, $p(y)$) are related to model performance. Our results, in Figures 3.5 and B.3, show that change in the label distribution Δ_y is correlated with shift gap Δ_{Acc} (Pearson correlation $\rho = 0.71$). This persists even after accounting for ID accuracy: a simple linear regression of OOD accuracy on [ID accuracy, Δ_y] obtains $R^2 = 0.996$. This suggests that the change in the label distribution is an important factor in understanding tabular shifts (for example, the outliers in Figure 3.1 are from the four tasks with largest label shift: Public Coverage, HELOC, ASSISTments, College Scorecard; see Figures 3.2, 3.3 and Table B.2). Label shift robustness methods in our study *did not* eliminate performance gaps under shift; in fact, label shift robustness methods often degraded both ID and OOD accuracy (e.g. Figure 3.4b). We provide similar analyses relating shift gap to (*i*) covariate shift and (*ii*) concept shift in Figure B.3, but find that they are not clearly related.

Changes in predictions are related to covariate shift. As an exploratory finding, we find some evidence that changes in the predictions for OOD data are correlated with changes in $p(x)$, shown in Figure B.2a ($\rho = 0.99$). This suggests that shift gaps in the benchmark datasets not explained by the combination of ID accuracy and Δ_y may be driven primarily by covariate shift (changes in $p(x)$) as opposed to concept shift (changes in $p(y|x)$). Further analysis is needed to confirm this exploratory finding. We note that relationships between other forms of shift showed much weaker correlation, roughly $\rho \approx -0.2$ (see Figure B.2).

3.6 Limitations

The conclusions in this study are limited to the specific datasets and models evaluated. While we intentionally selected a diverse suite of benchmark datasets along several axes (domain, distribution shift, dataset size,



(a) (b)

Figure 3.4: (a): Percentage of Maximum OOD Accuracy (PMA-OOD) across tasks (see Table B.13 for exact values). *: domain generalization models and Group DRO can only be trained on the subset of 10 tasks with multiple training subdomains (see "Domain Generalization" column in Table 3.1). (b): Average ID and OOD accuracy by model across domain generalization tasks only. We only show domain generalization tasks in order to compare all models on the same set of tasks. See Figure B.4 for results on all tasks. Exact values in Table B.14.

etc.), our conclusions can only be extended to other distribution shifts insofar as they are similar to the shifts in TableShift. More empirical validation is needed, including studies comparing our findings to other tabular shifts.

Our work does not exhaustively cover the space of all possible tabular data classifiers. In particular, "hybrid" methods combining some of the loss-based robustness interventions (i.e. Group DRO) with various tabular data-specific model architectures (e.g. FT-Transformer, ResNet) might lead to different results. Our initial exploratory evaluation of hybrid methods (see Section B.5.5), however, does not suggest that hybrid methods led to qualitative changes in our results, but these methods warrant a more extensive evaluation. Finally, our work does not establish *theoretical* connections between the factors analyzed (ID accuracy, OOD accuracy, Δ_y).

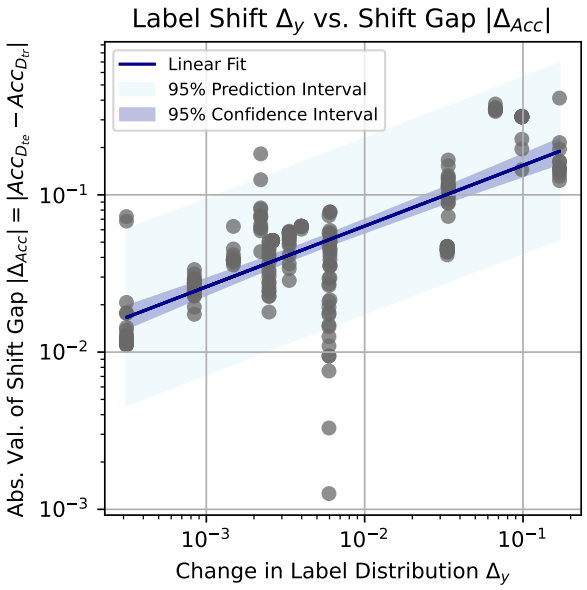


Figure 3.5: Label shift (Δ_y, measured via Equation (B.2)) and absolute shift gap Δ_{Acc} show moderate correlation across datasets and models (Pearson correlation $\rho = 0.70$). Exact Δ_y values in Table B.2.

3.7 Conclusion

We introduce the TableShift benchmark for studying distribution shift in tabular data. TableShift presents a diverse set of tasks for reliable study and benchmarking of tabular data models under distribution shift. We provide a Python API to access the datasets, along with implementations of several models including baselines, distributionally robust learners, and domain generalization methods. Finally, we present empirical results which form the first large-scale study of tabular data modeling under distribution shift.

Our results suggest multiple potential avenues for future work: First, improvements to *in-distribution* accuracy are likely to drive OOD accuracy gains. Second, improved robustness to *label shift* may reduce shift gaps. Third, *hybrid methods* which combine robustness-enhancing losses (such as Group DRO) with

improved neural network architectures may be able to further improve OOD performance. Beyond these proposed directions, we hope that TableShift opens new research frontiers for tabular machine learning research beyond those addressed in the current work.

Table 3.1: Summary of TableShift tasks and their associated distribution shifts. For details on each task, see Section B.5. "Domain Generalization" indicates whether there are multiple training subdomains ($|D^{\text{train}}| \geq 2$) and thus whether domain generalization models can be applied to this task. "Baseline gap" gives the "shift gap" Δ_{Acc} (difference between ID and OOD test accuracy, see Equation (3.1)) of the tuned XGBoost or LightGBM model with the best validation accuracy after following our hyperparameter tuning procedure (§3.4.2).

Task	Target	Shift	Domain Generalization	Baseline Gap Δ_{Acc}	SE(Δ_{Acc})
ASSISTmen	Next Answer Correct	School	✓	−34.49 %	0.011
College Scorecard	Low Degree Completion Rate	Institution Type	✓	−11.16 %	0.010
ICU Hospital Mortality	ICU patient expires in hospital during current visit	Insurance Type	✓	−6.30 %	0.008
Hospital Readmissior	30-day readmission of diabetic hospital patients	Admission source	✓	−5.94 %	0.002
Diabetes	Diabetes diagnosis	Race	✓	−4.48 %	0.001
ICU Length of Stay	Length of stay >= 3 hrs in ICU	Insurance Type	✓	−3.39 %	0.015
Voting	Voted in U.S. presidential election	Geographic Region	✓	−2.58 %	0.016
Food Stamps	Food stamp recipiency in past year for households with child	Geographic Region	✓	−2.39 %	0.002
Unemploym	Unemployment for non-social security-eligible adults	Education Level	✓	−1.28 %	0.001
Income	Income >= 56k for employed adults	Geographic Region	✓	−1.25 %	0.002
FICO	Repayment of Home	Est. third-		−22.58 %	0.029

Chapter 4

LARGE-SCALE TRANSFER LEARNING FOR TABULAR DATA VIA LANGUAGE MODELING

4.1 Introduction

Transfer learning - the ability of a model to accurately solve prediction tasks on data it was not trained on - is one of the defining hallmarks of recent foundation models in domains such as vision [144] and language [31]. Among their many advantages, transferable models expand the scope of problems that can be tackled via machine learning by reducing the need for curated, task-specific models and datasets. Such models also can provide both absolute performance and sample-efficiency gains over task-specific models when applied to new tasks [144, 148, 178].

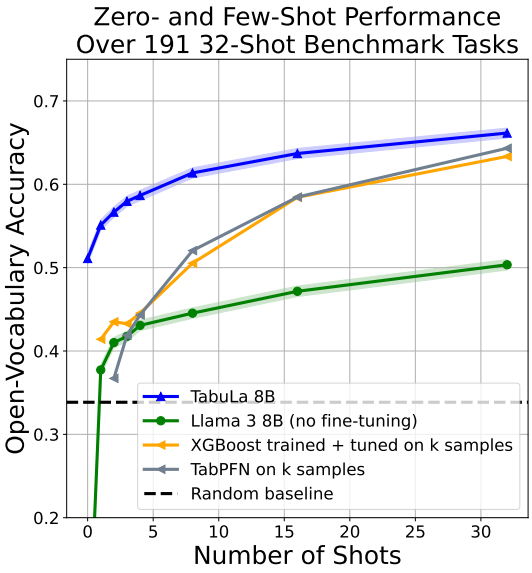


Figure 4.1: TabuLa-8B outperforms SOTA tabular baselines across $0 - 32$-shot tasks from five tabular benchmarks.

In this work, we introduce a new model and dataset for large-scale transfer learning on tabular data. Tabular, spreadsheet-style data underlies applications in healthcare, finance, government, and the natural sciences [169, 67, 29]. Yet, despite the potential impacts of transferable foundation models for tabular data [186], the core practices of machine learning on tabular data have remained largely unchanged. The prevailing paradigm is still to train single-task models (e.g., XGboost [38]) using a fixed schema on data from the same distribution on which the model will be deployed.

Here, we aim to bridge this gap. We introduce TabuLa-8B, a language model for tabular prediction which can flexibly solve classification tasks across unseen domains. Given a small number of examples (shots), without any fine-tuning, TabuLa-8B outperforms state-of-the-art gradient-boosted decision trees and

tabular deep learning methods *that are explicitly trained on the target data* (see Figure 4.1). Furthermore, TabuLa-8B is capable of zero-shot prediction, a behavior which is not possible using these prior methods. To enable these results, we build a new dataset for tabular prediction, The Tremendous TabLib Trawl (T4), that allows us to scale up training by several orders of magnitude (10,000× more data) relative to previous work.

4.1.1 Our Contributions

Summarizing, this paper has the following three main contributions:

TabuLa-8B, a Tabular Prediction Model: We build TabuLa-8B (Tabular LLama 3 - 8B), a model for prediction on tabular data. On an evaluation suite consisting of 329 tables drawn from five tabular benchmarks, TabuLa-8B has zero-shot accuracy 17 percentage points (pp) above random guessing. In the few-shot setting (1-32 examples), TabuLa-8B is 5-15 pp more accurate than state-of-the-art methods (XGboost, TabPFN) that are trained on equal number of shots, and these methods require 2-8× more data to achieve the performance of our model.

T4 - A Large Scale, High Quality Training Dataset: We build and release The Tremendous TabLib Trawl (T4), a filtered collection of 3.1M unique tables (consisting of over 1.6B rows, a total of 80B tokens) from TabLib [58]. We detail the recipe used to construct T4, including a suite of methods for filtering web-scale tabular data at several levels (table, row, column), removing unwanted information such as PII and code, and selecting unsupervised *prediction targets* from these tables.

Open-Source Release: As part of our publication, we release all relevant infrastructure (code, models, and data) with the hopes that the community will build on our work. We provide high-quality, efficient implementations of data pre-processing and model training pipelines, including our new row-causal tabular masking (RCTM) attention and packing scheme for training on tabular data. We also share the code used to filter T4 from TabLib, enabling future work that extends our dataset construction methodology.

4.1.2 Preliminaries & Project Scope

Our work is concerned with *prediction* models for *tabular data*. We define both below.

Tabular Data: For our purposes, tabular data has three main properties. (*i*) *Structured*: It consists of

elements with a "key-value" structure, often represented as a table with keys (or "headers") representing column names, and rows that consist of values. (*ii*) Heterogeneous: The values are of mixed types, including numeric, boolean, categorical, ordinal, text, date/time, etc. Missing values may be present. (*iii*) Exchangeable: The ordering of rows and columns in the dataset is arbitrary. In particular, any permutation of the rows, or columns, still represents the same tabular dataset.

Prediction Task Definition: The main focus of this work is *prediction* on tabular data. In tabular prediction, the goal is to predict the value y of a specific target column for a row in a dataset using the key-value pairs x from all other columns. More specifically, we focus on classification tasks where values y for the target column belong to a finite set C. Binned regression tasks, in which a real-valued y is discretized into a finite set of numeric values (as in [190]) also fit this definition.

4.2 Related Work

Our work builds on a line of foundation modeling, tabular data prediction, and natural language processing research. Given space constraints, here we focus on the most closely related literature.

Transfer Learning and Foundation Models: The idea of building general purpose models via autoregressive next-token prediction on large scale datasets was pioneered in a series of papers in natural language processing [146, 31, 138, 183, 48]. These results have since led to the development of foundation models capable of solving diverse tasks in other modalities including vision [144, 201], audio [209, 145], code [155, 77], time series [46, 70, 75], and graphs [196], as well as multi-modal models [178, 180]. Our work also build upon on the demonstrated capacity of transformers to perform few-shot or in context-learning [31, 69], which entails making predictions on examples from a previously unseen dataset, given only a few labeled examples from that task.

Large-Scale Dataset Curation: The construction of large, high-quality datasets has emerged as one of the most critical, and challenging, issues in the development of transferable models. Several major milestones in this space [148, 182, 179, 146, 31, 180, 178] stand out in their effort spent curating and cleaning web-scale datasets – often while using a model architecture and training recipe that only slightly differs from prior work. This has led to a number of modality-specific methods for large-scale dataset curation; for example, the use of heuristics [148] and model-based quality scoring to select high-quality text data [31, 182, 43]; methods for selecting aligned audio-transcript pairs for speech [145]; or the use of CLIP scores to filter for

aligned image-text pairs [165]. However, to the best of our knowledge no prior work has developed analogous methods for *tabular* data. This lack of large-scale training data has been a critical bottleneck toward the development of tabular foundation models.

Models for Tabular Prediction: Despite the fact that deep learning methods are now the norm in domains such as computer vision or NLP, methods based on gradient-boosted decision trees (GBDTs) [38, 142, 30] continue to be at or near state-of-the-art in tabular prediction [74]. Drawing upon recent breakthroughs in other modalities, the field has now developed deep learning-inspired approaches [73, 173] that are competitive with tree-based models *in-distribution*, where models are trained and evaluated on the same data, but the benefits of such approaches relative to GBDTs appears to be limited in practice [74, 122]. In particular, [90] introduces TabPFN, a transformer model for tabular data that outperforms XGBoost in certain regimes [122] and is capable of making predictions on unseen datasets (with some constraints on dataset size and label space; see C.4.2). A related recent work, CARTE [107], explores the use of graph-based architectures for tabular transfer, based on key-value encodings and pretrained on a large knowledge graph.

Several recent works [84, 50, 190, 209] have explored fine-tuning LLMs on individual tables, or on small collections of tables (< 200). The main idea in this closely related line of work is to reduce classification to next-token prediction by first *serializing* a row as text (see Figure 4.2b for an illustration) and then training an LLM to predict the serialized labels. These studies demonstrate that this LLM approach is often competitive with trees or tabular deep learning methods in-distribution [50, 190, 84]. However, in cases where models were evaluated out-of-distribution, they were less accurate than SOTA methods trained on these held-out tables [206]. Our work builds on this promising line of work. Relative to these prior efforts, we specifically address the (1) lack of large-scale and training data; and (2) the inability of exiting methods to be competitive when evaluated out-of-distribution.

4.3 TabuLa-8B - Model Design and Training

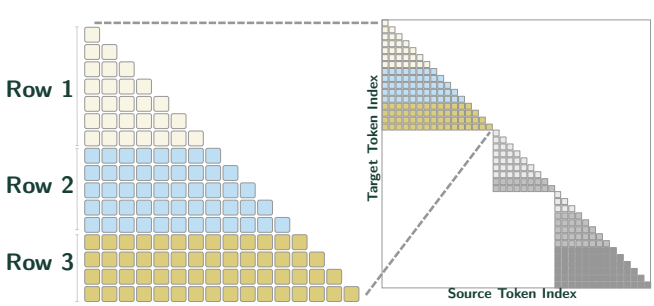
Our overall approach is to fine-tune the pretrained Llama 3-8B language model [179] on tabular prediction tasks using a new web-scale corpus, T4. We use Llama 3-8B as our starting point since it is a high-quality, open-source model trained on over 15T tokens that demonstrates strong performance on a diverse set of downstream tasks [179], particularly at its relatively modest size (which makes fine-tuning, inference, and deployment more accessible).

Serialization and Tabular Language Models: As discussed previously, our methodology extends ideas pioneered in previous work [50, 190, 84, 206] demonstrating how LLMs can be trained to perform tabular prediction tasks by serializing rows as text, converting the text to tokens, and then using the same loss function and optimization routines used in language modeling. *Serialization* refers to the procedure of converting a row of data into text, for instance by concatenating substrings of the form "the `<key>` is `<value>`". Prior works investigated the impact of different serialization formats [84, 50, 172], demonstrating that performance is largely insensitive to the exact mapping (e.g. using "{ `<key>`: `<value>` }") and other strategies do not improve upon this "the `<key>` is `<value>`" structure.

We adopt a similar serialization strategy, illustrated in Figure 4.2b. Given a row of data from a table, the corresponding serialization has three main parts: (*i*) a *prefix* containing a prompt (always "Predict the value of `<target column name>`") followed by a list of possible label values ("`val1 || ... || valNumClasses ||`"), (*ii*) the *example* consisting of all key, value pairs for the columns used as features, and (*iii*) a *suffix* prompting the model with a question ("What is the value of `<target column name>`?") again followed by the possible labels. For multiple-shot samples, we concatenate their serializations. We introduce three special tokens into the Llama 3 vocabulary to ensure these sequences are properly tokenized: `||`, to delimit answer choices; `<|endinput|>` to denote the end of an input sequence (the last token before the targets or model generation begin); and `<|endcompletion|>` to indicate the end of a completion.

Training Procedure: We train TabuLa-8B using a standard language modeling setup where the model is trained to minimize the cross-entropy over the sequence of target tokens. We only compute loss over the subsequence of target tokens: the tokens starting after the `<|endinput|>` token, up to and including `<|endcompletion|>`. This objective focuses training on learning the desired target label, as in [84, 50, 190], rather than developing a broader generative model of tabular data as in [206].

Relative to prior studies on tabular prediction with LLMs, our work has one main methodological innovation. We introduce an efficient attention masking scheme, row-causal tabular masking (RCTM), tailored to few-shot tabular prediction whereby the model is allowed to attend to all previous samples from the same table in a batch, but *not* to samples from other tables (this is sometimes referred to as "cross-contamination" in the language modeling literature [111]). However, by appropriately masking out values, RCTM also enables packing examples into the same batch (as effectively zero padding is required during training despite the large variance in the size of each tokenized table or row), thereby increasing model throughput (see



(a)

date	precipitation	temp_max	weather
2015-03-22	1.0	11.699	rain
2015-09-19	0.0	14.722	sun

Serialization

Predict the value of weather: ||sun||rain||snow|| The date is 2015-03-22. The precipitation is 1.0. The temp_max is 11.699. What is the value of weather? ||sun||rain||snow|| <|endinput|> rain<|endcompletion|>Predict the value of weather: ||sun||rain|| snow|| The date is 2015-09-19. The precipitation is 1.0. The temp_max is 14.722. What is the value of weather? ||sun||rain|| snow||<|endinput|>sun<|endcompletion|>

(b)

Figure 4.2: 4.2a: Illustration of the row-causal tabular mask (RCTM) representing a batch during training. Each triangular block represents potentially many rows from a *single* table (detail shown at left). Shaded groups within this block represent tokens from one row in the table. This structure implicitly trains the model for few-shot learning by permitting it to attend to previous rows from the table, but not to rows in other tables. 4.2b: Serialization of tabular data into text. The model is trained to produce the tokens following the `<|endinput|>` token.

Figure 4.2a). Taken together, these have the effect of training the model to use multiple "shots" during training and mitigates the potential loss of few-shot learning capabilities that has been observed to occur during fine-tuning [100, 205, 193].

The RCTM masking structure is shown in Figure 4.2a. Lower-triangular blocks correspond to rows from the same table that are present in the batch. This is similar to the "in-context pretraining" method from [167], except that (*i*) our procedure encourages the model to aggregate information across multiple *rows* of a given *table*, rather than attending across documents, and (*ii*) our procedure only trains the model to predict the *target* tokens, not the input features. We show that RCTM has a drastic impact on few-shot performance through an ablation experiment (see Section 4.5.5).

Training Details: The final model is trained for 40k steps with a global batch size of 24 (with sample packing, this is roughly equivalent to a global batch size of 600 rows of tabular data). The model sees

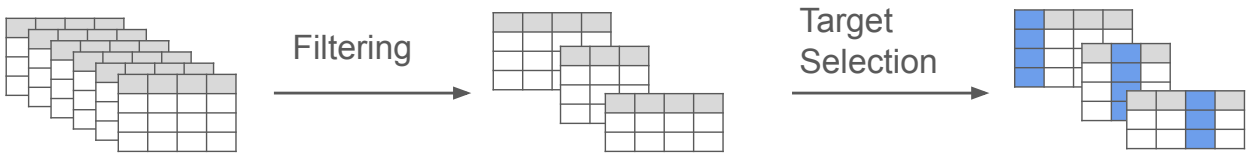


Figure 4.3: Sketch of dataset generation pipeline. 627M tables from TabLib [58] are filtered by applying rules at the table, row, and column level. Then, for each table, we identify valid and high-quality prediction targets in an unsupervised manner and use the results for training TabuLa-8B.

roughly 8B tokens during training; we note that is only 10% of the 80Btokens in T4, and less than one one thousandth of TabLib itself. We fully fine-tune all model parameters, as opposed to parameter-efficient fine-tuning, since full-fine tuning consistently benefits from scale [205, 87]. Reproducibility details are given in Appendix C.2.

4.4 Dataset Construction: Building The Tremendous TabLib Trawl (T4)

Beginning from a web-scale corpus of raw data (TabLib), we apply various filters to produce a high-quality subset, and transform the results into a set of prediction tasks for training. As the result of this procedure, we produce T4 (The Tremendous Tablib Trawl), which we release with this paper.

Original Raw Data Source: TabLib [58] is a publicly-available dataset consisting of 627M tables extracted from two main sources: Common Crawl and Github (see [58] for more details). Due to its scale and diversity, TabLib presents a unique opportunity for training foundation-scale models on the tabular data. However, like other web-scale datasets, the vast majority of its contents are low quality and not suitable for training. For instance, TabLib contains numerous system logs with inscrutable statistics, tables of software documentation, and call sheets with personally identifiable information (PII). To the best of our knowledge, no previous work has addressed the task of filtering TabLib into a usable training set, and no publicly-available models have been trained on this corpus.

Filtering Strategies: Filtering large collections of raw data to extract a higher-quality subset is an essential component in the development of foundation models [e.g. 148], yet to date, no previous work has addressed this core issue for tabular data. Filtering a web-scale dataset like TabLib into a usable subset of high-quality

tables is critical to leverage its diversity and scale, but also raises unique challenges specific to tabular data, such as missing data, web "content" that is formatted as HTML tables that does not satisfy our definition of tabular data, and PII. To turn TabLib into a usable training set, we develop a set of filtering methods to identify high-quality tables for prediction. Conceptually, our filtering occurs at three levels, each applied sequentially: *tables* (entire tables are removed from the pool), *columns* (individual columns are removed from a table), and *rows* (rows are removed from a table).

Similar to previous approaches [182, 179, 148, 146, 144], we use a mix of heuristics and rule-based methods to remove low-quality sources from our pool. We present the full list of our filtering rules in Section C.1. At a high level, our emphasis across all filtering strategies is to: (1) remove non-tabular data (such as text or PDFs incorrectly identified as tabular data during TabLib's collection), (2) ensure the *safety* of chosen tables (e.g. by removing PII), and (3) find sources with *high semantic content* (e.g. by removing tables with too many missing values). As part of this filtering process we develop and apply simple methods for deduplication, English language filtering, filtering for missing data, PII removal, code removal, and more.

Unsupervised Task Selection: As described in Section 4.1.2, we focus on methods for tabular prediction: predicting the value of a target column given the values of all other columns for an instance. Therefore, as part our data pipeline we develop new methods for selecting, in an unsupervised fashion, which column is the *target column* for each table in the corpus. Selecting targets of prediction for tabular data at scale is an under-explored problem. Prior work in this space operated on at most a few hundred tables and used either a combination of expensive queries to commercial LLMs or manual curation to identify tabular prediction targets [190, 206]. However, when operating on hundreds of millions of distinct tables with potentially no associated metadata, these strategies are not feasible.

For each table, we select a prediction target programmatically by first identifying a subset of columns that are suitable for prediction according to various heuristics, and then choosing a specific column at random from this set. The exact list of heuristics to arrive at this set is presented in Appendix C.1. Amongst others, these include excluding candidate columns if: the column name is numeric, it has only one unique value, or it has unique values for every row (excluding numeric columns).

Final T4 Dataset Summary: Running this entire filtering process (from raw data to serialized examples ready for training) on all 70TB of TabLib using our open-sourced implementation takes about 4 hours on

a CPU cluster. It yields a total of 3.1M tables, which equates to a table filtering rate of approximately 98.45%. The resulting dataset contains over 1.6B rows (approximately 80B Llama 3 tokens) for training of the downstream model, and occupies roughly 2TB compressed on disk. We note that 80B tokens is larger than the total number of tokens TabuLa-8B sees during training. Therefore, the model sees each distinct table at most once during training, and our pipeline could be scaled up to support larger models or longer training runs.

4.5 Experimental Results

4.5.1 Evaluation Methodology

We evaluate the transfer learning performance of TabuLa-8B on a diverse set of established benchmarks previously considered in prior work (see Section 4.5.2 for a list). For each dataset, we use the predefined prediction target from the original benchmark. Due to computational constraints, we evaluate TabuLa-8B on up to 128 test examples for each dataset and number of shots k.

The term "few-shot" is unfortunately overloaded. It is used both to refer to models that make predictions on instances never seen during training, and to models that directly *train* on these examples before predicting on unseen samples. We do not fine-tune our model on test examples, in contrast to [190, 84]. Our methodology only requires performing forward passes through the network to generate predictions and avoids the need for computationally-expensive gradient updates. In zero-shot evaluations, given a row of a dataset along with the corresponding set of columns and possible labels, we first serialize the row into the same format used during training, and feed it into the model to generate a prediction following the `<|endinput|>` token. For few-shot evaluations, we perform the same procedure, except that we preprend the serialized "shots" as in Figure 4.2b.

In contrast to methods like XGBoost that directly predict likelihoods of a set of labels, language models output likelihoods over a set of tokens (128k in the case of Llama 3). For each evaluation dataset, the values in the label set (e.g. "sun, rain, snow" in Figure 4.2b) can consist of a *sequence* of many individual tokens from this large vocabulary. Here, we use *open-vocabulary* (or "open-ended") accuracy [39, 3] as the main evaluation metric for our model. In this setup, once the model is prompted with a serialized example, it is allowed to generate an arbitrary sequence of tokens. Once it produces the `<|endofcompletion|>` token, the generated text is then directly compared to the correct completion. Only an exact match,

including the terminating `<|endofcompletion|>` token, is counted as accurate. This is more challenging than *closed*-vocabulary evaluation, where the model is only rated on assigning the highest probability to the correct completion from a predetermined set.

4.5.2 Evaluation Datasets

We evaluate our model's predictive performance across a collection of 329 publicly-available tabular datasets drawn from five tabular benchmarks (see Appendix C.7 for a full list). These include:

UniPredict Benchmark (169 datasets) [190]: We use the "supervised" subset of 169 datasets from the recently-introduced UniPredict benchmark. These are high-quality tabular datasets with generally informative column names and a mix of both categorical and continuous targets, drawn directly from Kaggle. While the model introduced in Wang et al. [190] was trained and tested on separate splits of these datasets, we only use them for testing. We make corrections to several datasets with targets erroneously treated as categorical in the original benchmark, described in Section 4.5.2.

Grinsztajn Benchmark (45 datasets) [74]: The Grinsztajn benchmark is a curated suite of datasets consisting of numeric and categorical features. This dataset is notable in that the original study by Grinsztajn et al. found that gradient boosted decision trees (GBDTs) consistently outperformed deep learning-based methods on these tasks.

AutoML Multimodal Benchmark (AMLB) (8 datasets) [168] : A suite of tables which include one or more free-text fields (such as an Airbnb description, or a product review). The benchmark is considered challenging for tree-based methods due to the non-standard text-based features. However, it also poses a challenge for LLMs since some columns can contain highly variable lengths of text.

OpenML CC-18 Benchmark (72 datasets) [25]: The OpenML Curated Classification Benchmark was created by applying filtering rules to extract a high-quality subset from the OpenML platform. The rules include: no artificial data sets, no subsets of larger data sets nor binarizations of other data sets, no data sets which are perfectly predictable by using a single feature or a simple decision tree.

OpenML CTR-23 Benchmark (35 datasets) [62]: The OpenML Curated Tabular Regression (CTR) Benchmark is a curated set of tables for regression drawn from OpenML. The curation process is similar to that of OpenML-CC18, for regression tasks. We note that, being primarily intended for the evaluation of

AutoML methods, the OpenML benchmarks are notable for lacking informative column names (i.e. names such as "`Var1`, `Var2`, ..." are common in OpenML benchmark datasets).

We transform all regression tasks into a 4-class classification based on quartiles, as in [190] (see Appendix C.1.2 for details). Many datasets contain rows with missing data. We leave these as-is and do not remove any data. On a computational note, some datasets contain a large numbers of features and performing k-shot evaluations on these datasets at large k can exceed the model's context window. Therefore, in few-shot evaluations, we always report results for the subset of datasets where k shots fit into the model's context window, for the entire specified range of k. We provide more details on the evaluation datasets in Appendix C.4.1 and report per-dataset results in Appendix C.7.

4.5.3 Baselines

When inspecting TabuLa-8B's performance, we compare against the following baselines:

Llama 3-8B [179]: This is the base model from which TabuLa-8B is fine-tuned. Comparing to the base model isolates the effects of the fine-tuning process. It also controls for any contamination of evaluation datasets that may be contained in pretraining data for Llama 3 (the exact training data for Llama 3 are not currently disclosed). We return to this point in Section 4.5.10.

XGBoost [38]: XGBoost is a supervised learning gradient-boosted decision tree (GBDT) method. It is widely considered to be highly competitive in tabular prediction tasks [74, 68, 122].

TabPFN [90]: This a transformed-based hypernetwork pretrained to reflect a set of inductive biases germane to tabular data. TabPFN is thus considered especially effective for few-shot learning [122, 90].

Whenever possible, we perform hyperparameter tuning on XGBoost and TabPFN in order to maximize their performance. See Appendix C.4.2 for further details on baseline implementation and tuning.

4.5.4 Main Results: Assessing TabuLa-8B's Transfer Learning

We present our main experiments evaluating the transfer learning ability of TabuLa-8B in Figure 4.4. As a whole, TabuLa-8B demonstrates strong transfer performance across the broad range of tasks.

In the zero-shot regime (seen in the left-most point for each plot in Figure 4.4) – where the model is presented

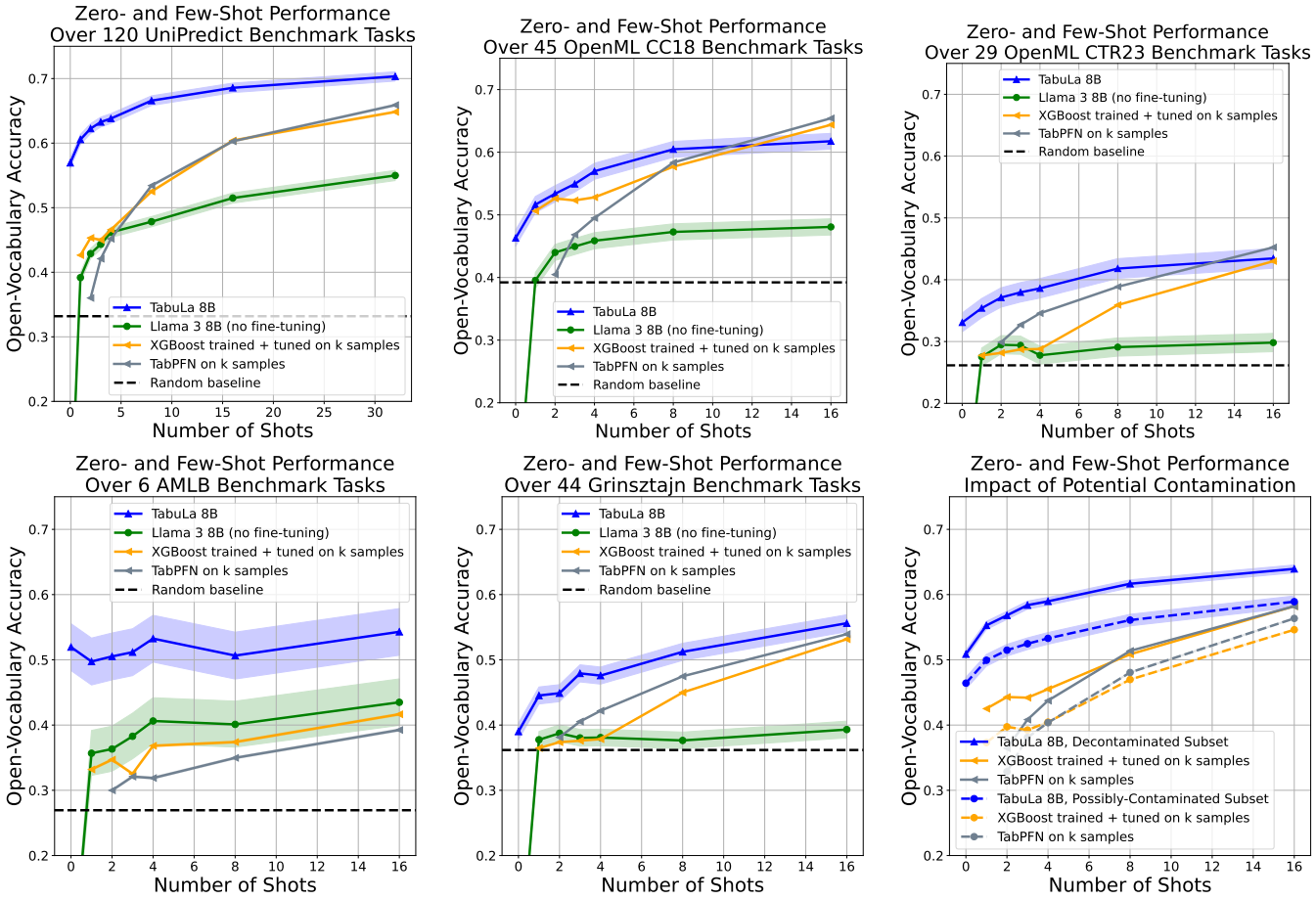


Figure 4.4: Zero- and few-shot accuracy across five tabular benchmarks. For each benchmark, we evaluate on all tasks, but in the figures above we only display the subset of tasks where k shots fit into the 8192-token context window of TabuLa-8B. Complete results are in Supplementary Section C.7. The final plot (lower right) shows curves separately over decontaminated vs. potentially-contaminated evaluation tasks (see Section 4.5.10); we find no impact on our overall findings due to contamination (and performance on tasks which may be in our training set is *lower* on average, across all models).

with no further information about the target dataset except for the serialized key-value pairs and set of possible labels for a single row –TabuLa-8B is between 5 to 25 pp more accurate than a random baseline and 50 pp more accurate than the base Llama 3 model. This illustrates one of the key benefits of using language models for tabular prediction: after fine-tuning, TabuLa-8B can leverage semantic information contained in the serialized data to make high-quality predictions.

While XGboost and TabPFN are not capable of zero-shot prediction, this behavior has been observed in the original Llama 3 model [179, 183]. However, in our evaluations, Llama 3 performs below random guessing in the zero-shot setting. We hypothesize that the base Llama 3 model requires a small number of samples to understand the input-output format and task (as indicated by the large leap in Llama 3 performance from $0 \rightarrow 1$ shot).

In the few-shot setting, where each method additionally sees a small number of labeled examples, TabuLa-8B's performance steadily improves with the number of shots. In the regime of 1 to 32 shots, it outperforms state-of-the-art models (XGBoost and TabPFN) that are directly trained (and hyperparameter tuned) on each specific dataset by 5-20pp. Once we evaluate performance on 32, 64, or 128 shots (see Figure C.1b), this gap begins to diminish, but the number of datasets that can fit > 32 shots into the 8192-token context window is both small and a relatively biased sample (due to their small number of features). TabuLa-8B is consistently 10 to 20pp above the Llama 3-8B base model for the full range of shots, highlighting the benefits of our training procedure on T4.

Improvements in Sample Efficiency: As discussed previously, the main benefit of transferable models is that they reduce the amount of data necessary to achieve good performance on new tasks. For instance, as seen in Figure 4.4, TabuLa-8B only needs one shot to achieve 60% average accuracy on UniPredict tasks. However, both TabPFN and XGBoost only reaches 60% accuracy after 16 shots. Therefore, TabuLa-8B reduces the amount of data necessary to achieve 60% accuracy by 16 fold relative to XGBoost annd TabPFN. We refer to this statistic as the relative sample efficiency (see C.4.3). TabuLa-8B in general achieves higher accuracy than the benchmarks using less data. Hence, the relative sample efficiency is always > 1 (the exact ratio varies across benchmarks).

Impact of Informative Column Headers: As shown in Figure 4.4, while TabuLa-8B generally has higher accuracy than the baselines, this accuracy gap varies across benchmarks and the number of shots.

For instance, for the UniPredict benchmark – which was specifically constructed to include datasets with semantically-meaningful column headers [190] – the gap to supervised baselines is much larger than in the OpenML benchmarks, which tend to have less semantically-meaningful column names. If meaningful column headers are absent, the model still performs well (matching or outperforming XGBoost at shots $k \leq 8$), but its advantage over these strong baselines is lessened. We investigate this effect in further detail with a controlled experiment in Section 4.5.6.

4.5.5 Ablation Study: Row-Causal Table Masking (RCTM)

In this section we conduct an ablation study of the row-causal table masking (RCTM) procedure used to train our model. To summarize the masking procedure (also described in Section 4.3 and Figure 4.2a): we explicitly allow the model to attend across samples within the same table in a batch. We hypothesize that this structure will encourage few-shot learning and will mitigate catastrophic forgetting which could cause the base model's few-shot capabilities to deteriorate during fine-tuning.

To do this, we design a controlled experiment. Both arms of the experiment use 10% of the compute of TabuLa-8B (trained for $4k$ steps) but are otherwise identical. In one run, we remove the RCTM strategy described in Figure 4.2a, replacing it with a per-sample causal attention mask (the model is not allowed to attend to any samples besides the target sample, regardless of which table they are derived from). We evaluate both models over the full test suite (all benchmark tasks).

The results of this study are shown in Figure 4.5. Our proposed masking scheme improves the models' ability to attend across samples, while removing this masking causes the model not to learn from additional shots (for $k \leq 8$) and to deteriorate as the number of shots grows. This figure also demonstrates the potential loss of few-shot capabilities during fine-tuning if it is not explicitly encouraged by the fine-tuning task – but also that these capabilities can be maintained or improved over the base model if they are a part of the fine-tuning task.

4.5.6 Robustness Evaluation: Informative Header Removal

A potential disadvantage of a language modeling approach to tabular data prediction is that language models may be particularly reliant on semantically-informative column "headers" (column names, the keys in the tabular key-value structure), in contrast to traditional supervised learning methods which do not utilize

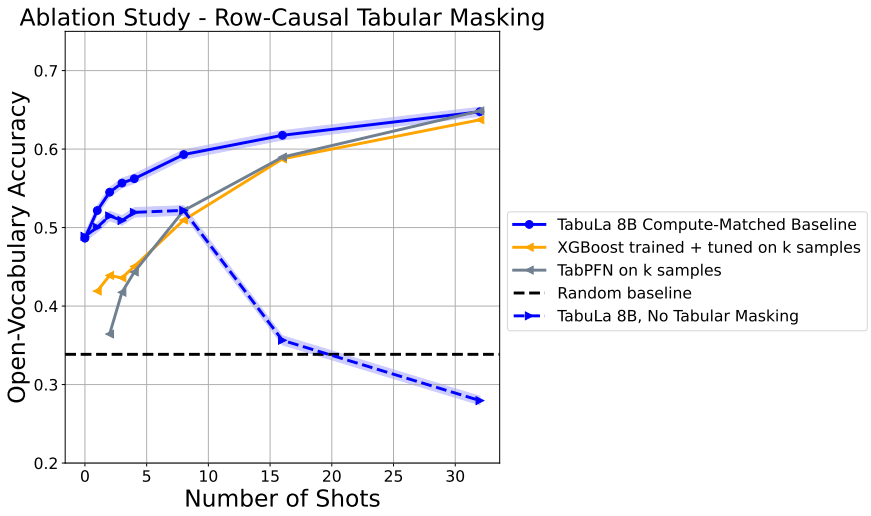


Figure 4.5: Results of an ablation study comparing a model trained without our novel row-causal tabular masking (RCTM) scheme (described in Section 4.3 and illustrated in Figure 4.2a) vs. a baseline compute-matched version of TabuLa-8B. RCTM improves the models' ability to attend across samples, while removing RCTM causes the model not to learn from additional shots (for $k \leq 8$) and to *deteriorate* as the number of shots grows. This result demonstrates the potential loss of few-shot capabilities during fine-tuning if it is not explicitly encouraged by the fine-tuning task.

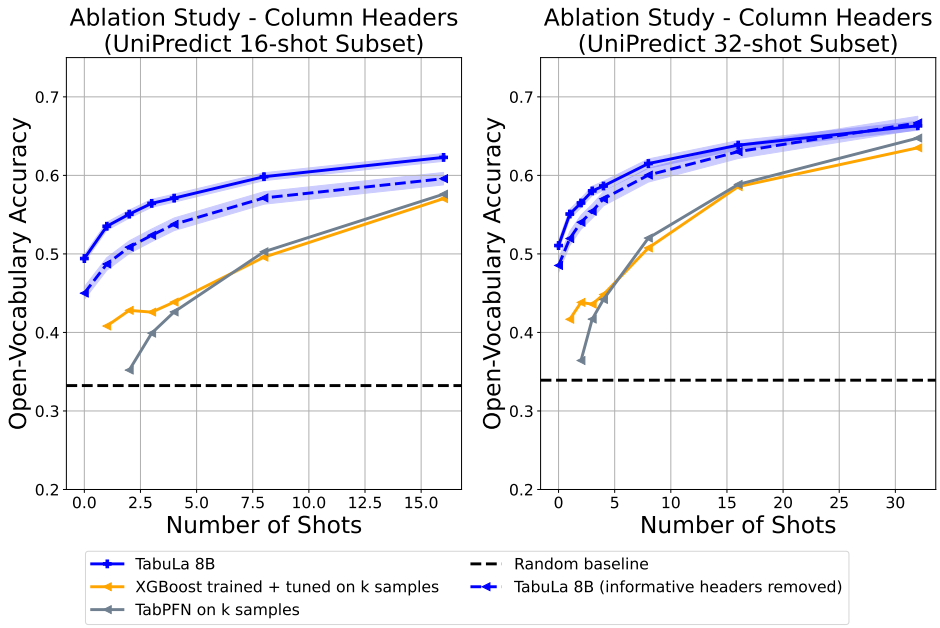


Figure 4.6: Results of column header ablation study described in Section 4.5.6. For low numbers of shots, there tends to be a positive effect from informative headers. However, as the number of shots increases, the utility of the headers decreases.

the headers at all. This risk has been noted in previous work; for example, UniPredict [190] suggests an approach that uses a commercial LLM to "rewrite" the headers of a table to make them more informative. In this work, however, we seek to avoid expensive preprocessing and use only the provided headers from our training data.

To understand this sensitivity to the semantic quality of the headers, we conduct a controlled experiment to test the effect of removing informative column headers from a dataset. Specifically, we do the following: starting from a benchmark with high-quality headers (UniPredict), we *replace* the original headers with "X1", "X2", ... and the target with "Y". Then, we evaluate TabuLa-8B on the data. We do not alter the data itself; only the feature names are replaced.

The results of this study are shown in Figure 4.6. We highlight a few key findings from these results. First, for small number of shots, the semantically-meaningful headers provide a performance benefit: for instance, in the 16-shot subset of Figure 4.6, semantically-meaningful headers provide a consistent accuracy

gain of 3-5pp. Second, TabuLa-8B can still outperform supervised baselines even without these headers: for example, Figure 4.6 shows that TabuLa-8B still outperforms all baselines on the benchmark even without semantically-meaningful headers. This finding is further supported by our results on the OpenML benchmarks (CC18, CTR23) in Figure 4.4; these datasets also tend to have uninformative headers. Third, we observe that the utility of semantically-meaningful headers decreases as the number of shots increase. For example, at 32 shots, the performance with and without the original headers is effectively identical. We hypothesize that, as the number of shots grows, the model is increasingly utilizing the *values* provided in the shots (and their distribution) and is less reliant on the *keys* for providing information about the task.

Collectively, the results of this ablation study suggest that TabuLa-8B is robust to the semantic content of the headers, and that TabuLa-8B is capable of providing effective tabular data predictions even in the absence of rich column headers.

4.5.7 Robustness Evaluation: Feature Dropout

A potential risk of using language models for tabular data prediction may be their brittleness: language models, particularly in the few-shot setting, are known to be sensitive to details which should be irrelevant to the task difficulty, including order of examples [117, 27], whitespace [166], and prompt formatting [166, 94, 164]. Here, we are particularly interested in probing robustness to the *removal* of features. Some supervised models, including XGBoost, can be trained in a way that allows them to handle missing features at inference time, at a cost of slightly decreased predictive accuracy due to the loss of information. However, whether language models possess similar characteristics is unknown.

We design an experiment to test how TabuLa-8B's zero- and few-shot performance degrades as features are removed from the evaluation datasets. First, on the training split of each dataset, we train a single XGBoost model using the same hyperparameter tuning procedure used for our baselines, and we extract the feature importance for all features in the dataset according to this model. Next, we evaluate TabuLa-8B on each dataset, removing the top k features for $k \in [1, 5]$. These features are removed in either descending or ascending order of importance ("important first" and "important last", respectively). For comparison, we also train and evaluate XGBoost models. For these XGBoost models, we set $1/k$ fraction of the data to missing, uniformly at random. Then we train a hyperparameter-tuned model on this data and evaluate it on clean test data where the top-k features are set to missing (no other data is set to missing in the test data);

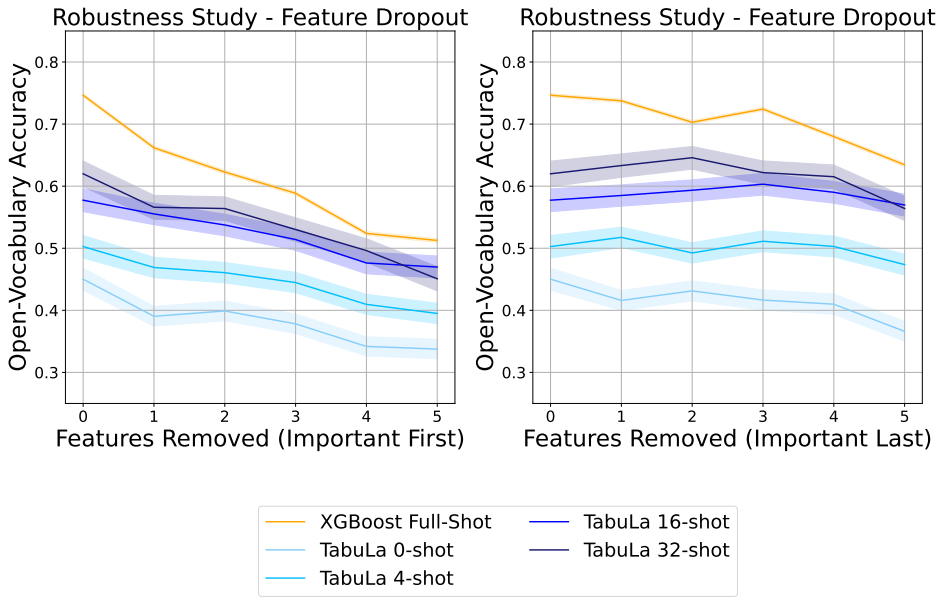


Figure 4.7: Feature dropout ablation study results. We compare the few-shot performance of TabuLa-8B to the performance of an XGBoost model trained on the full training split of each dataset, progressively removing features at evaluation time. Features are removed in order of decreasing ("important first") or increasing ("important last") variable importance (see Section 4.5.7 for details). TabuLa-8B's performance decreases at a rate consistent with XGBoost.

this allows us to naturally leverage XGBoost's robustness to missing data.

This experiment is conducted on the same random sample of 32 datasets described in Section 4.5.8. The results of this study are shown in Figure 4.7. Across 0-,4-,16-,and 32-shot evaluations, TabuLa-8B shows a similar or favorable rate of decline in performance, relative to XGBoost, as the number of removed features increases (as indicated by the similar slope of the lines). We note that the XGBoost models have better absolute performance because these are *full-shot* models trained on the full training split; we use these models to compare the rate of decrease in accuracy as dropout increases, not to compare the accuracy itself. The results in Figure 4.7 also provide further evidence of when TabuLa-8B may be favorable to standard supervised learning methods: namely, when the amount of missing data at test time is large. Finally, Figure 4.7 suggests that TabuLa-8B is *not* brittle with respect to the features present in our evaluation datasets; removing these features causes only the expected drop in performance (and removing *unimportant* features

is even associated with an increase in performance relative to the full feature set when the number of shots is larger than 8).

4.5.8 Robustness Evaluation: Column Order Invariance

Recent work has suggested that invariance to column ordering is an important property of tabular foundation models [186]. In this subsection, we conduct a controlled experiment to assess the sensitivity of our model to the *ordering* of the columns in the dataset.

To assess this, we conduct the following experiment. We first choose a random subset of 32 tasks from our evaluation suite (we exclude the tasks from AMLB due to their relatively irregular structure as discussed above since our goal is to compare performance on relatively standard tabular data). For each task, we evaluate the model on a *permuted* version of the original data: that is, we randomly permute the columns, but otherwise leave the data unchanged. Due to computational constraints, we conduct this evaluation only at a coarse grid of $(0, 8, 16, 32)$ shots. We compare the accuracy on these tasks to the accuracy on the original data. We use the same TabuLa-8B model for both the permuted and non-permuted evaluations.

The results of this study are shown in Figure 4.8. We observe a small drop (with Clopper-Pearson intervals overlapping at all points) after permuting the columns, but the general shape and rate of increase of the model under both cases is quite similar. We hypothesize that this drop is due to the fact that many tabular datasets, including those from our evaluation benchmark (which are drawn from manually-curated sources such as Kaggle, OpenML, and UCI Machine Learning Repository), have manually-selected column *orderings* that slightly improve prediction performance. This might include, for example, a "date" preceding the rest of the columns, or a "high" and "low" column located near each other in a stock dataset. These feature relationships, we hypothesize, can make it easier for models to pool information between related features and may contribute to the small drop observed on permuting the columns. We note that this sensitivity to order has been observed for language models in other contexts [117, 126].

Our results broadly show that TabuLa-8B maintains consistent performance above baselines even under feature permutation. However, they also suggest that the sensitivity to prompting and formatting that affects LLMs in other contexts [117, 126] could possibly affect tabular LLMs. We believe further research into this issue is necessary, and may indeed point toward future, more effective methods for leveraging

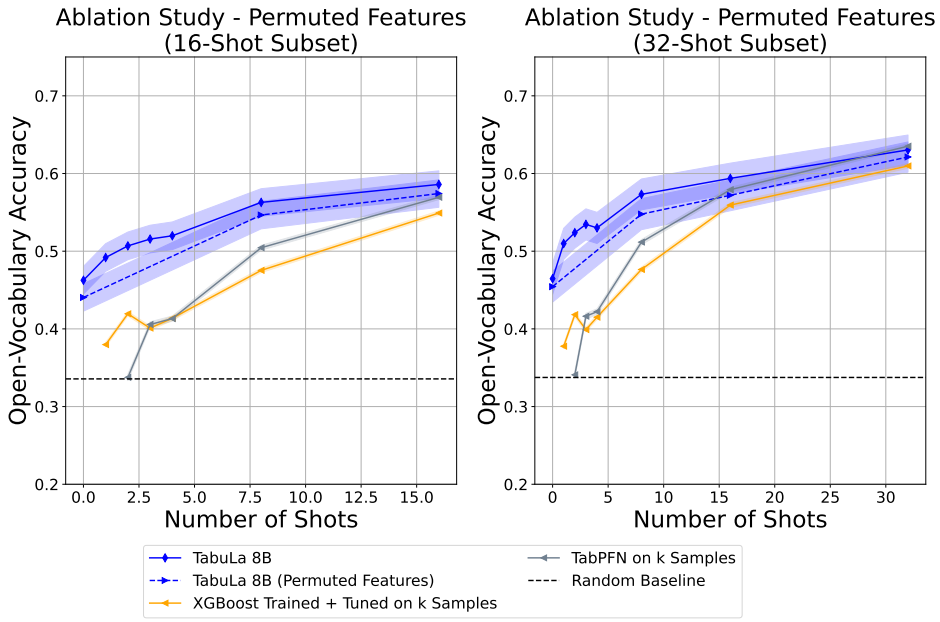


Figure 4.8: Results of column permutation study described in 4.5.8. We randomly permute the columns for a randomly-selected subset of 32 tasks, and evaluate TabuLa-8B on the permuted data. We hypothesize that the slight drop in model performance is due to manually-crafted and semantically meaningful feature orderings in the data. However, our results broadly show that TabuLa-8B maintains consistent performance above baselines even under feature permutation.

feature ordering to improve model performance.

4.5.9 Ablation Study: Data Filtering

We conduct a controlled experiment to assess the impact of our filtering strategies. In particular, we compare a dataset filtered according to the strategies described in Section 4.4 to a "minimally-filtered" TabLib dataset. We use a "minimally-filtered" dataset rather than an *unfiltered* dataset because applying no filtering could potentially result in a small number of very large tables dominating training. For the minimally-filtered baseline, we only apply the max-row count filter, max column filter, and max header length filter; the latter two help ensure that the resulting serializations are not too long so that the model is still able to perform few-shot training.

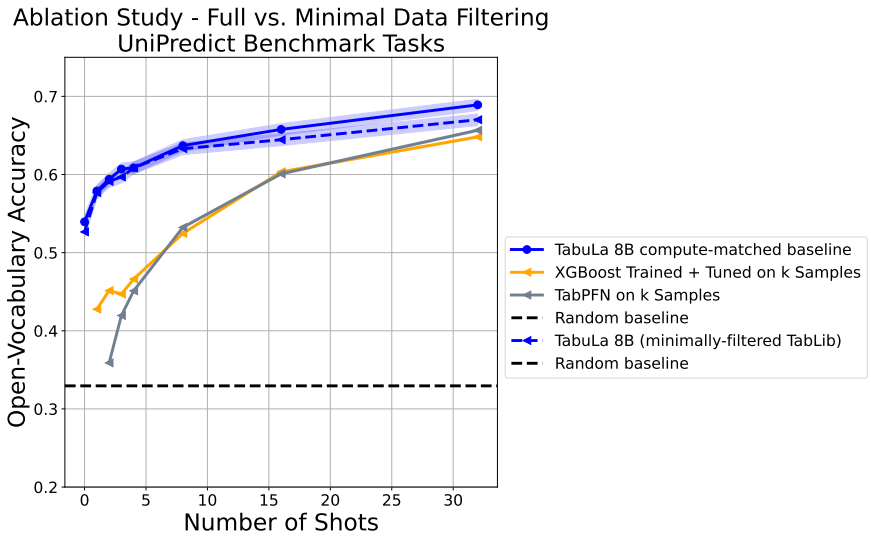


Figure 4.9: Compute-matched comparison of TabuLa-8B with our full filtering vs. an identical model trained on minimally-filtered TabLib (both models are trained for 10% of the number of steps as the final model, with identical hyperparameters).

The results of this study are shown in Figure 4.9. While our filtering strategy has a positive impact at larger numbers of shots ($k \geq 16$), there is no impact evident at $k < 16$, with the minimally-filtered baseline performing similarly. We hypothesize that this is a reflection of the relatively limited additional filtering performed by the rest of our pipeline relative to the "minimal" baseline (which consists mainly of language filtering, PII and code filtering, and heuristics to remove excessive amounts of missing data). Additionally, given the clear impact of improvements in data quality for language model pretraining (e.g. [179]), we hypothesize that further refinements of our filtering pipeline would be likely to achieve further gains over minimal filtering. Finally, we emphasize that some of our filtering strategies (in particular, the PII detection) were designed to improve the *safety* of the model, not the quality, and we believe that some form of safety filtering should still be used regardless of its downstream effects.

4.5.10 Assessing the Potential Impact of Data Contamination

Given that T4 consists of 3.1Mtables sourced from public data sources (Common Crawl, Github) and that our evaluations are also comprised of public benchmarks, we investigate the extent and possible impact of

data contamination – that is, training datasets that are part of the evaluation suite. In Section C.6, we explain our methodology to test for the potential presence of benchmark datasets in T4. Using a conservative identification strategy based on column matching (likely to include false positives). We find at most one-third of benchmark tables may occur at least once in the training set. When training large-scale models for transfer learning, it is not always clear *a priori* what the eventual application domains will be. Therefore, we believe that it is an important research question to understand the extent to which contamination may affect performance, as contamination may be difficult to prevent in some cases. Initial foundation modeling efforts in non-tabular domains adopted a similar approach, and found mixed or no impact from overlap [146, 144].

We evaluate the impact of contamination in our experimental setup by evaluating TabuLa-8B separately on "potentially contaminated" vs. uncontaminated tables. Our results are shown in the bottom right plot of Figure 4.4, as well as in Figures C.2 and C.3. Summarizing, we find no clear evidence that contamination affects model performance on the test suite, or that transfer ability is affected by contamination. In fact, as seen in Figure 4.4, the gap between TabuLa-8B and XGBoost is in fact *larger* if we restrict evaluation to the benchmark tables which we verify are not in T4. In addition to verifying that our results continue to hold over a diverse set of tables which we know the model did not see in training, it also shows that having some amount of potential contamination did not upwardly bias our estimate of TabuLa-8B's transfer learning ability. We hypothesize that the observed gap in Figure 4.4 is due to our conservative duplication procedure being more likely to flag datasets with generic or common column names, which also leads to *worse* baseline performance on these tasks. We present more comprehensive investigation on the effects of contamination in Appendix C.6.

4.6 Discussion

Limitations: TabuLa-8B has several limitations. First, TabuLa-8B has a limited context window of 8192 tokens. This restricts the number of examples that can be utilized for few-shot learning, as well as the additional information (such as text context or extended feature descriptions) that are available to the model. We expect that this limitation will be eased as the availability of longer-context models grows [e.g. 178]. Second, TabuLa-8B has 8B parameters, which makes serving and inference expensive and limits the environments it may be deployed in. Lastly, given that it uses a pretrained LLM as a base-model and fine

tunes on a web-scale corpora of historical datasets that likely contain various social biases, TabuLa-8B introduces new potential fairness considerations that are not present when using preexisting supervised methods such as XGBoost. We hope that by open sourcing the model, data, and code, we might enable future research addressing these important directions.

Future Work: Our work on transfer learning for tabular data is the first its kind at this scale, and there are several avenues for future research. These can be coarsely categorized as either improvements (of the existing dataset and model) or extensions (deeper investigations into the model and data itself).

On the *improvements* side, we see several promising directions. These include improvements in tabular data filtering (this has been the main axis of improvement in recent generations of language models); scaling the model + data + compute; exploring the use of inference-time strategies to improve prediction (such as self-consistency [192], prompt ensembling [144, 3], or in-context example selection [156]); and introducing extra information during both training and inference, such as contextual information or samples from different, but related, tables.

On the *extensions* side, we hope that our work opens avenues toward deeper understanding of tabular foundation models including: understanding potential biases or unwanted behavior with respect to sensitive features (features such as race, age, and gender are common in tabular datasets); using tabular foundation models to address small-sample problems which might be aided by a high-quality pretrained model (such as in the Fragile Families Challenge [161]); and extending this approach to new tasks beyond prediction, such as data generation, explanation, data wrangling, and more.

4.7 Accessing Open-Source Code, Data, and Model Weights

TabLib Preprocessing Code: We provide Python module for filtering TabLib, `tabliblib`, along with scripts and configurations used to perform the filtering, are available at https://github.com/mlfoundations/tabliblib.

Model Training and Inference Code: We provide `rtfm`, a Python module used to train TabuLa-8B, perform inference and evaluation, and process data, at https://github.com/mlfoundations/rtfm.

T4 Dataset: The T4 dataset is available via public credentialized access on Hugging Face datasets at https://huggingface.co/datasets/mlfoundations/t4-full. Because the dataset is derived from TabLib, users

must first obtain permission to access TabLib at https://huggingface.co/datasets/approximatelabs/tablib-v1-full.

Evaluation Datasets: The full evaluation suite used to evaluate TabuLa-8B is available via Hugging Face Datasets at https://huggingface.co/datasets/mlfoundations/tabula-8b-eval-suite. Each dataset includes: a CSV file containing the raw data; a TableShift [67] FeatureList JSON object; and a YAML file with associated metadata.

Model Weights: TabuLa-8B weights are provided via Hugging Face at https://huggingface.co/mlfoundations/tabula-8b.

4.8 Acknowledgements

This work was supported by a Microsoft Grant for Customer Experience Innovation. This work was also in part supported by the NSF AI Institute for Foundations of Machine Learning (IFML, CCF-2019844), Google, Open Philanthropy, and the Allen Institute for AI. JCP was supported by the Harvard Center for Research on Computation and Society.

We are grateful to Foundry[1] for providing the compute infrastructure used to train TabuLa-8B. Our research also utilized computational resources and services provided by the Hyak computing cluster at the University of Washington, and from Stability AI. The authors also gratefully acknowledge the Gauss Centre for Supercomputing[2] for funding this project by providing computing time on the GCS Supercomputer JUWELS[97] at Julich Supercomputing Centre (JSC). We are particularly appreciative of support from Jenia Jitsev at JSC.

We also acknowledge Approximate Labs[3] and express our appreciation for their development and release of TabLib, along with their communication and support as we utilized the dataset.

We are grateful to Jeffrey Li, Mike Merrill, Jonathan Hayase, and Nilesh Tripuraneni for feedback on a early versions of this paper. We also benefited greatly from advice from Matt Jordan, Alex Fang, and Jeffrey Li on large-scale data preprocessing.

---

[1] https://www.mlfoundry.com

[2] www.gauss-centre.eu

[3] https://www.approximatelabs.com

BIBLIOGRAPHY

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[2] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[4] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020.

[5] American Heart Association. The Facts About High Blood Pressure. https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure, 2017. Accessed: 2023-01-06.

[6] American Heart Association. Health Threats from High Blood Pressure. https://www.heart.org/en/health-topics/high-blood-pressure/health-threats-from-high-blood-pressure, 2022. Accessed: 2023-01-06.

[7] American National Election Studies (ANES). ANES Time Series Cumulative Data File, 1948-2020, 2020.

[8] Sercan O Arık and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *AAAI*, volume 35, pages 6679–6687, 2021.

[9] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[10] American Diabetes Association. Economic costs of diabetes in the us in 2017. *Diabetes care*, 41(5):917–928, 2018.

[11] Algernon Austin. A good credit score did not protect latino and black borrowers. 2012.

[12] Anas Awadalla, Mitchell Wortsman, Gabriel Ilharco, Sewon Min, Ian Magnusson, Hannaneh Hajishirzi, and Ludwig Schmidt. Exploring the landscape of distributional robustness for question answering models. *arXiv preprint arXiv:2210.12517*, 2022.

[13] Clare Bambra and Terje A Eikemo. Welfare state regimes, unemployment and health: a comparative study of the relationship between unemployment and self-reported health in 23 european countries. *Journal of Epidemiology & Community Health*, 63(2):92–98, 2009.

[14] Michelle Bao, Angela Zhou, Samantha Zottola, Brian Brubach, Sarah Desmarais, Aaron Horowitz, Kristian Lum, and Suresh Venkatasubramanian. It's compaslicated: The messy relationship between rai datasets and algorithmic fairness benchmarks. *arXiv preprint arXiv:2106.05498*, 2021.

[15] Matias Barenstein. Propublica's compas data revisited. *arXiv preprint arXiv:1906.04711*, 2019.

[16] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. 2017.

[17] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[18] Yahav Bechavod and Katrina Ligett. Learning fair classifiers: A regularization-inspired approach. 2017.

[19] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[20] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[21] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

[22] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.

[23] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

[24] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[25] Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Frank Hutter, Michel Lang, Rafael G. Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv:1708.03731v2 [stat.ML]*, 2019.

[26] Tony A Blakely, Sunny CD Collings, and June Atkinson. Unemployment and suicide. evidence for a causal association? *Journal of Epidemiology & Community Health*, 57(8):594–600, 2003.

[27] Rishi Bommasani, Percy Liang, and Tony Lee. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146, 2023.

[28] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *arXiv preprint arXiv:2110.01889*, 2021.

[29] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[30] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,

Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[32] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[33] N Cable, A Sacker, and M Bartley. The effect of employment on psychological health in mid-adulthood: findings from the 1970 british cohort study. *Journal of Epidemiology & Community Health*, 62(5):e10–e10, 2008.

[34] L Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *International Conference on Machine Learning*, pages 1349–1359. PMLR, 2020.

[35] Centers for Disease Control and Prevention. National Diabetes Statistics Report. https://www.cdc.gov/diabetes/data/statistics-report/index.html, 2022. Accessed: 2023-01-05.

[36] Centers for Disease Control and Prevention (CDC). National Health and Nutrition Examination Survey Questionnaire, Examination Protocol, and Laboratory Protocol (1999, 2001, 2003, 2005, 2007, 2009, 2011, 2013, 2015, 2017), 2017.

[37] Centers for Disease Control and Prevention (CDC). BRFSS Survey Data (2015, 2017, 2019, 2021), 2021.

[38] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[39] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2022.

[40] Jianfeng Chi, Yuan Tian, Geoffrey J Gordon, and Han Zhao. Understanding and mitigating accuracy disparity in regression. In *International Conference on Machine Learning*, pages 1866–1876. PMLR, 2021.

[41] Marshall H Chin, James X Zhang, and Katie Merrell. Diabetes in the african-american medicare population: morbidity, quality of care, and resource utilization. *Diabetes Care*, 21(7):1090–1095, 1998.

[42] Jung Hyun Choi, Alanna McCargo, Michael Neal, Laurie Goodman, and Caitlin Young. Explaining the black-white homeownership gap. *Washington, DC: Urban Institute.*, 25:2021, 2019.

[43] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[44] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[45] Giselle M Corbie-Smith. Minority recruitment and participation in health research. *North Carolina medical journal*, 65(6):385–387, 2004.

[46] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. 2024.

[47] Emily Denton, Mark Díaz, Ian Kivlichan, Vinodkumar Prabhakaran, and Rachel Rosen. Whose ground truth? accounting for individual and collective identities underlying dataset annotation. *arXiv preprint arXiv:2112.04554*, 2021.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[49] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[50] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

[51] Samuel Dooley, Ryan Downing, George Wei, Nathan Shankar, Bradon Thymes, Gudrun Thorkelsdottir, Tiye Kurtz-Miott, Rachel Mattson, Olufemi Obiwumi, Valeriia Cherepanova, et al. Comparing human and machine bias in face recognition. *arXiv preprint arXiv:2110.08396*, 2021.

[52] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[53] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.

[54] John Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses for latent covariate mixtures. *Operations Research*, 2022.

[55] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[56] Scientific American Editors. Clinical trials have far too little racial and ethnic diversity. *Scientific American*, 2018.

[57] Edward Metz. ASSISTments: From Research to Practice at Scale in Education. https://ies.ed.gov/blogs/research/post/assistments-from-research-to-practice-at-scale-in-education, 2020. Accessed: 2023-06-01.

[58] Gus Eggert, Kevin Huo, Mike Biven, and Justin Waugh. Tablib: A dataset of 627m tables with context. *arXiv preprint arXiv:2310.07875*, 2023.

[59] Federal Trade Commission. Press Release: Marketers of Blood-Pressure App Settle FTC Charges Regarding Accuracy of App Readings. https://www.ftc.gov/news-events/news/press-releases/2016/12/marketers-blood-pressure-app-settle-ftc-charges-regarding-accuracy-app-readings, 2016. Accessed: 2023-02-09.

[60] Li Fei-Fei, Jia Deng, and Kai Li. Imagenet: Constructing a large-scale image database. *Journal of vision*, 9(8):1037–1037, 2009.

[61] FICO. The Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge, 2019. Accessed: 2023-01-10.

[62] Sebastian Felix Fischer, Liana Harutyunyan Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*, 2023.

[63] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P

Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

[64] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[65] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[66] Flávio D Fuchs and Paul K Whelton. High blood pressure and cardiovascular disease. *Hypertension*, 75(2):285–292, 2020.

[67] Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36, 2024.

[68] Joshua P Gardner, Zoran Popovi, and Ludwig Schmidt. Subgroup robustness grows on trees: An empirical baseline investigation. In *Advances in Neural Information Processing Systems*, 2022.

[69] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[70] Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

[71] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[72] Pieter Gijsbers, Erin LeDell, Janek Thomas, Sébastien Poirier, Bernd Bischl, and Joaquin Vanschoren. An open source automl benchmark. *arXiv preprint arXiv:1907.00909*, 2019.

[73] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[74] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*, 2022.

[75] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 2024.

[76] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[77] Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

[78] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. Towards a critical race methodology in algorithmic fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 501–512, 2020.

[79] Hanson, Melanie. Average Cost of College & Tuition. https://educationdata.org/average-cost-of-college, 2023. Accessed: 2023-06-01.

[80] Vasyl Harasymiv. Lessons from 2 million machine learning models on kaggle, Dec 2015.

[81] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[82] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

[83] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[84] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR, 2023.

[85] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards

the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

[86] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[87] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

[88] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[89] Julia Hippisley-Cox and Carol Coupland. Development and validation of qdiabetes-2018 risk prediction algorithm to estimate future risk of type 2 diabetes: cohort study. *bmj*, 359, 2017.

[90] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

[91] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[92] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[93] Natalie Jacewicz. Why are health studies so white? *The Atlantic*, 2016.

[94] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 07 2020.

[95] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad

Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[96] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[97] Jülich Supercomputing Centre. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities*, 7(A138), 2021.

[98] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in Neural Information Processing Systems*, 34, 2021.

[99] Michael Kahn. Diabetes Data Set, 1994.

[100] Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*, 2024.

[101] Liran Katzir, Gal Elidan, and Ran El-Yaniv. Net-dnf: Effective deep modeling of tabular data. In *International Conference on Learning Representations*, 2020.

[102] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[103] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572. PMLR, 2018.

[104] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 100–109, 2019.

[105] Kevin L. Matthews II. There's a 'credit gap' between Black and white Americans, and it's holding Black Americans back from building wealth. https://www.businessinsider.com/personal-finance/credit-gap-black-americans-building-wealth-2021-1, 2021. Accessed: 2023-01-10.

[106] Fereshte Khani, Aditi Raghunathan, and Percy Liang. Maximum weighted loss discrepancy. *arXiv preprint arXiv:1906.03518*, 2019.

[107] Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. Carte: Pretraining and transfer for tabular learning. 2024.

[108] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[109] R. Kohavi and B. Becker. UCI adult data set., 1996.

[110] Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[111] Mario Michael Krell, Matej Kosec, Sergio P Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*, 2021.

[112] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[113] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33:8847–8860, 2020.

[114] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018.

[115] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[116] Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. Are we learning yet? a meta

review of evaluation failures across machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[117] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, 2022.

[118] Subha Maity, Debarghya Mukherjee, Mikhail Yurochkin, and Yuekai Sun. Does enforcing fairness mitigate biases caused by subpopulation shift? *Advances in Neural Information Processing Systems*, 34:25773–25784, 2021.

[119] Andrey Malinin, Neil Band, Yarin Gal, Mark Gales, Alexander Ganshin, German Chesnokov, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, et al. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2022.

[120] Kassandra Martinchek, Alex Carther, Breno Braga, Caleb Quakenbush, Signe-Mary McKernan, Allison Feldman, JoElla Carman, Luis Melgar, Liza Hagerman, and Laura Swanson. Credit health during the covid-19 pandemic. *The Urban Data Institute*, 2022.

[121] Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Douwe Kiela, David Jurado, et al. Dataperf: Benchmarks for data-centric ai development. *arXiv preprint arXiv:2207.10062*, 2022.

[122] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36, 2024.

[123] Megan Leonhart. Black and Hispanic Americans often have lower credit scores—here's why they're hit harder. https://www.cnbc.com/2021/01/28/black-and-hispanic-americans-often-have-lower-credit-scores.html, 2021. Accessed: 2023-01-10.

[124] John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution

shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR, 2020.

[125] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021.

[126] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022.

[127] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[128] Scott M Montgomery, Mel J Bartley, Derek G Cook, and Michael Ej Wadsworth. Health and social precursors of unemployment in young men in great britain. *Journal of Epidemiology & Community Health*, 50(4):415–422, 1996.

[129] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[130] National Institutes of Health National Hearth, Lung, and Blood Institute. High Blood Pressure: Causes and Risk Factors. https://www.nhlbi.nih.gov/health/high-blood-pressure/causes, 2022. Accessed: 2023-01-08.

[131] National Science Foundation. Directorate for Engineering Data Management Plans Guidance for Principal Investigators. https://www.nsf.gov/eng/general/ENG_DMP_Policy.pdf, 2018. Accessed: 2023-06-01.

[132] Douglas Noble, Rohini Mathur, Tom Dent, Catherine Meads, and Trisha Greenhalgh. Risk models and scores for type 2 diabetes: systematic review. *Bmj*, 343, 2011.

[133] National Academies of Sciences Engineering, Medicine, et al. Health-care utilization as a proxy in disability determination. 2018.

[134] Sam S Oh, Joshua Galanter, Neeta Thakur, Maria Pino-Yanes, Nicolas E Barcelo, Marquitta J White, Danielle M de Bruin, Ruth M Greenblatt, Kirsten Bibbins-Domingo, Alan HB Wu, et al. Diversity in clinical and biomedical research: a promise yet to be fulfilled. *PLoS medicine*, 12(12):e1001918, 2015.

[135] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[136] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[137] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[138] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 2018.

[139] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

[140] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

[141] PR Newswire. Black and Hispanic Americans on the U.S. financial system: "The odds were always against me," new Credit Sesame survey finds. https://www.prnewswire.com/news-releases/black-and-hispanic-americans-on-the-us-financial-system-the-odds-were-always-against-me-new-credit-sesame-survey-finds-301215072.html, 2021. Accessed: 2023-01-10.

[142] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[143] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, 83:112–134, 2018.

[144] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[145] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[146] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

[147] Edward Raff, Jared Sylvester, and Steven Mills. Fair forests: Regularized tree induction to minimize model bias. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 243–250, 2018.

[148] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[149] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[150] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, 21, 2008.

[151] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[152] Steffen Rendle, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395*, 2019.

[153] Matthew A Reyna, Chris Josef, Salman Seyedi, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D Clifford. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In *2019 Computing in Cardiology (CinC)*, pages Page–1. IEEE, 2019.

[154] Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[155] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[156] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, 2022.

[157] Cynthia Rudin, Caroline Wang, and Beau Coker. The age of secrecy and unfairness in recidivism prediction. *arXiv preprint arXiv:1811.00731*, 2018.

[158] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[159] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[160] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[161] Matthew J Salganik, Ian Lundberg, Alexander T Kindel, and Sara McLanahan. Introduction to the special collection on the fragile families challenge. *Socius*, 5:2378023119871580, 2019.

[162] Satish Misra. Blood pressure app study shows that top health app was highly inaccurate. https://www.imedicalapps.com/2016/03/instant-blood-pressure-app-study/, 2016. Accessed: 2023-02-09.

[163] Morgan Klaus Scheuerman, Aaron Jiang, Katta Spiel, and Jed R Brubaker. Revisiting gendered web forms: An evaluation of gender inputs with (non-) binary people. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–18, 2021.

[164] Timo Schick, Sahana Udupa, and Hinrich Schütze. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, 2021.

[165] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[166] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2023.

[167] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*, 2023.

[168] Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alex Smola. Multimodal automl on structured tables with text fields. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021.

[169] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021.

[170] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[171] Harvineet Singh, Rina Singh, Vishwali Mhasawade, and Rumi Chunara. Fairness violations and mitigation under covariate shift. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 3–13, 2021.

[172] Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. Tabular representa-

tion, noisy operators, and impacts on table structure understanding tasks in llms. *arXiv preprint arXiv:2310.10358*, 2023.

[173] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[174] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[175] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[176] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[177] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

[178] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[179] Meta Llama 3 Team. Introducing meta llama 3: The most capable openly available llm to date. Available at https://ai.meta.com/blog/meta-llama-3/ (2024/05/01), 2024.

[180] OpenAI GPT-4 Team. Gpt-4 technical report, 2024.

[181] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, pages 266–282. Springer, 2020.

[182] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[183] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[184] Guillermo E Umpierrez, Scott D Isaacs, Niloofar Bazargan, Xiangdong You, Leonard M Thaler, and Abbas E Kitabchi. Hyperglycemia: an independent marker of in-hospital mortality in patients with undiagnosed diabetes. *The Journal of Clinical Endocrinology & Metabolism*, 87(3):978–982, 2002.

[185] United States Department of Justice. Press Release: Justice Department Reaches $335 Million Settlement to Resolve Allegations of Lending Discrimination by Countrywide Financial Corporation. https://www.justice.gov/opa/pr/justice-department-reaches-335-million-settlement-resolve-allegations-lending-discrimination, 2011. Accessed: 2023-01-10.

[186] Boris van Breugel and Mihaela van der Schaar. Why tabular foundation models should be a research priority. *arXiv preprint arXiv:2405.01147*, 2024.

[187] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[188] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[189] Angelina Wang, Vikram V Ramaswamy, and Olga Russakovsky. Towards intersectionality in machine learning: Including more identities, handling underrepresentation, and performing evaluation. *arXiv preprint arXiv:2205.04610*, 2022.

[190] Ruiyu Wang, Zifeng Wang, and Jimeng Sun. Unipredict: Large language models are universal tabular predictors. *arXiv preprint arXiv:2310.03266*, 2023.

[191] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes,

and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235, 2020.

[192] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[193] Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. Two-stage llm fine-tuning with less specialization and more generalization. *arXiv preprint arXiv:2211.00635*, 2022.

[194] Michael Wick, Jean-Baptiste Tristan, et al. Unlocking fairness: a trade-off revisited. *Advances in neural information processing systems*, 32, 2019.

[195] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

[196] Lianghao Xia, Ben Kao, and Chao Huang. Opengraph: Towards open graph foundation models. *arXiv preprint arXiv:2403.01121*, 2024.

[197] Zidian Xie, Olga Nikolayeva, Jiebo Luo, and Dongmei Li. Building risk prediction models for type 2 diabetes using machine learning techniques. *Preventing Chronic Disease*, 16, 2019.

[198] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6502–6509, 2020.

[199] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

[200] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems*, 25, 2012.

[201] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan,

Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

[202] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[203] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[204] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Doro: Distributional and outlier robust optimization. In *International Conference on Machine Learning*, pages 12345–12355. PMLR, 2021.

[205] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024.

[206] Han Zhang, Xumeng Wen, Shun Zheng, Wei Xu, and Jiang Bian. Towards foundation models for learning on tabular data. *arXiv preprint arXiv:2310.07338*, 2023.

[207] Jingzhao Zhang, Aditya Krishna Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. Coping with label shift via distributionally robust optimisation. In *International Conference on Learning Representations*, 2020.

[208] Xueru Zhang, Mohammadmahdi Khaliligarekani, Cem Tekin, et al. Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness. *Advances in Neural Information Processing Systems*, 32, 2019.

[209] Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*, 2023.

Appendix A

SUBGROUP ROBUSTNESS GROWS ON TREES: AN EMPIRICAL BASELINE INVESTIGATION

A.1 Dataset Details

This section provides further detail on the datasets used in this work, along with their preprocessing.

For each dataset, we use an 80%/10%/10% train/validation/test split. The only exceptions are ACS datasets and LARC, where we use equally-sized train/test/validation splits (see below), and Adult, where we use the official train-test split.

For each dataset, we use two binary sensitive attributes derived from existing features. These attributes are primarily selected to both align with real-world sensitive attributes in practice, and, where possible, to match the implementations in previous works. For methods which use group information, each intersection of the sensitive attributes are considered a separate group (for a total of 2^2 nonoverlapping subgroups).

- **ACS Income**[1]**:** Proposed in [49], consists of approximately 160k responses to the 2018 American Community Survey. Since it is proposed as a replacement for the Adult dataset, we perform income prediction task, where the label is an indicator for whether an individuals' income exceeds the median ($\$56,000$). Sensitive attributes are race and gender. We use a larger set of features than that described in [49], which we found to improve classifier performance. The sensitive feature for race is coded as "white alone" vs. other categories, as in [49]. We use a subsample of the full 1.6M records for our experiments due to computational constraints (for example, fairness methods scale linearly or quadratically in dataset size and feature size; robustness methods also incur extra costs as data dimensionality increases).
- **ACS Public Coverage:** Derived from the same raw data source as ACS Income, described above. The task is to predict whether an individual is covered by public health insurance. Sensitive attributes

[1] https://github.com/zykls/folktables

are race and gender. We use identical feature set to [49]. The sensitive features and subsampling are handled as in ACS Income.

- **Adult**[2]**:** A widely-used benchmark dataset derived from 1994 US Census data [110]. The task is to predict whether an individuals' income exceeded $\$50,000$. Sensitive attributes are race and gender. We use the standard train-test split for Adult, following the preprocessing code of [17][3], and we further split the set partition evenly into validation and test.

- **BRFSS:** The Behavioral Risk Factors and Surveillance System[4] is a large-scale phone survey conducted annually from a random sample of adults in the United States by the Centers for Disease Control and Prevention [5]. The objective of the BRFSS is to collect uniform, state-specific data on preventive health practices and risk behaviors that are linked to chronic diseases, injuries, and preventable infectious diseases in the adult population. Respondents answer questions related to personal health, lifestyle habits, and health care coverage. BRFSS completes more than 400,000 adult interviews each year, making it the largest continuously conducted health survey system in the world. We use the BRFSS sample from 2015.

- **Communities and Crime**[6]**:** A set of features describing a community, and the prediction target is whether the community has an elevated crime rate. Following [106, 103, 104], we predict a binary label for whether the violent crime rate exceeds a threshold of 0.08. Sensitive attributes are race and income level. We use the preprocessing of [106] for this dataset, where the race feature is a binary indicator for whether the feature `racePctWhite` > 0.85 and the income level is an indicator for whether the income is above the median community income.

- **COMPAS**[7]**:** Parole records from Florida, USA, where the task is to predict whether an individual will recidivate within two years. Sensitive attributes are race and gender.

---

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/

[3] https://fairmlbook.org/code/adult.html

[4] https://www.kaggle.com/datasets/cdc/behavioral-risk-factor-surveillance-system

[5] https://www.cdc.gov/brfss/annual_data/annual_data.htm

[6] https://archive.ics.uci.edu/ml/datasets/communities+and+crime

[7] https://github.com/propublica/compas-analysis

While every labeled dataset contains human biases inherent in collecting and categorizing the data, the COMPAS dataset in particular has been the subject of valid critiques, and is mostly included for comparisons to prior work. We note that the COMPAS dataset reflects the patterns of policing and social processes in a particular community (South Florida) at a specific time, and as others have noted [194], each stage of the COMPAS dataset's creation introduces opportunities for bias [157, 15] and that measurement biases and errors with COMPAS have been widely documented [14].

- **German Credit**[8]**:** Credit application records, where the goal is to predict whether an individual has low or high credit risk. Sensitive attributes are age and gender. There are two versions of the German Credit dataset, a "numeric" version which contains binarized versions of most categorical features (with some removed), and a non-numeric version, which also contains categorical features. We do *not* use the "numeric" version of the dataset used by several other works. We found the numeric version of the dataset to be poorly-documented, lack useful features which were present in the "non-numeric" version, and contain features which actually mixed multiple variables . Using the non-numeric version, we extract separate features for sex and marital status (which are combined under a single feature in the numeric dataset).

- **LARC**[9]**:** The Learning Analytics Data Architecture (LARC) Data Set is a research-focused data set containing information about students who have attended the University of Michigan since the mid-1990s. The data includes features related to students, their enrollment, and performance, similar to the electronic records stored by many institutions of higher education. The data is divided into that which is constant throughout a student's academic career (e.g., ethnicity, SAT test scores, high school GPA, and earned degrees), that which can change from term to term (e.g., academic level, academic career, term GPA, and enrolled credits), and that which can change from class to class (e.g., subject, catalog number, earned grade, etc.). The prediction target is an indicator for whether a student will receive a grade above the median in a course; this is commonly referred to as "at-risk" grade prediction and is used to identify students at risk of struggling in a course. Sensitive attributes are "Underrepresented Minority" status (a common indicator of diversity reported by all accredited institutions in the United States) and students' self-reported sex.

---

[8] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[9] https://enrollment.umich.edu/data/learning-analytics-data-architecture-larc

It is critical to note that any notion of fairness or robustness in real-world societal contexts involves much more than even the sets of two demographic attributes considered here [189]. We also note that our treatment of these sensitive attributes as binary, while consistent with the majority of the fairness and robustness literature upon which this work is based, is necessarily reductionistic, and in practice many of these sensitive attributes are neither binary [163] nor fixed [78].

Furthermore, we urge readers to consider that, while these datasets are commonly used as benchmarks for fair or subgroup-learning, there are important social and contextual factors that must be considered for any real-world deployment of a model in these tasks to be considered "fair" [14].

A.2 Model Details

Fairness-Enhancing Models: Following [49], we evaluate one method each of pre-, in-, and postprocessing. The preprocessing method of [203] attempts to learn a transformation of the original inputs which minimally distorts the original data and its relationship to the labels, while ensuring that both group fairness (the proportion of members in a protected group receiving positive classification is identical to the proportion in the overall population) and individual fairness (similar individuals are treated similarly). The inprocessing method of [1] attempts to reduce fair classification to a cost-sensitive classification problem, where the goal is to minimize the prediction error subject to one or more fairness constraints. Finally, the postprocessing method we use is from [81], which randomizes the predictions of a fixed f_θ to satisfy equalized odds criterion. All fairness methods are implemented to simultaneously satisfy their constraints across both sensitive attributes in our datasets. We use the implementations of `aif360` [20] and `fairlearn` [24] for all fairness interventions.

None of the fairness methods encompasses a prediction model, so following [49], we pair each method (in-, pre-, and postprocessing) with a gradient-boosted tree (GBM) [64, 136]. However, unlike [49], we perform hyperparameter sweeps for both the fairness methods *and* the GBM.

Distributionally Robust Models: We draw from several classes of modern robust learning techniques. We utilize two variants of Distributionally Robust Optimization (DRO), which attempts to solve

$$\min_{\theta} \sup_{Q \in \mathcal{U}(D)} \mathbb{E}_{S \sim Q}\big[\mathcal{L}(f_\theta)\big] \tag{A.1}$$

where $\mathcal{U}$ defines an uncertainty set with respect to the training distribution D. We evaluate two widely-used formulations of the uncertainty set $\mathcal{U}$ in Equation (A.1). The first is the set of distributions with bounded likelihood ratio to D, such that (A.1) defines the conditional value at risk (CVaR). The second is the set of distributions with bounded χ^2-divergence to D. We refer to these as CVaR-DRO and χ^2-DRO, respectively. We use the efficient implementation of [113] for these methods.

We also evaluate the "Distributional and Outlier Robust Optimization" (DORO) of [204], which adds an ϵ parameter to (A.1) such that only the ϵ-smallest fraction of the data, when sorted by $\mathcal{L}$, are considered at each update step; this has the effect of ignoring outliers.

We additionally evaluate Group DRO [160], seeks to minimize the worst-group loss by solving

$$\min_{\theta} \sup_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim \mathcal{D}_g} \left[\mathcal{L}(f_\theta) \right] \tag{A.2}$$

. Finally, we evalute the Maximum Weighted Loss Discrepancy (MWLD) of [106]. This formulation adds an extra term, $\mathcal{L}_{\text{LV}} \coloneqq \text{Var}(\mathcal{L}(f_\theta(x_i) \in X)$, to the ERM objective during training. The MWLD objective is shown in [106] to be related to group fairness and robustness to subgroup shifts.

Tree-Based Models: As we note in Section 2.2.3, several modern tree-based methods achieve effectively identical performance on many tasks, with several flavors of gradient-boosted trees (GBM, LightGBM, CatBoost, XGBoost) widely being considered the state-of-the art models for tabular data. Therefore, we evaluate GBM (in order to compare directly to [49], which combines GBM with our fairness methods of interest) and LightGBM (due to its scalability, which is required for large-scale hyperparameter tuning over the datasets in this work). We also evaluate Random Forests in order to compare to a non-gradient-boosted tree-based classifier.

Baseline Supervised Learning Models: In addition to the above-described methods, we also include the following standard supervised learning methods for comparison: L_2-regularized logistic regression, and Support Vector Machines (SVM). For the SVM methods, because learning nonlinear kernels for large datasets with many features can be prohibitively expensive, we instead use two kernel approximation methods for learning: the Nystroem kernel method [53, 195] and random Fourier features [149, 150, 200]. For all baseline methods, we use the implementation of [137].

A.3 Additional Metrics

In addition to the metrics reported and defined in Section 2.3.4, we also use the following metrics in our supplementary results reported below:

Accuracy: The accuracy is defined as the fraction of labels correctly predicted at a given threshold: $\mathcal{L}_{\text{Acc}} \coloneqq \mathbb{1}(\hat{y} == y)$, where $\hat{y} = \mathbb{1}(f_\theta(x) >\geq t)$ is the predicted label of x using threshold t. We use $t = 0.5$ throughout.

Cross-Entropy: We use the standard binary cross-entropy measure, defined as $\mathcal{L}_{\text{ce}}(f_\theta; x, y) = -y \log(f_\theta(x)) - (1-y)\log(1 - f_\theta(x))$.

DORO CVaR Risk: The DORO CVaR risk is a version of CVaR risl (2.2) which excludes the ϵ-largest-loss elements in D in an effort to avoid outliers. Formally, the DORO CVaR risk is:

$$\mathcal{L}_{\text{CVaR-DORO}}(D, \mathcal{P}, \epsilon) \coloneqq \inf_{D'} \mathcal{L}_{\text{CVaR}}(D, \mathcal{P}') : \exists \tilde{\mathcal{P}}' \quad \text{s.t. } \mathcal{P} = (1-\epsilon)\mathcal{P}' + \epsilon \tilde{P}' \tag{A.3}$$

where ϵ is a hyperparameter corresponding to the fraction of outliers in the dataset. We note that (A.3) is the loss function directly optimized by the DORO-CVaR model, but it has been used as a more general evaluation of the outlier-robust tail risk of a classifier (cf. [204]).

Demographic Parity Difference: Demographic Parity (DP) is a fairness criterion that indicates the positive prediction rates across two disjoint subgroups $a, a' \in A$ are equal [17]. That is, when demographic parity is satisfied, $P(f_\theta(x_i; a_i = a) = 1) = P(f_\theta(x_j; a_j = a') = 1) \forall i, j$. The Demographic Parity Difference measures the degree to which this constraint is violated, and for nonbinary sensitive subgroups, it measures the worst-case difference:

$$\mathcal{L}_{DP-Diff} \coloneqq \max_{a,a'} \Big| P(f_\theta(x_i; a_i = a) = 1) - P(f_\theta(x_j; a_j = a') = 1) \Big| \tag{A.4}$$

Equalized Odds Difference: Equalized Odds (EO) is a fairness criterion indicating that the true positive and false positive rates are equal across two groups. The Equalized Odds Difference measures worst-case violation of Equalized Odds, across sensitive subgroups (this is $\mathcal{L}_{DISP}$ with $\mathcal{L}$ as DP). Equalized Odds

Difference can be formulated as the greater of two metrics: the true positive rate difference, and the false positive rate difference [24].

A.4 Model Performance Frontier Curves

This section describes how Model Performance Frontier curves are computed. We note that this work is not the first to use convex envelopes as a way to understand model performance; for example, [1] uses convex envelopes to understand the relationship between fairness constraint violations and model error.

To compute a model envelope in 2D, we use an algorithm to compute the convex hull, and then trace the relevant edge of the convex hull corresponding to the best-achieved performance tradeoffs under the two metrics (depending on whether these metrics are maximized, or minimized).

We provide Python code to compute these curves in the code repository associated with this paper; for completeness, we also provide the full algorithm in Algorithm 1.

A.5 Training Details

We train all models using the original optimizer, SGD, used in their original works [204, 113, 106, 160]. For all models, we train for a fixed number of epochs, but keep the weights from the best epoch based on the loss on the validation set. The number of epochs used for each dataset is as follows: ACS Income 50 epochs; Adult 300 epochs; Communities and Crime 100 epochs; COMPAS 300 epochs; German 50 epochs.

For all neural network-based models, we used a fixed batch size of 128 as in [98]; we found that varying the batch size did not affect performance but significantly increased the computational cost of hyperparameter sweeps. This also ensures that each model trained with batching sees the same number of examples during training on a given dataset.

Neural-network-based models were trained on GPU, either NVIDIA RTX 2080 Ti GPUs with 11GB of RAM, or NVIDIA Tesla M60 GPUs with 8 GB of RAM.

Algorithm 1 Model performance frontiers. This shows the computation where higher values are better for both metrics $m^{(1)}$, $m^{(2)}$.

Input: $\left(m_i^{(1)}, (m_i^{(2)}\right)_{i=i}^{|\mathcal{G}|}$ ▷ Model performance metrics $m^{(1)}, m^{(2)}$ for each configuration in grid $\mathcal{G}$

Input: ConvexHull, a method which computes the convex hull for a set of points and returns them in clockwise order.

Output: Idxs $\subseteq 1, \ldots, |\mathcal{G}|$ ▷ Indices of inputs on frontier.

vertices $\leftarrow$ ConvexHull(Input)

idxs $\leftarrow$ [ ] ▷ Initialize array of frontier points.

top_idx = $\text{argmax}_{m_+ i^{(2)}} \left(m_i^{(1)}, (m_i^{(2)}\right)_i \in$ vertices ▷ Add uppermost point to frontier

idx $\leftarrow$ top_idx

idxs $\leftarrow$ idxs + [idx]

$m_{curr}^{(1)}, m_{curr}^{(2)} \leftarrow$ vertices[idx]

next $\leftarrow$ (|vertices| + 1)mod|vertices|

$m_{next}^{(1)}, m_{next}^{(2)} \leftarrow$ vertices[next]

while $m_{next}^{(1)} < m_{curr}^{(1)}$ **do** ▷ Trace frontier counterclockwise

idxs $\leftarrow$ next

idx $\leftarrow$ (idx + 1) mod |vertices|

next $\leftarrow$ (next + 1) mod |vertices|

$m_{curr}^{(1)}, m_{curr}^{(2)} \leftarrow$ vertices[idx]

$m_{next}^{(1)}, m_{next}^{(2)} \leftarrow$ vertices[next]

end while

idx $\leftarrow$ top_idx

next $\leftarrow$ (|vertices| − 1) mod |vertices|

$m_{curr}^{(1)}, m_{curr}^{(2)} \leftarrow$ vertices[idx]

$m_{next}^{(1)}, m_{next}^{(2)} \leftarrow$ vertices[next]

while $m_{next}^{(1)} > m_{curr}^{(1)}$ **do** ▷ Trace frontier clockwise

idxs $\leftarrow$ next

idx $\leftarrow$ (idx − 1) mod |vertices|

next $\leftarrow$ (next − 1) mod |vertices|

$m_{curr}^{(1)}, m_{curr}^{(2)} \leftarrow$ vertices[idx]

$m_{next}^{(1)}, m_{next}^{(2)} \leftarrow$ vertices[next]

end while

return idxs

A.6 Hyperparameter Grids

A.6.1 Grid Definition

We detail the hyperparameter grids for each experiment in Table A.1. For each dataset, we perform a full hyperparameter grid sweep.

For methods which are built on a "base" model (i.e. Group DRO, which uses an MLP model, or Preprocessing, which is paired with GBM as in [49]), we tune the full grid of hyperparameters for the base model in addition to the hyperparameters for that method, as indicated in Table A.1. We note that this is not always the case in previous works; for example, [49] uses the default hyperparameters for GBM with fairness methods, and many DRO-based works use a fixed architecture or optimization hyperparameters for their published comparisons.

We use default parameters with the given implementation for all methods except where indicated.

Tree-based methods: For GBM and random forest, we use the implementation of [137]. For LightGBM we use the original implementation of [102][10]. For XGBoost we use the original implementation of [38][11]. For each method, we construct hyperparameter grids by beginning with large sweeps around the default hyperparameters of each method, and then pruning the sweeps to a tractable size manually by inspecting accuracy, robustness, and fairness metrics. For XGBoost, we only use training methods available with GPU-based training to ensure scalability (note that only CPU-based training was used for XGBoost models in our experiments).

Robustness-enhancing methods: For robustness-enhancing methods, our hyperparameter grids combine our large default MLP grid with the hyperparameter grids for method-specific parameters used in previous works (e.g. [204, 113]. We use the implementation of [204] for DORO and [113] for DRO methods. Note that we do not conduct sweeps for DORO models on the Adult and ACS datasets due to computational limitations, as full sweeps for both methods are prohibitively expensive to run on these large datasets.

Fairness methods: We use standard implementations and the largest-possible grids while meeting our computational constraints, as many of the fairness methods scale worse than linearly with dataset size,

[10] https://github.com/microsoft/LightGBM

[11] https://github.com/dmlc/xgboost

feature size, or both. For LFR, we fix Ax as in [203], and otherwise use the center portion of the grid from [49] which we found to be sufficient for our sweeps when also tuning the GBM parameters (which was not performed in [49]) across our datasets. Our grid for inprocessing uses the same constraints explored in [49] but also tunes the constraint slack and GBM parameters, which were not tuned in [49]. We use the implementation of [19] for LFR and postprocessing, and [24] for inprocessing.

A.6.2 Hyperparameter Sensitivity Analysis

This section provides exploratory results regarding the *sensitivity* of the models evaluated to hyperparameter configurations. For each model, we take the full set of hyperparameter configurations evaluated. then for each hyperparameter, we compute the set of values of the best-performing model (here, using worst-group accuracy) over all datasets, dropping values from continuous hyperparameter grids outside this range. This truncation step eliminates ranges of each hyperparameter which performed poorly for *all* datasets. Finally, we order the remaining hyperparameter configurations by worst-group accuracy to construct the plots below.

In the top panel of Figure A.1, we show only the DRO χ^2, XGBoost, and Group DRO models, following our running example in the main text. We include 95% Clopper-Pearson confidence intervals using the *smallest* sensitive subgroup as the sample size for the CI.[12] In the bottom panel of Figure A.1, scale the results according to the *best* performance achieved by each model class on the target dataset.

We provide similar results in the top and bottom rows of Figure A.2 which include all models; here we omit the confidence intervals due to space (although the intervals would have the same width as in Figure A.1).

A.7 Additional Results

This section contains additional experimental results not included in the main text, along with fine-grained displays of the results summarized in the main figures.

A.7.1 Training Compute Cost

In Section 2.6, we discuss hyperparameter sensitivity and training time of the various algrorithms present in our study. Here, we provide estimations of the result of conducting our hyperparameter sweeps on modern

[12]This makes the confidence intervals in A.1 conservative, as they would be narrower when the worst group is not the smallest group; we do this so that the interval width is consistent for all model configurations.

Model	Grid Size	Hyperparameter	Values
Baseline Methods			
♣ MLP	405	Learning Rate	$\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$
		Weight Decay	$\{0, 0.1, 1\}$
		Num. Layers	$\{1, 2, 3\}$
		Hidden Units	$\{64, 128, 256\}$
		Momentum	$\{0., 0.1, 0.9\}$
		Batch Size	$\{128\}$
SVM	576	C	$\{0.01, 0.1, 1., 10., 100., 1000.\}$
		Kernel Appx.	$\{\text{Nystroem}, \text{RKS}\}$
		Loss	Squared Hinge
		γ	$\{0.5, 1.0, 2.0\}$
		Num. Components	$\{64, 128, 256, 512\}$
		Nystroem Kernel Degree	$\{2, 3\}$
		Nystroem Kernel	$\{\text{RBF}, \text{poly}\}$
Logistic Regression	8	L_2 penalty	$\{0.001, 0.01, 0.1, 1., 10., 100., 1000., 10000.\}$
Tree-Based Methods			
♢ GBM	100	Learning Rate	$\{0.01, 0.1, 0.5, 1.0, 2.0\}$
		Num. Estimators	$\{64, 128, 256, 512, 1024\}$
		Max Depth	$\{2, 4, 8, 16\}$
		Min. Samples Split	2
		Min. Samples Leaf	1
Random Forest	640	Num. Estimators	$\{64, 128, 256, 512\}$
		Max Features	$\{\text{sqrt}, \text{log2}\}$
		Min. Samples Split	$\{2, 4, 8, 16\}$
		Min. Samples Leaf	$\{1, 2, 4, 8, 16\}$
		Cost-Complexity α	$\{0., 0.001, 0.01, 0.1\}$
XGBoost	1944	Learning Rate	$\{0.1, 0.3, 1.0, 2.0\}$
		Min. Split Loss	$\{0, 0.1, 0.5\}$
		Max. Depth	$\{4, 6, 8\}$
		Column Subsample Ratio (tree)	$\{0.7, 0.9, 1\}$
		Column Subsample Ratio (level)	$\{0.7, 0.9, 1\}$
		Max. Bins	$\{128, 256, 512\}$
		Growth Policy	$\{\text{Depthwise}, \text{Loss Guide}\}$
LightGBM	12544	Learning Rate	$\{0.01, 0.1, 0.5, 1.\}$
		Num. Estimators	$\{64, 128, 256, 512\}$
		L_2-reg.	$\{0., 0.00001, 0.0001, 0.001, 0.01, 0.1, 1.\}$
		Min. Child Samples	$\{1, 2, 4, 8, 16, 32, 64\}$

cloud-based computing platforms, in order to estimate the costs of individual training runs, and the full hyperparameter sweeps, in our study.

Figure A.12 displays the estimated median cost of a single training run of each model in (DRO χ^2, XGBoost, Group DRO, LightGBM). We note that while XGBoost and LightGBM are trained on CPU in this study (although GPU implementations of each training algorithm are available), training of the DRO models at scale is only feasible on GPU.

For our cost estimation, we use prices of \$7.20 per compute-hour for GPU and \$3.072 per compute-hour for CPU, which reflect the hourly price of cloud-based GPU and CPU hardware used in this study.

Figure A.12 shows that, compared to DRO methods, tree-based methods (XGBoost, LightGBM) achieve comparable cost or considerable savings for all datasets in our study. The lone exception is XGBoost on the German Credit dataset, which we hypothesize is due to potential overloading of the CPU cluster during these training experiments (we note that German is, by far, the smallest dataset in our study with $n = 1000$, and the XGBoost algorithm generally performs well on small datasets).

Collectively, these results, combined with the demonstration that tree-based models also require fewer iterations to tune due to their decreased hyperparameter sensitivity (see Section 2.6), suggest that tree-based models are a considerably more resource-efficient way to achieve state-of-the-art subgroup robustness for tabular data classification.

A.7.2 Peak Performance Summary

We provide the best performance per model, in terms of both overall accuracy and worst-group accuracy, in Tables A.2 and A.3, respectively.

A.7.3 Performance of Default Tree Hyperparameters

For the tree-based models in our summary, we give the performance of the default hyperparameters in Table A.4.

	Income	**Pub. Cov.**	**Adult**	**BRFSS**	**C&C**	**COMPAS**	**German**	**LARC**
DORO CVaR	0.816	N/A	0.852	N/A	0.905	0.725	0.86	N/A
DORO χ^2	0.813	N/A	0.851	N/A	N/A	0.743	N/A	N/A
DRO CVaR	0.677	0.719	0.778	0.898	0.879	0.627	0.83	0.644
DRO χ^2	0.694	0.742	0.774	0.896	0.869	0.691	0.82	0.667
GBM	0.824	0.786	0.873	0.896	0.854	0.712	0.82	0.747
Group DRO	0.816	0.77	0.85	0.897	0.894	0.702	0.82	0.729
Inprocessing + GBM	0.823	0.789	0.87	0.898	0.879	0.732	0.87	0.75
L2LR	0.815	0.768	0.852	0.895	0.844	0.7	0.82	0.739
LightGBM	0.827	0.791	0.874	0.901	0.91	0.734	0.9	0.751
MLP	0.821	0.782	0.857	0.897	0.879	0.724	0.82	0.743
MWLD	0.823	0.784	0.864	0.898	0.894	0.739	0.85	0.744
Marginal DRO	N/A	N/A	N/A	N/A	0.849	0.72	0.82	N/A
Postprocessing + GBM	0.775	0.772	0.842	0.897	0.749	0.689	0.85	0.714
Preprocesing + GBM	0.771	0.765	0.827	0.897	0.879	0.724	0.86	0.678
Random forest	0.82	0.786	0.865	0.897	0.905	0.728	0.85	0.746
SVM	0.821	0.785	0.857	0.898	0.874	0.723	0.83	0.74
XGBoost	0.824	0.79	0.875	0.899	0.894	0.725	0.88	0.748

Table A.2: Best observed overall accuracy per model, by dataset.

	Income	**Pub. Cov.**	**Adult**	**BRFSS**	**C&C**	**COMPAS**	**German**	**LARC**
DORO CVaR	0.788	N/A	0.81	N/A	0.836	0.71	0.8	N/A
DORO χ^2	0.783	N/A	0.808	N/A	N/A	0.718	N/A	N/A
DRO CVaR	0.582	0.672	0.712	0.857	0.803	0.606	0.762	0.492
DRO χ^2	0.641	0.683	0.707	0.863	0.789	0.676	0.789	0.61
GBM	0.796	0.738	0.833	0.851	0.787	0.684	0.737	0.734
Group DRO	0.783	0.718	0.807	0.855	0.816	0.682	0.789	0.702
Inprocessing + GBM	0.796	0.742	0.845	0.859	0.803	0.702	0.842	0.735
L2LR	0.785	0.718	0.808	0.849	0.754	0.644	0.727	0.72
LightGBM	0.798	0.745	0.837	0.87	0.836	0.718	0.842	0.739
MLP	0.791	0.738	0.814	0.854	0.787	0.715	0.75	0.721
MWLD	0.794	0.739	0.826	0.859	0.82	0.729	0.774	0.728
Marginal DRO	N/A	N/A	N/A	N/A	0.77	0.707	0.774	N/A
Postprocessing + GBM	0.72	0.723	0.809	0.857	0.639	0.623	0.75	0.632
Preprocesing + GBM	0.737	0.717	0.781	0.861	0.803	0.702	0.816	0.647
Random forest	0.79	0.741	0.824	0.858	0.836	0.702	0.8	0.726
SVM	0.791	0.735	0.815	0.858	0.763	0.707	0.789	0.719
XGBoost	0.796	0.744	0.836	0.868	0.836	0.711	0.818	0.736

Table A.3: Best observed worst-group accuracy per model, by dataset.

	Income	Pub. Cov.	Adult	BRFSS	C&C	COMPAS	German	LARC
Overall Accuracy								
GBM	0.81	0.777	0.871	0.892	0.859	0.7	0.79	0.739
LightGBM	0.823	0.787	0.874	0.887	0.819	0.682	0.75	0.747
Random Forest	0.812	0.765	0.852	0.891	0.854	0.669	0.74	0.754
XGBoost	0.825	0.788	0.872	0.893	0.854	0.698	0.73	0.748
Worst-Group Accuracy								
GBM	0.779	0.725	0.83	0.86	0.711	0.684	0.65	0.718
LightGBM	0.793	0.738	0.834	0.834	0.658	0.667	0.6	0.725
Random Forest	0.781	0.713	0.808	0.848	0.738	0.596	0.71	0.742
XGBoost	0.797	0.735	0.834	0.833	0.754	0.654	0.55	0.728

Table A.4: Performance of default hyperparameters for tree-based models.

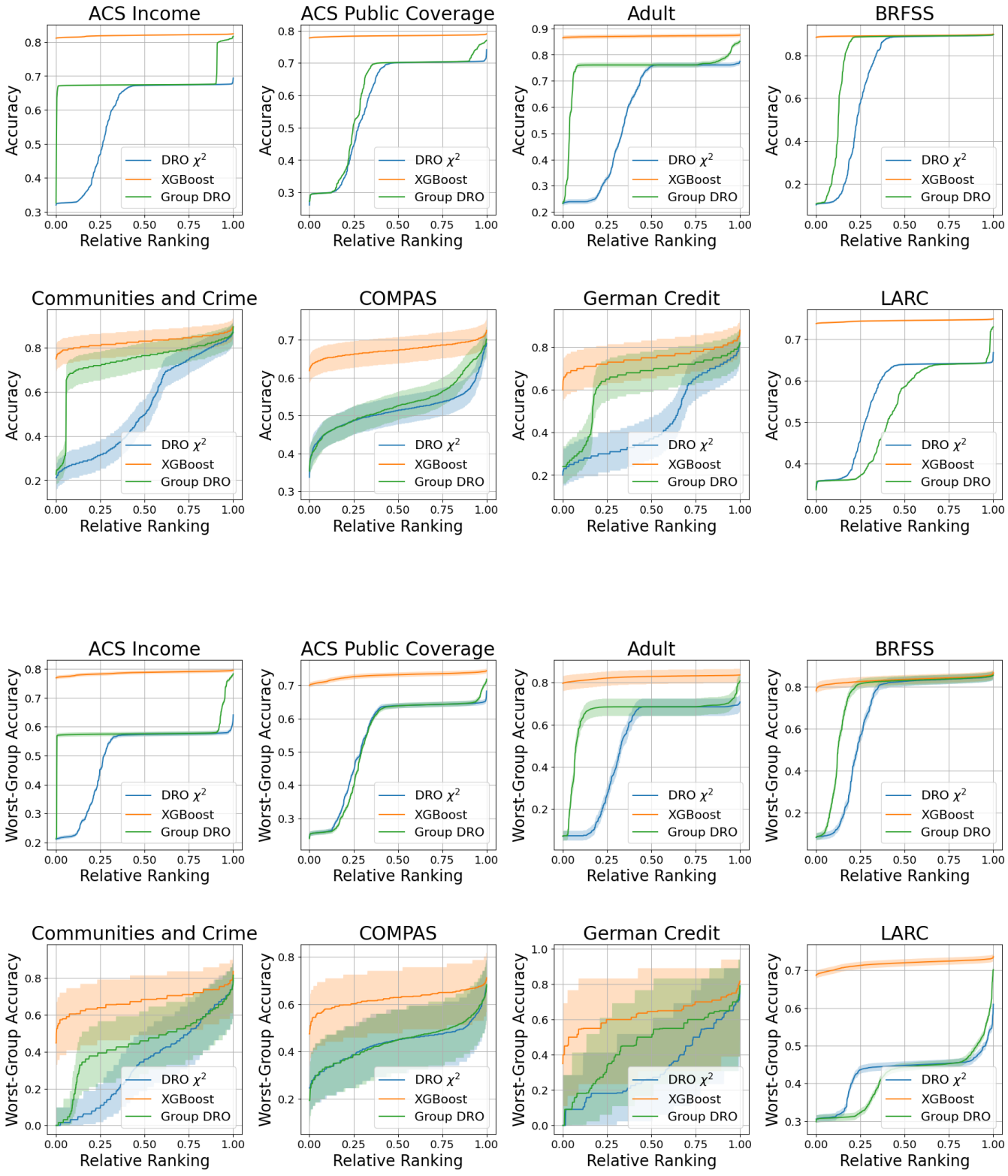


Figure A.1: Hyperparameter sensitivity plots for χ^2 DRO, Group DRO, and XGBoost models. The top 8 panels show Accuracy; the lower 8 panels show worst-group accuracy (this is identical to Figure 2.7, reproduced here for clarity). XGBoost shows considerably lower sensitivity to hyperparameter tuning.

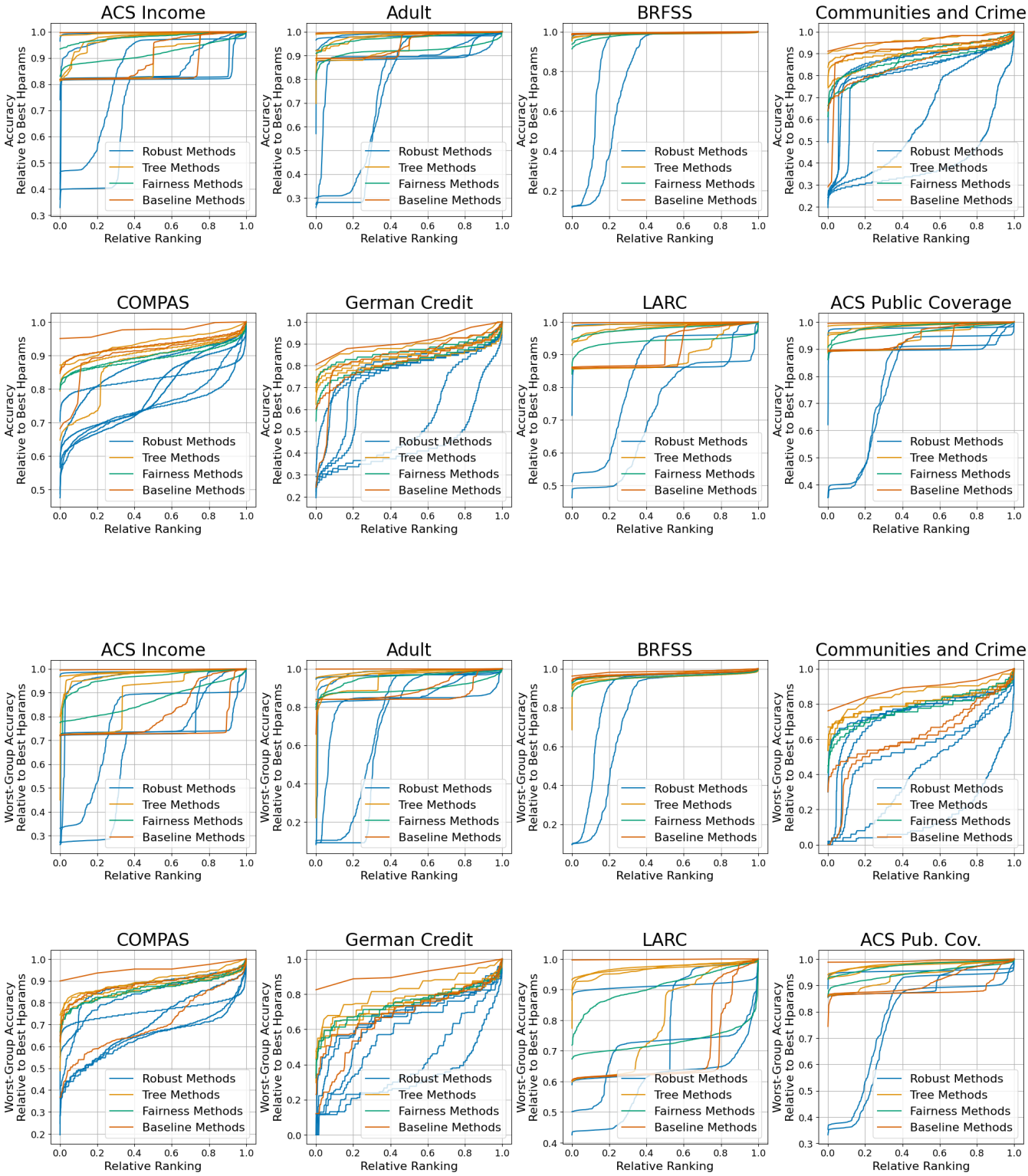


Figure A.2: Hyperparameter sensitivity plots for all models evaluated. (Clopper-Pearson CIs omitted due to space).

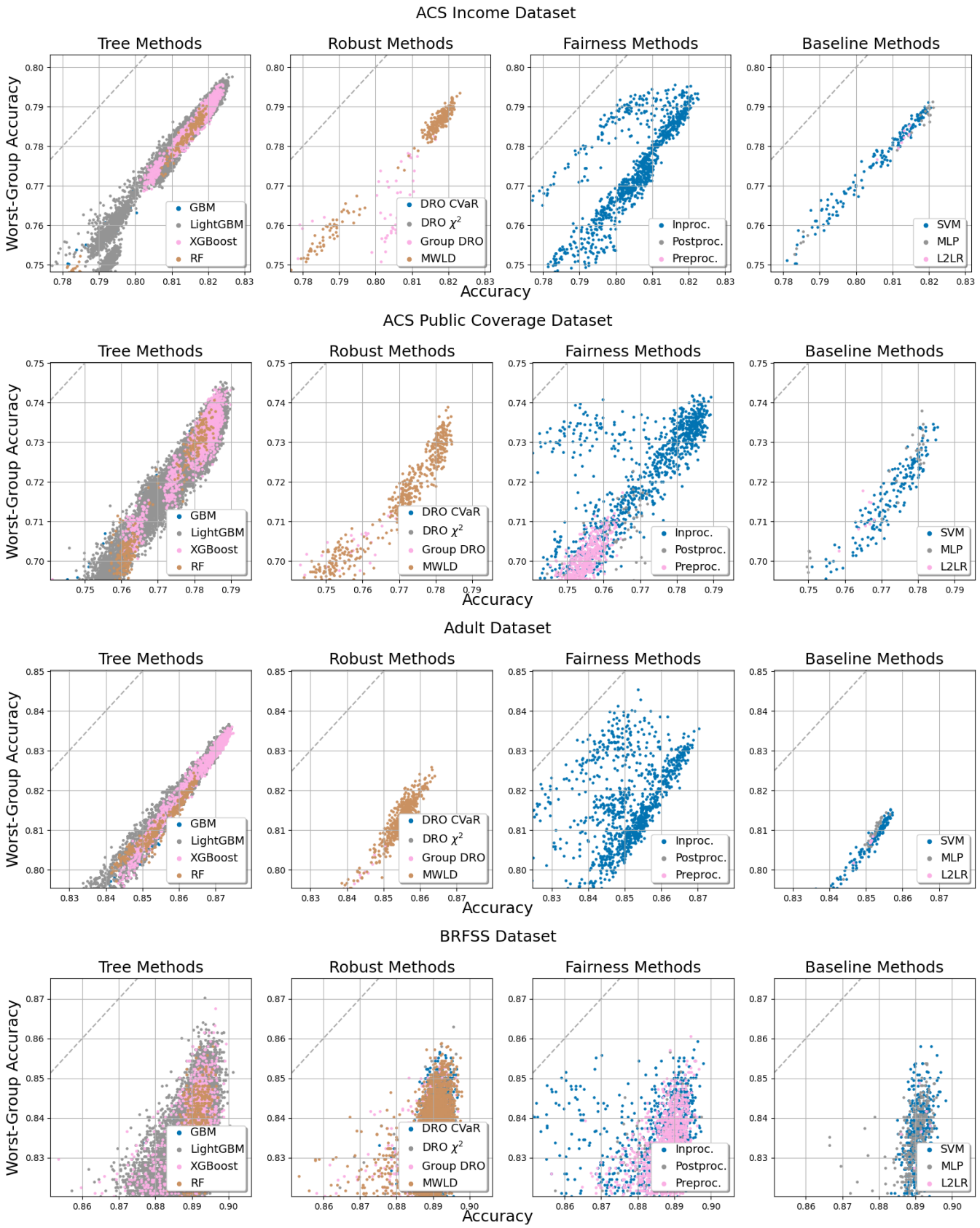


Figure A.3: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and

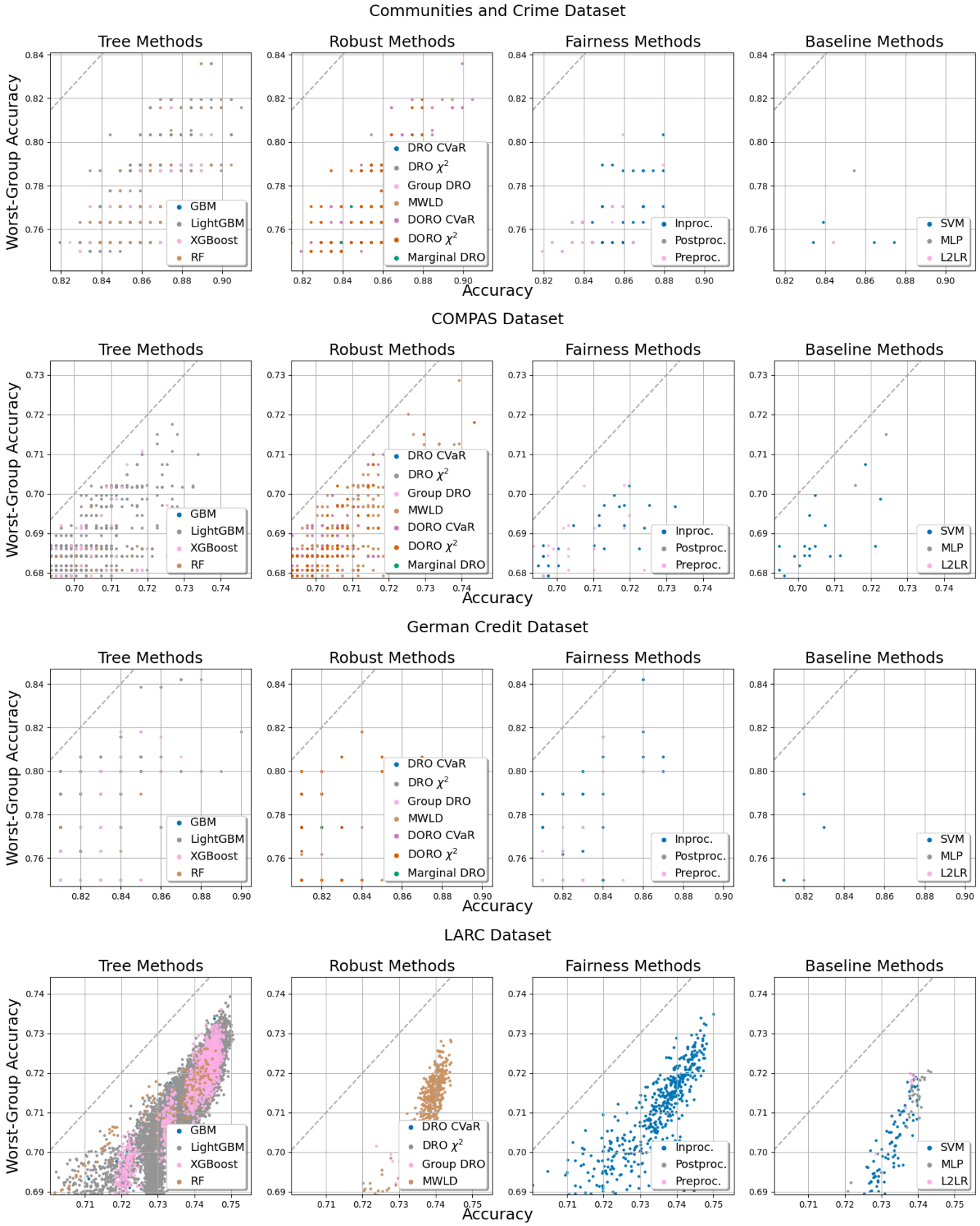


Figure A.4: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and

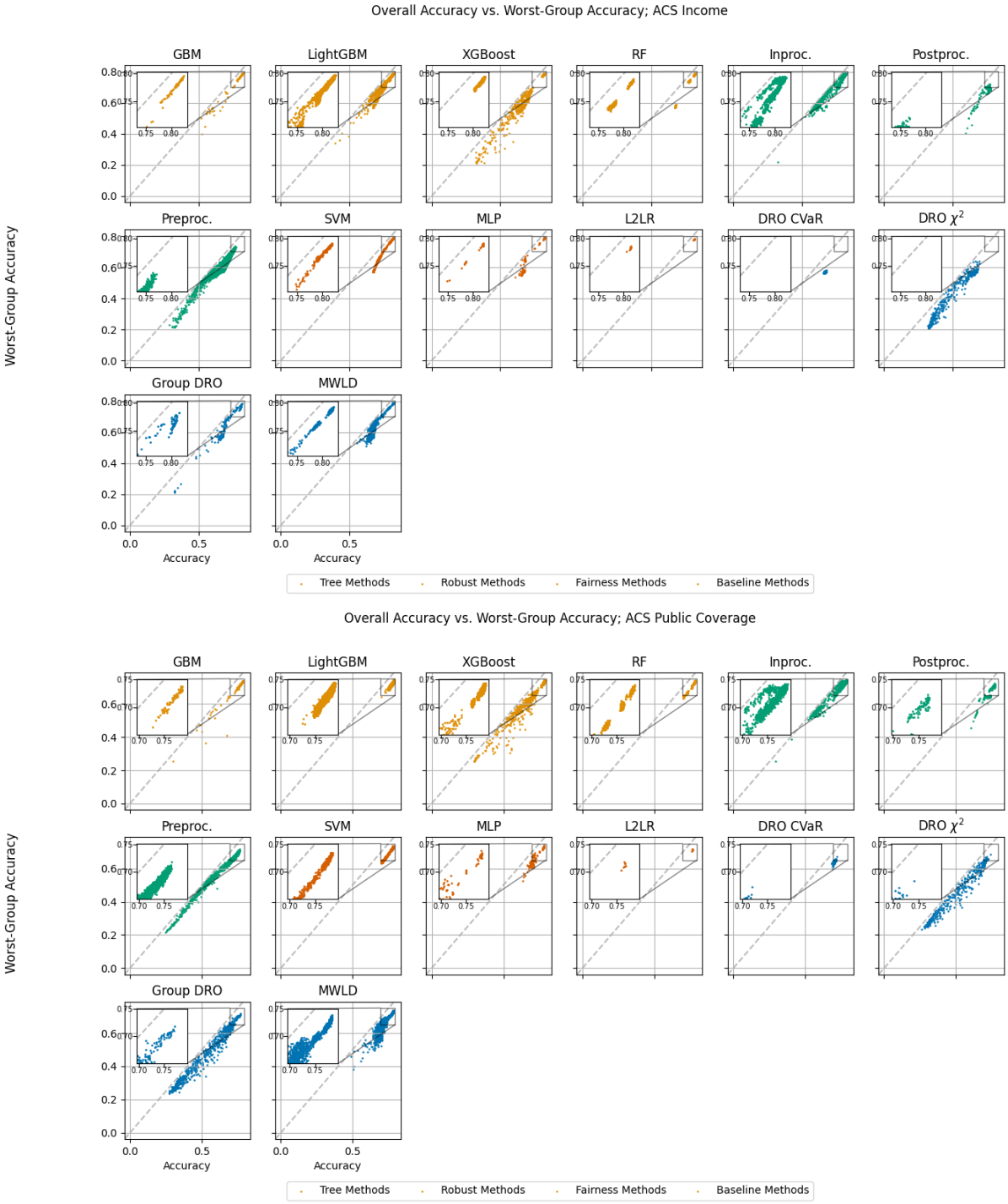


Figure A.5: Detailed accuracy vs. worst-group accuracy for ACS Income and ACS Public Coverage datasets.

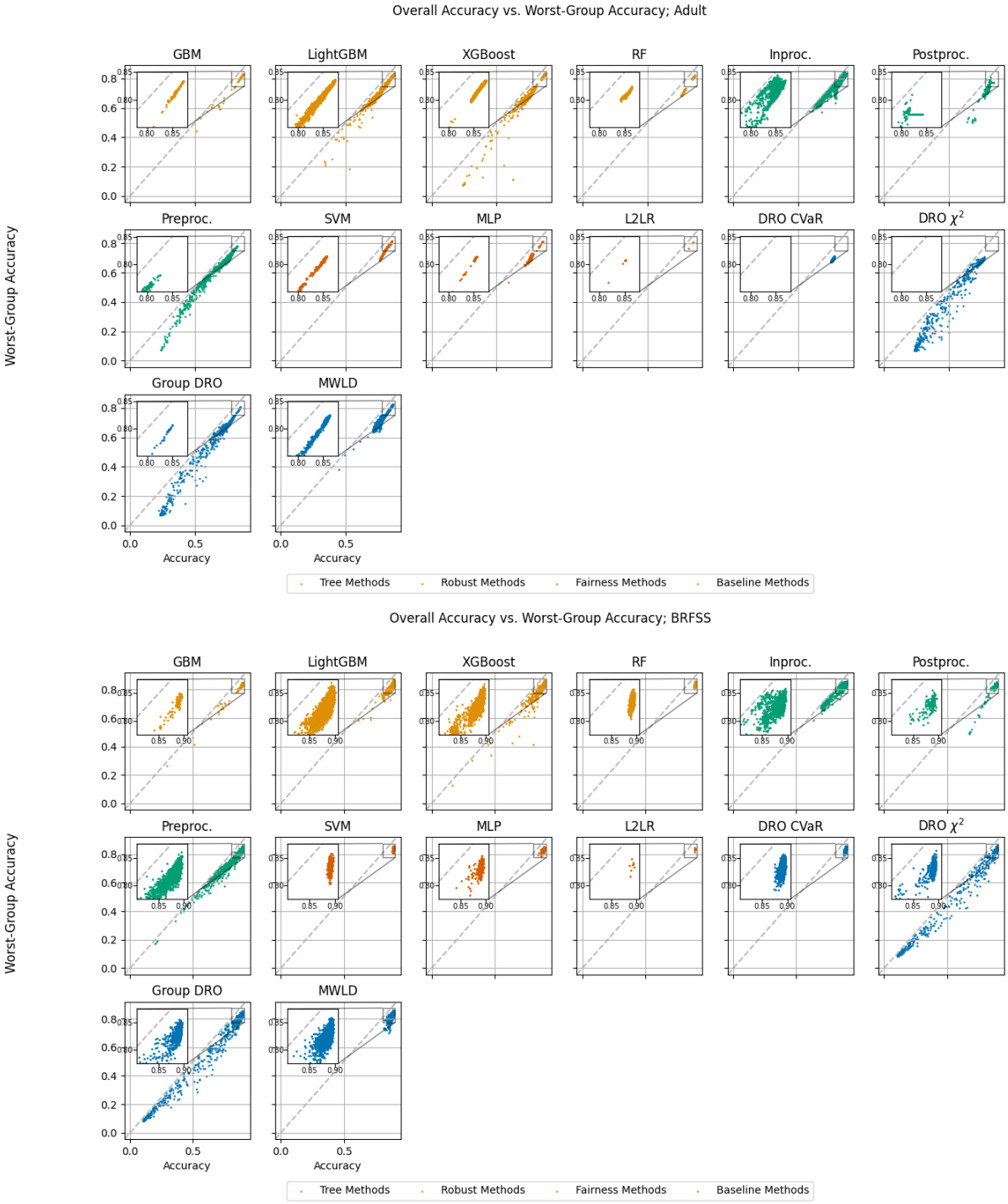


Figure A.6: Detailed accuracy vs. worst-group accuracy for Adult and BRFSS datasets.

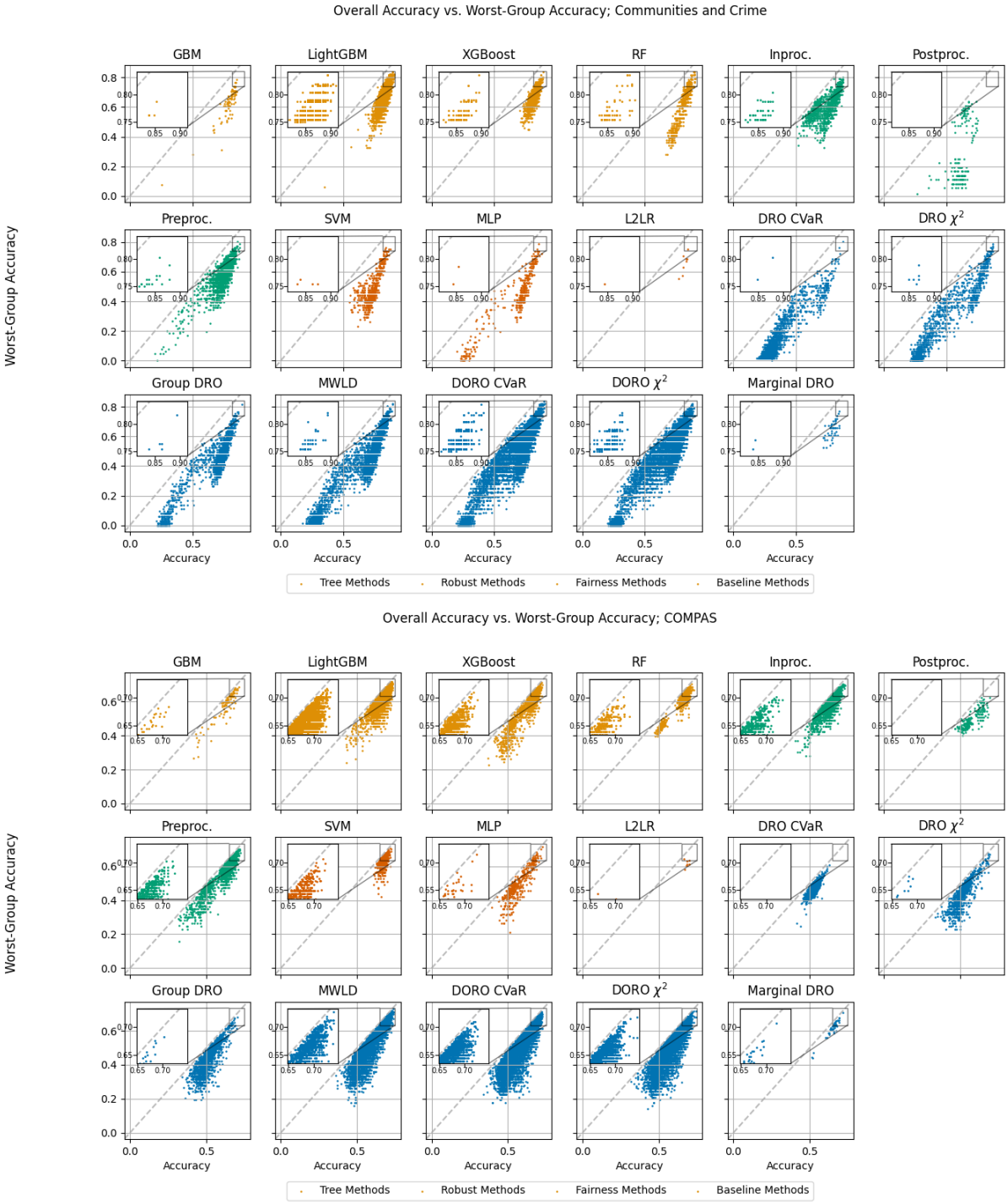


Figure A.7: Detailed accuracy vs. worst-group accuracy for Communities and Crime and COMPAS datasets.

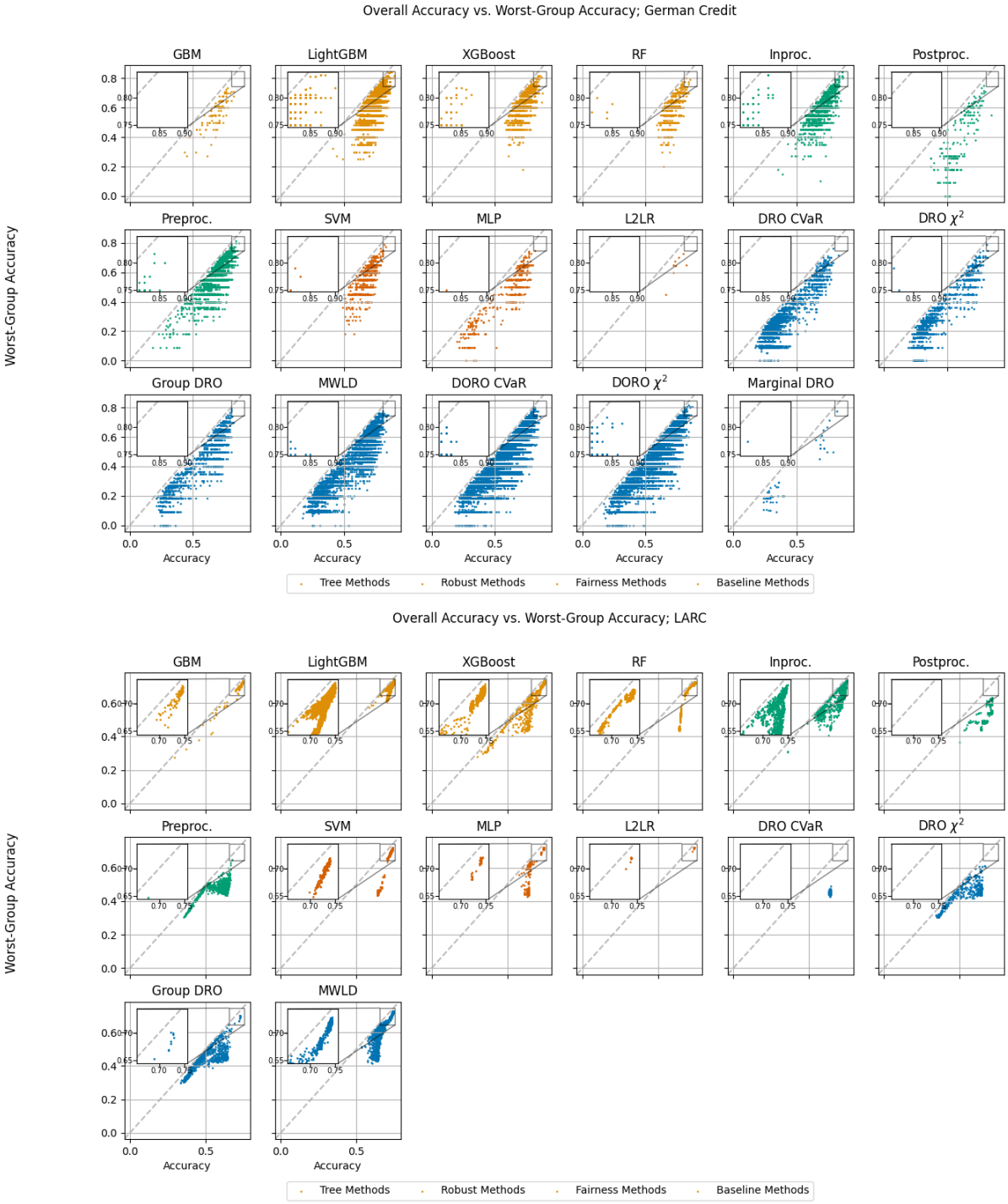


Figure A.8: Detailed accuracy vs. worst-group accuracy for German and LARC datasets.

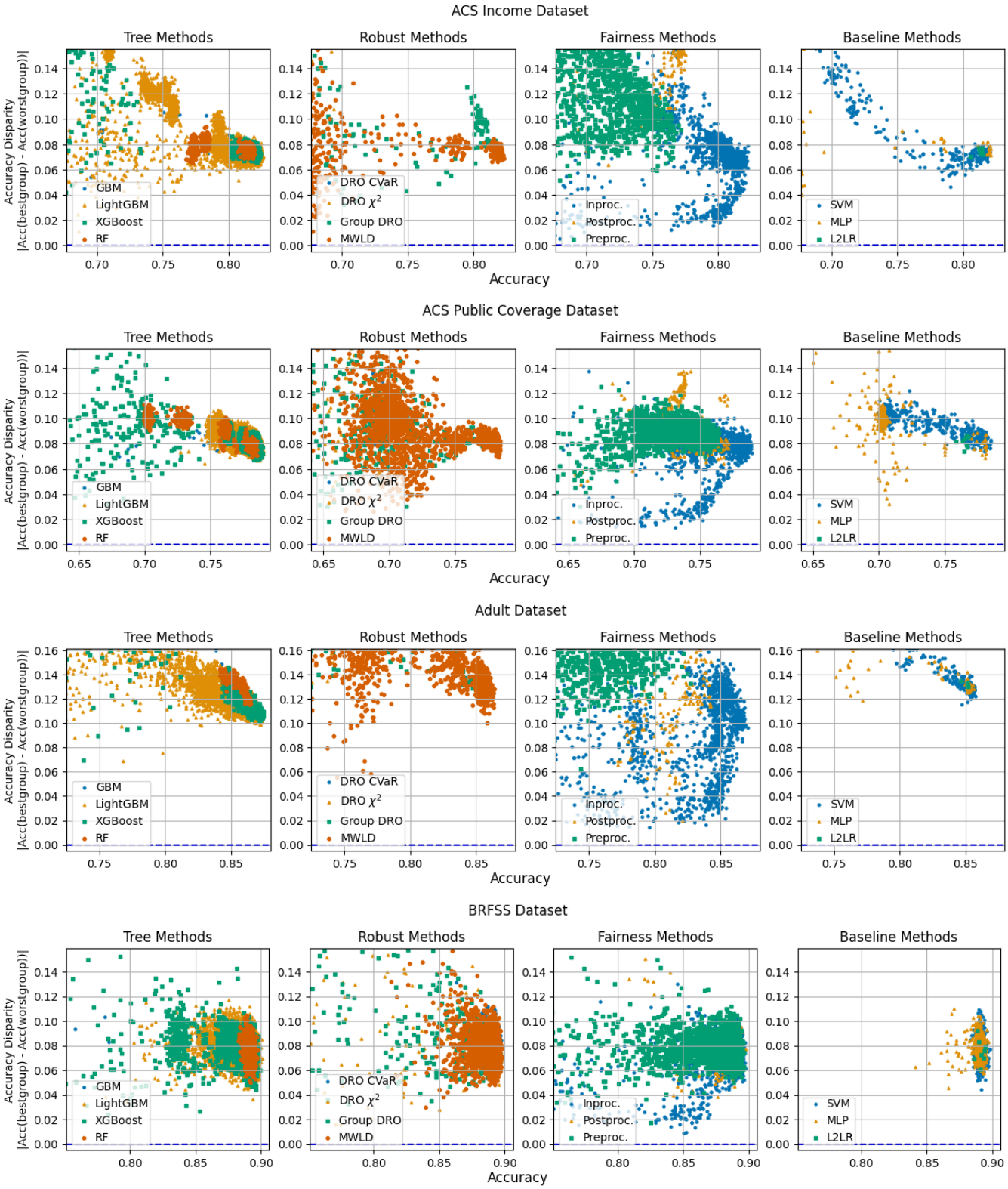


Figure A.9: Overall Accuracy vs. Accuracy Disparity of robust, fairness-enhancing, tree-based, and baseline models over datasets ACS Income, ACS Public Coverage, Adult, BRFSS. See also Figure A.10.

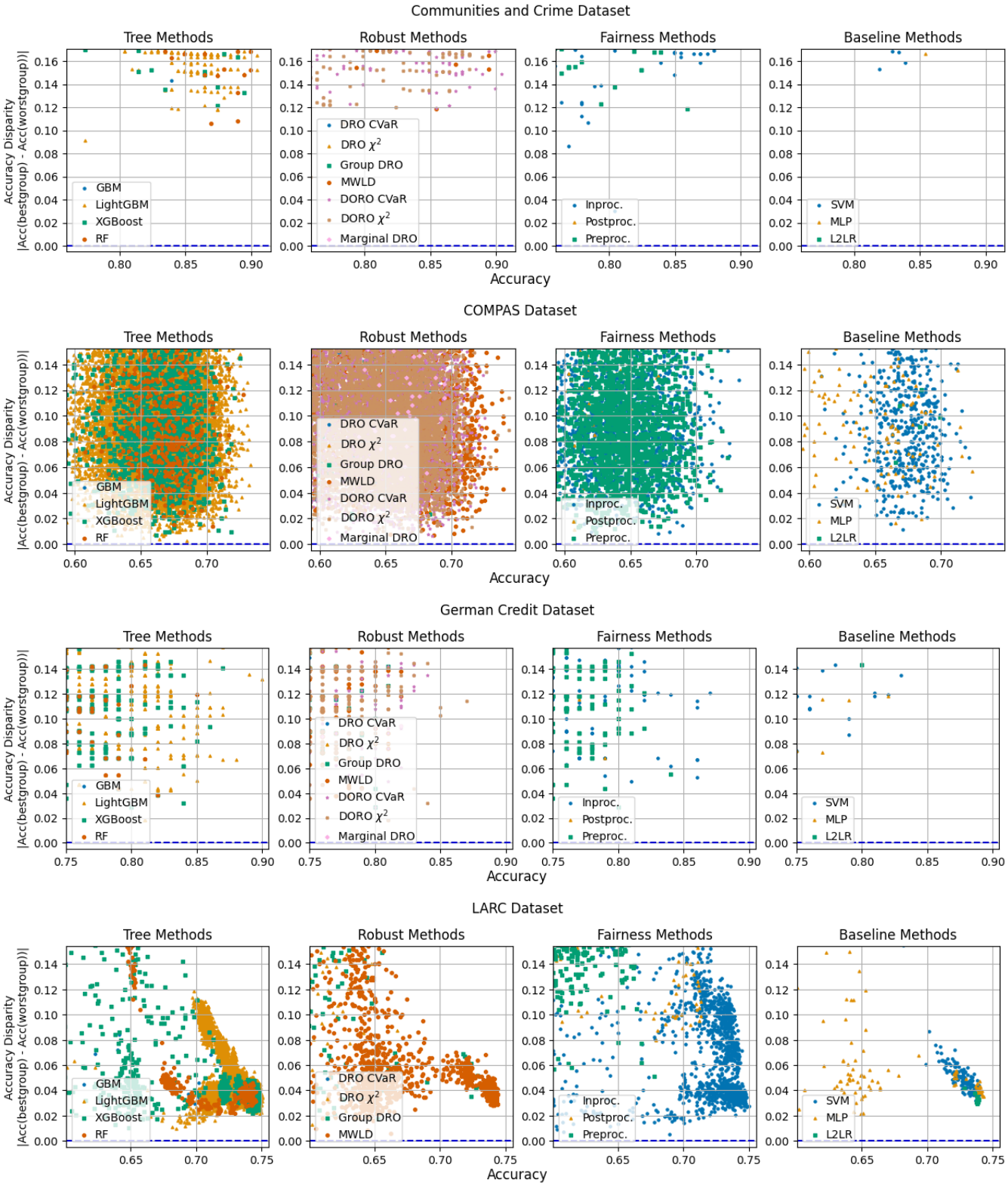


Figure A.10: Overall Accuracy vs. Accuracy Disparity of robust, fairness-enhancing, tree-based, and baseline models over datasets Communities and Crime, COMPAS, German, LARC. See also Figure A.9.

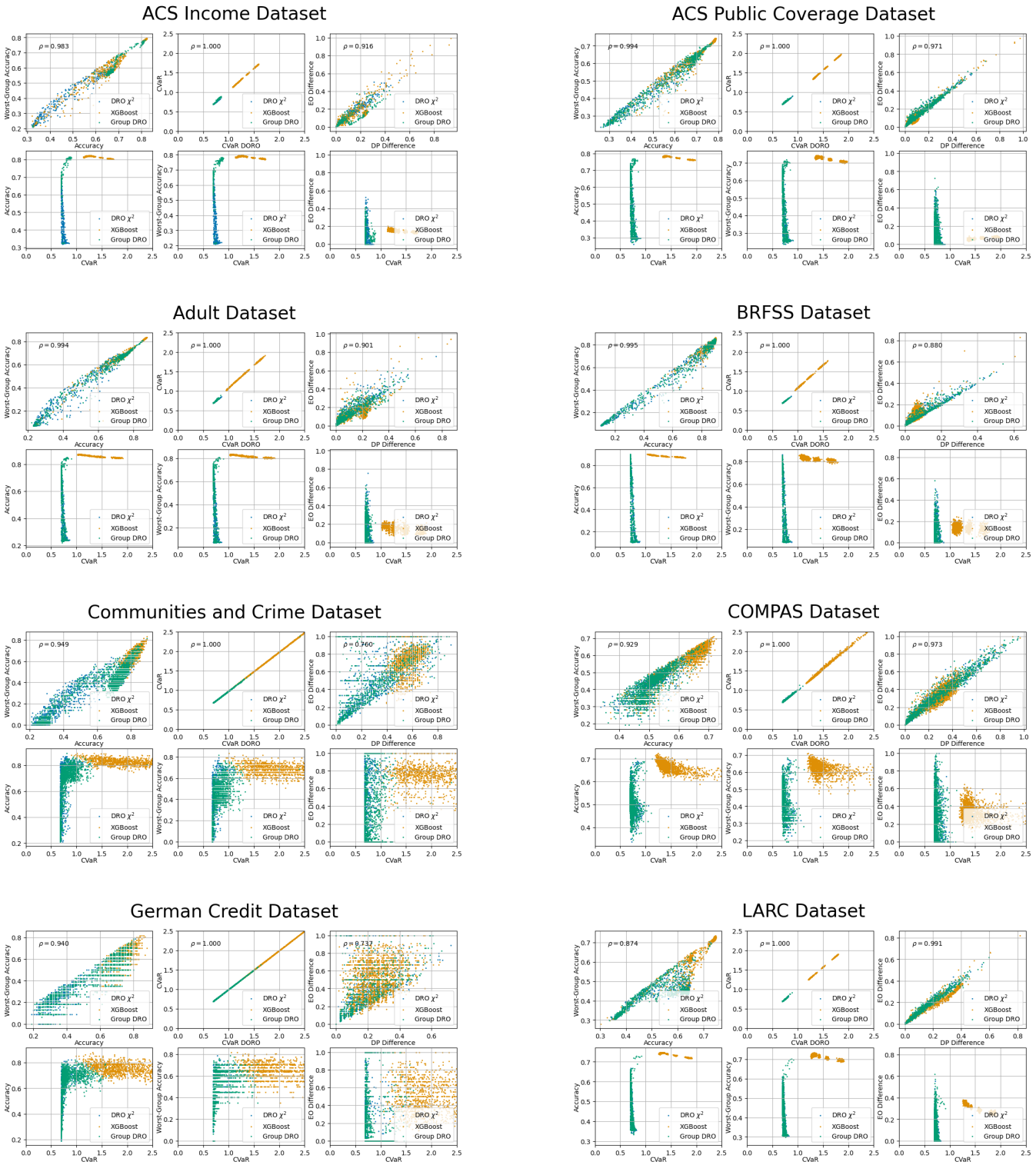


Figure A.11: Correlation between complementary metrics (top row) and non-complementary metrics (bottom row) for each dataset, for DORO, XGBoost, and Group DRO models. Complementary metrics show strong correlations for all models, while non-complementary metrics do not. Pearson's r correlation coefficient for each pair of complementary metrics shown in the upper-left of each plot.

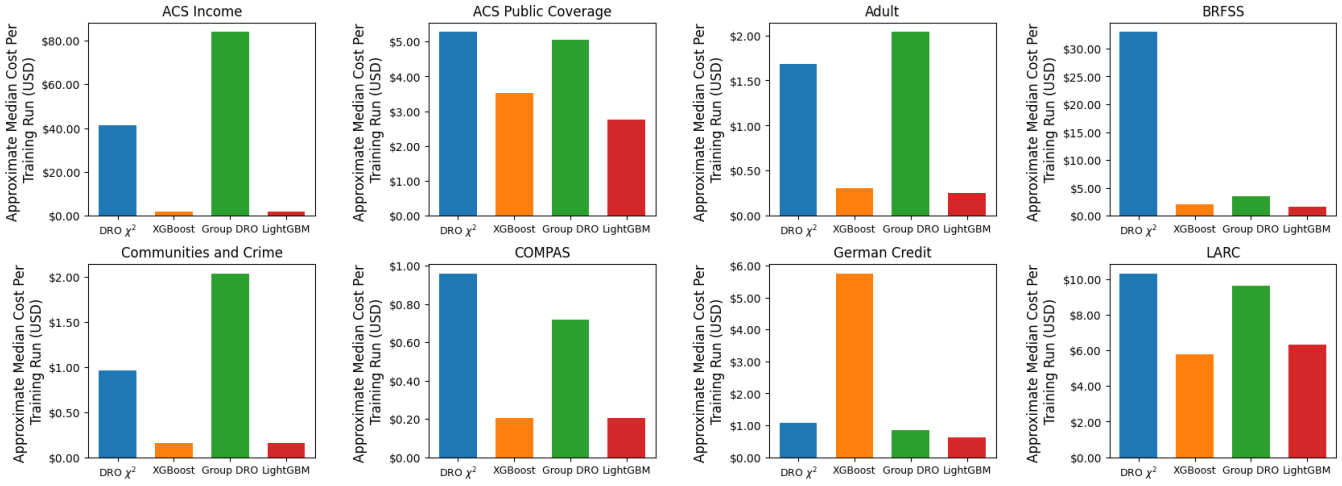


Figure A.12: Estimated cost per training run, based on the median train time over the iterations in our study and the price of cloud-based computing infrastructure.

Appendix B

BENCHMARKING DISTRIBUTION SHIFT IN TABULAR DATA WITH TABLESHIFT

B.1 Acknowledgements

This work was supported by a Microsoft Grant for Customer Experience Innovation. This work was also in part supported by the NSF AI Institute for Foundations of Machine Learning (IFML, CCF-2019844), Google, Open Philanthropy, and the Allen Institute for AI.

Our research utilized computational resources and services provided by the Hyak computing cluster at the University of Washington.

B.2 Benchmark Task Details

This section provides details on each of the benchmark tasks in TableShift. While we describe the data source for each task, we emphasize that *TableShift does not host or distribute the data*; each data source is publicly available (some require training or authorization, but all are available to the public). Based on our review of the datasets, we believe that the datasets do not contain personally identifiable information, offensive content, or proprietary information. For data collected from human subjects, the conditions of collection and the ethics approval under which the data were collected are described in the documentation associated with each dataset.

B.2.1 Food Stamps

Background: Food insecurity is a problem affecting more than 10% of households (13.5 million) across the United States in 2021[1]. Various programs exist to provide families and individuals with supplemental income to reduce food insecurity. However, diminished social support services in many U.S. states limit

[1] https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-u-s/key-statistics-graphics/

the ability of outreach providers to ensure all aligible individuals are receiving available benefits. Low-cost, low-friction screening tools powered by machine learning models might provide useful information whether an individual is receiving food stamps in order to identify lilely candidates for both food security programs ("food stamps") and as a proxy for eligibility and need for additional support services.

Data Source: We use person-level data from the American Community Survey (ACS)[2]. We filter the data for low-income adults aged 18-62 (i.e. selecting only adults below the social security eligibility age) in households with at least one child in the household. We use an income threshold of $30000 based on the U.S. poverty threshold for a family with one child.

Distribution Shift: In the United States, food stamps programs are managed at the state level. We apply domain shift over states, at the *regional* level. Specifically, we use the ACS census region as the split. The ACS includes 10 regions, which are: Puerto Rico; New England (Northeast region); Middle Atlantic (Northeast region); East North Central (Midwest region); West North Central (Midwest region); South Atlantic (South region); East South Central (South region); West South Central (South Region); Mountain (West region); Pacific (West region). We use East South Central (South region) as the holdout domain for this task.

This split parallels the case where a system is trained on a subset of states in a specific geographic area (perhaps in a localized study that draws participants or respondents from some geographic areas, but excludes other areas), and then applied to another. It also parallels the case where there is an interest in simulating the effect of a policy change. Finally, it mirrors the challenge of predicting an effect of a policy outcome (food stamps eligibility/recipiency) where differences in the underlying policy (different programs or eligibility across states) are a confounder.

B.2.2 Income

Background: Income is a widely-used measure of social stability. In addition, income is often used as a criteria for various social support programs. For example, in the United States, income is used to measure poverty, and can be used determine eligibility for various social services such as food stamps and medicaid. Income prediction has obvious commercial utility. Finally, income prediction has a rich and unique history

[2] https://www.census.gov/programs-surveys/acs/about.html

in the machine learning community, dating back to the "adult income" census dataset [109, 49].

Data Source: We use person-level data from the American Community Survey (ACS), as described in Task B.2.1. However, for the income prediction task, we use different filtering. We use the filtering described in [49], which filters for adults aged at least 16 years old, who report working more than zero hours in the past month with reported income at least \$100.00. We use an income threshold of $\$56,000$, which is the median income, as in [68].

Distribution Shift: Income patterns can vary in many ways. Here, we focus on domain shift at the *regional* level. We use the same splitting variable (US Census Region) described in Task B.2.1. However, for the income prediction task, we use New England (Northeast region) as the held-out domain.

B.2.3 Public Coverage

Background: People use health-care services to diagnose, cure, or treat disease or injury; to improve or maintain function; or to obtain information about their health status and prognosis [133]. In the United States, health insurance is a critical component of individuals' ability to access health care. Public health insurance exists, among other reasons, to provide affordable and accessible health insurance options for individuals not willing or able to purchase insurance through the private insurance market. However, not all individuals have health insurance; only 88% of individuals in the U.S had health insurance in 2019 according to the National Health Interview Survey (NHIS). Increasing the proportion of people in the United States with health insurance is one of the four healthcare objectives of the U.S. Department of Health and Human Services "Healthy People 2030" initiative[3]. In this task, the goal is to predict whether an individual is covered by public health insurance.

Data Source: We use person-level data from the American Community Survey (ACS), as described in Task B.2.1. However, for this task, we filter the data to include only low-income individuals (those with income less than $\$30,000$) who are below the age of 65 (at which age all persons in the United States are covered by Medicare). This is the same filtering used in [49, 68].

Distribution Shift: Many factors can influence individuals' ability to access or utilize health insurance

[3] https://health.gov/healthypeople/objectives-and-data/browse-objectives/health-care-access-and-quality/increase-proportion-people-health-insurance-ahs-01

and healthcare services. These include spoken language skills, mobility (whether an individual has recently relocated), education, ease of obtaining services, and discriminatory practices among providers [133]. We focus on *disability status*, as this is a widely-known factor in obtaining access to adequate health care [133]. Disability is also a particularly realistic factor in that disability status is likely to contribute to nonresponse to certain forms of data collection for many tabular data sources (including the four methods used to collect the ACS data: internet, mail, telephone, and in-person interviews) that can disadvantage persons with certain disabilities and decrease likelihood of participation or cause them to be excluded from study population.

For this task, the holdout domain D^{test} consists of persons with disabilities; the training domain D^{train} consists of persons who do not have disabilities. This simulates a situation where data collection practices excluded disabled persons, potentially through the factors described above.

B.2.4 ACS Unemployment

Background: Unemployment is a key macroeconomic indicator and a measure of individual well-being. Unemployment is also linked to a variety of adverse outcomes, including socioeconomic, psychological, and health impacts [13, 33, 26, 128].

Data Source: We use person-level data from the American Community Survey (ACS), as described in Task B.2.1. However, for this task, we filter the data to include only individuals over the age of 18 and below the age of 62 (at which age persons in the United States are eligible to receive Social Security income).

Distribution Shift: Many factors are known to be related to unemployment. We focus on a form of subpopulation shift, and use *education level* as the domain split. We use individuals with educational attainment of GED (high school diploma equivalent) or higher as the training population D^{train}, and individuals without high school-level education as D^{test}. This simulates a survey collection with a biased sample that systematically excludes such persons.

B.2.5 Diabetes

Background: Diabetes is a chronic disease that affects at least 37.7million people in the United States (11.3% of the U.S population); it is estimated that an additional 96 million adults have prediabetes.[4] Diabetes increases the risk of a variety of other health conditions, including stroke, kidney failure, renal complications, peripheral vascular disease, heart disease, and death. The economic cost of diabetes is also significant: The total estimated cost of diagnosed diabetes in 2017 is $327 billion [10]. Care for people with diagnosed diabetes accounts for 1 in 4 health care dollars in the U.S. – more than half of that expenditure is directly attributable to diabetes [10].

Early detection of diabetes thus stands to have a significant impact, allowing for clinical intervention and potentially reducing the prevalence of diabetes. Further, even prediabetes is ackowledged to have significant impacts both on health outcomes and quality of life [10], and early detection if high diabetes risk could serve to identify prediabetic individuals. There exists a considerable prior literature on models for early diabetes prediction, e.g. [197, 132, 89]

Data Source: We use data provided by the Behavioral Risk Factors Surveillance System (BRFSS)[5]. BRFSS is a large-scale telephone survey conducted by the Centers of Disease Control and Prevention. BRFSS collects data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services. BRFSS collects data in all 50 states as well as the District of Columbia and three U.S. territories. BRFSS completes more than 400,000 adult interviews each year, making it the largest continuously conducted health survey system in the world. BRFSS annual survey data from 2017-2021 is currently available from the CDC.

The BRFSS is composed of three components: 'fixed core' questions, asked every year, 'rotating core', asked every other year, and 'emerging core'. Since some of our features come from the rotating core, we only use every-other-year data sources; otherwise many features would be empty for the intervening years.

For the Diabetes prediction task, we use a set of features related to several known indicators for diabetes derived from [197]. These risk factors are general physical health, high cholesterol, BMI/obesity, smoking, the presence of other chronic health conditions (stroke, coronary heart diseas), diet, alcohol consumption,

[4] https://www.cdc.gov/diabetes/health-equity/diabetes-by-the-numbers.html

[5] https://www.cdc.gov/brfss/index.html

exercise, household income, marital status, time since last checkup, education level, health care coverage, and mental health. For each risk factor, we extract a set of relevant features from the BRFSS foxed core and rotating core questionnaires. We also use a shared set of demographic indicators (race, sex, state, survey year, and a question related to income level). The prediction target is a binary indicator for whether the respondent has ever been told they have diabetes.

Distribution Shift: While diabetes affects a large fraction of the overall population, diabetes risk varies according to several demographic factors. One such factor is race/ethnicity [89, 35], with all other race-ethnicity groups reported in the 2022 CDC National Diabetes Statistics Report displaying higher risk than 'White non-Hispanic' individuals[35]. Compounding this issue, it has been widely acknowledged that health studiy populations tend to be biased toward white European-Americans [45, 134, 56, 93]. As a result, these studies have tended to focus on risk factors affecting white populations at the expense of identifying risk factors for nonwhite populations [93], despite distinct differences in how these populations are affected by various disease risk factors, differences in individuals' genetic factors, and differences in how they respond to medication across racial and ethnic populations. This disparity is a contributing factor to race-based disparities in treatment for diabetes [41].

In order to simulate the domain gap induced by these real-world differences in study vs. deployment populations, we partition the benchmark task by race/ethnicity. We use "White non-Hispanic"-identified individuals as the training domain, and all other race/ethnicity groups as the target domain.

B.2.6 Hypertension

Background: Hypertension, or systolic blood pressure (typically systolic pressure 130 mm Hg or higher or diastolic 80 or higher) affects nearly half of Americans [5]. Hypertension is sometimes called a "silent killer" because in most cases, there are no obvious symptoms of hypertension [5]; this would make an accurate at-risk model of hypertension useful. When left untreated, hypertension is associated with the strongest evidence for causation of all risk factors for heart attack and other cardiovascular disease [66]. Hypertension also increases the risk of stroke, kidney damage, vision loss, insulin resistance, and other adverse outcomes [6]. While existing tools have attempted to predict blood pressure without the use of a cuff (the gold-standard measurement of blood pressure), these tools are still significantly less accurate (see e.g. [162, 59]), and there is an ongoing need for effective blood pressure measurement.

Data Source: We use BRFSS as the raw data source, as described in Task B.2.5 above. However, for the hypertension prediction task, we use features related to the following set of risk factors for hypertension via [130]: Age, family history and genetics, other medical conditions (e.g. diabetes, various forms of cancer), race/ethnicity, sex, and social and economic factors (income, employment status). We collect all survey questions related to these risk factors and use them as the predictors for this task, along with a shared set of demographic indicators (race, sex, state, survey year, and a question related to income level).

Distribution Shift: We use BMI category as the domain splitting variable. Individuals with BMI identified as "overweight" or "obese" are in the held-out domain, and those identified as "underweight" or "normal weight" are in the training domain. This simulates a model being deployed under subpopulation shift, where the target population has different (higher) BMI than the training population.

B.2.7 Voting

Background Understanding participation in elections is a critical task for policymakers, politicians, and those with an interest in democracy. In the 2020 United States presidential election, for example, voter turnout reached record levels, but it is estimated that only 66.8% of eligible individuals voted according to the U.S. Census[6]. Additionally, so-called "likely voter models," that predict which individuals will vote in an electio, are widely acknowledged as critical to polling and campaigning in U.S. politics. Predicting whether an individual will vote is notoriously difficult; one reason for this challenge is that domain shift is a fundamental reality of such modeling (presidential elections only occur every four years, after which significant political and demographic changes occur prior to the next presidential election).

The prediction target for this dataset is to determine whether an individual will vote in the U.S presidential election, from a detailed questionnaire.

Data Source We use data from the American National Election Studies (ANES)[7]. Since 1948, ANES has conducted surveys, usually administered as in-person interviews, during most years of national elections. This series of studies, known as the ANES "Time Series," constitutes a pre-election interview and a post-election interview during years of Presidential elections, along with other data sources. Topics cover voting behavior

[6] https://www.census.gov/library/stories/2021/04/record-high-turnout-in-2020-general-election.html

[7] https://electionstudies.org/

and the elections, together with questions on public opinion and attitudes.

We use features derived from the ANES Time Series. From the pool of over 500 questions in the ANES Time Series, we extract a set of features related to Americans' voting behavior, including their social and political attitudes, opinions about elected leaders, and media consumption habits.

Domain Shift We introduce a domain split by geographic region. We use the ANES Census Region feature, where the out-of-domain region is the region representing the southern United States (AL, AR, DE, D.C., FL, GA, KY, LA, MD, MS, NC, OK, SC,TN, TX, VA, WV). This simulates a study in which voter data is collected in one part of the country, and the goal is to infer voting behavior in another geographic region; this is a common occurence with polling data, particularly during the U.S. primaries, which occur over a period of several weeks at the state level.

B.2.8 Childhood Lead Exposure

In this task, the goal is to identify children 18 or younger with elevated lead blood levels.

Background: Lead is a known environmental toxin that has been shown to affect deleteriously the nervous, hematopoietic, endocrine, renal, and reproductive systems[8]. In young children, lead exposure is a particular hazard because children more readily absorb lead than adults, and children's developing nervous systems also make them more susceptible to the effects of lead. However, most children with any lead in their blood have no obvious immediate symptoms.[9] The risk for lead exposure is disproportionately higher for children who are poor, non-Hispanic black, living in large metropolitan areas, or living in older housing.

The CDC sets a national standard for blood lead levels in children. This value was established in 2012 to be 3.5 micrograms per decileter (μg/dL) of blood.[10] This value, called the blood lead reference value (BLRV) for children, corresponds to the 97.5 percentile and is intended to identify lead exposure in order to allow parents, doctors, public health officials, and communities to act early to reduce harmful exposure to lead in children. Thus, early prediction of childhood lead exposure, as well as accurate just-in-time prediction for children where obtaining actual laboratory blood test results is too costly or infeasible, is of high utility to

[8] https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/P_PBCD.htm

[9] https://www.cdc.gov/nceh/lead/prevention/blood-lead-levels.htm

[10] https://www.cdc.gov/nceh/lead/data/blood-lead-reference-value.htm

many stakeholders.

Early detection of lead exposure can trigger many potentially impactful interventions, including: environmental and home analysis for early identification of sources of lead; testing and treatment for nutritional factors influencing susceptibility to lead exposure (such as calcium and iron intake); developmental analysis and support; and additional medical diagnostic tests.[11]

Using the laboratory blood test results from the NHANES (see 'Data Source' below), the task is to identify whether a respondents' blood level exceeds the BLRV *using only questionnaire data*. We use respondents of age 18 or younger as the target population (note that respondent data for ages 1-5 is restricted and thus not available to our benchmarking study). This simulates the prediction of expensive and time-consuming laboratory testing using a quick and inexpensive questionnaire. Laboratory testing is conducted by the CDC at the National Center for Environmental Health, Centers for Disease Control and Prevention, Atlanta, GA[12]

Data Source: The data are drawn from the CDC National Health and Nutrition Examination Survey (NHANES)[13], a program of the National Center for Health Statistics (NCHS) within the Centers for Disease Control and Prevention (CDC). NHANES is a program of studies designed to assess the health and nutritional status of adults and children in the United States. The survey is unique in that it combines extensive interviews with physical examinations and high-quality laboratory testing. The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions. The survey examines a nationally representative sample of about 5,000 persons each year. The examination component consists of medical, dental, and physiological measurements, as well as laboratory tests administered by highly trained medical personnel.

Findings from NHANES are used to determine the prevalence of major diseases and risk factors for diseases; to assess nutritional status and its association with health promotion and disease prevention; and are the basis for national standards for such measurements as height, weight, and blood pressure. Data from this

[11] https://www.cdc.gov/nceh/lead/advisory/acclpp/actions-blls.htm

[12] A detailed description of the methods and procedures used for laboratory testing for lead in the 2017-2018 NHANES survey is given at https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/P_PBCD.htm; similar descriptions are available for each year of data collection.

[13] https://wwwn.cdc.gov/Nchs/Nhanes/

survey are widely used in epidemiological studies and health sciences research.

We use only questionnaire-based NHANES features as the predictors, but use a prediction target from the NHANES' lab-based component. This simulates the development of a screening questionnaire to predict blood lead levels.

Distribution Shift: We use poverty as a domain-splitting variable. Children from low-income households and those who live in housing built before 1978 are at the greatest risk of lead exposure[14]. However, due to factors mentioned above, impoverished populations can be less likely to be included in medical studies, including those that may involve in-person visits for blood laboratory testing, which is the primary method for lead exposure detection. We use the poverty-income ratio (PIR) measurement in NHANES. The PIR is calculated by dividing total annual family (or individual) income by the poverty guidelines specific to the survey year. The Department of Health and Human Services (HHS) poverty guidelines are used as the poverty measure to calculate this ratio. These guidelines are issued each year, in the Federal Register, for determining financial eligibility for certain federal programs, such as Head Start, Supplemental Nutrition Assistance Program (SNAP), Special Supplemental Nutrition Program for Women, Infants, and Children (WIC), and the National School Lunch Program. The poverty guidelines vary by family size and geographic location (with different guidelines for the 48 contiguous states and the District of Columbia; Alaska; and Hawaii).

The training domain is composed of individuals with PIR of at least 1.3; persons with PIR ≤ 1.3 are in the held-out domain. The threshold of 1.3 is selected based on the PIR categorization used in NHANES, where PIR ≤ 1.3 is the lowest level.

B.2.9 Hospital Readmission

Background: Effective management and treatment of diabetic patients admitted to the hospital can have a significant impact on their health outcomes, both short-term and long-term [184]. Several factors can affect the quality of treatment patients receive [175]. One of the costliest and potentially most adverse outcomes after a patient is released from the hospital is for that patient to be *readmitted* soon after their initial release; this can both be a sign of a condition that is not improving, and, at times, ineffective initial treatment.

[14] https://www.cdc.gov/nceh/lead/prevention/populations.htm

Thus, predicting the readmission of patients is a priority from both a medical and economic perspective.

In this task, the goal is to predict whether a diabetic patient is *readmitted* to the hospital within 30 days of their initial release.

Data Source: We use the dataset provided by [175][15]. The dataset represents 10 years (1999-2008) of clinical care at 130 US medical facilities, including hospitals and other networks. It includes over 50 features representing patient and hospital outcomes. The dataset includes observations for records which meet the following criteria: (1) It is an inpatient encounter (a hospital admission). (2) It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis. (3) The length of stay was at least 1 day and at most 14 days. (4) Laboratory tests were performed during the encounter. (5) Medications were administered during the encounter.

The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medication, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc. We use the full set of features in the initial dataset, which is described in [175].

Distribution Shift: Patients can be (re)admitted to hospitals from a variety of sources. The source of a patient admission canbe correlated with many demographic and other risk factors known to be related to health outcomes (e.g. race, income level, etc.).

We use the "admission source" as the domain split for TableShift. There are 21 distinct admission sources in the dataset, including "transfer from a hospital", "physician referral", etc. After conducting a sweep over various held-out values, we use "emergency room" as the held-out domain split. This matches a potential scenario where a model is constructed using a variety of admission sources, but a patient from a novel source is added; it is also possible e.g. that data from emergent patients could not be collected when training a readmission model. We note that this domain split provides 20 unique training subdomains (the other admission sources), which is the largest $|D^{\text{train}}|$ in TableShift.

[15] https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

B.2.10 Sepsis

Background: Sepsis is a life-threatening condition that arises when the body's response to infection causes injury to its own tissues and organs. Sepsis is a major public health concern with significant morbidity, mortality, and healthcare expenses; each year, 1.7 million adults in America develop sepsis, of which at least $350,000$ die during their hospitalization or are discharged to hospice. The CDC estimates that 1 in 3 people who dies in a hospital had sepsis during that hospitalization[16].

Early detection and antibiotic treatment of sepsis improve patient outcomes. While advances have been made in early sepsis prediction, there is a fundamental unmet clinical need for improved prediction [153]. The goal in this task is to predict, from a set of fine-grained ICU data (including laboratory measurements, sensor data, and patient demographic information), whether a patient will experience sepsis onset within the next 6 hours.

Data Source: We use the data source from the PhysioNet/Computing in Cardiology Challenge [153], which was designed by clinicians and other healthcare experts to facilitate the development of automated, open-source algorithms for the early detection of sepsis from clinical data. The dataset is derived from ICU patient records for over $60,000$ patients from two hospitals with up to 40 clinical variables collected during each hour of the patient's ICU stay.

Distribution Shift: We explored multiple domain shifts for this dataset; we note that, in particular, splitting domains by *hospital* did *not* lead to a shift gap for tuned baseline models (although there is a third, held-out hospital that was used in the original challenge for this dataset, it is not publicly available and is not part of the TableShift benchmark). Instead, we use "length of stay" as a domain shift variable. We bifurcate the dataset based on how long a patient has been in the ICU, with patients having been in ICU for ≤ 47 hours in the training domain, and patients having been in ICU more than 47 hours in the test domain. This matches a scenario where a medical model is trained only on observed stays of a fixed duration (no more than two full days), but then used beyond its initial observation window to predict sepsis in patients with longer stays. We note that length of stay of 47 hours corresponds to the 80th percentile of the data for that feature.

[16] https://www.cdc.gov/sepsis/what-is-sepsis.html

B.2.11 ICU Patient Length-of-Stay

Background: According to [143], length of hospital stay is, along with patient mortality, "the most important clinical outcome" for an ICU admission. Accurately predicting the length of stay of a patient can aid in assessment of the severity of a patient's condition. Of particular clinical relevance, making these predictions *early* and with a *non-zero time gap*between the prediction and the outcome is of real-world importance: predictions must be made sufficiently early such that a patient's treatment can be adjusted to potentially avoid a negative outcome. The importance of this prediction task for real-world clinical care is underscored by the many previous works in the medical literature addressing this prediction topic (see e.g. [82, 143, 191].

In our benchmark, the specific task is to predict, from the first 24 hours of patient data, an ICU patient's stay will exceed 3 days (a binary indicator for whether length of stay > 3). We note that this is directly adopted from MIMIC-extract.

Data Source: We use the MIMIC-extract dataset [191]. MIMIC-extract is an open-source pipeline for transforming raw electronic health record (EHR) data from the Medical Information Mart for Intensive Care (MIMIC-III) dataset [95].

MIMIC-III, the underlying data source, captures over a decade of intensive care unit (ICU) patient stays at Beth Israel Deaconess Medical Center in Boston, USA. An individual patient might be admitted to the ICU at multiple times in the dataset; however, MIMIC-extract focuses on each subject's first UCI visit only, since those who make repeat visits typically require additional considerations with respect to modeling and care [191]. MIMIC-extract includes all patient ICU stays in the MIMIC-III database that where the following criteria are met: (*i*) the subject is an adult (age of at least 15 at time of admission), (*ii*) the stay is the first known ICU admission for the subject, and (*iii*) the total duration of the stay is at least 12 hours and less than 10 days.

MIMIC-extract is designed by EHR domain experts with clinical validity (of data) and relevance (of prediction tasks) in mind. In addition to the filtering described above, MIMIC-extract's pipeline includes steps to standardize units of measurement, detect and correct outliers, and select a curated set of features that reduce data missingness in the preprocessed data; for details on the steps taken by the original authors to achieve this, see [191]. We use the preprocessed version of MIMIC-extract made available by the authors

[17]. This includes the static demographic variables, alongside the time-varying vitals and labs described in [95]. Because event he preprocessed data contains missing values, we use the authors' default methods for handling missing data.

The resulting dataset contains approximately $24,000$ observations.

Distribution Shift: We split the domains by health insurance type. We train on patients with all insurance types except Medicare, and use patients with Medicare insurance as the target domain.

B.2.12 ICU Patient In-Hospital Mortality

Background: As discussed in the background of §B.2.11, hospital mortality is considered to be one of the most important outcomes for ICU patients. The clinical relevance of hospital mortality is perhaps even more clear than for length-of-stay prediction, as preventing patient mortality is one of the primary goals for many patients. Again, as discussed in §B.2.11, making this prediction *early* is of particular importance, as early predictions can provide a proxy for overall patient risk and can be used to intervene to avoid mortality.

We note that in this task, we are predicting *hospital* morality (that the patient dies at any point during this visit, even if they are discharged from the ICU to another unit in the hospital). Hospital mortality events are distinct from (and a superset of) ICU mortality events. As mentioned above, the importance of this prediction task for real-world clinical care is underscored by the many previous works addressing this prediction topic (see e.g. [82, 143, 191].

Data Source: This task uses the same data source and feature set from MIMIC-extract described above in §B.2.11.

Distribution Shift: We split the domains by health insurance type. We train on patients with all insurance types except { Medicare, Medicaid } and use patients with { Medicare, Medicaid } insurance as the target domain.

[17]The publicly-accessible dataset (which requires credentialed MIMIC-III access through PhysioNet due to privacy restrictions) is described at https://github.com/MLforHealth/MIMIC_Extract

B.2.13 FICO Home Equity Line of Credit (HELOC)

Background: FICO (legal name: Fair Isaac Corporation) is a US-based company that provides credit scoring services. The FICO score, a measure of consumer credit risk, is a widely used risk assessment measure for consumer lending in the united states.

The Home Equity Line of Credit (HELOC) is a line of credit, secured by the applicant's home. A HELOC provides access to a revolving credit line to use for large expenses or to consolidate higher-interest rate debt on other loans such as credit cards. A HELOC often has a lower interest rate than some other common types of loans. To assess an applicant's suitability for a HELOC, a lender evaluates an applicants' financial background, including credit score and financial history. The lender's goal is to predict, using this historical customer information, whether a given applicant is likely to repay a line of credit and, if so, how much credit should be extended.

In addition to desiring accurate credit risk predictions for their overall utility for both lenders and borrowers, lending institutions are incentivized (and, in some cases, legally required) to use models which achieve some degree of robustness: institutions can face severe penalties when borrowers are not treated equitably (e.g. [185]).

Data Source: We use the dataset from the FICO Commmunity Explainable AI Challenge[18], an open-source dataset containing features derived from anonymized credit bureau data. The binary prediction target is an indicator for whether a consumer was 90 days past due or worse at least once over a period of 24 months from when the credit account was opened. The features represent various aspects of an applicant's existing financial profile, including recent financial activity, number of various transactions and credit inquiries, credit balance, and number of delinquent accounts.

Distribution Shift: It is widely acknowledged that the dominant approach to credit scoring using financial profiles can unintentionally discriminate against historically marginalized groups (credit bureau data do not include explicit information about race [120]). For example, since FICO scores are based on payment history and credit use and many marginalized groups in the United States have lower or less reliable incomes, these marginalized groups can suffer from systematically lower credit scores [123, 141, 11, 120]; this has been referred to as the "credit gap" [105, 42]. In particular, debt and savings level play a role in credit scores

[18] https://community.fico.com/s/explainable-machine-learning-challenge

and can systematically disadvantage Black and Hispanic applicants, even when demographic data are not formally used in the credit rating process [123, 120].

For this task, we partition the dataset based on the 'External Risk Estimate', a feature in the dataset corresponding to the risk estimate assigned to an applicant by a third-party service. This estimate was identified in the original FICO explanable ML challenge [19]. We use individuals with a high external risk estimate (where "high" estimate is defined as exceeding an external risk estimate of 63, a threshold identified in the original challenge-winning model linked above) as the training domain, and individuals with estimate ≤ 63 as the held-out domain.

B.2.14 College Scorecard Degree Completion Rate

Background: Higher education is increasingly critical to securing strong job and income opportunities for persons in the United States. At the same time, the cost of obtaining a four-year college degree is extremely high: The average cost of college* in the United States is $\$35,551$ per student per year, including books, supplies, and daily living expenses and this cost has more than doubled in the 21st century alone, with an annual growth rate of 7.1% [79].

However, not all institutions have similar outcomes for students. Graduation rates across institutions in the U.S. vary widely, and failure to complete a degree can leave a student with significant debt and a reduced ability to repay it. Understanding factors related to degree completion is an area of active research.

For this task, our goal is to predict whether an institution has a low completion rate, based on other characteristics of that institution. While the definition of a "low" completion rate is ultimately subjective and context-dependent, we use a thredhold of 50%, which is approximately equivalent to the median graduate rate across the institutions in the dataset. We use the completion rate for first-time, full-time students at four-year institutions (150% of expected time to completion/6 years).

Data Source: We use the College Scorecard[20]. The College Scorecard is an institution-level dataset compiled by the U.S. Department of Education from 1996-present. The College scorecard includes detailed institutional factors, including information about each institutions' student population, course offerings, and

[19] https://community.fico.com/s/blog-post/a5Q2E0000001czyUAA/fico1670

[20] https://collegescorecard.ed.gov

outcomes.

Distribution Shift: Institutions vary widely in their profiles, student populations, educational approach, and target industries or student pathways. We partition universities according to the `CCBASIC` variable[21], which gives the Carnegie Classification (Basic)[22]. This classification uses a framework developed by the Carnegie Commission on Higher Education in the early 1970s to support its research program. Partitioning our data according to this variable measures the robustness over institutional subpopulations, and is thus a form of subpopulation shift. We use the following set of institutions as the target domain (all other institutional types are in the training domain): 'Special Focus Institutions–Other special-focus institutions', 'Special Focus Institutions–Theological seminaries, Bible colleges, and other faith-related institutions', "Associate's–Private For-profit 4-year Primarily Associate's", 'Baccalaureate Colleges–Diverse Fields', 'Special Focus Institutions–Schools of art, music, and design', "Associate's–Private Not-for-profit", "Baccalaureate/Associate's Colleges", "Master's Colleges and Universities (larger programs)". Exact definitions of each institution class are available via the Carnegie Commission on Higher Education[23].

B.2.15 ASSISTments Tutoring System Correct Answer Prediction

Background: Machine learning systems are increasingly being adopted in digital learning tools for students of all ages. The ASSISTments tutoring platform[24] is a free, web-based, data-driven tutoring platform for students in grades 3-12. As of 2020, ASSISTments has been used by approximately 60,000 students with over 12 million problems solved [57]. ASSISTments also periodically releases open-source data snapshots from their platform to support educational research.

Data Source: We use the open-source ASSISTments 2012-2013 dataset. This is a dataset from school year 2012-2013 which contains submission-level features (each row in the dataset represents one submission by a student attempting to answer a problem on the ASSISTments tutoring platform). In addition to containing student-, problem-, and school-level features, the dataset also contains affect predictions for students based

[21]The data dictionary for the College Scorecard is available at https://collegescorecard.ed.gov/assets/CollegeScorecardDataDictionary.xlsx

[22]https://carnegieclassifications.acenet.edu

[23]https://carnegieclassifications.acenet.edu

[24]https://new.assistments.org

on an experimental affect detector implemented in ASSISTments. (These affect predictions are intended to be useful in identifying affective states such as boredom, confusion, frustration, and engaged problem-solving behavior).

Distribution Shift: We partition the datasets by school. Approximately 700 schools are in the training set, and 10 schools are used as the target distribution. This simulates the process of deploying ASSISTments at a new school.

B.3 Dataset Availability

All datasets in TableShift meet the definition of "available and accessible" as described in [131]; namely, the data can be obtained without a personal request to the PI. All datasets are obtained from reliable, high-quality sources (United States government agencies, UCI Machine Learning Repository, Kaggle). We selected high-quality data sources which we expect to ensure keep the relevant data available for the foreseeable future. We provide a single script that can be used to download and preprocess TableShift data for all tasks in the git repository.

The data sources used to construct the TableShift benchmark datasets vary, and necessarily so do the restrictions or agreements required to access this data. All data sources have an established credentialization procedure that is open to the public, provides rapid access to the data, and is expected to be maintained for many years. An overview of the restrictions for each dataset is given below. A link to the data use agreement or credentialization procedure for each dataset marked "open credentialized access" is available in the README of our github repo; we will maintain this list over time if the access agreements change.

B.4 Related Work

B.4.1 Distribution/Domain Shift

The (non)robustness of modern machine learning models to distribution shift has been extensively studied, but primarily in non-tabular domains, such as vision and language [125, 124]. Through the use of diverse and high-quality benchmarking suites, several recent works have demonstrated that many existing robust learning or domain generalization methods do not outperform standard supervised training such as SGD [76, 108]. Recent evidence has also suggested that in-distribution (ID) test performance is a very strong predictor of out-of-distribution (OOD) test performance in the domains of image classification [125], language modeling

Table B.1: Dataset availability.

Task	Public Access	Open Credentialized Access	Source
ASSISTments	✓		Kaggle
College Scorecard	✓		Department of Education
ICU Hospital Mortality		✓	MIMIC Clinical Database
Hospital Readmission	✓		UCI Machine Learning Repository
Diabetes	✓		Centers for Disease Control/BRFSS
ICU Length of Stay		✓	MIMIC Clinical Database
Voting		✓	American National Election Survey
Food Stamps	✓		American Community Survey
Unemployment	✓		American Community Survey
Income	✓		American Community Survey
FICO HELOC		✓	FICO
Public Health Ins.	✓		American Community Survey
Sepsis		✓	PhysioNet
Childhood Lead	✓		Centers for Disease Control/NHANES
Hypertension	✓		Centers for Disease Control/BRFSS

[115], and question answering [12], but whether these relationships hold for tabular data is unknown.

Several families of methods have been proposed to address this sensitivity to distribution shifts, including methods for distributional robustness [159, 113] and domain generalization [2, 9, 199, 198, 114, 98] . However, these methods are largely evaluated in non-tabular domains, and several "standard" domain generalization methods have never been applied to tabular data, to our knowledge. Formal analyses of robustness to any kind of shift in the tabular domain have been lacking [68].

B.4.2 Tabular Data Modeling

Tabular data – data defined by structured, heterogeneous features – is common in many real-world applications, including medical diagnosis, finance, social science, and recommender systems [28, 98, 170]. In many respects, tabular data is different from the other modalities where deep learning models have had great success in the past decade. In contrast to these other modalities, where deep learning is the undisputed state of the art, deep learning-based models have tended to underperform on tabular data, and the state of the art is often considered to be tree-based ensemble models, such as XGBoost, LightGBM, or CatBoost [28, 73, 170, 68].

Deep learning-based models have been proposed for tabular data modeling, including carefully-regularized deep multilayer perceptrons (MLPs) [98], tabular variants of ResNet [73] and Transformer architectures [92, 73, 173], and differentiable tree-inspired models [140]. However, it is unclear whether there is any benefit from these sophisticated architectures, which are often derived from models which were designed for non-tabular tasks. Subsequent evaluations of deep learning-based tabular data models have often shown tree-based models to achieve superior performance [170, 28, 68]. However, their robustness to *distribution shift* has not been thoroughly evaluated (a notable exception is [68], which strictly evaluates subgroup robustness).

B.4.3 Benchmarking for Machine Learning

Benchmarking – the use of standardized, publicly-available, high-quality datasets to evaluate performance on one or more tasks – is a critical practice contributing to progress the machine learning [116]. Distribution shift benchmarks in particular have been critical in assessing progress in the robustness of vision and language models, e.g. [108, 76, 174]. Because these benchmarks often require interfacing with many distinct

data sources, successful and widely-used benchmarks also typically include a lightweight software API for interfacing with benchmarking datasets in a consistent manner[25]. In the IID setting, benchmark datasets have also been crucial to assessing and driving progress, such as ImageNet [60] for vision, LibriSpeech [135] for speech, AudioSet [71] for audio classification, or GLUE for NLP [188]. Critically, evaluations have shown that reuse of these high-quality benchmarks such as CIFAR-10, ImageNet, and even widely-used Kaggle datasets has not led to "overfitting" to performance on the benchmarks [154], and, in fact, progress on these benchmarks generalizes beyond the benchmark tasks [151].

High-quality benchmarks for *tabular* data are lacking, as has been noted in many previous works [73, 28, 68, 74, 170, 119, 72]. Existing datasets used for *de facto* tabular data "benchmarking" are often of low quality. For example, the German Credit dataset contains only $1k$ observations; the COMPAS and Adult datasets have data quality and bias issues [14, 15, 49]. While a small number of general tabular benchmarks have been proposed [28, 74], they have not seen widespread adoption, do not include the software utilities that have driven adoption of benchmarks in language and vision [108, 76], and do not contain distribution shifts (we make more detailed comparisons between TableShift and existing benchmarks in Section B.7). Critically, these tabular benchmarks also often lack feature-level documentation, which can be critical for tabular data.

Thus, while limited individual benchmarks do exist for tabular data modeling (without distribution shift) or for distribution shift (without tabular data), there is no existing benchmark that provides a high-quality set of tabular datasets *and* associated distribution shifts.

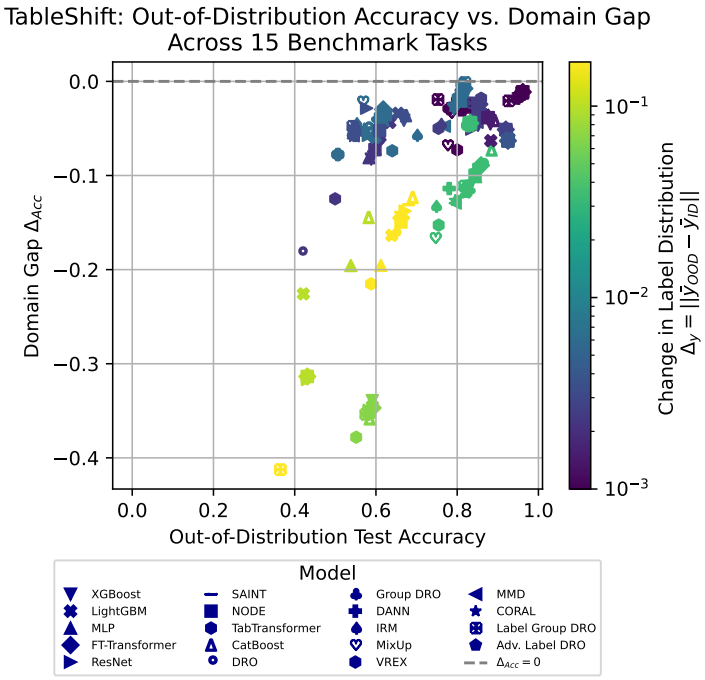
B.5 Additional Dataset Details and Results

In this section we provide a brief tour of exploratory results regarding the domain shift datasets in TableShift, and additional expeirmental results.

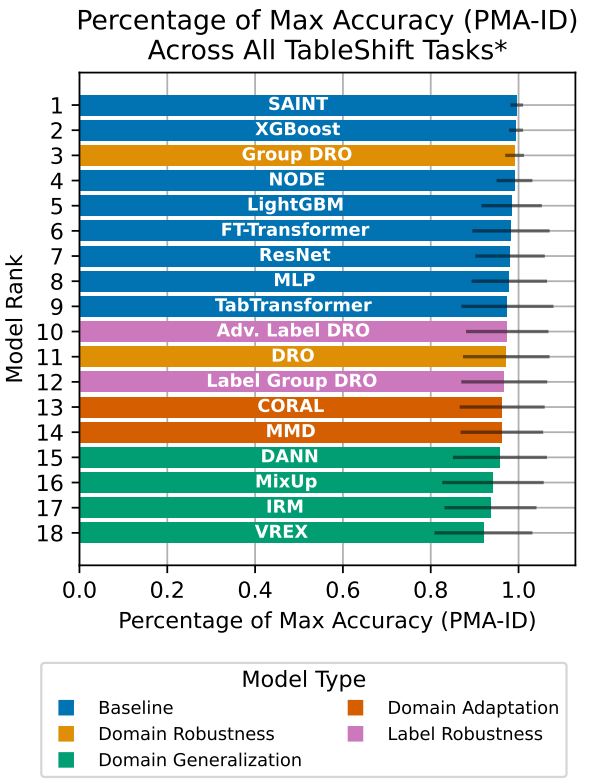
B.5.1 Domain Split Selection

For many tasks in TableShift, there exist clear motivations for selecting certain splitting variables, and for selecting which values of these variables to use as out-of-domain value(s) for our benchmarks. However, for oehters, there might bemultiple plausible splitting variables, or no obvious way to choose which specific

[25] e.g. DomainBed https://github.com/facebookresearch/DomainBed, WILDS https://wilds.stanford.edu/, BIG-bench https://github.com/google/BIG-bench



(a) Out-of-Distribution Accuracy vs. Shift Gap Δ_{Acc} across all TableShift tasks.



(b) Percentage of Maximum In-Distribution Accuracy (PMA-ID) across all TableShift tasks.

Figure B.1: Additional results.

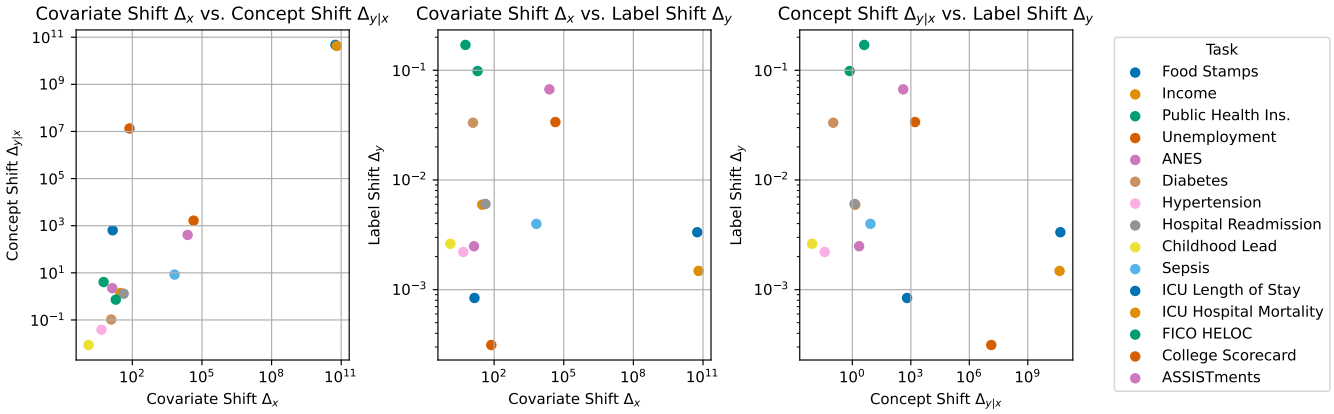


Figure B.2: Pairwise scatterplots of shifts. Each point represents one dataset. Metrics are computed according to the domain split for each dataset (ID test vs. OOD test) according to the metric definitions for Δ_x, $\Delta_{y|x}$, Δ_y in Section B.5. (a) left: Covariate shift Δ_x (computed via Optimal Transport Data Distance) vs. concept shift $\Delta_{y|x}$ (computed via Frechet Dataset Distance); $\rho = 0.99$. (b) center: Covariate shift Δ_x vs. label shift Δ_y; $\rho = -0.20$. (c) left: Concept shift $\Delta_{y|x}$ vs. label shift Δ_y; $\rho = -0.20$.

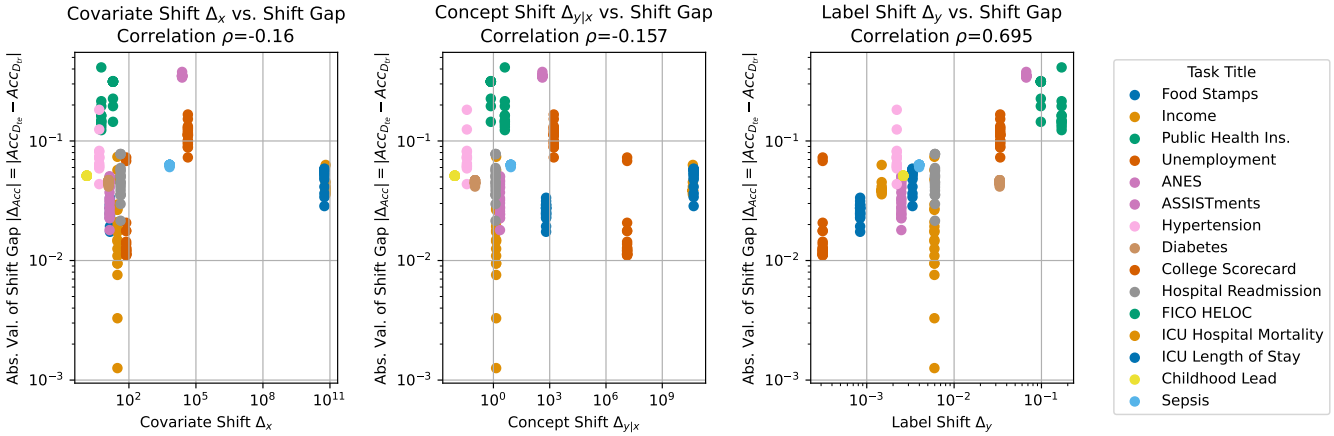


Figure B.3: Domain shift metrics vs. (absolute) shift gap. Each point represents a tuned model on a given dataset. Only Label shift shows a strong correlation with shift gap ($\rho = 0.73$).

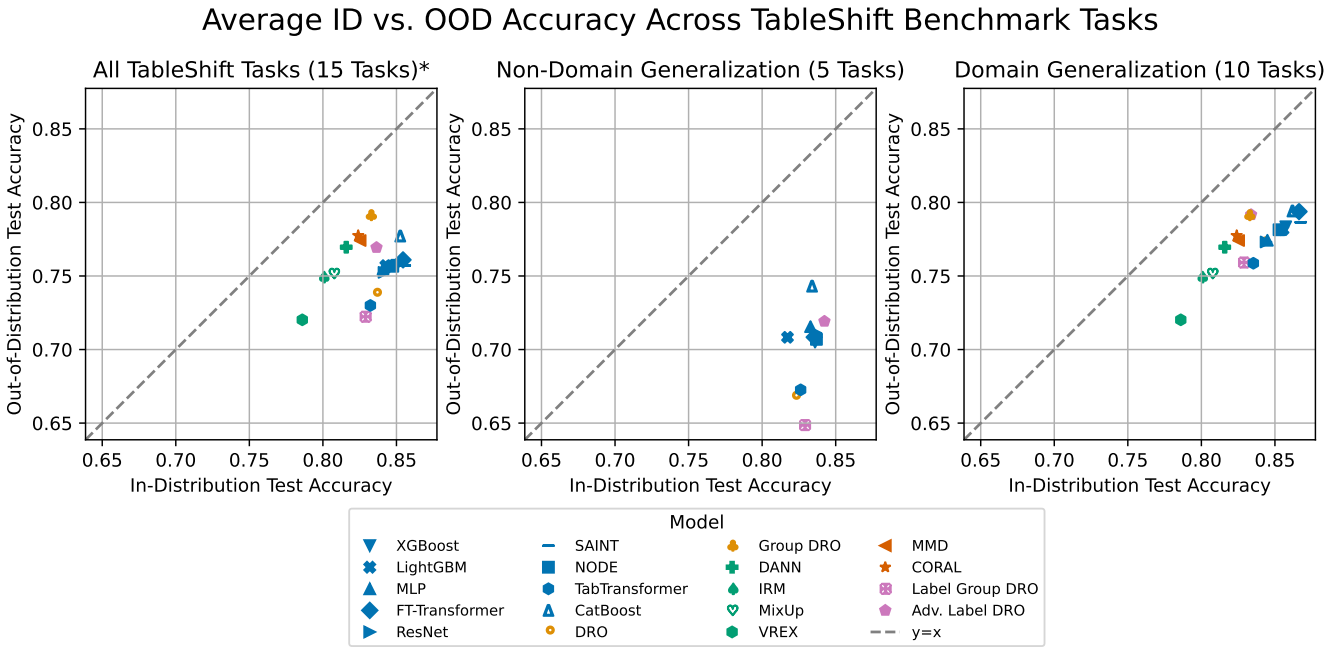


Figure B.4: TableShift benchmark results, mean per model (left: non-domain generalization tasks, $\rho = 0.834$; right: domain generalization tasks, $\rho = 963$). In-domain and out-of-domain accuracy show a general linear trend. Baseline models (blue) consistently match or outperform domain robustness and domain generalization methods.

value(s) to use as out of domain (e.g., any geographic region in ACS might be equallyplausible as a holdout domain for the Feed Stamps task).

For tasks where there were known domain splits that were likely to induce performance gaps that matches a real-world domain shift scenario, we began by selecting these. When tuned baselines (LightGBM and XGBoost) showed a shift gap Δ_{Acc} of at least 1%, we used that split. However, for tasks without a clear domain split or where mutliple plausible splitting values exist, we do the following. First, we identified a variable(s) that was likely to contribute to an actual shift in a real-world production through reviewing the relevant literature. Then, for each value $d \in \{d_1, \dots d_D\} = \mathcal{D}$, we train on $\{\mathcal{D} \setminus d\}$ and evaluate on d. We select the split(s) that induced the highest performance gap in our baseline tree methods). We repeat this process for each dataset until a split that is both real-world relevant and also leads to a shift gap is found.

B.5.2 Domain Shift Metrics (Covariate, Concept, and Label Shift)

As noted above, the domain shift Δ_{Acc} incurred when training a classifier is comprised of three distinct forms of shift: changes in $p(x)$ ("covariate shift"), changes in $p(y|x)$ ("concept shift"), and changes in $p(y)$ ("label shift"). It is not possible to measure the true shifts for any given dataset, because doing so would require knowing the true (ID, OOD) distributions. As a result, in order to still explore the influence of these various forms of shift on tabular data models, we propose metrics to approximately measure each form of shift.

We propose these metrics while noting that each is only an approximation of the actual degree of a certain form of shift in our dataset; measuring the actual underlying shift (e.g. the true change in $p(x)$ for covariate shift) is not possible from a finite sample. Thus, while these metrics can provide exploratory evidence of the relationship between a given type of shift (covariate, comcept, label) and model performance, they cannot provide direct evidence that any given shift type is (not) causing changes in model performance.

Table B.12 gives the exact In- and Out-of-Distribution label proportions for each task, which are used to compute the label shift Δ_y.

Measuring covariate shift with OTDD: We propose to use the following measure to approximate the degree of covariate shift between the (ID, OOD) test sets of a given task:

$$\Delta_x = \text{OTDD}(D^{\text{train}}, D^{\text{test}}) \tag{B.1}$$

Table B.2: Summary of tasks in the TableShift benchmark and their associated distribution shifts.

Task	Δ_x (Eqn. (B.1))	$\Delta_{y\|x}$ (Eqn (B.2))	Δ_y (Eqn (B.2))
Food Stamps	14.20	640.82	0.0008
Income	30.60	1.40	0.0060
Public Health Ins.	5.79	4.06	0.1701
Unemployment	75.47	13,389,512.51	0.0003
ANES	13.60	2.23	0.0025
Diabetes	12.28	0.10	0.0332
Hypertension	4.69	0.04	0.0022
Hospital Readmission	42.37	1.30	0.0060
Childhood Lead	1.30	0.01	0.0026
Sepsis	6609.73	8.44	0.0040
ICU Length of Stay	56,439,324,672.00	47,042,729,585.25	0.0033
ICU Hospital Mortality	64,479,092,736.00	42,639,188,407.47	0.0015
FICO HELOC	19.35	0.73	0.0983
ASSISTments	24,054.59	1137.42	0.0670
College Scorecard	43,566.39	2116.63	0.0337

where $D^{\text{train}}, D^{\text{test}}$ are the holdout (test) sets from the source and target domains, respectively. Here OTDD represents the Optimal Transport Dataset Distance with the Gaussian approximation as described in [4].

Measuring concept shift with Frechet Dataset Distance (FDD): We propose a straightforward measure of the change in $p(y|x)$ across two distributions. Inspired by measures of distributional difference widely used in the machine learning (Frechet Inception Distance, [88]) which leverage changes in the intermediate representations of a reference classifier for comparing distributions, we propose 'Frechet dataset

distance" (FDD) for comparing two distributions.

This metric is computed as follows: First, we train a classifier on the source domain using the best tuned hyperparameters from our hyperparameter sweep to obtain a fixed classifier f_θ. Then, for each domain, we compute $\tilde{x} \coloneqq f_{\theta[i]}(x)$, for each $x \in \mathcal{D}$, where i indicates that we compute the activations at the i^{th} layer of the model (this is sometimes referred to as the coding vector or feature vector for an input). Finally, we compute the Frechet dataset distance, which measures the distance between these two distributions (also called the Wasserstein-2 distance), as:

$$\text{DFD}(D^{\text{train}}, D^{\text{test}}) = ||\mu_{D^{\text{train}}} \breve{} \mu_{D^{\text{test}}}||^2 + Tr(\Sigma_{D^{\text{train}}} + \Sigma_{D^{\text{test}}} \breve{} 2 * \sqrt{\Sigma_{D^{\text{train}}} * \Sigma_{D^{\text{test}}}})$$

where $\mu_{\mathcal{D}}$ indicates the set of feature vectors from dimain $\mathcal{D}$ and $\Sigma_{\mathcal{D}}$ indicates the covariance matrix of $\mu_{\mathcal{D}}$. We refer to this measure as $\Delta_{y|x}$ below. A lower FDD score indicates a smaller distance between $x_i : x_i \in D^{\text{train}}$ and $x_j : x_j \in D^{\text{test}}$.

We parameterize the models used for FDD as MLPs. For each dataset, we use the MLP hyperparameters associated with the best validation accuracy for that model over our experiments; the model trained using these parameters is used for computing the feature activations for FDD.

Measuring label shift: We propose a simple measure of label shift. While label shift is clearly one factor influencing shift gaps and is perhaps the most straightforward to empirically estimate, it receives surprisingly little attention in existing literature on domain shift. We use the following measure to quantify the label shift between the source and target distributions:

$$\Delta_y = ||\bar{y}_{D^{\text{train}}} - \bar{y}_{D^{\text{test}}}||^2 \tag{B.2}$$

where $\bar{y}_{\mathcal{D}} = \frac{1}{|\mathcal{D}|} \sum_{i \in D} y_i$ is the empirical sample mean of a given domain. Since all tasks in TableShift are binary classification tasks, this measures the L_2 difference in the base rates across the two domains.

Using these metrics, we provide one perspective on the amount of each respective form of shift in Table B.2. Additionally, we provide scatter plots showing the pairwise relationships between these metrics in Figure B.2, and scatter plots showing the relationship between each individual metric and the shift gap in Figure B.3 (see also Figure 3.5 discussed in Section 3.5).

B.5.3 Detailed Results Per Task

We provide detailed task-specific results and data in this section. In particular, we list the complete set of main results for the (In-Distribution, Out-Of-Distribution) scatter plots shown in Figures 3.1, 3.2, 3.3, along with the 95% Clopper-Pearson confidence intervals for these results, in Tables B.3, B.4, B.5, B.7, B.8, B.6, B.9, B.10. We also give summary metrics describing the size of each dataset split in Table B.11.

Table B.3: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3. ♢: the large number of training subdomains (over 700 for ASSISTments) makes training domain generalization models impractical. □: the large dataset size makes training adversarial label DRO models impractical (since per-example gradients must be computed). We leave these experiments to future work.

Estimator	**ASSISTments**				**Childhood Lead**			
	ID Acc. (95% CI)		**OOD Acc. (95% CI)**		**ID Acc. (95% CI)**		**OOD Acc. (95% CI)**	
Adv. Label DRO	□	□	□	□	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
CatBoost	0.943	(0.942, 0.944)	0.584	(0.562, 0.607)	0.971	(0.961, 0.979)	0.92	(0.914, 0.925)
DRO	0.932	(0.931, 0.933)	0.583	(0.561, 0.606)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
FT-Transformer	0.939	(0.938, 0.94)	0.592	(0.569, 0.614)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
Label Group DRO	0.928	(0.927, 0.929)	0.574	(0.551, 0.596)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
LightGBM	0.936	(0.935, 0.937)	0.591	(0.568, 0.613)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
MLP	0.933	(0.932, 0.934)	0.583	(0.561, 0.606)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
NODE	0.935	(0.934, 0.936)	0.583	(0.561, 0.606)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
ResNet	0.933	(0.932, 0.934)	0.583	(0.561, 0.606)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
SAINT	0.935	(0.934, 0.936)	0.584	(0.562, 0.607)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
TabTransformer	0.93	(0.929, 0.93)	0.551	(0.529, 0.574)	0.971	(0.961, 0.979)	0.92	(0.915, 0.925)
XGBoost	0.93	(0.929, 0.931)	0.591	(0.568, 0.613)	0.971	(0.961, 0.979)	0.92	(0.914, 0.925)
CORAL	♢	♢	♢	♢	⋆	⋆	⋆	⋆
DANN	♢	♢	♢	♢	⋆	⋆	⋆	⋆
Group DRO	♢	♢	♢	♢	⋆	⋆	⋆	⋆
IRM	♢	♢	♢	♢	⋆	⋆	⋆	⋆
MMD	♢	♢	♢	♢	⋆	⋆	⋆	⋆
MixUp	♢	♢	♢	♢	⋆	⋆	⋆	⋆
VREX	♢	♢	♢	♢	⋆	⋆	⋆	⋆

Table B.4: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3.

Estimator	**College Scorecard**				**Diabetes**			
	ID Acc. (95% CI)		**OOD Acc. (95% CI)**		**ID Acc. (95% CI)**		**OOD Acc. (95% CI)**	
Adv. Label DRO	0.937	(0.933, 0.942)	0.826	(0.805, 0.846)	0.877	(0.875, 0.878)	0.832	(0.83, 0.833)
CatBoost	0.957	(0.954, 0.961)	0.885	(0.866, 0.901)	0.877	(0.876, 0.879)	0.833	(0.831, 0.835)
DRO	0.95	(0.946, 0.954)	0.862	(0.842, 0.88)	0.876	(0.875, 0.878)	0.832	(0.83, 0.834)
FT-Transformer	0.948	(0.944, 0.952)	0.859	(0.839, 0.877)	0.877	(0.875, 0.879)	0.832	(0.831, 0.834)
Label Group DRO	0.928	(0.924, 0.933)	0.817	(0.796, 0.838)	0.876	(0.874, 0.878)	0.831	(0.83, 0.833)
LightGBM	0.939	(0.935, 0.943)	0.822	(0.8, 0.841)	0.876	(0.874, 0.878)	0.833	(0.831, 0.835)
MLP	0.947	(0.942, 0.95)	0.845	(0.825, 0.864)	0.877	(0.875, 0.879)	0.832	(0.83, 0.833)
NODE	0.944	(0.939, 0.948)	0.844	(0.823, 0.863)	0.877	(0.875, 0.879)	0.833	(0.832, 0.835)
ResNet	0.947	(0.943, 0.951)	0.854	(0.834, 0.872)	0.874	(0.872, 0.876)	0.829	(0.828, 0.831)
SAINT	0.936	(0.931, 0.94)	0.814	(0.792, 0.834)	0.877	(0.875, 0.879)	0.833	(0.831, 0.834)
TabTransformer	0.942	(0.938, 0.946)	0.845	(0.825, 0.864)	0.875	(0.873, 0.877)	0.83	(0.829, 0.832)
XGBoost	0.942	(0.938, 0.946)	0.83	(0.809, 0.85)	0.877	(0.875, 0.879)	0.832	(0.83, 0.834)
CORAL	0.922	(0.917, 0.926)	0.795	(0.773, 0.816)	0.874	(0.872, 0.875)	0.832	(0.83, 0.834)
DANN	0.894	(0.889, 0.9)	0.78	(0.757, 0.802)	0.873	(0.871, 0.875)	0.826	(0.824, 0.827)
Group DRO	0.944	(0.939, 0.948)	0.829	(0.808, 0.849)	0.877	(0.875, 0.879)	0.832	(0.83, 0.833)
IRM	0.879	(0.873, 0.885)	0.746	(0.721, 0.769)	0.873	(0.871, 0.875)	0.826	(0.824, 0.827)
MMD	0.925	(0.92, 0.929)	0.795	(0.773, 0.816)	0.873	(0.871, 0.875)	0.826	(0.825, 0.828)
MixUp	0.912	(0.907, 0.917)	0.746	(0.721, 0.769)	0.873	(0.871, 0.875)	0.826	(0.824, 0.827)
VREX	0.907	(0.902, 0.912)	0.754	(0.731, 0.777)	0.873	(0.871, 0.875)	0.826	(0.824, 0.827)

Table B.5: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3.

Estimator	**FICO HELOC**				**Food Stamps**			
	ID Acc. (95% CI)		**OOD Acc. (95% CI)**		**ID Acc. (95% CI)**		**OOD Acc. (95% CI)**	
Adv. Label DRO	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.843	(0.84, 0.846)	0.812	(0.808, 0.815)
CatBoost	0.727	(0.67, 0.778)	0.582	(0.57, 0.594)	0.849	(0.847, 0.852)	0.825	(0.821, 0.828)
DRO	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.844	(0.841, 0.846)	0.819	(0.815, 0.822)
FT-Transformer	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.843	(0.841, 0.846)	0.816	(0.812, 0.819)
Label Group DRO	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.771	(0.768, 0.774)	0.752	(0.748, 0.756)
LightGBM	0.647	(0.584, 0.7)	0.421	(0.409, 0.433)	0.836	(0.833, 0.838)	0.808	(0.805, 0.812)
MLP	0.734	(0.678, 0.785)	0.538	(0.526, 0.55)	0.841	(0.838, 0.844)	0.815	(0.812, 0.819)
NODE	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.849	(0.847, 0.852)	0.822	(0.819, 0.825)
ResNet	0.748	(0.693, 0.798)	0.431	(0.42, 0.443)	0.843	(0.84, 0.845)	0.82	(0.817, 0.824)
SAINT	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.849	(0.846, 0.851)	0.821	(0.818, 0.825)
TabTransformer	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.836	(0.834, 0.839)	0.807	(0.803, 0.81)
XGBoost	0.745	(0.689, 0.795)	0.431	(0.419, 0.443)	0.844	(0.842, 0.847)	0.82	(0.817, 0.824)
CORAL	⋆	⋆	⋆	⋆	0.818	(0.815, 0.82)	0.793	(0.79, 0.797)
DANN	⋆	⋆	⋆	⋆	0.809	(0.806, 0.812)	0.78	(0.776, 0.784)
Group DRO	⋆	⋆	⋆	⋆	0.84	(0.838, 0.843)	0.817	(0.814, 0.821)
IRM	⋆	⋆	⋆	⋆	0.812	(0.81, 0.815)	0.795	(0.791, 0.798)
MMD	⋆	⋆	⋆	⋆	0.813	(0.81, 0.816)	0.786	(0.782, 0.789)
MixUp	⋆	⋆	⋆	⋆	0.819	(0.816, 0.821)	0.785	(0.782, 0.789)
VREX	⋆	⋆	⋆	⋆	0.809	(0.806, 0.812)	0.78	(0.776, 0.784)

Table B.6: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3.

Estimator	**Hospital Readmission**				**Hypertension**			
	ID Acc.	**(95% CI)**	**OOD Acc.**	**(95% CI)**	**ID Acc.**	**(95% CI)**	**OOD Acc.**	**(95% CI)**
Adv. Label DRO	0.655	(0.641, 0.669)	0.603	(0.599, 0.607)	0.666	(0.66, 0.672)	0.601	(0.6, 0.603)
CatBoost	0.659	(0.645, 0.674)	0.618	(0.614, 0.623)	0.67	(0.665, 0.676)	0.599	(0.597, 0.6)
DRO	0.628	(0.613, 0.642)	0.578	(0.574, 0.582)	0.598	(0.592, 0.604)	0.416	(0.414, 0.417)
FT-Transformer	0.648	(0.633, 0.662)	0.618	(0.614, 0.622)	0.666	(0.661, 0.672)	0.604	(0.603, 0.605)
Label Group DRO	0.652	(0.637, 0.666)	0.616	(0.612, 0.62)	0.665	(0.659, 0.671)	0.604	(0.603, 0.605)
LightGBM	0.658	(0.643, 0.672)	0.598	(0.594, 0.602)	0.678	(0.672, 0.683)	0.634	(0.633, 0.635)
MLP	0.648	(0.633, 0.662)	0.612	(0.608, 0.617)	0.664	(0.658, 0.67)	0.583	(0.582, 0.584)
NODE	0.659	(0.645, 0.673)	0.624	(0.62, 0.628)	0.67	(0.664, 0.676)	0.597	(0.596, 0.599)
ResNet	0.639	(0.624, 0.653)	0.581	(0.577, 0.586)	0.667	(0.661, 0.672)	0.608	(0.606, 0.609)
SAINT	0.654	(0.639, 0.668)	0.61	(0.606, 0.615)	0.669	(0.664, 0.675)	0.595	(0.594, 0.596)
TabTransformer	0.584	(0.569, 0.599)	0.507	(0.502, 0.511)	0.624	(0.618, 0.63)	0.499	(0.498, 0.501)
XGBoost	0.651	(0.636, 0.665)	0.605	(0.601, 0.61)	0.671	(0.665, 0.677)	0.588	(0.587, 0.59)
CORAL	0.622	(0.607, 0.637)	0.571	(0.567, 0.576)	⋆	⋆	⋆	⋆
DANN	0.584	(0.569, 0.599)	0.506	(0.502, 0.51)	⋆	⋆	⋆	⋆
Group DRO	0.639	(0.624, 0.653)	0.6	(0.596, 0.605)	⋆	⋆	⋆	⋆
IRM	0.595	(0.58, 0.61)	0.55	(0.546, 0.555)	⋆	⋆	⋆	⋆
MMD	0.626	(0.611, 0.64)	0.57	(0.565, 0.574)	⋆	⋆	⋆	⋆
MixUp	0.589	(0.574, 0.604)	0.567	(0.563, 0.572)	⋆	⋆	⋆	⋆
VREX	0.584	(0.569, 0.599)	0.506	(0.502, 0.51)	⋆	⋆	⋆	⋆

Table B.7: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3. ♡: the large feature dimensionality of both ICU datasets makes training Transformer-based models impractical (e.g. even a single-layer SAINT model requires >13B trainable parameters on both ICU datasets)

Estimator	**ICU Hospital Mortality**				**ICU Length of Stay**			
	ID Acc. (95% CI)		**OOD Acc. (95% CI)**		**ID Acc. (95% CI)**		**OOD Acc. (95% CI)**	
Adv. Label DRO	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.602	(0.572, 0.631)	0.544	(0.535, 0.553)
CatBoost	0.934	(0.914, 0.948)	0.892	(0.887, 0.897)	0.71	(0.682, 0.737)	0.674	(0.665, 0.682)
DRO	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.601	(0.571, 0.63)	0.544	(0.535, 0.553)
FT-Transformer	♡	♡	♡	♡	♡	♡	♡	♡
Label Group DRO	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.59	(0.56, 0.619)	0.542	(0.533, 0.551)
LightGBM	0.946	(0.928, 0.959)	0.883	(0.877, 0.888)	0.689	(0.66, 0.716)	0.655	(0.646, 0.663)
MLP	0.912	(0.891, 0.929)	0.877	(0.871, 0.882)	0.599	(0.569, 0.628)	0.544	(0.535, 0.553)
NODE	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.661	(0.632, 0.689)	0.609	(0.6, 0.618)
ResNet	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.606	(0.576, 0.635)	0.577	(0.568, 0.586)
SAINT	♡	♡	♡	♡	♡	♡	♡	♡
TabTransformer	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.604	(0.574, 0.633)	0.549	(0.54, 0.558)
XGBoost	0.927	(0.908, 0.943)	0.882	(0.876, 0.887)	0.71	(0.682, 0.737)	0.669	(0.66, 0.677)
CORAL	0.915	(0.893, 0.931)	0.875	(0.869, 0.881)	0.603	(0.573, 0.632)	0.544	(0.535, 0.553)
DANN	0.915	(0.893, 0.931)	0.876	(0.871, 0.882)	0.594	(0.564, 0.624)	0.545	(0.536, 0.554)
Group DRO	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.602	(0.572, 0.631)	0.544	(0.535, 0.553)
IRM	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.601	(0.571, 0.63)	0.544	(0.535, 0.553)
MMD	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.602	(0.572, 0.631)	0.544	(0.535, 0.553)
MixUp	0.915	(0.893, 0.931)	0.876	(0.87, 0.882)	0.602	(0.572, 0.631)	0.544	(0.535, 0.553)
VREX	0.913	(0.893, 0.931)	0.876	(0.87, 0.882)	0.597	(0.567, 0.627)	0.545	(0.536, 0.554)

Table B.8: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. □: the large dataset size makes training adversarial label DRO models impractical (since per-example gradients must be computed). We leave these experiments to future work. See also Figures 3.1, 3.2,3.3.

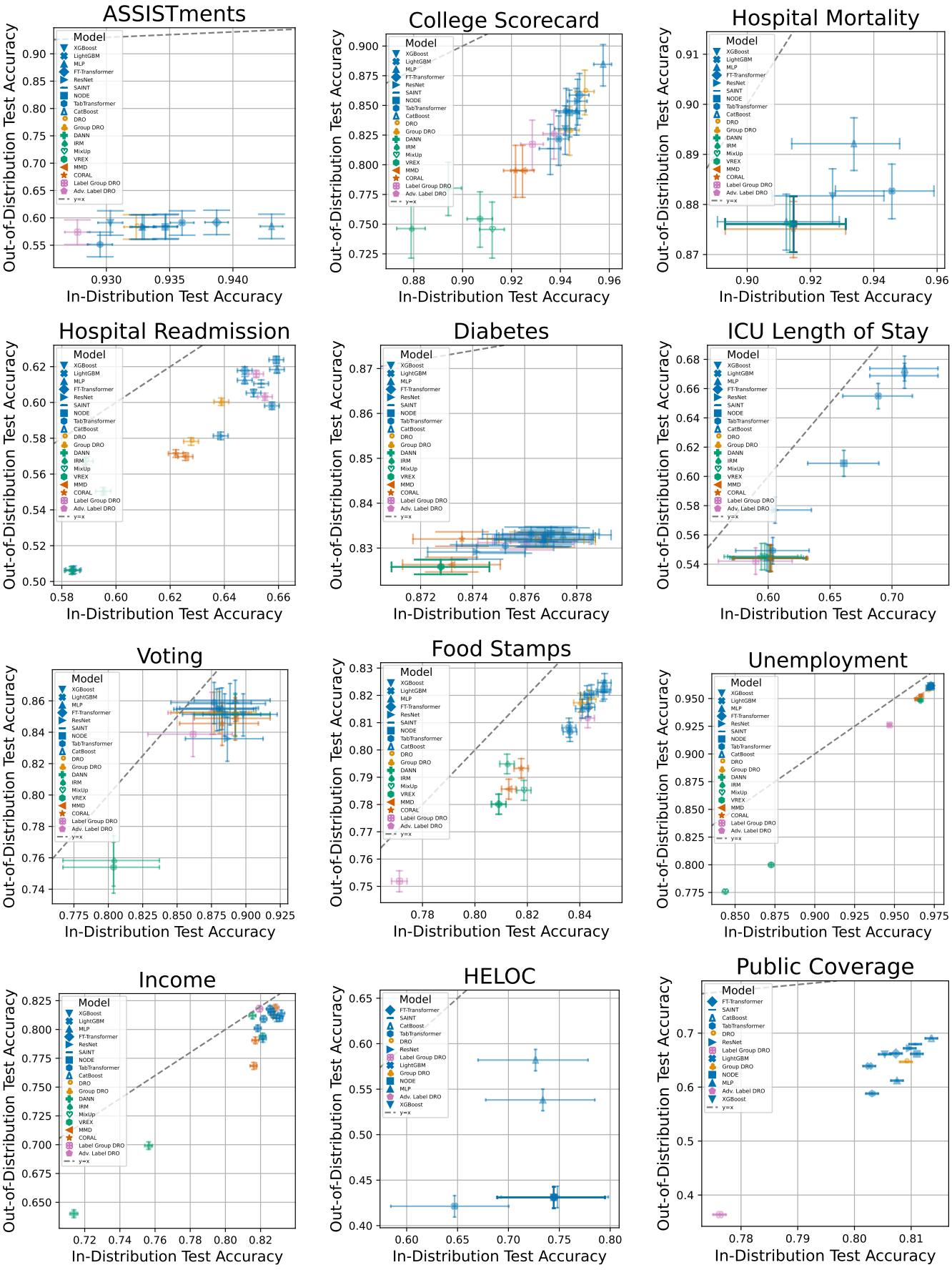
Estimator	**Income**				**Public Health Ins.**			
	ID Acc. (95% CI)		**OOD Acc. (95% CI)**		**ID Acc. (95% CI)**		**OOD Acc. (95% CI)**	
Adv. Label DRO	0.829	(0.827, 0.83)	0.819	(0.816, 0.822)	□	□	□	□
CatBoost	0.832	(0.83, 0.834)	0.814	(0.811, 0.817)	0.814	(0.812, 0.815)	0.69	(0.689, 0.691)
DRO	0.828	(0.826, 0.83)	0.818	(0.816, 0.821)	0.809	(0.808, 0.81)	0.647	(0.646, 0.648)
FT-Transformer	0.825	(0.823, 0.827)	0.818	(0.815, 0.821)	0.807	(0.806, 0.808)	0.662	(0.661, 0.663)
Label Group DRO	0.819	(0.817, 0.821)	0.818	(0.815, 0.821)	0.776	(0.775, 0.777)	0.364	(0.363, 0.365)
LightGBM	0.822	(0.82, 0.824)	0.809	(0.806, 0.812)	0.803	(0.802, 0.804)	0.639	(0.638, 0.64)
MLP	0.828	(0.826, 0.829)	0.813	(0.81, 0.816)	0.808	(0.806, 0.809)	0.612	(0.611, 0.613)
NODE	0.831	(0.829, 0.833)	0.81	(0.807, 0.813)	0.811	(0.81, 0.812)	0.662	(0.661, 0.663)
ResNet	0.826	(0.824, 0.828)	0.815	(0.812, 0.818)	0.81	(0.809, 0.811)	0.672	(0.671, 0.673)
SAINT	0.829	(0.827, 0.831)	0.81	(0.807, 0.812)	0.811	(0.81, 0.812)	0.68	(0.679, 0.681)
TabTransformer	0.818	(0.816, 0.82)	0.801	(0.798, 0.804)	0.803	(0.802, 0.804)	0.588	(0.587, 0.589)
XGBoost	0.821	(0.819, 0.823)	0.792	(0.789, 0.795)	0.805	(0.804, 0.806)	0.661	(0.66, 0.662)
CORAL	0.817	(0.815, 0.819)	0.791	(0.788, 0.793)	⋆	⋆	⋆	⋆
DANN	0.815	(0.813, 0.817)	0.812	(0.809, 0.815)	⋆	⋆	⋆	⋆
Group DRO	0.827	(0.826, 0.829)	0.813	(0.81, 0.815)	⋆	⋆	⋆	⋆
IRM	0.756	(0.754, 0.758)	0.699	(0.696, 0.702)	⋆	⋆	⋆	⋆
MMD	0.816	(0.814, 0.818)	0.768	(0.765, 0.771)	⋆	⋆	⋆	⋆
MixUp	0.821	(0.819, 0.823)	0.794	(0.791, 0.797)	⋆	⋆	⋆	⋆
VREX	0.714	(0.712, 0.716)	0.64	(0.637, 0.644)	⋆	⋆	⋆	⋆

Table B.9: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. Note that domain generalization models can only be trained on datasets with more than one training subdomain (see Table 3.1 for domain generalization datasets and Section 3.4.1 for a list of domain generalization models). ⋆: domain generalization models cannot be trained when only one training subdomain is present. See also Figures 3.1, 3.2,3.3.

Estimator	**Sepsis**				**Unemployment**			
	ID Acc.	**(95% CI)**	**OOD Acc.**	**(95% CI)**	**ID Acc.**	**(95% CI)**	**OOD Acc.**	**(95% CI)**
Adv. Label DRO	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.972	(0.971, 0.973)	0.96	(0.959, 0.961)
CatBoost	0.988	(0.987, 0.989)	0.925	(0.923, 0.926)	0.973	(0.973, 0.974)	0.962	(0.961, 0.963)
DRO	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.973	(0.972, 0.973)	0.961	(0.96, 0.962)
FT-Transformer	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.973	(0.972, 0.974)	0.962	(0.961, 0.962)
Label Group DRO	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.947	(0.946, 0.948)	0.926	(0.925, 0.927)
LightGBM	0.988	(0.987, 0.989)	0.928	(0.926, 0.929)	0.973	(0.972, 0.974)	0.96	(0.96, 0.961)
MLP	0.988	(0.987, 0.989)	0.925	(0.923, 0.926)	0.973	(0.972, 0.973)	0.96	(0.959, 0.961)
NODE	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.973	(0.972, 0.974)	0.962	(0.961, 0.963)
ResNet	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.972	(0.971, 0.972)	0.959	(0.958, 0.96)
SAINT	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.973	(0.972, 0.974)	0.962	(0.961, 0.963)
TabTransformer	0.988	(0.987, 0.989)	0.925	(0.924, 0.926)	0.972	(0.971, 0.973)	0.961	(0.96, 0.962)
XGBoost	0.988	(0.987, 0.989)	0.925	(0.923, 0.926)	0.973	(0.972, 0.973)	0.961	(0.961, 0.962)
CORAL	⋆	⋆	⋆	⋆	0.964	(0.963, 0.965)	0.95	(0.949, 0.951)
DANN	⋆	⋆	⋆	⋆	0.966	(0.965, 0.967)	0.948	(0.947, 0.95)
Group DRO	⋆	⋆	⋆	⋆	0.971	(0.97, 0.972)	0.958	(0.957, 0.959)
IRM	⋆	⋆	⋆	⋆	0.966	(0.965, 0.967)	0.948	(0.947, 0.95)
MMD	⋆	⋆	⋆	⋆	0.966	(0.966, 0.967)	0.953	(0.952, 0.954)
MixUp	⋆	⋆	⋆	⋆	0.844	(0.842, 0.846)	0.776	(0.774, 0.778)
VREX	⋆	⋆	⋆	⋆	0.873	(0.871, 0.874)	0.8	(0.798, 0.802)

Table B.10: Best (In-Distribution and Out-Of-Distribution) accuracy pair observed on each benchmark task. See also Figures 3.1, 3.2,3.3.

Estimator	**Voting**			
	ID Acc.	**(95% CI)**	**OOD Acc.**	**(95% CI)**
Adv. Label DRO	0.875	(0.843, 0.902)	0.852	(0.839, 0.865)
CatBoost	0.883	(0.852, 0.909)	0.855	(0.842, 0.868)
DRO	0.881	(0.85, 0.907)	0.853	(0.839, 0.866)
FT-Transformer	0.879	(0.848, 0.906)	0.855	(0.841, 0.868)
Label Group DRO	0.862	(0.829, 0.89)	0.839	(0.825, 0.852)
LightGBM	0.881	(0.85, 0.907)	0.855	(0.841, 0.868)
MLP	0.892	(0.862, 0.918)	0.86	(0.847, 0.873)
NODE	0.885	(0.854, 0.911)	0.851	(0.838, 0.864)
ResNet	0.887	(0.856, 0.912)	0.836	(0.822, 0.849)
SAINT	0.888	(0.858, 0.914)	0.858	(0.845, 0.871)
TabTransformer	0.877	(0.846, 0.904)	0.859	(0.845, 0.872)
XGBoost	0.898	(0.869, 0.923)	0.851	(0.838, 0.864)
CORAL	0.883	(0.852, 0.909)	0.846	(0.832, 0.859)
DANN	0.892	(0.862, 0.918)	0.852	(0.838, 0.865)
Group DRO	0.877	(0.846, 0.904)	0.852	(0.839, 0.865)
IRM	0.804	(0.767, 0.837)	0.758	(0.742, 0.774)
MMD	0.892	(0.862, 0.918)	0.849	(0.835, 0.862)
MixUp	0.892	(0.862, 0.918)	0.851	(0.837, 0.864)
VREX	0.804	(0.767, 0.837)	0.754	(0.737, 0.77)


ASSISTments
College Scorecard
Hospital Mortality
Hospital Readmission
Diabetes
ICU Length of Stay
Voting
Food Stamps
Unemployment
Income
HELOC
Public Coverage
Out-of-Distribution Test Accuracy
In-Distribution Test Accuracy
Model
XGBoost
LightGBM
MLP
FT-Transformer
ResNet
SAINT
NODE
TabTransformer
CatBoost
DRO
Group DRO
DANN
IRM
MixUp
VREX
MMD
CORAL
Label Group DRO
Adv. Label DRO
y=x

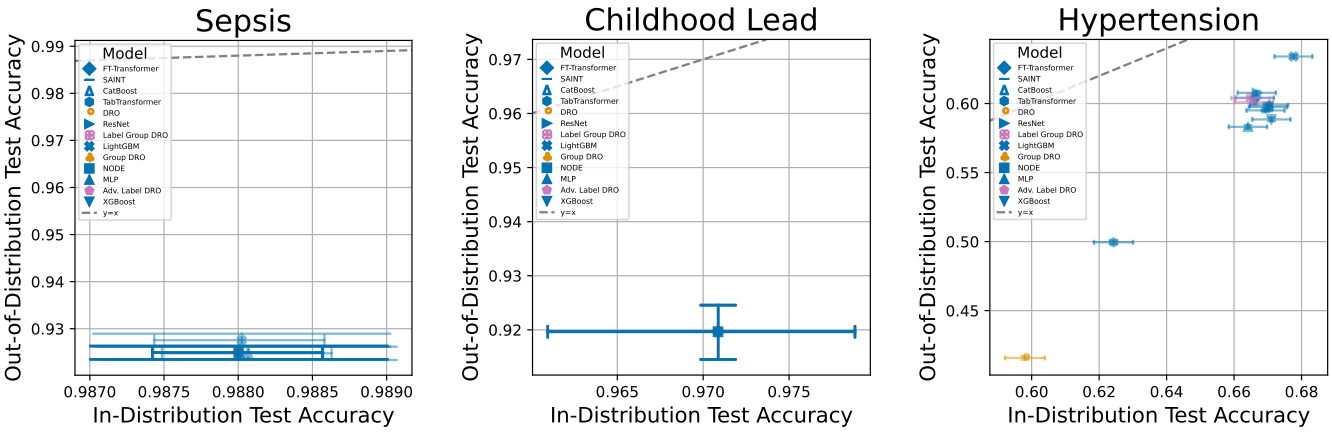


Figure B.6: Alternate version of Figure 3.3 with adjusted scaling for increased detail.

Table B.11: Sample sizes by split. In particular, large *test* sizes are desirable for benchmarking, as they reduce the statistical uncertainty of comparing model performance.

Task	ID Test	OOD Test	OOD Validation	Train	Validation	Total
Food Stamps	78,628	48,878	5431	629,018	78,627	840,582
Income	158,016	75,911	8435	1,264,123	158,015	1,664,500
Public Coverage	500,782	817,877	90,876	4,006,249	500,781	5,916,565
Unemployment	161,365	163,611	18,180	1,290,914	161,364	1,795,434
Voting	520	2772	309	4159	520	8280
Hypertension	27,052	518,622	57,625	216,411	27,051	846,761
Diabetes	121,154	209,375	23,264	969,229	121,154	1,444,176
Readmission	4287	50,968	5664	34,288	4286	99,493
HELOC	278	6914	769	2220	278	10,459
ICU Length of Stay	1080	11,835	1316	8634	1079	23,944
ICU Hospital Mortality	890	13,544	1505	7116	889	23,944
Sepsis	140,288	134,402	14,934	1,122,299	140,287	1,552,210
Childhood Lead	1476	11,466	1274	11,807	1476	27,499
ASSISTments	266,566	1906	212	2,132,526	266,566	2,667,776
College Scorecard	12,320	1352	151	98,556	12,320	124,699

Table B.12: Label summary statistics for test sets in TableShift. $\bar{Y}$ gives the proportion of positive labels for the respective split, and $\mathrm{Var}(\bar{Y})$ the variance of the sample proportion.

	ID		OOD	
Task	$\bar{Y}_{\mathrm{ID}}$	$\mathrm{Var}(\bar{Y}_{\mathrm{ID}})$	$\bar{Y}_{\mathrm{OOD}}$	$\mathrm{Var}(\bar{Y}_{\mathrm{OOD}})$
Voting	0.804	0.017	0.754	0.008
ASSISTments	0.695	0.001	0.437	0.011
Childhood Lead	0.029	0.004	0.080	0.003
College Scorecard	0.127	0.003	0.311	0.013
Diabetes	0.127	0.001	0.174	0.001
FICO HELOC	0.255	0.026	0.569	0.006
Food Stamps	0.191	0.001	0.220	0.002
Hospital Readmission	0.416	0.008	0.494	0.002
Hypertension	0.402	0.003	0.584	0.001
ICU Hospital Mortality	0.085	0.009	0.124	0.003
ICU Length of Stay	0.398	0.015	0.456	0.005
Income	0.321	0.001	0.398	0.002
Public Health Ins.	0.224	0.001	0.636	0.001
Sepsis	0.012	0.000	0.075	0.001
Unemployment	0.034	0.000	0.052	0.001

Table B.13: PMA-OOD results (cf. Figure) and standard deviation of PMA-OOD over benchmark tasks. (Cf. Figure 3.4a.)

Estimator	**PMA-OOD Mean**	**PMA-OOD Std.**
VREX	0.876	0.076
IRM	0.910	0.069
Label Group DRO	0.915	0.129
MixUp	0.917	0.076
TabTransformer	0.922	0.091
DANN	0.932	0.075
DRO	0.932	0.107
MMD	0.940	0.059
CORAL	0.944	0.059
Adv. Label DRO	0.950	0.080
ResNet	0.956	0.069
MLP	0.961	0.054
Group DRO	0.962	0.059
NODE	0.963	0.066
SAINT	0.964	0.070
LightGBM	0.964	0.070
XGBoost	0.964	0.065
FT-Transformer	0.968	0.069
CatBoost	0.994	0.014

Table B.14: Label summary statistics for test sets in TableShift. $\bar{Y}$ gives the proportion of positive labels for the respective split, and $\mathrm{Var}(\bar{Y})$ the variance of the sample proportion. (Cf. Figure 3.4b.)

	ID		**OOD**	
Method	ID Accuracy	Std.	OOD Accuracy	Std.
Adv. Label DRO	0.834	0.125	0.792	0.132
CatBoost	0.862	0.105	0.794	0.126
DANN	0.816	0.137	0.770	0.148
CORAL	0.824	0.129	0.777	0.134
DRO	0.843	0.129	0.773	0.147
FT-Transformer	0.866	0.103	0.794	0.126
Group DRO	0.832	0.129	0.791	0.133
IRM	0.800	0.130	0.749	0.136
Label Group DRO	0.829	0.123	0.759	0.134
LightGBM	0.856	0.108	0.781	0.124
MixUp	0.807	0.125	0.752	0.118
MLP	0.845	0.126	0.774	0.141
MMD	0.825	0.130	0.774	0.135
NODE	0.853	0.110	0.781	0.129
ResNet	0.844	0.126	0.773	0.139
SAINT	0.868	0.099	0.787	0.127
TabTransformer	0.835	0.136	0.759	0.160
VREX	0.786	0.127	0.720	0.128
XGBoost	0.857	0.105	0.783	0.122

B.5.4 Results with Additional Random Seeds

Our experiments on each model-dataset pair comprise a single run of 100 rounds of our hyperparameter tuning protocol described in Section 3.4.2. Here, we provide the results of additional experiments conducted using different random seeds, in order to evaluate the sensitivity of our results to the random variation inherent in the training and hyperparameter tuning process.

For these experiments, we conduct an identical procedure to the experiments described in the main text of our paper, but only change the random seed. This process affects the random initialization of model weights, random initialization of hyperparameter tuning, and training data shuffling, among other procedures. We note that it does *not* affect the train/test splitting in our datasets, as the train/test splits are defined by distribution shifts and are fixed to ensure comparability of the benchmark across experiments.

The results are shown in Table B.15. Table B.15 shows that, across the five models and three datasets evaluated, there is minimal variation in performance due to random seeds. Of the 90 measurements covering 45 trials represented in Table B.15, the 95% Clopper-Pearson CIs for both ID and OOD accuracy overlap in all cases, with only four exceptions (LightGBM, iteration 0, Food Stamps ID and OOD accuracy; LightGBM, iteration 2, Hypertension OOD accuracy; MLP, Hypertension, iteration 0, OOD accuracy; FT-Transformer, iteration 0, OOD accuracy). These results provide evidence that our results are robust to variation due to random seed.

B.5.5 Results with Hybrid Methods

Our main study design is focused on benchmarking existing previously-proposed methods for tabular modeling. The methods we evlauate span *models*, which prescribe the functional form of a predictor f_θ, and also *objective functions*, which describe the loss $\mathcal{L}$ to be minimized while learning the parameters θ of a fixed predictor f. Concretely, for example, FT-Transformer or MLP specify the form of f, while some robustness interventions, such as Group DRO, specify an objective that can be monimized over any smooth continuous function.

Our study does not explore potential combinations of different models and objective functions from the preexisting literature. In this section, we conduct an exploratory investigation into whether "hybrid methods" – combinations of different models and objective functions explored in our study – might improve robustness,

for the best-performing compatible combinations of models and objective functions in our study.

In particular, our hybrid model study explores the use of the Group DRO objective function, in combination with three models from our study: FT-Transformer, NODE, and ResNet. Group DRO was selected as it is the highest-performing objective-based technique in our study (see Figure 3.4a), and the three models were selected as they are the highest-performing Transformer-based model, tree-based model, and baseline supervised model, repsectively, in our study. We note that Group DRO cannot be easily combined with CatBoost, XGBoost, or LightGBM, as these are not smooth differentiable continuous functions, which is a requirement for the use of the Group DRO objective.

Our methodology in this section is as follows: for each estimator (FT-Transformer, NODE, ResNet), we train the model with both ERM (the standard procedure used in our main experiments above) and Group DRO. We follow the same hyperparameter tuning procedure as described in Section 3.4.2) above. We use the same hyperparameter grid defined in Section B.9 for each model, but also include a full sweep over the Group DRO step size parameter, using the Group DRO grid described in Section B.9 (thus, for model X, we take the union of the two hyperparameter grids: { grid(X) $\cup$ grid(Group DRO) }). We conduct this procedure for five benchmark datasets: Childhood Lead, College Scorecard, Food Stamps, Hypertension, and Voting.

The results of our hybrid model experiments are shown in Table B.16. The results show little or no evidence that Group DRO reduces shift gaps for the models evaluated, as indicated by the fact that OOD test accuracy intervals tend to be overlapping, or *higher*, for ERM relative to Group DRO. Keeping in mind that Group DRO was parameterized over MLP models in our main experiments (as all prior works only use Group DRO with MLP), the results in Table B.16 suggest that Group DRO may primarily improve weak (MLP) models but does not improve robustness for stronger models, explaining the improvements for Group DRO over vanilla MLP models in the main text.

Table B.15: Results with additional random seeds. Varying random seeds has a minimal impact on the final results of our hyperparameter tuning procedure, indicating that our findings are robust to variation due to random seeds. See Section B.5.4 for details on experimental design.

			ID Test Accuracy		OOD Test Accuracy	
Task	**Base Estimator**	**Iteration**	**Value**	**95% CI**	**Value**	**95% CI**
College Scorecard	CatBoost	0	0.957	(0.954, 0.961)	0.885	(0.866, 0.901)
		1	0.959	(0.955, 0.962)	0.879	(0.861, 0.896)
		2	0.959	(0.956, 0.963)	0.882	(0.863, 0.898)
	FT-Transformer	0	0.948	(0.944, 0.952)	0.859	(0.839, 0.877)
		1	0.946	(0.942, 0.95)	0.850	(0.83, 0.868)
		2	0.940	(0.936, 0.945)	0.830	(0.809, 0.85)
	LightGBM	0	0.939	(0.935, 0.943)	0.822	(0.8, 0.841)
		1	0.943	(0.938, 0.947)	0.839	(0.819, 0.859)
		2	0.943	(0.939, 0.947)	0.837	(0.816, 0.856)
	MLP	0	0.947	(0.942, 0.95)	0.845	(0.825, 0.864)
		1	0.949	(0.944, 0.952)	0.859	(0.84, 0.878)
		2	0.945	(0.941, 0.949)	0.859	(0.839, 0.877)
	XGBoost	0	0.942	(0.938, 0.946)	0.830	(0.809, 0.85)
		1	0.946	(0.942, 0.95)	0.842	(0.821, 0.861)
		2	0.947	(0.943, 0.951)	0.845	(0.824, 0.864)
Food Stamps	CatBoost	0	0.849	(0.847, 0.852)	0.825	(0.821, 0.828)
		1	0.850	(0.847, 0.852)	0.824	(0.821, 0.827)
		2	0.849	(0.847, 0.852)	0.824	(0.82, 0.827)
	FT-Transformer	0	0.843	(0.841, 0.846)	0.816	(0.812, 0.819)
		1	0.848	(0.846, 0.851)	0.824	(0.82, 0.827)
		2	0.844	(0.842, 0.847)	0.817	(0.814, 0.82)
	LightGBM	0	0.836	(0.833, 0.838)	0.808	(0.805, 0.812)
		1	0.844	(0.841, 0.846)	0.818	(0.814, 0.821)
		2	0.843	(0.84, 0.846)	0.817	(0.814, 0.821)
		0	0.841	(0.838, 0.844)	0.815	(0.812, 0.819)

Table B.16: Hybrid method results. We compare Group DRO to standard ERM for the highest-performing Transformer, tree-based, and baseline models in our study (FT-Transformer, NODE, and ResNet, respectively) over five TableShift tasks, following our hyperparameter tuning procedure. There is little or no evidence that Group DRO reduces shift gaps for these models, indicating that Group DRO may primarily improve weak (MLP) models but does not improve robustness for stronger models. See Section B.5.5 for details on experimental design.

			ID Test Accuracy		OOD Test Accuracy	
Task	**Base Estimator**	**Method**	**Value**	**95% CI**	**Value**	**95% CI**
Childhood Lead	FT-Transformer	ERM	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
		Group DRO	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
	NODE	ERM	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
		Group DRO	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
	ResNet	ERM	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
		Group DRO	0.971	(0.961, 0.979)	0.920	(0.915, 0.925)
College Scorecard	FT-Transformer	ERM	0.948	(0.944, 0.952)	0.859	(0.839, 0.877)
		Group DRO	0.935	(0.93, 0.939)	0.815	(0.793, 0.835)
	NODE	ERM	0.944	(0.939, 0.948)	0.844	(0.823, 0.863)
		Group DRO	0.946	(0.942, 0.95)	0.835	(0.814, 0.854)
	ResNet	ERM	0.947	(0.943, 0.951)	0.854	(0.834, 0.872)
		Group DRO	0.947	(0.942, 0.95)	0.824	(0.803, 0.844)
Food Stamps	FT-Transformer	ERM	0.843	(0.841, 0.846)	0.816	(0.812, 0.819)
		Group DRO	0.826	(0.823, 0.829)	0.795	(0.792, 0.799)
	NODE	ERM	0.849	(0.847, 0.852)	0.822	(0.819, 0.825)
		Group DRO	0.845	(0.842, 0.847)	0.822	(0.819, 0.825)
	ResNet	ERM	0.843	(0.84, 0.845)	0.820	(0.817, 0.824)
		Group DRO	0.848	(0.846, 0.851)	0.818	(0.815, 0.822)
Hypertension	FT-Transformer	ERM	0.666	(0.661, 0.672)	0.604	(0.603, 0.605)
		Group DRO	0.665	(0.659, 0.67)	0.608	(0.607, 0.609)
	NODE	ERM	0.670	(0.664, 0.676)	0.597	(0.596, 0.599)

B.6 Model Details

This section describes the models used in our study. For the hyperparameters used in our experiments, see Section B.9.

Our implementations of these models, along with associated code to train models with fixed hyperparameters or to tune hyperparameters at scale via the Ray framework, are available at https://github.com/mlfoundations/tableshift.

B.6.1 Baseline Models

XGBoost: XGBoost is a popular library for learning gradient-boosted trees. We use the original XGBoost implementation [38]. XGBoost introduced column subsampling, weight regularization, and introduced major improvements in efficiency for training gradient boosted models on large or out-of-core datasets.

LightGBM: LightGBM is a library for learning gradient-boosted trees which extends the success of XGBoost in working fast and with large datasets [102]. LightGBM introduces novel techniques such as converting continuous features to histograms (for computational efficiency and for to reduce overfitting), combining certain features using Exclusive Feature Bundling (EFB), and through the use of Gradient-based One-Side Sampling (GOSS).

CatBoost: CatBoost [52] is a library for learning gradient-boosted trees which includes novel techniques for leveraging categorical features. This includes heuristics to replace numeric or one-hot encoding of categorical features with label-derived heuristics; "appearance" (count) features for categorical features; and efficient greedy feature recombination techniques.

MLP: We use standard multilayer perceptrons, via the implementation in `RTDL`[26]. MLPs have been shown to be highly effective models for tabuilar data, particularly when a large model search space is used and regularization is carefully tuned [98].

[26] https://github.com/Yura52/rtdl

B.6.2 Tabular Neural Networks

FT-Transformer: FT-Transformer is a transformer-based model that learns separate feature tokenizers for numeric and categorical data, and applies a transformer model [187] to the tokenized features.

Tabular ResNet: We use the version of Tabular ResNet proposed in [73]. We note that, despite the fact that this approach is shown to have competitive performance with many existing tabular data models in [73], it has not been widely used in the literature.

NODE Neural Oblivious Decision Ensembles (NODE) [140] is a method that leverages oblivious ensembling methods to train "tree-like" neural networks.

TabTransformer: TabTransformers [92] is a model that uses learned embeddings of categorical features, which are then passed through standard Transformer layers, alongside layer normalization of continuous features.

SAINT: SAINT [173] uses an enhanced embedding method for categorical features, alongside (optional) attention over both rows and columns, in a Transformer architecture. We note that, due to its use of featurewise feedforward layers, SAINT was impractical to use for our datasets with the largest numbers of features (ICU Hospital Mortality, ICU Length of Stay; both contain over 1000 features which resulted in over 13B parameters for even a single-layer SAINT model).

B.6.3 Robustness Models

Distributionally Robust Optimization (DRO): We use two variants of DRO, both via [113]. For both methods, the model attempts to optimize a worst-care risk within a bounded distribution of the training data via a projected gradient descent procedure.

Group DRO: Originally introduced as a subgroup robustness method in [159], Group DRO is a DRO method which attempts to optimize the worst-group loss during training. Group DRO can also, however, be used as a domain robustness method by treating the domains as "group labels", which is how we use it in our study. We note that this use of Group DRO has been applied previously; e.g. [76].

B.6.4 Domain Generalization Models

Invariant Risk Minimization (IRM): IRM [9] uses a modified training objective to learn models which a feature representation such that the optimal linear classifier on top of that representation matches across domains.

MixUp: Inter-Domain MixUp [199, 198] uses combinations of data points from random pairs of domains and their labels during training.

Domain-Adversarial Neural Networks (DANN): DANN [2] uses adversarial training to achieve domain robustness, where a discriminator attempts to predict the domain of a training example in order to match feature distributions across domains.

Risk Extrapolation (REx): REx [112] attempts to reduce differences in risk across training domains, in order to reduce a model's sensitivity to distributional shifts.

CORAL: CORAL (CORrelation ALignment) [176] attempts to ensure that feature activations are similar across domains; this can be used as either a domain generalization method or a domain adaptation method.

MMD: Similar to CORAL using a different kernel, MMD attempts to minimize the Maximum Mean Discrepancy (MMD) between domains.

B.6.5 Label Shift Robustness Models

Group DRO: here, we use Group DRO [159] with class labels as the grouping attribute.

Adversarial Label DRO: This method, proposed in [207], uses a distributionally robust objective to optimize for the worst-case weighting over label groupings. We note that this approach is computationally expensive, requiring sample-level gradients even following the authors' original implementation, and so was not practical for our datasets with very large n (ASSISTments, Public Health Insurance).

B.7 Comparison To Other Benchmarking Toolkits

In this section, we provide a brief comparison of TableShift to other relevant benchmarking toolkits. We note that our goal in this section is *not* to fully characterize the functionality of other benchmarking platforms; it is only to compare and contrast their relevant attributes with TableShift and to motivate the creation of a

novel benchmark and API for TableShift (as opposed to incorporating TableShift into an existing toolkit).

As noted above, there is *no* existing benchmark for domain shift in tabular data. However, in this section we compare to three main categories of relevant related toolkits: (1) domain shift benchmarks for non-tabular data (DomainBed, WILDS); (2) IID (non-domain-shift) benchmarks for tabular data ([74], OpenML); and (3) generic data-hosting platforms (Huggingface Datasets, TensorFlow Datasets. We briefly introduce and compare to each of these below.

B.7.1 Domain Shift Benchmarks for Non-Tabular Data

WILDS: WILDS[27] is perhaps the closest benchmark to TableShift, but only uses non-tabular data. WILDS demonstrates a lightweight, useful set of programming abstractions for benchmarking models and sharing results across a diverse set of datasets for domain shift. WILDS interfaces with image and text datasets, and includes a rich variety of datasets with real-world sensitive attributes, carefully selected domain shifts, and has wide adoption in the robustness community. WILDS includes a high-quality Python API, which has led to wide integration with researchers' open-source code and widespread adoption. However, WILDS is currently not compatible with tabular datasets and does not include any tabular datasets in its benchmark suite. The needs for tabular datasets are different than the datasets currently used in WILDS (i.e. non-Torch models must be supported by the benchmark; subgroup and domain-shift information are handled differently for our use cases; data preprocessing is also different for tabular data as noted above).

DomainBed: DomainBed[28] is a benchmark that contains several reference implementations, including some that have been adapted for use in TableShift. In addition to these model implementations, DomainBed serves as an interface to several existing datasets through a Python API. However, DomainBed is specifically adapted to image data. It only supports image datasets with a specific folder structure (which would make extending to tabular datasets nontrivial) and includes many image-specific augmentation components of its pipelines. It also uses ResNet50/ResNet18 networks designed specifically for image classification, and therefore does not currently support either deep learning models suited to image data, nor (more importantly) the effective non-DL baselines described above such as XGBoost and LightGBM.

[27] See https://wilds.stanford.edu/ and [108].

[28] See https://github.com/facebookresearch/DomainBed and [76].

Shift Happens: The "shift happens" benchmark[29] is a community-built benchmark suite for image models. It specifically aims to feature datasets with domain shift, for tasks including image classification under domain shift, and out-of-distribution detection. The benchmark includes a Python API. This benchmark does not support tabular datasets, and is much less widely used, perhaps due to the community-driven effort (as opposed to benchmarks such as WILDS and DomainBed, which come packaged with preselected datasets and domain splits).

Shifts 2.0: Shifts ([119], recently upgraded to Shifts 2.0 [119]) is a collection of multimodal tasks with domain shifts. The Shifts benchmark is a part of the Shifts Project, an international collaboration of academic and industrial researchers dedicated to studying distributional shift.[30] Shifts 2.0, the current version, includes five tasks: tabular weather prediction, tabular marine cargo vessel power consumption prediction, machine translation, self-driving car vehicle motion prediction, and segmentation of white matter Multiple Sclerosis lesions in 3D magnetic resonance brain images. While shifts does contain two tabular data tasks, its relatively small number of tasks makes it a less reliable benchmark compared to the rich set of tabular datasets comprising TableShift. Shifts also does not include any tasks with real-world sensitive subgroups (such as age, race, or gender) which are of particular interest in many tabular classification tasks. Additionally, the domains represented in the tabular tasks of Shifts do not cover many critical domains widely recognized as using tabular data (e.g. finance, medicine, etc.; see Section B.4.2).

B.7.2 IID Benchmarks for Tabular Data

Benchmark of [74]: The unnamed benchmark proposed in this work is intended to provide a consistent benchmarking suite for tabular dataset classification tasks, and was motivated by some of the same gaps described in this work. However, the datasets comprising the benchmark of [74] do not meet our specifications, for several reasons. First, the datasets are limited to be of *maximum* size $10k$ observations; this is too small for reliable and repeated benchmarking comparisons. Additionally, the datasets are label-balanced; in contrast, we use the naturaly-occurring label distributions for all datasets (and we show that these label distributions are importantly related to shift gap). Most critically, the benchmark datasets in [74] do not contain domain shifts; for most or all of the datasets, it is does not appear that a domain shift could be

[29] https://shift-happens-benchmark.github.io/index.html

[30] https://shifts.ai/

induced from splitting the existing data on an existing feature.

OpenML: OpenML has some overlap with the proposed functionality. However, OpenML both lacks functionality we seek to provide (subgroup robustness and domain shift utilities; a curated set of benchmarking datasets; lightweight and standardized control over common tabular preprocessing methods) and also provides extraneous functionality not needed for a lightweight tabular benchmarking library (tools for OpenML-hosted model/pipeline/evaluation sharing and collaboration; API/utilities for model training) . Additionally, OpenML is not yet widely used in the tabular data community, as demonstrated by the wide calls for effective tabular data benchmarking tools ([28], [74], [170]) and the lack of usage of OpenML in most robustness works, even recent works (e.g. [204]), which largely focus on canonical tabular datasets such as COMPAS and Adult.

DataPerf: DataPerf is "a benchmark package for evaluating ML datasets and dataset working algorithms" [121]. Similar to WILDS and DomainBed, DataPerf covers many domains, not only tabular data. DataPerf has a much broader set of goals relative to TableShift, and includes a collection of tasks, metrics and rules that are intended to benchmark all stages of an ML pipeline, from raw data to test set selection amd model selection. However, DataPerf does *not* offer domain shifts, and while it is possible domain shifts could be integrated into DataPerf, it does not natively support the kind of benchmarking that we intend to support with TableShift.

B.7.3 Other Data Hosting Platforms

Hugging Face Datasets (HFDS): HFDS is a generic dataset hosting utility provided by the company Hugging Face. It serves as a large, open dataset repository; however, these datasets are not curated for size, featurization, or quality. HFDS is a public platform where datasets can be contributed openly. However, of the tabular datasets on HFDS, few if any meet the specifications described in §3.3.1; in particular, most are not domain shift datasets.

TensorFlow Datasets (TFDS): TFDS is similar in many ways to HFDS. It is a public, open repository of datasets, and new datasets can be contributed via git. However, TFDS also has the same shortcomings as HFDS; in particular, its open format leads to a collection of datasets mostly not useful for tabular data benchmarking and almost no datasets with meaningful distribution shifts.

Table B.17: Comparison of relevant benchmarks. DB: DomainBed. OML: OpenML. GOV22: [74]. SH: Shift Happens. HFDS: Hugging Face Datasets. TFDS: TensorFlow Datasets.

		DB	OML	GOV22	WILDS	SH	Shifts 2.0	HFDS	TFDS	TableShift
Output	Supports tabular data input formats (e.g. .csv)		✓				✓	✓	✓	✓
	Supports tabular output formats (e.g. `pd.DataFrame`)						✓	✓	✓	✓
	Support for large/out-of-core datasets	✓			✓	✓		✓	✓	✓
Preprocess	Supports tabular preprocessing: categorical encoding; missing value handling									✓
	Provides shared utilities for user-defined preprocessing per dataset	✓			✓	✓		✓	✓	✓
Metadata	Feature-level metadata									✓
	Dataset-level metadata							✓	✓	✓
Benchmark Tasks	Includes domain shift (non-IID) splits	✓			✓	✓	✓	some		✓
	Meets criteria in §3.3.1	✓			✓					✓
	Large test sets ($\geq 10k$)	✓				✓				
	Includes label-imbalanced	✓	✓		✓	✓	✓	✓	✓	✓

B.8 Training Details

Neural network-based models were trained on GPU, either NVIDIA RTX 2080 Ti GPUs with 11GB of RAM, or NVIDIA Tesla M60 GPUs with 48 GB of RAM. We used a batch size of 4096 for training all models, except where this was not possible due to memory limitations (see code for details).

Where possible, gradient boosted tree models were trained using CPU (not GPU).

B.9 Hyperparameter Grids

The hyperparameter tuning grid for each model is shown in Table B.18. We make the full hyperparameter tuning code available as part of the release of this work, at `https://github.com/mlfoundations/tableshift`.

We made an effort to ensure our hyperparameter grids always included at least the full grid described in the original work(s) cited for each learning method used in our study. For some methods, our grid is a superset of the hyperparameter grid in the original study. This is to ensure, where possible, that we tune a similar range of certain parameters (i.e. learning rate) across all methods. For domain generalization methods, since we are not aware of any prior application to these methods to tabular data, we use the hyperparameter grids from [76].

Table B.18: Hyperparameter grids used in all experiments. ♣: all MLP parameters also tuned. Continued in Table B.19.

Model	Hyperparameter	Values
Baseline Methods		
♣ MLP	Learning Rate	LogUniform$(1e^{-5}, 1e^{-1})$
	Weight Decay	LogUniform$(1e^{-6}, 1)$
	Num. Layers	$\{1, 2, \ldots, 8\}$
	Hidden Units	$\{64, 128, 256, 512, 1024\}$
	Num Epochs	QRandInt$(5, 100, 5)$
	Dropout	Unif$(0, 0.5)$
	Batch Size	$\{4096\}$
XGBoost	Learning Rate	LogUniform$\{1e-5, 1\}$
	Max. Depth	$\{3, \ldots, 10\}$
	Min Child Weight	LogUniform$\{1e-8, 1e5\}$
	Row Subsample	Uniform$\{0.5, 1.\}$
	Column Subsample (Tree)	Uniform$\{0.5, 1.\}$
	Column Subsample (Level)	Uniform$\{0.5, 1.\}$
	γ	LogUniform$\{1e-8, 1e2\}$
	λ	LogUniform$\{1e-8, 1e2\}$
	α	LogUniform$\{1e-8, 1e2\}$
	Max. Bins	$\{128, 256, 512\}$
LightGBM	Learning Rate	LogUniform$\{1e-5, 1\}$
	Min. Child Samples	$\{1, 2, 4, 8, 16, 32, 64\}$
	Min. Child Weight	LogUniform $(1e-8, 1e5)$
	Row Subsample	Uniform$\{0.5, 1.\}$
	Max. Depth	$\{\text{None}, 1, 2, \ldots, 31\}$
	Column Subsample (Tree)	Uniform$\{0.5, 1.\}$
	Column Subsample (Level)	Uniform$\{0.5, 1.\}$
	λ	LogUniform$\{1e-8, 1e2\}$
	α	LogUniform$\{1e-8, 1e2\}$
CatBoost	Learning Rate	LogUniform$\{1e-3, 1\}$
	Depth	QRandInt$\{3, 10\}$
	Bagging Temp.	LogUniform $(1e-6, 1)$
	L_2 Leaf Reg.	LogUniform $(1, 100)$
	Leaf Estimation Iterations	QRandInt$\{1, 10\}$
Domain Generalization Methods		
	LR_G	LogUniform$\{1e-5, 1e-1\}$

Model	Hyperparameter	Values
Tabular Neural Networks		
ResNet ♣	Num. Blocks	$\{1, 2, 3, 4\}$
	d main	RandInt$(64, 1024)$
	Hidden Factor	RandInt$(1, 4)$
	Dropout First	Uniform$(0, 0.5)$
	Dropout Second	Uniform$(0, 0.5)$
FT-Transformer ♣	Num. Blocks	$\{1, 2, 3, 4\}$
	Residual Dropout	Uniform$(0, 0.2)$
	Attention Dropout	Uniform$(0, 0.5)$
	FFN Dropout	Uniform$(0, 0.5)$
	FFN Factor	Uniform$(2/3, 8/3)$
	FFN Factor	$\{64, 128, 256, 512\}$
TabTransformer	Num. Blocks	$\{1, 2, 3, 4\}$
	Learning Rate	LogUniform$(1e^{-5}, 1e^{-1})$
	Weight Decay	LogUniform$(1e^{-6}, 1)$
	Num Epochs	QRandInt$(5, 100, 5)$
	FFN Dropout	Uniform$(0, 0.5)$
	Attention Dropout	Uniform$(0, 0.5)$
	Model Dimension	$32, 64, 128, 256$
	Depth	$3, 4, 5, 6$
	Num. Heads	$2, 4, 8$
NODE	Num. Epochs	$\{1, 2, 3, 4, 5\}$
	Num. Layers	$2, 4, 8$
	Total Tree Count	$1024, 2048$
	Tree Depth	$6, 8$
	Tree Output Dim.	$2, 3$
	FFN Factor	$\{64, 128, 256, 512\}$
	Learning Rate	LogUniform$(1e^{-5}, 1e^{-1})$
	Weight Decay	LogUniform$(1e^{-6}, 1)$
SAINT	Num. Epochs	$\{1, 2, 3, 4, 5\}$
	Depth	$4, 6$
	Model Dimension	$4, 8, 12, 16, 32$
	Learning Rate	LogUniform$(1e^{-5}, 1e^{-1})$
	Weight Decay	LogUniform$(1e^{-6}, 1)$
	FFN Dropout	Uniform$(0.1, 0.8)$
	Heads	$4, 8$
	Attention Type	Row, Col, RowCol
Domain Robustness Methods		

Appendix C

LARGE-SCALE TRANSFER LEARNING FOR TABULAR DATA VIA LANGUAGE MODELING

C.1 T4 Data Filtering Details

As discussed previously, in order to generate T4, we filter TabLib [58] at the table, column and row level. Having filtered down the tables, we then programatically select a target column for prediction according to another set of heuristics. The remaining columns are used as features, but not as prediction targets. We select only a single target column for each table.

C.1.1 Table, Column, and Row Filtering Rules

The following tables describe the entire set of heuristics we use as filters. A precise implementation may be found as a part of our software release associated with this paper. Our pipeline involves a langauge identification step; for language detection, we make use of the `fasttext` library [96].[1]

List of Table Filtering Rules

[1] https://pypi.org/project/fasttext-langdetect/

Level	Name	Description	Motivation / Hypothesis
Table	English Filtering	Drop where a langauge ID model score is below a fixed threshold	All downstream benchmark datasets are in English
Table	Schema Heterogeneity	Drop tables where every cell is of the same type	Encourages understanding of mixed data types
Table	Row Count	Drop table with fews than k rows	Anecdotally, many "very small" tables in TabLib are general web-text tables not useful/suitable for ML.
Table	Column Count	Drop tables with fewer than k columns *after column filters are applied*	Exclude tables that lack a reasonable amount of features
Table	Parse Error	Drop tables where the headers suggest there was a parsing error.	These tables are likely the result of bad table detection, and they almost definitely contain low-quality headers.
Table	Drop PII	Drop table where $> x\%$ of the cells match a regex for phone number or email	Don't want to train on PII for privacy reasons. Also not likely to be present in downstream tasks.
Table	Drop Code	Drop table with any cell that has probability $> p$ of containing code.	Lots of the data in TabLib is from Github and other technical documentation. Code is common. Code also confuses the model a lot, apparently due to special characters and whitespace. This code can be unevenly broken/spread across cells due to the tablib parser.
Table	Too many unnamed columns	Drop table if the fraction of "Unnamed: " columns is greater than a threshold.	Discard low-quality data; unnamed columns tend to be of significantly lower quality based on manual data inspection.

List of Row and Column Filtering Rules

Level	Name	Description	Motivation / Hypothesis
Column	Drop Free-Text	Drop columns with long headers (> 256 characters)	The TabLib process used to scrape the tables can result in tables with "headers" that are actually just rows of data. One indicator of this is very long headers (e.g. a text column that ends up as a header).
Column	Drop Numeric	Drop columns with names that are numeric.	TabLib's parsing removed tables with all-numeric headers "for most file formats". This means that (a) some formats were missed, and (b) tables with many numeric headers and even one non-numeric header were still included.
Column	Drop Missing	Drop any column with $> x\%$ values that are None, NaN, whitespace or empty string values	Columns that are mostly missing will waste compute processing headers; empty cells usually won't be informative (although sometimes a header alone can be useful).
Column	English Filtering	Drop any text columns where average probability of English over rows is less than p	Some tables contain English headers and non-English data. All of our downstream data is English. It's hard to assess quality of non-English data.
Column	Drop Constant	Drop columns where all values are the same.	Constant features are not useful for prediction
Row	Drop Missing	Drop any row with too many values that are None, NaN, whitespace or empty string values	Rows with mostly missing data are likely to be uninformative.
Row	Drop Duplicates	Drop duplicate rows	This is non-standard in downstream tasks
Row	Drop PII	Drop any row where PII is de-	Tables with small numbers of rows contain-

List of Target Column Selection Rules

Name	**Description**	**Motivation / Hypothesis**
Drop All Unique	Drop non-numeric columns if all values all distinct	This is probably not a classification target (most likely a unique identifier, a date, a number, etc.)
Drop "Unnamed:"	Drop any column whose name starts with "Unnamed:"	"Unnamed:" is a special prefix used for unnamed columns in an Arrow table; we avoid making predictions on columns where there is no clear semantic information about the prediction target.
Drop dates	Drop any column with any date or time data type	Not useful as classification targets in most cases
Drop Too Long	Drop any column which, when serialized, is greater than 256 characters	This helps avoid choosing free-text columns as targets.
Drop dates	Drop any column with any date or time data type	Not useful as classification targets in most cases

C.1.2 Target Column Selection

Our procedure for target column selection is based on the rules described in the above tables. Given a table, we consider all columns to be potentially valid targets unless they do not satisfy one of the listed criteria.

Once target candidates are identified for a table, we choose a single candidate at random and use this as the prediction target. When there are both continuous and categorical candidates present, we choose a categorical candidate with probability $p = 0.9$ and a continuous candidate with probability $1 - p$. This decision reflects our qualitative observation that our selection method tends to produce higher-quality categorical columns than numeric columns, and has the effect of showing the model classification tasks more often than (binned) regression tasks during training.

In the case where we select a continuous target, we discretize it into a discrete set by selecting a number

of quantiles uniformly at random over the interval $[3, 8]$, and then discretizing the target value into these columns. We serialize the resulting quantiles as "less than 1," "between 1 and 2.5", "greater than 2.5" etc.

Even after filtering, the tables in our pool contain columns which may not be meaningful prediction *targets* (for example, timestamps or UIDs). For a given table, we propose and apply a series of heuristics for identifying suitable candidates for target columns. These heuristics include excluding columns if: the name is numeric; only one unique value; has unique values for every row (excluding numeric columns); any row has a value longer than 256 characters; the column is of a date or timestamp type. We provide a detailed list of the exact target selection rules in Section C.1.2, and an implementation in our code. We do *not* drop such columns from the table; columns not meeting these criteria are simply kept as predictors but will never be used as prediction targets.

Once target candidates are identified for a table, we choose a single candidate at random and use this as the prediction target. When there are both continuous and categorical candidates present, we choose a categorical candidate with probability $p = 0.9$ and a continuous candidate with probability $1 - p$. This decision reflects our qualitative observation that our selection method tends to produce higher-quality categorical columns than numeric columns, and has the effect of showing the model classification tasks more often than (binned) regression tasks during training.

C.1.3 Implementation Details

Our data processing implementation uses Ray Datasets[2] to process tables in parallel. Our pipeline utilizes TabLib's existing `content_hash` feature to read only a set of unique tables; this avoids performing an expensive deduplication step during online processing. The shards of TabLib are processed in chunks to avoid extra overhead due to very large collections in Ray. Our pipeline includes certain additional optimizations: for example, in TabLib, data is stored as serialized Arrow bytes (and the deserialized bytes cannot easily be passed between stages of the Ray pipeline); as a result, our pipeline avoids repeatedly deserializing these bytes and only does so at a single filtering step, and at the write step.

Our preprocessing implementation is provided as a separate library, `tabliblib`, along with the release of this paper and in the supplementary material.

[2] https://docs.ray.io/en/latest/data/overview.html

C.2 Training Details

TabuLa-8B is trained for 40 thousand steps with a global batch size of 24. We use a peak learning rate of $1e-5$ warmed up over 10% of the total training to the peak learning rate, and then use a cosine decay schedule to zero. We do not use weight decay. We found that our model was quite sensitive to the selection of the learning rate, and that evaluation loss during training correlated strongly with performance on downstream tasks.

C.3 Description of Compute Resources Used

Our final training run for TabuLa-8B took approximately 6 days on a single node of 8 NVIDIA 80GB A100 GPUs on a commercial cloud provider. For our TabLib filtering and XGboost experiments, we used an academic CPU cluster. Our evaluations were distributed across two academic GPU clusters consisting of NVIDIA 40GB A40 GPUs and NVIDIA A100 GPUs. As a rule of thumb, evaluating the model on a single dataset over a grid of 8 values for k (number of shots) consumes around 4 GPU-hours. We estimate that the total number of GPU hours used across training, evaluation, and development is approximately $5,000 - 10,000$.

C.4 Evaluation Details

In this section we provide more detail on the benchmarks datasets in our evaluation suite. We do not preprocess the datasets (no normalization, one-hot encoding, etc), as many datasets have been filtered for specific properties, and some of the downstream methods (e.g. TabPFN) perform best when preprocessing is not applied [90].

C.4.1 Evaluation Datasets

UniPredict Benchmark

We use the "supervised" subset of 169 datasets from the recently-introduced UniPredict [190] benchmark, obtained through correspondence with the authors. These are high-quality tabular datasets with generally informative column names and a mix of both categorical and continuous targets, drawn directly from Kaggle. The datasets are prostprocessed in [190] using a commercial LLM; however, we do not have access to the results of this postprocessing and instead use the datasets exactly as they are obtained from the Kaggle API.

Note that while the model introduced in [190] was trained and tested on separate splits of these datasets, we only use them for testing.

We obtain the complete set of tasks, and the corresponding target columns for each task, for UniPredict, via correspondence with the authors of UniPredict. However, we found that several tasks in the benchmark were incorrectly labeled as categorical when, in fact, these tasks were continuous (this is likely due to the use of an LLM to determine the target columns and their attributes in [190]). We manually verify the correct target type (continuous vs. categorical) for each of the 169 datasets, and apply corrections to TODO datasets in the benchmark. The exact set of benchmarks, target columns, and target type (continuous vs. non-continuous) used in our paper are provided in the supplementary material.

We note that our results are *not* directly comparable to the original results in UniPredict. This is for at least two reasons: (1) due to the modifications described above, where there are errors present in the original categorization of the targets (continuous vs. non-continuous) and (2) because some of the Kaggle datasets in UniPredict are continuously updated (e.g. stock datasets) and the data or access dates are not reported in [190]. We do provide our full evaluation suite, including all UniPredict datasets, in the supplementary material in order for future comparisons to our work.

Grinsztajn Benchmark

The Grinsztajn benchmark [74] is a curated suite of 45 datasets consisting of numeric and categorical features. This dataset is notable in that the original study by Grinsztajn et al. found that gradient boosted decision trees (GBDTs) consistently outperformed deep learning-based methods on these tasks. The benchmark is comprised of a mix of classification and regression tasks; for all regression tasks, we apply the discretization method used in UniPredict [190] and discretize the targets into quartiles.

AutoML Multimodal Benchmark

The AutoML Multimodal Benchmark [168][3] is a suite of tables which include one or more free-text fields (such as an Airbnb description, or a product review). The benchmark is considered challenging for tree-based methods due to the non-standard text-based features. However, it also poses a challenge for LLMs since some columns can contain highly variable lengths of text.

[3] https://github.com/sxjscience/automl_multimodal_benchmark/tree/main

OpenML CC-18 Benchmark

The OpenML Curated Classification Benchmark [25] was created by applying filtering rules to extract a high-quality subset from the OpenML platform. The rules include: no artificial data sets, no subsets of larger data sets nor binarizations of other data sets, no data sets which are perfectly predictable by using a single feature or a simple decision tree.

OpenML CTR-23 Benchmark

The OpenML Curated Tabular Regression (CTR) Benchmark [62] is a curated set of tables for regression drawn from OpenML. The curation process is similar to that of OpenML-CC18, for regression tasks. As in [190], we convert the continuous regression targets into a finite set of discrete labels.

We remove the `solar_flare` task from the benchmark, as 82% of the observations have the same regression target value (0) and thus we cannot apply our quartile-transformation method.

C.4.2 Baselines

Describe any baseline models and any special techniques or tuning that differ from what we use in the main text.

For the supervised learning baselines (XGBoost, TabPFN), we conduct 10 independent trials, drawing separate training sets of size k shots, for each value of k. We use the full remaining dataset as the test set.

Hyperparameter tuning: For each of the 10 independent trials, we tune the hyperparameters of the model. For XGBoost, we conduct 10 iterations of hyperparameter tuning using the HyperOpt hyperparameter optimization library and the hyperparameter grid defined in [67]. For TabPFN, we conduct a full grid search (since there is only a single hyperparameter).

Llama 3 Base Model

We use the pretrained Llama 3 model available on Hugging Face. This model, we do not modify the tokenizer (i.e. by adding special tokens that are used by TabuLa-8B), but the serialized data format is identical to the format seen by our model during training (Figure 4.2b).

XGBoost

For every dataset and number of shots, we tune the hyperparameters according to the grid from [67]. We use 10 iterations of the adaptive hyperparameter tuning method HyperOpt on this grid with 3-fold cross-validation, whenever the number of samples is greater than or equal to 3; when the number of samples is less than 3, we use the default settings.

TabPFN

TabPFN [90] is a hypernetwork that predicts the parameters of a neural network that can be used to classify the data. TabPFN is a pretrained Transformer model that takes a training dataset as input, and produces the parameters of a network as output; that network is then used to classify the test data. TabPFN is widely considered to be among the state-of-the-art methods for prediction on tabular data [122, 90], and has been shown to be especially effective for few-shot learning.

We use the official TabPFN implementation[4]. TabPFN has one tunable hyperparameter, the number of model predictions that are ensembled with feature and class rotations (`N_ensemble_configurations` in the TabPFN codebase). We sweep over all values in the range $[3, 2 \cdot d]$ where d is the number of features. As noted in the package documentation, when `N_ensemble_configurations` $> 2 \cdot d$ for a binary classification task, no further averaging is applied.

The official implementation of TabPFN has three limitations relevant to our experiments. First, TabPFN is limited to 100 features.[5] As a result, when the number of features is greater than 100, we use TabPFN's feature subsampling, which randomly selects 100 features. Second, TabPFN cannot be trained on datasets that have more than 10 input classes. We do not report the results of experiments where TabPFN cannot be trained on at least one of the 10 random iterates of each value of k. Third, TabPFN cannot be trained on fewer than $|C|$ examples, where C is the set of potential classes.

[4] https://github.com/automl/TabPFN

[5] https://github.com/automl/TabPFN/blob/main/tabpfn/scripts/transformer_prediction_interface.py#L105

C.4.3 Relative Sample Efficiency

For two classifiers f, g, let $N_D(f, \alpha)$ and $N_D(g, \alpha)$ denote the number of samples required for the classifiers to reach a performance level α on data D. The *relative efficiency* of f relative to g on dataset D at level α is equal to

$$N_D(f, \alpha)/N_D(g, \alpha). \tag{C.1}$$

C.4.4 Generation Procedure

We use the default generation settings of the Llama 3 Hugging Face model.[6]. This includes: temperature of 0.6, top-p 0.9. We do not tune these generation settings.

C.5 Detailed Results

In this section, we provide additional results for larger numbers of shots not provided in the main text. Figure C.1 provides extended results for both the baseline models, and for TabuLa-8B.

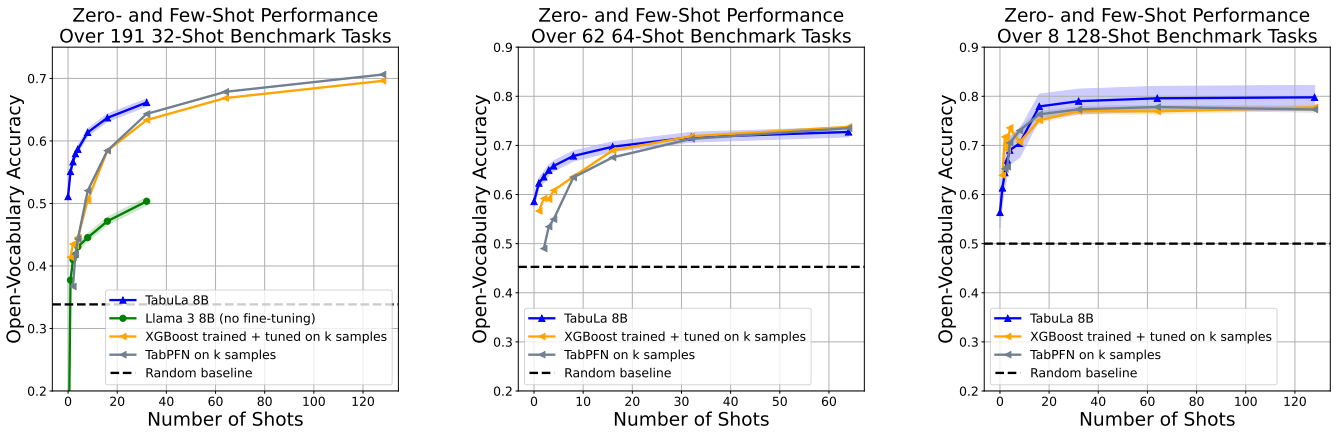
As shown in Figure C.1, all models tend to improve with more shots. However, on the subset of datasets that can be used for 64- and 128-shot learning, we observe a narrower gap between TabuLa-8B and baselines. We hypothesize that this is due to the fact that there is a selection bias: only datasets with small numbers of features and short column headers are candidates for 128-shot learning (due to the limited context window size of TabuLa-8B). As a result, this biases those evaluations away from semantically-rich datasets where we expect TabuLa-8B to excel.

C.6 Benchmark Contamination Study

C.6.1 Identifying Potential Contamination

Identifying contamination in tabular data is complex. As mentioned above, tabular data is invariant to permutations of rows and columns; as a result, the "same" tabular dataset could appear in a number of different permutations of its respective rows or columns. As a result, so-called "exact" deduplication methods are likely to be imperfect for tabular data (we also perform exact deduplication in our filtering step, so at most each individual dataset should only appear once in T4).

[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B



(a) (b) (c)

Figure C.1: C.1a: Results for 32-shot tasks, with extended curves for baselines. C.1b, C.1c: Results for 64- and 128-shot tasks. All models continue to improve as k increases, but the gap between methods may narrow. However, the *nature* of the datasets that can be used for 64-shot learning with TabuLa-8B are considerably different – fewer features, with shorter, less semantically-meaningful column names – which may downwardly bias the observed performance of TabuLa-8B with large k.

Benchmark	Potentially Leaked Tables Count (% of Benchmark Tables)	
	Fuzzy Deduplication	Strict Deduplication
OpenML-CC18	23 (31.9%)	13 (18.1%)
OpenML-CTR23	12 (34.2%)	6 (17.1%)
AMLB	1 (12.5%)	0
UniPredict	50 (29.5%)	39 (23%)
Grinsztajn	16 (35.5%)	4 (8.9%)

Table C.1: Counts of potentially contaminated tables according to "fuzzy" and "strict" procedures described. Note that "fuzzy" decontamination is stricter and is more likely to generate both true and false positives when checking for contamination.

When assessing the impact of duplication on our downstream evaluations, we are particularly concerned about the possibility of a model overfitting to the *overall features and distribution* of a dataset, not individual points in that distirbution – a model could learn unwanted information about a test set, for example, by observing points strictly from the training set of an i.i.d. split. As a result, we focus on eliminating datasets which have the same *schema* as a proxy for a dataset; we do not search for individual *data points* within that schema.

We propose two levels of searching for contamination; we refer to these methods as "fuzzy" and "strict". In *fuzzy* deduplication, we search for whether *every column name* in an evaluation dataset is present in the training dataset. In *strict* deduplication, we further add the condition that the number of columns must be identical. Note that we do not compute a matching directly between the columns of the two datasets, as checking for membership is much more efficient than checking for a compatible mapping between columns and our datasets can be up to 512 columns. Note that *fuzzy deduplication will potentially exclude more datasets* at the risk of potentially more false positives, since strict deduplication only adds conditions to the fuzzy deduplication. Fuzzy is therefore a *more conservative* deduplication mechanism than strict.

We search over all 1.55M tables in T4 and apply both "fuzzy" and "strict" checks. Some descriptive metrics for this search are shown in Table C.1.

It is perhaps expected that, in most cases, benchmark tables pass our T4 filtering procedure and make it into T4 (one notable exception is AMLB, where most tables likely fail to pass our rules which filter for cells with large numbers of characters). Indeed, these are public benchmarks designed for learning on tabular data, and TabLib includes a significant component of datasets sourced from GitHub; many users likely work with these datasets and have uploaded them to GitHub.

C.6.2 Impact of Contamination

In this section we investigate the impact of potential contamination in our training data. In particular, we investigate whether the number of repetitions of a dataset in the training data is correlated with the downstream performance on that task, and we assess performance on "potentially contaminated" vs. "decontaminated" tasks.

Figure C.3 shows the relationship between the number of potential contamination instances for each dataset, and the TabuLa-8B accuracy on that dataset. We find no clear relationship between this potential contamination and downstream performance, and believe that this reflects, at least in part, the conservative nature of our contamination test, which is likely to have may false positives for datasets with generic column names that may occur frequently in TabLib (for example, columns "Date", "Open", "Close" are common among stock datasets; columns "v1" "v2" are common generic variable names).

Additionally, we believe that other considerations likely explain the lack of correlation between (potential) contamination in the training data and downstream benchmark berformance. For example:

- TabuLa-8B does not train on the full corpus; datasets that appear a small number of times may in fact not be seen during training.
- Prior work has suggested mixed impact of contamination [e.g. 146, 31]; contamination does not always improve performance and can sometimes reduce performance.
- We are only fine-tuning the model which can be thought of as an alignment step. It is possible that contamination in *pretraining* affects this more and that fine-tuning is less susceptible to memorization.
- Due to our use of deduplication, if a dataset does recur, it is not identical – so the model never really sees an identical table more than once unless we make multiple passes over the training set or sample

with replacement.

- Due to our use of row-level deduplication and random shuffling, the exact evaluation datasets in the same order are unlikely to ever be seen by the model even if provided during training; this may be a guard against memorization.

We also separately report our results on the "possibly contaminated" vs. "decontaminated" subset of our evaluation suite, in Figure C.2. Figure C.2 shows that our model (and all baseline models) tend to perform *better* on the decontaminated subset. We hypothesize that this reflects a few factors. In particular, datasets in the "decontaminated" subset are likely to have unique column names. This "uniqueness" likely correlates positively with semantic quality; our model will tend to perform better on such datasets. Furthermore, we hypothesize that this reflects the strictness of our fuzzy decontamination check: it is likely that many of the tables flagged as "potentially contaminated" are false positives (our manual inspection confirmed this in many cases, although it is difficult to verify that two shuffled tabular datasets are identical; we leave such an analysis to future work).

C.7 List of Evaluation Datasets and Per-Task Results

We provide results for TabuLa-8B on each individual task, along with the accuracy of a random-class predictor (which is equivalent to 1 / number of classes) in this section. For the complete results for all values of shots and all baselines, see the supplementary material. Note that 'NA' results for TabuLa-8B indicate that the serialized data with a given value of shots k exceeds the model's context window size.

	TabuLa-8B			XGBoost		Random
Task	0	4	32	4	32	
amlb/data_scientist_salary	0.328	0.398	0.391	0.202	0.283	0.167
amlb/imdb_genre_prediction	0.844	0.742	0.828	0.528	0.554	0.500
amlb/jigsaw_unintended_bias100K	0.938	0.922	NA	0.942	NA	0.500
amlb/kick_starter_funding	0.594	0.680	0.609	0.644	0.649	0.500
amlb/melbourne_airbnb	0.727	NA	NA	NA	NA	0.100
amlb/news_channel	0.297	0.227	NA	0.213	NA	0.167
amlb/product_sentiment_machine_hack	0.406	0.438	0.531	0.542	0.762	0.250

amlb/wine_reviews	0.648	0.711	NA	0.080	NA	0.033
grinsztajn/cat_clf/albert	0.445	0.500	NA	0.499	NA	0.500
grinsztajn/cat_clf/compas-two-years	0.547	0.539	0.602	0.499	0.582	0.500
grinsztajn/cat_clf/covertype	0.422	0.570	NA	0.500	NA	0.500
grinsztajn/cat_clf/default-of-credit-card-clients	0.508	0.531	0.539	0.500	0.618	0.500
grinsztajn/cat_clf/electricity	0.461	0.641	0.664	0.503	0.644	0.500
grinsztajn/cat_clf/eye_movements	0.508	0.430	NA	0.504	NA	0.500
grinsztajn/cat_clf/road-safety	0.508	0.477	NA	0.502	NA	0.500
grinsztajn/cat_reg/Bike_Sharing_Demand	0.352	0.344	0.508	0.250	0.424	0.250
grinsztajn/cat_reg/Brazilian_houses	0.344	0.547	0.797	0.256	0.769	0.250
grinsztajn/cat_reg/Mercedes_Benz_Greener_Manufa...	0.266	NA	NA	NA	NA	0.250
grinsztajn/cat_reg/OnlineNewsPopularity	0.289	0.273	NA	0.253	NA	0.250
grinsztajn/cat_reg/SGEMM_GPU_kernel_performance	0.094	0.602	0.898	0.249	0.891	0.250
grinsztajn/cat_reg/analcatdata_supreme	0.461	0.453	0.914	0.624	0.980	0.333
grinsztajn/cat_reg/black_friday	0.203	0.234	0.297	0.250	0.331	0.250
grinsztajn/cat_reg/diamonds	0.344	0.547	0.688	0.248	0.736	0.250
grinsztajn/cat_reg/house_sales	0.258	0.453	NA	0.249	NA	0.250
grinsztajn/cat_reg/nyc-taxi-green-dec-2016	0.242	0.445	NA	0.249	NA	0.250
grinsztajn/cat_reg/particulate-matter-ukair-2017	0.305	0.375	NA	0.248	NA	0.250
grinsztajn/cat_reg/visualizing_soil	0.281	0.383	0.688	0.248	0.789	0.250
grinsztajn/cat_reg/yprop_4_1	0.203	0.219	NA	0.251	NA	0.250
grinsztajn/num_clf/Diabetes130US	0.531	0.508	0.484	0.500	0.523	0.500
grinsztajn/num_clf/MagicTelescope	0.555	0.703	0.695	0.494	0.670	0.500
grinsztajn/num_clf/MiniBooNE	0.461	0.539	NA	0.505	NA	0.500
grinsztajn/num_clf/bank-marketing	0.500	0.578	0.656	0.502	0.695	0.500
grinsztajn/num_clf/california	0.453	0.578	0.742	0.500	0.720	0.500
grinsztajn/num_clf/covertype	0.500	0.484	0.477	0.500	0.553	0.500
grinsztajn/num_clf/credit	0.562	0.578	0.656	0.505	0.659	0.500
grinsztajn/num_clf/default-of-credit-card-clients	0.516	0.477	0.523	0.500	0.601	0.500

grinsztajn/num_clf/electricity	0.547	0.609	0.602	0.503	0.655	0.500
grinsztajn/num_clf/eye_movements	0.469	0.547	NA	0.504	NA	0.500
grinsztajn/num_clf/heloc	0.500	0.508	NA	0.494	NA	0.500
grinsztajn/num_clf/house_16H	0.477	0.609	NA	0.503	NA	0.500
grinsztajn/num_clf/jannis	0.500	0.594	NA	0.500	NA	0.500
grinsztajn/num_clf/pol	0.523	0.484	0.773	0.512	0.810	0.500
grinsztajn/num_reg/Ailerons	0.266	0.289	NA	0.278	NA	0.250
grinsztajn/num_reg/Bike_Sharing_Demand	0.234	0.414	0.398	0.250	0.424	0.250
grinsztajn/num_reg/Brazilian_houses	0.250	0.570	0.812	0.256	0.769	0.250
grinsztajn/num_reg/MiamiHousing2016	0.250	0.352	NA	0.250	NA	0.250
grinsztajn/num_reg/california	0.289	0.406	NA	0.247	NA	0.250
grinsztajn/num_reg/cpu_act	0.359	0.320	NA	0.272	NA	0.250
grinsztajn/num_reg/diamonds	0.391	0.602	0.688	0.248	0.742	0.250
grinsztajn/num_reg/elevators	0.188	0.266	NA	0.266	NA	0.250
grinsztajn/num_reg/fifa	0.250	0.406	0.469	0.248	0.434	0.250
grinsztajn/num_reg/house_16H	0.336	0.320	NA	0.249	NA	0.250
grinsztajn/num_reg/house_sales	0.328	0.492	NA	0.249	NA	0.250
grinsztajn/num_reg/houses	0.258	0.312	0.445	0.247	0.418	0.250
grinsztajn/num_reg/isolet	0.262	NA	NA	NA	NA	0.250
grinsztajn/num_reg/medical_charges	0.328	0.555	0.727	0.249	0.801	0.250
grinsztajn/num_reg/nyc-taxi-green-dec-2016	0.273	0.406	NA	0.249	NA	0.250
grinsztajn/num_reg/pol	0.602	0.617	0.867	0.683	0.815	0.500
grinsztajn/num_reg/sulfur	0.258	0.227	0.383	0.253	0.441	0.250
grinsztajn/num_reg/superconduct	0.242	0.438	NA	0.247	NA	0.250
grinsztajn/num_reg/wine_quality	0.578	0.508	0.516	0.538	0.622	0.333
grinsztajn/num_reg/year	0.227	0.289	NA	0.262	NA	0.250
openml_cc18/Fashion-MNIST	0.070	NA	NA	NA	NA	0.100
openml_cc18/GesturePhaseSegmentationProcessed	0.266	0.312	NA	0.270	NA	0.200
openml_cc18/MiceProtein	0.125	0.211	NA	0.118	NA	0.125

openml_cc18/PhishingWebsites	0.516	0.656	NA	0.500	NA	0.500
openml_cc18/adult	0.727	0.812	0.828	0.763	0.764	0.500
openml_cc18/analcatdata_authorship	0.234	0.438	NA	0.365	NA	0.250
openml_cc18/analcatdata_dmft	0.125	0.156	0.125	0.182	0.188	0.167
openml_cc18/bank-marketing	0.508	0.758	0.844	0.889	0.880	0.500
openml_cc18/banknote-authentication	0.477	0.680	0.922	0.509	0.862	0.500
openml_cc18/blood-transfusion-service-center	0.516	0.664	0.586	0.787	0.749	0.500
openml_cc18/breast-w	0.938	0.961	0.984	0.569	0.900	0.500
openml_cc18/car	0.789	0.898	0.898	0.763	0.755	0.250
openml_cc18/churn	0.703	0.758	0.914	0.870	0.863	0.500
openml_cc18/cmc	0.320	0.383	0.352	0.330	0.450	0.333
openml_cc18/cnae-9	0.172	NA	NA	NA	NA	0.111
openml_cc18/connect-4	0.297	0.414	NA	0.522	NA	0.333
openml_cc18/credit-approval	0.484	0.555	0.742	0.428	0.764	0.500
openml_cc18/credit-g	0.508	0.523	0.703	0.500	0.725	0.500
openml_cc18/diabetes	0.617	0.656	0.680	0.661	0.722	0.500
openml_cc18/dna	0.312	0.430	NA	0.467	NA	0.333
openml_cc18/electricity	0.438	0.633	0.711	0.544	0.695	0.500
openml_cc18/eucalyptus	0.273	0.352	0.359	0.211	0.407	0.200
openml_cc18/first-order-theorem-proving	0.211	0.195	NA	0.284	NA	0.167
openml_cc18/har	0.125	NA	NA	NA	NA	0.167
openml_cc18/isolet	0.022	NA	NA	NA	NA	0.038
openml_cc18/jm1	0.812	0.750	0.805	0.746	0.784	0.500
openml_cc18/jungle_chess_2pcs_raw_endgame_complete	0.477	0.555	0.516	0.409	0.589	0.333
openml_cc18/kc1	0.852	0.836	0.820	0.775	0.843	0.500
openml_cc18/kr-vs-kp	0.469	0.508	NA	0.490	NA	0.500
openml_cc18/letter	0.039	0.070	0.266	0.038	0.200	0.038
openml_cc18/madelon	0.531	NA	NA	NA	NA	0.500
openml_cc18/mfeat-factors	0.078	0.172	NA	0.092	NA	0.100

openml_cc18/mfeat-fourier	0.086	0.172	NA	0.092	NA	0.100
openml_cc18/mfeat-karhunen	0.078	0.141	NA	0.092	NA	0.100
openml_cc18/mfeat-morphological	0.141	0.188	0.555	0.092	0.493	0.100
openml_cc18/mfeat-pixel	0.109	NA	NA	NA	NA	0.100
openml_cc18/mfeat-zernike	0.141	0.141	NA	0.092	NA	0.100
openml_cc18/mnist_784	0.148	NA	NA	NA	NA	0.100
openml_cc18/nomao	0.398	0.742	NA	0.542	NA	0.500
openml_cc18/numerai28.6	0.539	0.469	NA	0.495	NA	0.500
openml_cc18/optdigits	0.070	0.094	NA	0.092	NA	0.100
openml_cc18/ozone-level-8hr	0.445	0.852	NA	0.969	NA	0.500
openml_cc18/pc1	0.766	0.945	0.898	0.964	0.964	0.500
openml_cc18/pc3	0.820	0.875	NA	0.834	NA	0.500
openml_cc18/pc4	0.609	0.914	NA	0.890	NA	0.500
openml_cc18/pendigits	0.039	0.242	0.523	0.099	0.534	0.100
openml_cc18/phoneme	0.453	0.641	0.734	0.624	0.723	0.500
openml_cc18/qsar-biodeg	0.508	0.523	NA	0.579	NA	0.500
openml_cc18/satimage	0.180	0.320	NA	0.186	NA	0.167
openml_cc18/segment	0.195	0.438	NA	0.144	NA	0.143
openml_cc18/semeion	0.148	NA	NA	NA	NA	0.100
openml_cc18/sick	0.938	0.914	NA	0.950	NA	0.500
openml_cc18/spambase	0.555	0.680	NA	0.560	NA	0.500
openml_cc18/splice	0.336	0.328	NA	0.378	NA	0.333
openml_cc18/steel-plates-fault	0.141	0.273	NA	0.201	NA	0.143
openml_cc18/texture	0.070	0.180	NA	0.090	NA	0.091
openml_cc18/tic-tac-toe	0.383	0.609	0.523	0.454	0.702	0.500
openml_cc18/vehicle	0.219	0.258	0.484	0.239	0.495	0.250
openml_cc18/vowel	0.055	0.125	0.172	0.106	0.358	0.091
openml_cc18/wall-robot-navigation	0.250	0.305	NA	0.398	NA	0.250
openml_cc18/wilt	0.594	0.812	0.969	0.959	0.957	0.500

openml_ctr23/Moneyball	0.477	0.375	0.609	0.243	0.595	0.250
openml_ctr23/QSAR_fish_toxicity	0.266	0.305	0.484	0.245	0.462	0.250
openml_ctr23/abalone	0.273	0.430	0.406	0.296	0.483	0.250
openml_ctr23/airfoil_self_noise	0.258	0.305	0.383	0.242	0.379	0.250
openml_ctr23/auction_verification	0.266	0.344	0.398	0.252	0.652	0.250
openml_ctr23/brazilian_houses	0.438	0.578	0.633	0.256	0.551	0.250
openml_ctr23/california_housing	0.320	0.375	0.461	0.247	0.419	0.250
openml_ctr23/cars	0.359	0.336	0.656	0.228	0.637	0.250
openml_ctr23/concrete_compressive_strength	0.328	0.484	0.562	0.246	0.446	0.250
openml_ctr23/cps88wages	0.391	0.367	0.422	0.255	0.332	0.250
openml_ctr23/cpu_activity	0.250	0.375	NA	0.272	NA	0.250
openml_ctr23/diamonds	0.438	0.680	0.742	0.247	0.751	0.250
openml_ctr23/energy_efficiency	0.391	0.461	0.719	0.238	0.749	0.250
openml_ctr23/fifa	0.352	0.359	NA	0.243	NA	0.250
openml_ctr23/fps_benchmark	0.242	0.367	NA	0.250	NA	0.250
openml_ctr23/geographical_origin_of_music	0.281	NA	NA	NA	NA	0.250
openml_ctr23/grid_stability	0.242	0.297	NA	0.239	NA	0.250
openml_ctr23/health_insurance	0.406	0.500	0.633	0.482	0.591	0.333
openml_ctr23/kin8nm	0.312	0.242	0.297	0.251	0.342	0.250
openml_ctr23/kings_county	0.305	0.422	NA	0.249	NA	0.250
openml_ctr23/miami_housing	0.273	0.391	NA	0.250	NA	0.250
openml_ctr23/naval_propulsion_plant	0.211	0.227	NA	0.249	NA	0.250
openml_ctr23/physiochemical_protein	0.242	0.289	0.281	0.250	0.334	0.250
openml_ctr23/pumadyn32nh	0.219	0.273	NA	0.251	NA	0.250
openml_ctr23/red_wine	0.523	0.508	0.656	0.476	0.619	0.333
openml_ctr23/sarcos	0.227	0.203	NA	0.254	NA	0.250
openml_ctr23/socmob	0.391	0.523	0.695	0.248	0.598	0.250
openml_ctr23/space_ga	0.203	0.219	0.305	0.261	0.412	0.250
openml_ctr23/student_performance_por	0.375	0.258	NA	0.274	NA	0.250

openml_ctr23/superconductivity	0.250	0.352	NA	0.247	NA	0.250
openml_ctr23/video_transcoding	0.250	0.336	NA	0.249	NA	0.250
openml_ctr23/wave_energy	0.234	0.273	NA	0.256	NA	0.250
openml_ctr23/white_wine	0.484	0.594	0.594	0.651	0.660	0.333
unipredict/aakashjoshi123/exercise-and-fitness-...	0.383	0.641	0.766	0.253	0.761	0.250
unipredict/aakashjoshi123/spotify-top-hits-data	0.555	0.797	0.781	0.750	0.694	0.077
unipredict/abcsds/pokemon	0.977	0.984	0.992	0.060	0.128	0.059
unipredict/adityakadiwal/water-potability	0.562	0.469	0.516	0.524	0.591	0.500
unipredict/agirlcoding/all-space-missions-from-...	0.891	0.875	0.867	0.910	0.874	0.333
unipredict/ahsan81/food-ordering-and-delivery-a...	0.273	0.266	0.242	0.296	0.323	0.250
unipredict/ahsan81/superstore-marketing-campaig...	0.516	0.727	0.766	0.848	0.836	0.500
unipredict/akshaydattatraykhare/diabetes-dataset	0.586	0.742	0.672	0.661	0.704	0.500
unipredict/alexisbcook/pakistan-intellectual-ca...	0.609	0.656	NA	0.592	NA	0.083
unipredict/alirezachahardoli/bank-personal-loan-1	0.758	0.773	0.914	0.920	0.913	0.500
unipredict/altruistdelhite04/gold-price-data	0.562	0.680	0.867	0.241	0.603	0.250
unipredict/amirhosseinmirzaie/countries-life-ex...	0.766	0.719	NA	0.244	NA	0.250
unipredict/amirhosseinmirzaie/pistachio-types-d...	0.547	0.570	NA	0.523	NA	0.500
unipredict/ananthr1/weather-prediction	0.703	0.828	0.812	0.369	0.787	0.200
unipredict/andrewmvd/fetal-health-classification	0.211	0.734	NA	0.736	NA	0.333
unipredict/andrewmvd/udemy-courses	1.000	1.000	0.992	0.311	0.433	0.250
unipredict/arashnic/time-series-forecasting-wit...	0.992	1.000	0.984	0.251	0.903	0.250
unipredict/arnabchaki/data-science-salaries-2023	0.898	0.961	0.969	0.256	0.898	0.250
unipredict/arnabchaki/indian-restaurants-2023	0.281	0.289	0.242	0.257	0.308	0.250
unipredict/arnavsmayan/netflix-userbase-dataset	0.344	0.305	0.352	0.362	0.438	0.333
unipredict/arnavsmayan/vehicle-manufacturing-da...	0.055	0.047	0.062	0.077	0.094	0.059
unipredict/arslanr369/bitcoin-price-2014-2023	1.000	0.992	0.984	0.247	0.899	0.250
unipredict/ashishkumarjayswal/diabetes-dataset	0.516	0.680	0.742	0.661	0.712	0.500
unipredict/ashishkumarjayswal/movies-updated-data	0.656	0.555	0.641	0.158	0.276	0.091
unipredict/atharvaingle/crop-recommendation-dat...	0.055	0.320	0.695	0.049	0.438	0.045

unipredict/awaiskaggler/insurance-csv	0.398	0.547	0.695	0.245	0.519	0.250
unipredict/azminetoushikwasi/-lionel-messi-all-...	0.219	0.414	0.469	0.634	0.594	0.111
unipredict/barun2104/telecom-churn	0.844	0.836	0.836	0.717	0.857	0.500
unipredict/bhanupratapbiswas/bollywood-actress-...	0.258	0.430	0.656	0.567	0.440	0.125
unipredict/bhanupratapbiswas/fashion-products	0.359	0.367	0.406	0.328	0.349	0.333
unipredict/bhanupratapbiswas/uber-data-analysis	0.617	0.820	0.922	0.931	0.928	0.500
unipredict/bhanupratapbiswas/world-top-billiona...	0.383	0.531	NA	0.254	NA	0.250
unipredict/bharath011/heart-disease-classificat...	0.531	0.641	0.844	0.564	0.906	0.500
unipredict/bhavkaur/hotel-guests-dataset	0.430	0.594	0.867	0.865	0.836	0.333
unipredict/bhavkaur/simplified-titanic-dataset	0.555	0.523	0.734	0.647	0.735	0.500
unipredict/blastchar/telco-customer-churn	0.711	0.719	0.711	0.696	0.748	0.500
unipredict/bretmathyer/telemedicine-used	0.500	0.555	NA	0.494	NA	0.500
unipredict/buntyshah/auto-insurance-claims-data	0.602	0.594	NA	0.700	NA	0.500
unipredict/carolzhangdc/imdb-5000-movie-dataset	0.484	0.516	NA	0.247	NA	0.250
unipredict/chirin/africa-economic-banking-and-s...	0.953	0.953	0.977	0.877	0.969	0.500
unipredict/christinestevens/cstevens-peloton-data	0.984	0.992	1.000	0.151	0.164	0.143
unipredict/cpluzshrijayan/milkquality	0.406	0.406	0.711	0.361	0.836	0.333
unipredict/crxxom/manhwa-dataset	0.695	0.859	NA	0.589	NA	0.250
unipredict/dansbecker/aer-credit-card-data	0.609	0.742	0.930	0.545	0.967	0.500
unipredict/deependraverma13/diabetes-healthcare...	0.602	0.680	0.734	0.661	0.706	0.500
unipredict/desalegngeb/german-fintech-companies	0.758	0.789	NA	0.280	NA	0.250
unipredict/dileep070/heart-disease-prediction-u...	0.789	0.750	0.805	0.768	0.827	0.500
unipredict/dsfelix/us-stores-sales	0.547	0.648	NA	0.258	NA	0.250
unipredict/elakiricoder/gender-classification-d...	0.594	0.719	0.883	0.502	0.935	0.500
unipredict/fedesoriano/stroke-prediction-dataset	0.945	0.977	0.945	0.951	0.950	0.500
unipredict/gabrielsantello/cars-purchase-decisi...	0.383	0.609	0.820	0.580	0.843	0.500
unipredict/gauravduttakiit/resume-dataset	0.906	0.969	NA	0.043	NA	0.040
unipredict/geomack/spotifyclassification	0.422	0.641	0.984	0.536	0.981	0.500
unipredict/gyanprakashkushwaha/laptop-price-pre...	0.328	0.562	0.570	0.251	0.550	0.250

unipredict/hansrobertson/american-companies-pro...	0.250	0.297	0.242	0.250	0.314	0.250
unipredict/harishkumardatalab/medical-insurance...	0.336	0.586	0.750	0.235	0.681	0.250
unipredict/harshitshankhdhar/imdb-dataset-of-to...	0.430	0.445	NA	0.310	NA	0.250
unipredict/hashemi221022/bank-loans	0.773	0.844	0.867	0.920	0.887	0.500
unipredict/hashemi221022/diabetes	0.508	0.750	0.656	0.661	0.721	0.500
unipredict/hawkingcr/airbnb-for-boston-with-fra...	0.789	0.719	0.797	0.780	0.796	0.500
unipredict/hemanthhari/psycological-effects-of-...	0.414	0.391	0.570	0.245	0.615	0.143
unipredict/hesh97/titanicdataset-traincsv	0.797	0.734	0.703	0.587	0.706	0.500
unipredict/iamsumat/spotify-top-2000s-mega-dataset	0.305	0.406	0.516	0.258	0.276	0.250
unipredict/iqmansingh/company-employee-dataset	0.641	0.539	0.586	0.073	0.199	0.050
unipredict/ishadss/productivity-prediction-of-g...	0.391	0.430	NA	0.233	NA	0.250
unipredict/jainilcoder/netflix-stock-price-pred...	1.000	0.992	0.992	0.261	0.884	0.250
unipredict/jillanisofttech/brain-stroke-dataset	0.969	0.961	0.953	0.948	0.948	0.500
unipredict/kabure/german-credit-data-with-risk	0.594	0.586	0.695	0.500	0.720	0.500
unipredict/kandij/diabetes-dataset	0.562	0.758	0.711	0.661	0.717	0.500
unipredict/kanths028/usa-housing	0.195	0.281	NA	0.242	NA	0.250
unipredict/kingabzpro/cosmetics-datasets	0.859	0.836	NA	0.164	NA	0.167
unipredict/kreeshrajani/human-stress-prediction	0.547	0.633	0.664	0.500	0.496	0.500
unipredict/kumargh/pimaindiansdiabetescsv	0.117	0.172	0.141	0.156	0.171	0.077
unipredict/larsen0966/student-performance-data-set	0.297	0.266	NA	0.088	NA	0.077
unipredict/lightonkalumba/us-womens-labor-force...	0.891	0.992	1.000	0.508	0.986	0.500
unipredict/mahnazarjmand/bank-personal-loan	0.758	0.828	0.922	0.920	0.918	0.500
unipredict/maryalebron/life-expectancy-data	0.023	0.023	NA	0.030	NA	0.028
unipredict/maryammanoochehry/bank-personal-loan	0.844	0.891	0.883	0.920	0.914	0.500
unipredict/mathchi/diabetes-data-set	0.500	0.695	0.680	0.661	0.704	0.500
unipredict/mayankpatel14/second-hand-used-cars-...	0.234	0.203	0.297	0.226	0.659	0.250
unipredict/mayurdalvi/simple-linear-regression-...	0.453	0.516	0.594	0.548	0.543	0.500
unipredict/mayuriawati/bangalore-chain-restaura...	0.891	0.875	NA	0.086	NA	0.045
unipredict/mazlumi/ielts-writing-scored-essays-...	0.109	0.227	NA	0.122	NA	0.083

unipredict/mfaisalqureshi/spam-email	0.703	0.891	0.984	0.846	0.846	0.500
unipredict/mirichoi0218/insurance	0.250	0.633	0.719	0.258	0.708	0.250
unipredict/nancyalaswad90/review	0.617	0.641	0.680	0.661	0.703	0.500
unipredict/naveenkumar20bps1137/predict-student...	0.039	0.086	NA	0.061	NA	0.059
unipredict/nikhil1e9/netflix-stock-price	0.984	0.977	0.984	0.246	0.898	0.250
unipredict/noordeen/insurance-premium-prediction	0.461	0.531	0.734	0.258	0.677	0.250
unipredict/oles04/bundesliga-seasons	1.000	1.000	NA	0.529	NA	0.500
unipredict/oles04/top-leagues-player	0.484	0.641	0.609	0.264	0.271	0.250
unipredict/patelprashant/employee-attrition	0.805	0.820	NA	0.837	NA	0.500
unipredict/pavansubhasht/ibm-hr-analytics-attri...	0.820	0.883	NA	0.837	NA	0.500
unipredict/phangud/spamcsv	0.664	0.883	0.969	0.846	0.846	0.500
unipredict/prevek18/ames-housing-dataset	0.445	0.625	NA	0.252	NA	0.250
unipredict/primaryobjects/voicegender	0.445	0.516	NA	0.500	NA	0.500
unipredict/prkhrawsthi/bitcoin-usd-daily-price-...	0.961	0.969	0.984	0.248	0.899	0.250
unipredict/rajyellow46/wine-quality	0.352	0.453	0.438	0.372	0.426	0.143
unipredict/ravibarnawal/mutual-funds-india-deta...	0.203	0.227	0.219	0.228	0.310	0.167
unipredict/receplyasolu/6k-weather-labeled-spot...	0.141	0.172	0.273	0.124	0.211	0.125
unipredict/redwankarimsony/heart-disease-data	0.273	0.484	0.555	0.430	0.528	0.200
unipredict/reihanenamdari/breast-cancer	0.344	0.289	0.297	0.250	0.295	0.250
unipredict/rishikeshkonapure/hr-analytics-predi...	0.820	0.836	NA	0.837	NA	0.500
unipredict/rkiattisak/student-performance-in-ma...	0.453	0.477	0.547	0.250	0.494	0.250
unipredict/rounakbanik/pokemon	0.977	0.969	NA	0.963	NA	0.500
unipredict/rpaguirre/tesla-stock-price	0.977	0.992	0.984	0.247	0.882	0.250
unipredict/rtatman/chocolate-bar-ratings	0.109	0.141	0.227	0.138	0.149	0.100
unipredict/ruchi798/student-feedback-survey-res...	0.117	0.062	0.070	0.086	0.099	0.100
unipredict/ruchi798/tv-shows-on-netflix-prime-v...	0.250	0.312	0.477	0.283	0.457	0.167
unipredict/sabasaeed1953/stock-prices-of-2023	0.984	0.969	0.977	0.274	0.854	0.250
unipredict/saloni1712/chatgpt-app-reviews	0.625	0.602	0.633	0.365	0.519	0.200
unipredict/sanjanchaudhari/bankloan	0.656	0.648	0.594	0.628	0.669	0.500

unipredict/sanjanchaudhari/netflix-dataset	0.602	0.500	0.500	0.155	0.293	0.100
unipredict/sanjanchaudhari/user-behavior-on-ins...	0.469	0.547	0.766	0.500	0.796	0.500
unipredict/saunakghosh/nba-players-dataset	0.828	0.719	0.836	0.472	0.763	0.125
unipredict/saurabh00007/diabetescsv	0.578	0.695	0.703	0.661	0.703	0.500
unipredict/sbhatti/financial-sentiment-analysis	0.594	0.602	0.680	0.471	0.527	0.333
unipredict/shashankshukla123123/marketing-campaign	0.266	0.812	NA	0.888	NA	0.500
unipredict/shivamb/disney-movies-and-tv-shows	0.883	0.969	0.984	0.752	0.899	0.500
unipredict/shivamb/hm-stores-dataset	0.211	0.547	NA	0.458	NA	0.250
unipredict/shreyanshverma27/imdb-horror-chillin...	0.398	0.484	0.500	0.257	0.301	0.250
unipredict/shreyapurohit/anime-data	0.266	0.789	0.906	0.246	0.889	0.250
unipredict/shroukgomaa/babies-food-ingredients	0.289	0.320	NA	0.274	NA	0.250
unipredict/shubhamgupta012/titanic-dataset	0.742	0.703	0.734	0.697	0.682	0.500
unipredict/siddharthss/crop-recommendation-dataset	0.102	0.227	0.625	0.049	0.438	0.045
unipredict/sidhus/crab-age-prediction	0.047	0.148	0.195	0.088	0.192	0.053
unipredict/suraj520/dairy-goods-sales-dataset	0.617	0.508	NA	0.268	NA	0.250
unipredict/surajjha101/stores-area-and-sales-data	0.250	0.234	0.250	0.253	0.244	0.250
unipredict/surajjha101/top-youtube-channels-data	0.508	0.508	0.469	0.142	0.183	0.077
unipredict/tahzeer/indian-startups-by-state	0.062	0.141	NA	0.227	NA	0.012
unipredict/tarkkaanko/amazon	0.469	0.766	NA	0.750	NA	0.200
unipredict/team-ai/spam-text-message-classifica...	0.750	0.867	0.961	0.846	0.846	0.500
unipredict/teertha/ushealthinsurancedataset	0.312	0.617	0.711	0.258	0.708	0.250
unipredict/tejashvi14/employee-future-prediction	0.516	0.531	0.477	0.608	0.659	0.500
unipredict/tejashvi14/engineering-placements-pr...	0.617	0.594	0.773	0.487	0.724	0.500
unipredict/thedevastator/cancer-patients-and-ai...	0.039	0.109	NA	0.010	NA	0.029
unipredict/thedevastator/employee-attrition-and...	0.805	0.812	NA	0.837	NA	0.500
unipredict/thedevastator/higher-education-predi...	0.672	0.867	NA	0.654	NA	0.500
unipredict/therealsampat/predict-movie-success-...	0.617	0.883	NA	0.762	NA	0.500
unipredict/timoboz/tesla-stock-data-from-2010-t...	0.992	0.984	1.000	0.244	0.907	0.250
unipredict/uciml/mushroom-classification	0.555	0.734	0.961	0.502	0.952	0.500

unipredict/uciml/pima-indians-diabetes-database	0.648	0.664	0.703	0.661	0.705	0.500
unipredict/uciml/red-wine-quality-cortez-et-al-...	0.344	0.344	0.461	0.385	0.479	0.200
unipredict/varpit94/tesla-stock-data-updated-ti...	0.984	1.000	1.000	0.255	0.924	0.250
unipredict/vedavyasv/usa-housing	0.242	0.305	NA	0.242	NA	0.250
unipredict/vijayvvenkitesh/microsoft-stock-time...	0.984	0.977	0.953	0.260	0.859	0.250
unipredict/vikramamin/customer-churn-decision-t...	0.711	0.711	0.664	0.696	0.746	0.500
unipredict/vikramamin/time-series-forecasting-u...	0.211	0.359	0.578	0.256	0.247	0.250
unipredict/vstacknocopyright/blood-transfusion-...	0.484	0.570	0.672	0.787	0.749	0.500
unipredict/warcoder/earthquake-dataset	0.820	0.820	NA	0.259	NA	0.250
unipredict/whenamancodes/predict-diabities	0.672	0.695	0.781	0.661	0.718	0.500
unipredict/whenamancodes/students-performance-i...	0.500	0.453	0.500	0.252	0.481	0.250
unipredict/yasserh/titanic-dataset	0.711	0.727	0.805	0.587	0.721	0.500
unipredict/yasserh/wine-quality-dataset	0.391	0.359	0.500	0.346	0.511	0.200

C.8 Model Card

We provide a Model Card for TabuLa-8B, as outlined in [127].

C.8.1 Model Details

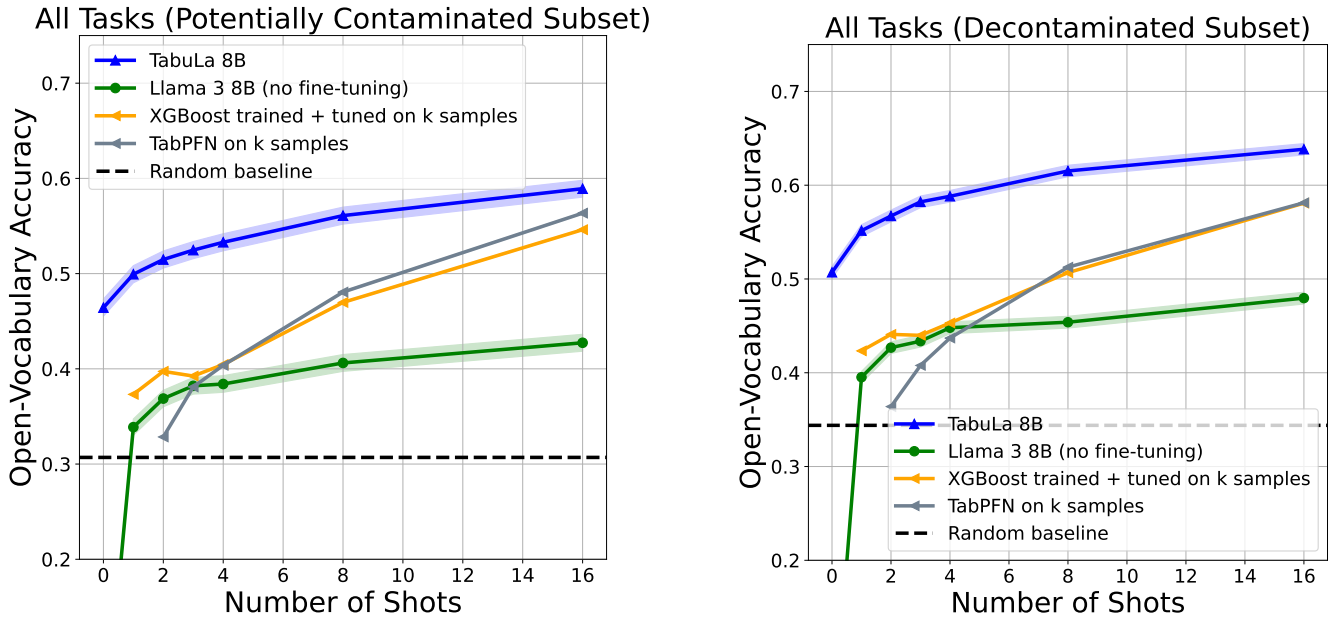
Person or organization developing model: This model was developed by the authors of this paper. Organizations providing computational support are listed in the Acknowledgements, but this model is not officially developed as part of any organization. The author affiliations are listed on the first page of this paper.

Model date: This paper describes the May 2024 version of TabuLa-8B.

Model version: This paper describes version 1.0 of TabuLa-8B.

Model type: TabuLa-8B is an autoregressive language model, identical in architecture to Llama 3 [179].

Information about training algorithms, parameters, fairness constraints or other applied approaches, and features: Our training procedure is described in Section 4.3. Our procedure for dataset construction, which includes methods for removing sensitive PII, is described in Sections 4.4 and C.1.



(a) (b)

Figure C.2: C.2a: Results curves on the subset of tasks identified as potentially contaminated according to our fuzzy decontamination procedure. C.2b: Results curves on the subset of tasks not identified as potentially contaminated according to our fuzzy decontamination procedure.

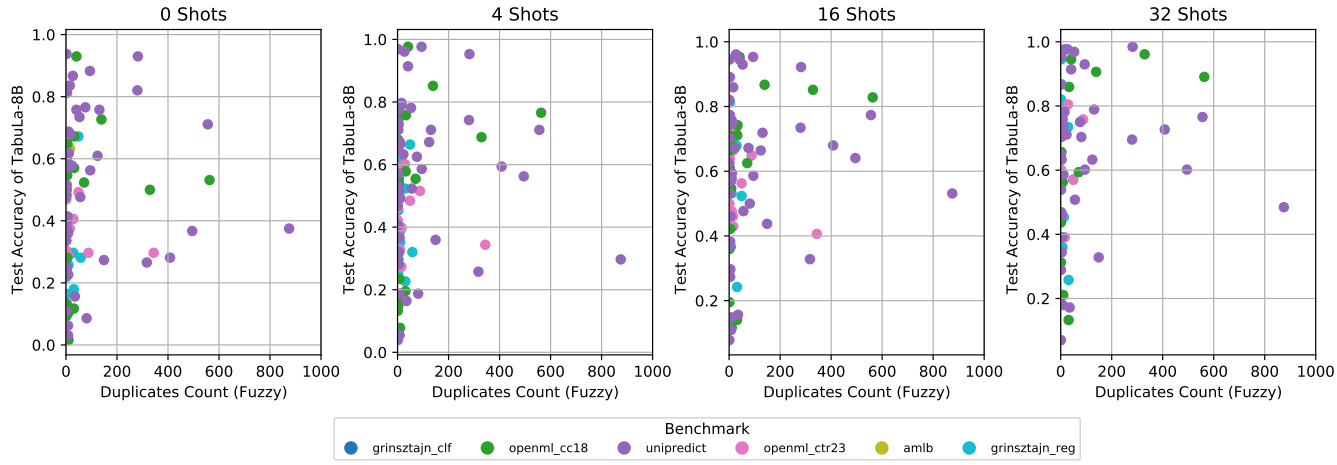


Figure C.3: Contamination rates vs. accuracy across varying numbers of shots. We find no clear relationship between contamination in the training set and downstream task performance.

Paper or other resource for more information: This paper is the primary resource for information about TabuLa-8B. Implementation details can also be found at the open-source code release associated with the project.

Citation details: See the first page of this paper.

License: The model uses the Meta Llama 3 license (see https://llama.meta.com/llama3/license/).

Where to send questions or comments about the model: Send questions or comments directly to the corresponding authors, or file issues on the project git repo.

C.8.2 Intended Use

Primary intended uses: This is a research-only release. The primary intended use of this model is for research on tabular data modeling, or for research applications on tabular data.

Primary intended users: The primary intended users are scientific researchers interested in understanding, training, and applying tabular foundation models.

Out-of-scope use cases: Commercial use, use of the model to attempt to identify, harm, or violate the privacy of individuals represented in the training data, and any other behavior that violates the Meta Llama 3 license is out of scope.

C.8.3 Factors

Relevant factors: The original Model Cards paper [127] identifies factors as "groups, instrumentation, and environments" relevant to summaries of model performance. One group relevant to our models' performance is the task type (classification vs. binned regression). We report performance on these tasks separately; our results are discussed in Section 4.5. Broadly, we find that TabuLa-8B's overall performance profile relative to baselines is similar for both classification and binned regression tasks. Similarly, the different benchmarks may be viewed as different *environments*, each testing a different type of dataset. For example, UniPredict tests performance on datasets with informative headers; OpenML-CC18 tests performance on datasets without such headers and where traditional supervised learning methods can be tuned to good performance; Grinsztajn tests performance on datasets where GBDTs tend to perform best; and AMLB tests performance

on tasks including free-form text. Our main results show that TabuLa-8B's overall performance relative to baselines is similar across these tasks; we analyze the differences in detail in the paper.

Evaluation factors: Evaluating language models (LMs) is different from evaluating standard supervised learning methods: while the latter directly output a score or probability over the set of target labels, LMs only output next-token probabilities over their vocabularies; as a result, predicted probabilities are not directly available (although these can be obtained through the use of various heuristics). In order to avoid introducing additional degrees of freedom into the evaluation process, we do not use score-based evaluation methods that rely on evaluating predicted probabilities; we only evaluate based on exact matching (as in several works both in the tabular literature [50, 84] and in the broader language modeling literature [39, 3]. As a consequence, our evaluation does not use metrics which are sometimes used to evaluate tabular classification models, such as Area Under the Receiver Operating Characteristic Curve (AUC).

C.8.4 Metrics

Model performance measures: Our primary evaluation measures are based on accuracy. We use exact-match accuracy for language model generations, and top-1 accuracy for supervised learning model predictions.

Decision thresholds: We use top-1 accuracy for supervised learning model predictions, but do not apply a specific threshold.

Variation approaches: N/A

C.8.5 Evaluation Data

Datasets: We use a suite of five previously-proposed tabular benchmarks, comprising a total of 329 tables. Our evaluation datasets are described in detail in Sections 4.5.2 and C.4.1.

Motivation: Using preexisting benchmark datasets allows us to compare the performance of our models to prior work in the tabular prediction literature. Additionally, using high-quality, curated benchmarks ensures that we are able to make reliable conclusion about overall model quality and performance relative to baselines.

Preprocessing: Our preprocessing is described in Sections 4.5.2 and C.4.1. We perform minimal pre-

processing on the datasets (no one-hot encoding, standardization, etc.) except for the logistic regression baseline, which requires this for best performance.

C.8.6 Training Data

Our training data is described in 4.4, with further details in the supplementary.

C.8.7 Quantitative Analyses

Unitary results: Our unitary results are summarized in Section 4.5. We provide detailed analysis in the supplementary section and give per-dataset results in Section C.7.

Intersectional results: We do not explicitly investigate tasks which include sensitive attributes, and so do not consider intersectional analysis in this work. We call for future work understanding the fairness properties of tabular foundation models in the future work (and our first-of-its-kind model will enable such research).

C.8.8 Ethical Considerations

There are important ethical considerations of both the data and model presented in this work. We discuss these in our Impact Statement.

C.8.9 Caveats and Recommendations

This model is for research use only. We recommend that more thorough research on both the impact of tabular training datasets, and the downstream performance of fine-tuned language models, be conducted before the deployment of tabular foundation models for real-world decisionmaking deployments.

C.9 Datasheet

C.9.1 Motivation

For what purpose was the dataset created?

The dataset was created for training tabular data foundation models, serving a purpose similar to C4 [148] or other large-scale corpuses in the natural language processing community.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The dataset was created by the authors of this paper in their roles at the institutions listed in the affiliations section of this paper.

Who funded the creation of the dataset?

No funding was provided with the explicit purpose of creating this dataset. However, JG was supported by a Microsoft Grant for Customer Success. JCP was supported by the Harvard Center for Research on Computation and Society.

C.9.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?

The instances that comprise the dataset represent tables extracted from the web (or individual rows of tables, depending on the downstream use of the data). All tables are publicly available and extracted from the Internet; in particular, all tables are available in the TabLib dataset from which T4 is filtered.

How many instances are there in total (of each type, if appropriate)?

The dataset consists of 3.1M tables where each table has many rows. The total number of rows across tables is 1.6B.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?

The dataset consists of a deterministically filtered subset of the TabLib dataset.

What data does each instance consist of?

The data consists of tables drawn from Github and CommonCrawl, as initially captured by the TabLib authors.

Is there a label or target associated with each instance?

Not by default. As part of our work, we select a target at random from filtered subset of the columns for each dataset. See C.1.2.

Is any information missing from individual instances?

Yes, many tables contain missing values for certain rows and columns.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?

In some cases, the tables have informative column headers describing the relationships between features for individual rows.

Are there recommended data splits (e.g., training, develop- ment/validation, testing)?

Not by default. We implement these as part of our training.

Are there any errors, sources of noise, or redundancies in the dataset?

We deduplicate the T4 dataset so that each table appears at most once. However, as is the case with most internet scale datasets, many of the tables contain noisy values whose correctness we do not manually inspect.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?

It is self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non- public communications)?

We do our best to remove any kind of data that might be considered confidential or personally identifying. If someone finds tables with confidential information that still remain, we would appreciate if they contact us so we might remove them.

Does the dataset contain data that, if viewed directly, might be offen- sive, insulting, threatening, or might otherwise cause anxiety?

It is possible that there are tables with information that might be anxiety inducing. We do not explicitly filter for this type of information, but believe it is not common in our dataset.

Does the dataset identify any subpopulations (e.g., by age, gender)?

The instances (table rows) in the data represent a variety of entities, and the majority of these do not represent persons. However, for the subset of tables where each row does represent an individual person, it is possible that the dataset does identify subpopulations.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?

While we aim to reduce obviously personally identifying data, we do not use techniques like differential privacy to formally defend against reidentification attacks.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals race or ethnic origins, sexual orienta- tions, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?

We aim to remove this kind of information.

Any other comments?

C.9.3 Collection Process

How was the data associated with each instance acquired?

The data was filtered from the original TabLib dataset [58].

What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)?

The data was filtered programatically according to hand picked heuristics as described in 4.4.

If the dataset is a sample from a larger set, what was the sam- pling strategy (e.g., deterministic, probabilistic with specific sam- pling probabilities)?

It was deterministically chosen according to hand coded filtering rules.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

The authors performed the data collection process. No external crowdsourcing or contractors were employed.

Over what timeframe was the data collected?

The original TabLib dataset was collected in 2023 by the original authors and contains tables published from a wide range of years. Our filtering was conducted during the spring of 2024.

Were any ethical review processes conducted (e.g., by an institu- tional review board)?

No.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

Data was filtered from TabLib and hence not collected directly.

Were the individuals in question notified about the data collection?

We notified the TabLib authors of our effort, but not the owners of the publicly available tables in the original corpus.

Did the individuals in question consent to the collection and use of their data?

To the best of our knowledge, the data scraped in TabLib did not have a consent procedure.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?

NA

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?

No.

C.9.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?

Yes.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?

The raw data is available as part of the TabLib release [58].

Is the software that was used to preprocess/clean/label the data available?

Yes, all the code used to filter the original corpus is available as part of our open source release.

C.9.5 Uses

Has the dataset been used for any tasks already?

We are not aware of any other uses of T4 apart from the training of TabuLa-8B.

Is there a repository that links to any or all papers or systems that use the dataset?

Models trained on the dataset can be found using the Hugging Face dataset page.

What (other) tasks could the dataset be used for?

The dataset could be used to train generative models, LLM data science assistants, amongst others.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?

Our filtering choices were optimized to train a tabular prediction (classification) model and may lead to suboptimal behavior for other tasks.

Are there tasks for which the dataset should not be used?

The dataset should not be used to try and identify private individuals.

C.9.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?

The dataset will be made publicly available on HuggingFace.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?

It will be published at HuggingFace.

When will the dataset be distributed?

June 2024

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?

The dataset is subject to the same usage and copyright restrictions as the original TabLib release.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?

Yes, the original TabLib authors place usage restrictions. See [] for more details.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?

The dataset is subject to the same usage and copyright restrictions as the original TabLib release.

C.9.7 Maintenance

Who will be supporting/hosting/maintaining the dataset?

The authors of the paper.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

Please contact jpgard@cs.washington.edu.

Is there an erratum?

The current manuscript, as published on the arxiv server, will serve as the main source of documenting errors.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?

We do not foresee any updates.

If the dataset relates to people, are there applicable limits on the re- tention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)?

NA

Will older versions of the dataset continue to be supported/hosted/maintained?

NA

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?

Others are free to build on the dataset as long as they adhere to the original terms of use put forth by [58].