Subgroup Robustness Grows On Trees: An Empirical Baseline Investigation

Josh Gardner **Zoran Popović** **Ludwig Schmidt**

Paul G. Allen School of Computer Science & Engineering
University of Washington
{jpgard, zoran, schmidt}@cs.washington.edu)

Abstract

Researchers have proposed many methods for fair and robust machine learning, but thorough empirical evaluation of their subgroup robustness is lacking. In this work, we address this gap in the context of tabular data, where sensitive subgroups are clearly-defined, real-world fairness problems abound, and prior works often fail to compare to state-of-the-art tree-based models. We conduct an empirical comparison of several previously-proposed methods for fair and robust learning alongside state-of-the-art tree-based methods and other baselines. Via experiments with more than $340{,}000$ model configurations on eight datasets, we show that tree-based methods have strong subgroup robustness, even when compared to robustness- and fairness-enhancing methods. Moreover, the best tree-based models tend to show good performance over a range of metrics, while robust or group-fair models can show brittleness, with significant performance differences across different metrics for a fixed model. We also demonstrate that tree-based models show less sensitivity to hyperparameter configurations, and are less costly to train. Our work suggests that tree-based ensemble models make an effective baseline for tabular data, and are a sensible default when subgroup robustness is desired.

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 [46]. 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 [40, 62, 31]. 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 share the (sometimes implicit) criteria that models should have low *disparity* (variation in performance over all subgroups should be minimized; cf. [71, 17, 30, 39]) and good *worst-group performance* (the lowest accuracy over all subgroups should be maximized; cf. [70, 58]) – that is, 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 [34, 60]. Second, tabular data often directly encodes sensitive attributes with respect to which subgroup robustness is desired (race,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

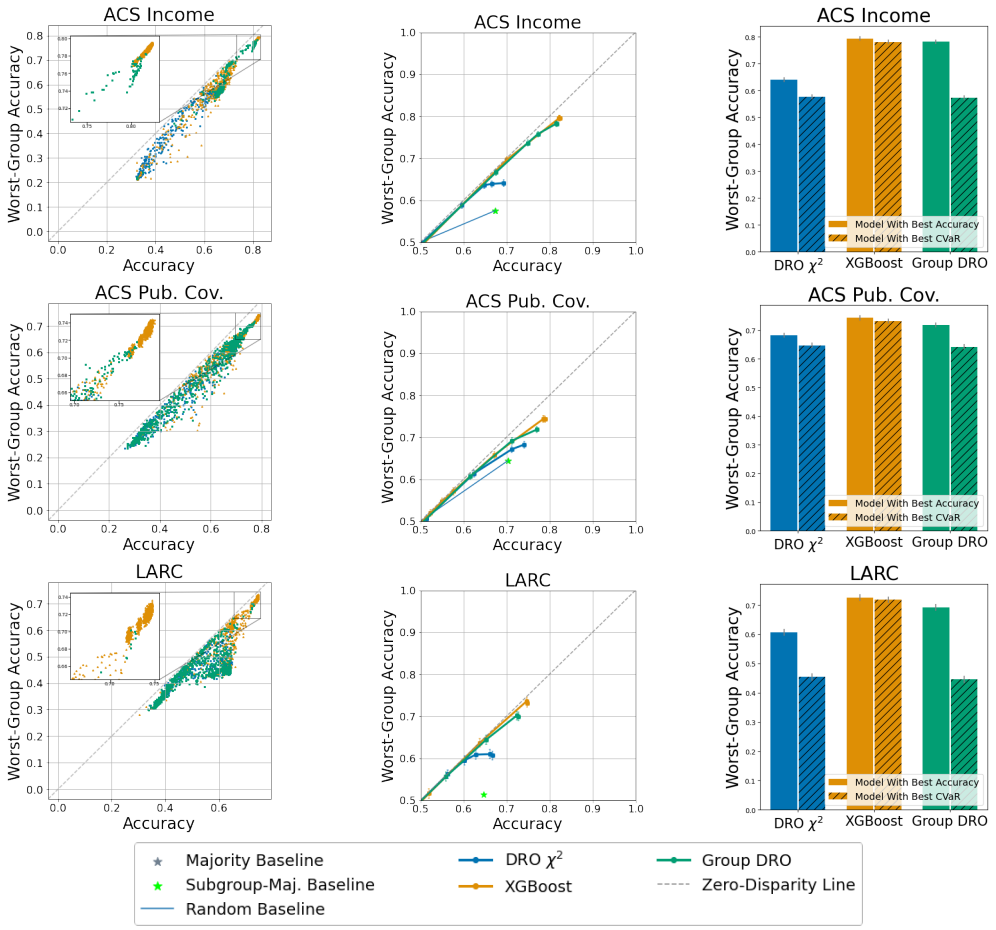

Figure 1: Results from three datasets in our study. (a) Left: Tree-based methods such as XGBoost show similar subgroup robustness, with sometimes better overall performance, as robustness-enhancing or disparity-mitigation methods. (b) Center: Model performance frontiers corresponding to (a) show similar accuracy-robustness frontiers for XGBoost and DRO/Group DRO. (c) Right: Tree-based methods' worst-group accuracy is robust to model selection metrics.

gender, income level), and these attributes are often declared by the individuals represented in the dataset. Finally, tabular 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 [43]. We 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 [34]. 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 performance "frontiers" for each technique. 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: We show that modern tree-based methods are surprisingly robust to subgroup shift despite lacking any explicit robustness intervention, achieving competitive or better performance against state-of-the-art methods for fairness and robustness (Fig. 1a,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 (Fig. 1b,d) and hyperparameter sensitivity analysis (Figs. 8,9).

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 (Fig. 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.

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 Related Work

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 [5, 18]. 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* [69, 15] 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, 68], or including explicit regularizers designed to encourage fairness [7, 11, 39, 52]. *Postprocessing* [29, 49] operates on the predictions of a model, modifying them to achieve fairness criteria.

The impact of fairness under subpopulation shift is analyzed in [44], which shows theoretically that enforcing fairness during training can harm or improve the model, under certain conditions, and [61] conducts a causal analysis of fairness under covariate shift. Perhaps the work most closely related to the current study is Friedler et al. [23], 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. [23] 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 Distributionally Robust Learning

While efforts in robust optimization date back several decades [10], 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 [32, 42, 58, 57, 70]. 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 [70, 58, 30].

A particular form of shift (and robustness) evaluated by these works is *subgroup shift* (also called subpopulation shift) [46], 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 [14].

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 [34, 13, 60]. 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 [28, 50, 60, 13]. This includes GBM [24, 25], LightGBM [36], XGBoost [16], and CatBoost [20, 51], 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 [34, 2, 33, 50, 35]. 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, [60] and [13] 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 [26, 33].

The overall predictive performance of these tabular methods is evaluated in e.g. [13, 26, 34, 50, 60]. 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. [32, 39, 70, 58]). Notably, [1] evaluates GBM in conjunction with their proposed inprocessing method, but does not evaluate subpopulation robustness; [19] evaluates GBM with fairness methods but does not compare to robust methods and does not tune the GBM with fairness methods.

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.

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 [18] and robustness [70, 39], 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

$$\mathcal{L}_{\text{WorstGroup}} \coloneqq \max_{a \in \mathcal{A}} \mathbb{E}_{D \sim \mathcal{P}}\left[\mathcal{L}_a(f_\theta)\right] \quad \text{and} \quad \mathcal{L}_{\text{DISP}} \coloneqq \max_{a, a' \in \mathcal{A}} \mathbb{E}_{D \sim \mathcal{P}}\left[|\mathcal{L}_a(f_\theta) - \mathcal{L}_{a'}(f_\theta)|\right] \quad (1)$$

. 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 [46]).

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 B.

Fairness-enhancing models: We evaluate the LFR preprocessing method of [69]; the inprocessing method of[1], and the postprocessing method of [29]. As in [1, 19], we use GBM as the base learner for each model; however, unlike [19], we also tune the base learner parameters.

Robust models: We evaluate both the CVaR and χ^2 constraint forms of DRO via [42]; the CVaR and χ^2 constraint forms of DORO [70]; Group DRO [58]; Marginal DRO [22]; and Maximum Weighted Loss Discrepancy [39]. Each robust optimization method is used with a multilayer perceptron (MLP), as in most previous works when using tabular data, e.g. [39, 42, 58, 70].

Tree-based models: We evaluate GBM, LightGBM [36], XGBoost [16], 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.

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 F.

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. [70, 34]), perform manual tuning ([42]), or do not tune hyperparameters of the base model when using fairness-enhancing techniques ([19]).

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 (1). Worst-group and disparity measures, for various formulations of the loss functions above, are a widely-used measure of fairness and robustness [12, 70, 58, 57, 30, 40]. 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:

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
BRFSS	175,745	28	Diabetes	Race, Sex
Comm. & Crime	1994	113	High Crime	Income Level, Race
COMPAS	7,215	10	Recidivism	Race, Sex
German Credit	1,000	22	Credit Risk	Age, Sex
LARC	169,032	26	At-Risk (Grade)	URM Status, Sex

Table 1: Overview of datasets used.

$$\mathcal{L}_{\text{CVaR}}(D, \mathcal{P}) := \sup_{q \in \Delta^N} \sum_{i=1}^{N} q_i \ell(f_\theta; x) \quad \text{s.t. } ||q||_\infty \leq \frac{1}{\alpha N} \tag{2}$$

(2) is the risk function optimized by the DRO CVaR model [42], but it is also used more widely as a measure of the tail risk of a classifier (cf. [70]).

Additional fairness and robustness metrics are reported in Section G and defined in Section C.

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 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.

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.

A summary of our results is shown in Figures 1, 2, 3, and 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 Pub. Cov., LARC), XGBoost achieves significantly *better* $\mathcal{L}$ and $\mathcal{L}_wg$ than DRO-based methods (Group DRO, χ^2 DRO), as indicated by nonoverlapping Clopper-Pearson CIs ($\alpha = 0.05$).

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 16, 17, and a complete set of model performance observations from each algorithm and dataset, are in Supplementary Section G.

To summarize our results over the large hyperparameter sweeps, we use *model performance frontiers* 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 1b and 1d, with the remaining datasets in Figures 3 and 4. The frontiers allow us to compare the best-possible tradeoff between accuracy and worst-group accuracy (Fig 3; higher is better) or between accuracy and accuracy disparity (Fig 4; lower is better).

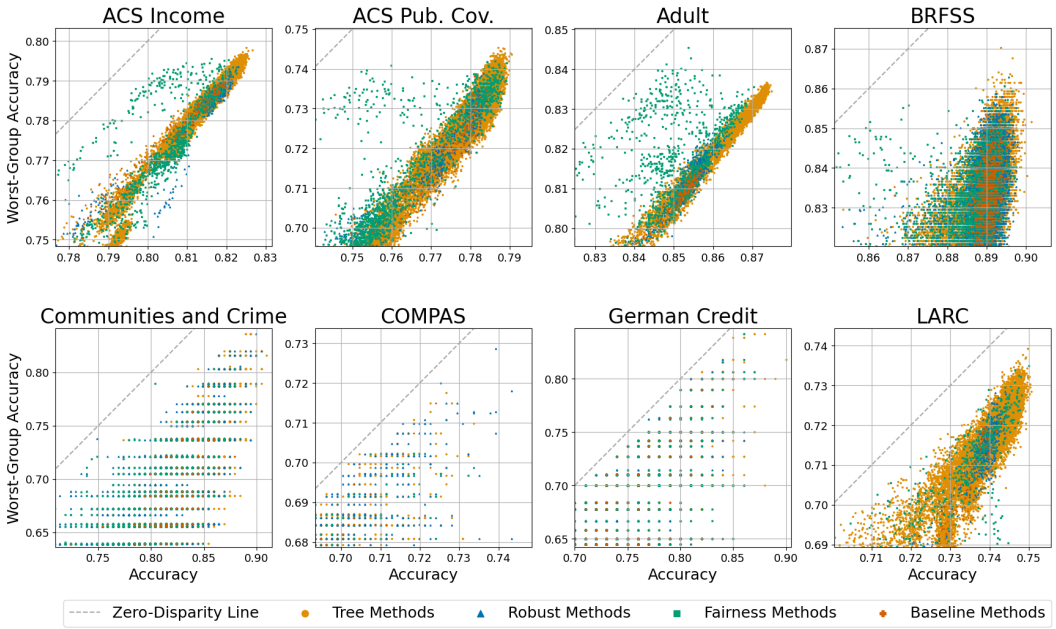

Figure 2: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and baseline models over eight tabular datasets. Dashed lines indicate $y = x$, when worst-group accuracy equal to overall accuracy (zero accuracy disparity). See Figures 12-14 for more detailed results by algorithm. Note: "discretization" artifacts are due to small test/dataset sizes.

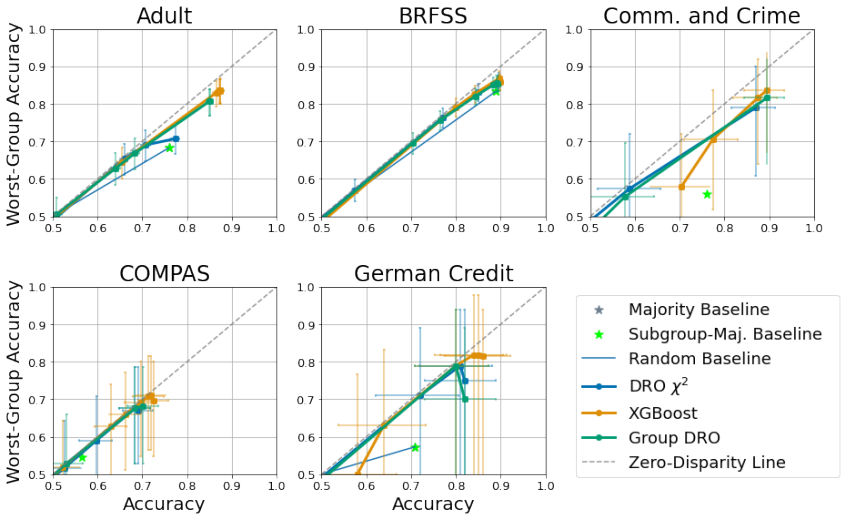

Figure 3: Model performance frontiers, formed by tracing the convex envelope of model performance. Tree-based models achieve comparable and sometimes improved frontiers with the highest-performing robustness methods. (See also Fig. 1 for the remaining three datasets)

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 these correlations on Adult in Fig. 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

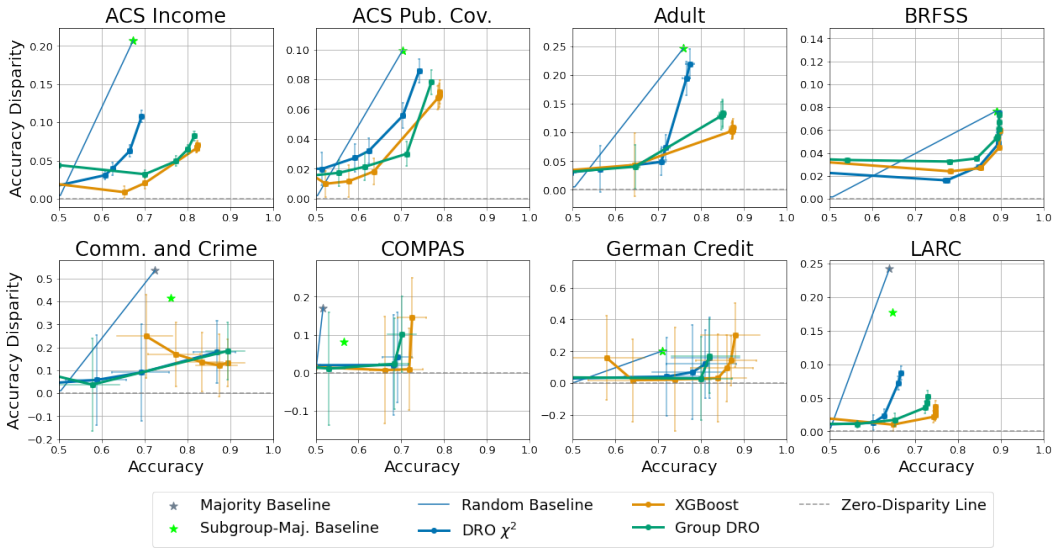

Figure 4: Model disparity frontiers, formed by tracing the convex envelope of model disparity. Tree-based models achieve comparable and sometimes improved frontiers compared to the highest-performing robustness methods, particularly in high-accuracy regions.

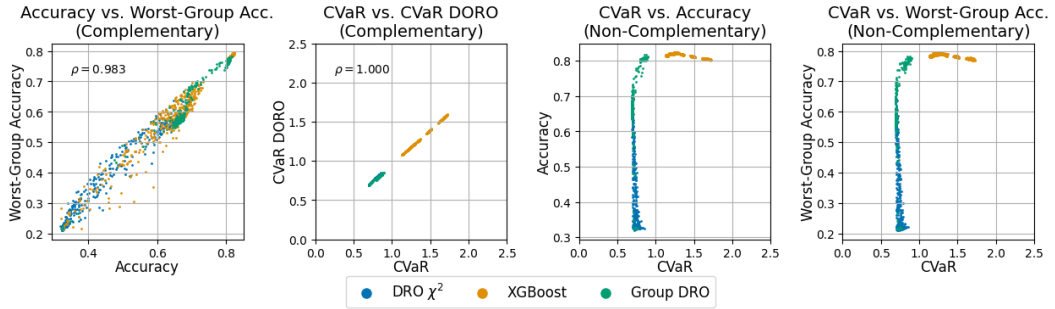

Figure 5: While "complementary" metrics (those measuring the same event with different conditioning) are closely correlated for all models, non-complementary metrics behave differently by model. Accuracy can vary widely for "robust" models with a fixed robust (CVaR) risk, while for tree models, the best (lowest) CVaR risk is typically associated with the best (highest) accuracy. ACS Income dataset shown; see Supplementary Figure 18 and §G for additional datasets and metrics.

complete set of correlations for each model and metric in Figure 18). 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 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 [70] – 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 6. While these results confirm the finding in previous works that fairness metrics are correlated [23], 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. [70]), 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

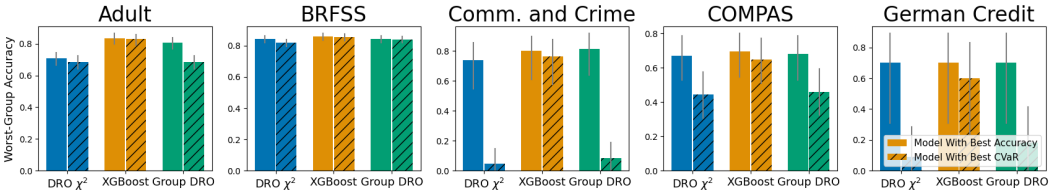

Figure 6: Worst-group accuracy for models with best overall accuracy (solid), and best CVaR (shaded). Tree-based models show better performance across non-complementary metrics (as indicated by similar heights of orange bars within each plot), whereas robustness- and fairness-based models do not (statistically-significant 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 Fig. 1).

problem is implicitly addressed in almost all previous works, the decision is often handled differently. For example, models are tuned and selected by hand [42], chosen based on worst-group performance [70], or based on robust risk metrics [58]. 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 6. Figure 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).

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 [CITES]. 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 7 Section F.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 19, show that tree-based models are also considerably less expensive to train and tune.

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

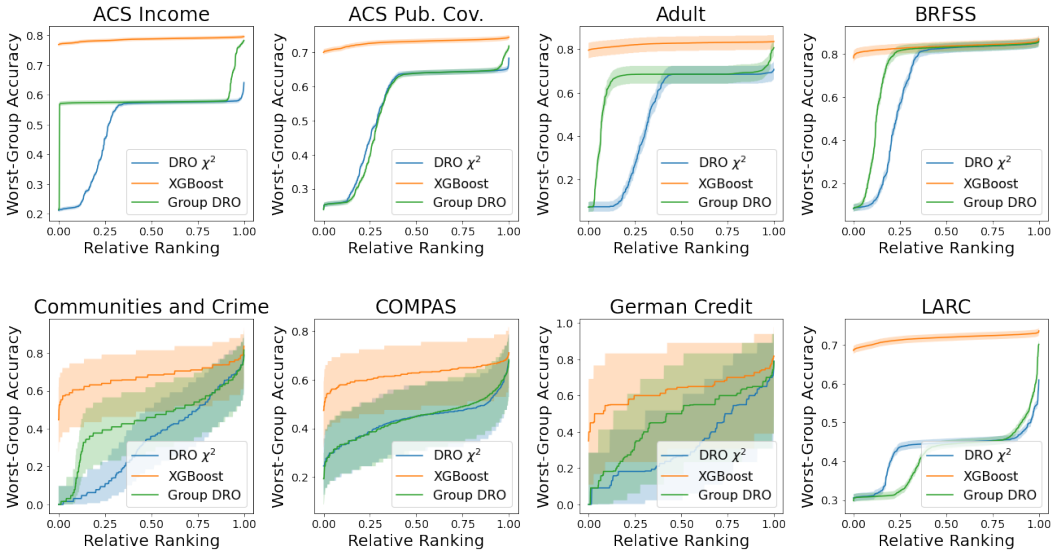

Figure 7: Performance over (truncated) hyperparameter grids on all datasets. Tree-based models (orange) show significantly less sensitivity to hyperparameter settings, for both subgroup and overall performance metrics (worst-group accuracy shown here), as indicated by nonoverlapping Clopper-Pearson CIs ($\alpha = 0.05$). For additional results and methodology for constructing these plots, see Section F.2.

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 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 [55, 63, 43], 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. [58, 70]). 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. 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 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 [45] 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.

8 Acknowledgements

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

This research was supported in part through 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.

References

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

[3] 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.

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

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

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

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

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

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

[26] 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.

[27] 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.

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

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

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

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

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

[47] 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.

[48] 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.

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

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

[51] 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.

[52] 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.

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

[54] 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.

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

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

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

[58] 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.

[59] 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.

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

[61] 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.

[62] 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.

[63] 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.

[64] 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.

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

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

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. F.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Our work is not concerned with point estimates, so we do not display error bars in e.g. our scatterplots. However, we do fix random seeds for reproducibility and provide extensive empirical results over a wide grid of experimental parameters in order to provide a sense of the variability of our findings over different models, datasets, and learning hyperparameters.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section E.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use existing datasets, all of which are piblicly available, and the sources are cited and linked.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code to reproduce experiments.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We discuss issues with data representation in our introduction (which motivates the use of tabular data), as well as potential concerns raised regarding COMPAS and other datasets in Section A.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A 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 [19], 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 [19], which we found to improve classifier performance. The sensitive feature for race is coded as "white alone" vs. other categories, as in [19]. 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 are race and gender. We use identical feature set to [19]. The sensitive features and subsampling are handled as in ACS Income.
- **Adult**[2]**:** A widely-used benchmark dataset derived from 1994 US Census data [41]. 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 [6][3], and we further split the set partition evenly into validation and test.
- **BRFSS:** The Behavioral Risk Factors and Surveillance Survey[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 [39, 37, 38], 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 [39] 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.

  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

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

[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`

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 [65], each stage of the COMPAS dataset's creation introduces opportunities for bias [56, 4] and that measurement biases and errors with COMPAS have been widely documented [3].

- **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.

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 [64]. 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 [59] nor fixed [27].

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" [3].

B Model Details

Fairness-Enhancing Models: Following [19], we evaluate one method each of pre-, in-, and postprocessing. The preprocessing method of [69] 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 [29], 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` [9] and `fairlearn` [12] for all fairness interventions.

None of the fairness methods encompasses a prediction model, so following [19], we pair each method (in-, pre-, and postprocessing) with a gradient-boosted tree (GBM) [24, 47]. However, unlike [19], we perform hyperparameter sweeps for both the fairness methods *and* the GBM.

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

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

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

$$\min_{\theta} \sup_{Q \in \mathcal{U}(D)} \mathbb{E}_{S \sim Q} \left[\mathcal{L}(f_\theta) \right] \tag{3}$$

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 (3). The first is the set of distributions with bounded likelihood ratio to D, such that (3) 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 [42] for these methods.

We also evaluate the "Distributional and Outlier Robust Optimization" (DORO) of [70], which adds an ϵ parameter to (3) 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 [58], seeks to minimize the worst-group loss by solving

$$\min_{\theta} \sup_{g \in \mathcal{G}} \mathbb{E}_{(x,y) \sim \mathcal{D}_g} \left[\mathcal{L}(f_\theta) \right] \tag{4}$$

. Finally, we evalute the Maximum Weighted Loss Discrepancy (MWLD) of [39]. 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 [39] to be related to group fairness and robustness to subgroup shifts.

Tree-Based Models: As we note in Section 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 [19], 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 [21, 66] and random Fourier features [53, 54, 67]. For all baseline methods, we use the implementation of [48].

C Additional Metrics

In addition to the metrics reported and defined in Section 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) which excludes the ϵ-largest-loss elements in D in an effort to avoid outliers. Formally, the DORO CVaR risk is:

$$\mathcal{L}_{\text{CVaR-DORO}}(D, \mathcal{P}, \epsilon) \coloneqq \inf_{D'} \mathcal{L}_{\text{CVaR}}(D, \mathcal{P}') : \exists \tilde{\mathcal{P}}' \quad \text{s.t. } \mathcal{P} = (1-\epsilon)\mathcal{P}' + \epsilon \tilde{P}' \tag{5}$$

where ϵ is a hyperparameter corresponding to the fraction of outliers in the dataset. We note that (5) 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. [70]).

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 [6]. 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:

$$\mathcal{L}_{DP-Diff} \coloneqq \max_{a,a'} \Big| P(f_\theta(x_i; a_i = a) = 1) - P(f_\theta(x_j; a_j = a') = 1) \Big| \tag{6}$$

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 [12].

D 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.

E Training Details

We train all models using the original optimizer, SGD, used in their original works [70, 42, 39, 58]. 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 [34]; 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.

F Hyperparameter Grids

F.1 Grid Definition

We detail the hyperparameter grids for each experiment in Table 2. 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 [19]), we tune the full grid of hyperparameters for the base model in addition to the hyperparameters for that method, as indicated in Table 2. We note that this is not always the case in previous works; for example, [19] 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.

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 ← ConvexHull(Input)
idxs ← [ ]                                          ▷ Initialize array of frontier points.
top_idx = argmax_{m_+ i^(2)} (m_i^(1), (m_i^(2))_i ∈ vertices   ▷ Add uppermost point to frontier
idx ← top_idx
idxs ← idxs + [idx]
m_curr^(1), m_curr^(2) ← vertices[idx]
next ← (|vertices| + 1)mod|vertices|
m_next^(1), m_next^(2) ← vertices[next]
while m_next^(1) < m_curr^(1) do                    ▷ Trace frontier counterclockwise
    idxs ← next
    idx ← (idx + 1) mod |vertices|
    next ← (next + 1) mod |vertices|
    m_curr^(1), m_curr^(2) ← vertices[idx]
    m_next^(1), m_next^(2) ← vertices[next]
end while
idx ← top_idx
next ← (|vertices| − 1) mod |vertices|
m_curr^(1), m_curr^(2) ← vertices[idx]
m_next^(1), m_next^(2) ← vertices[next]
while m_next^(1) > m_curr^(1) do                    ▷ Trace frontier clockwise
    idxs ← next
    idx ← (idx − 1) mod |vertices|
    next ← (next − 1) mod |vertices|
    m_curr^(1), m_curr^(2) ← vertices[idx]
    m_next^(1), m_next^(2) ← vertices[next]
end while
return idxs
```

Tree-based methods: For GBM and random forest, we use the implementation of [48]. For LightGBM we use the original implementation of [36][10]. For XGBoost we use the original implementation of [16][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. [70, 42]. We use the implementation of [70] for DORO and [42] 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, feature size, or both. For LFR, we fix Ax as in [69], and otherwise use the center portion

[10] `https://github.com/microsoft/LightGBM`
[11] `https://github.com/dmlc/xgboost`

of the grid from [19] which we found to be sufficient for our sweeps when also tuning the GBM parameters (which was not performed in [19]) across our datasets. Our grid for inprocessing uses the same constraints explored in [19] but also tunes the constraint slack and GBM parameters, which were not tuned in [19]. We use the implementation of [8] for LFR and postprocessing, and [12] for inprocessing.

F.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 8, 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 8, 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 9 which include all models; here we omit the confidence intervals due to space (although the intervals would have the same width as in Figure 8).

G 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.

G.1 Training Compute Cost

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

Figure 19 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 19 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 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.

[12] This makes the confidence intervals in 8 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.	$\{$Nystroem, RKS$\}$
		Loss	Squared Hinge
		γ	$\{0.5, 1.0, 2.0\}$
		Num. Components	$\{64, 128, 256, 512\}$
		Nystroem Kernel Degree	$\{2, 3\}$
		Nystroem Kernel	$\{$RBF, 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	$\{$sqrt, 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	$\{$Depthwise, 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	$\{1, 2, 4, 8, 16, 32, 64\}$
		Max. Depth	$\{$None, $2, 4, 8\}$
		Column Subsample Ratio (tree)	$\{0.4, 0.5, 0.8, 1.\}$
Robustness-Enhancing Methods			
DORO χ^2 ♣	12150	Uncertainty set size α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
		Outlier proportion ϵ	$\{0.001, 0.01, 0.1, 0.2, 0.3\}$
DORO CVaR ♣	12150	Uncertainty set size α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
		Outlier proportion ϵ	$\{0.001, 0.01, 0.1, 0.2, 0.3\}$
DRO χ^2 ♣	2835	Uncertainty set size α	$\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
DRO CVaR ♣	2835	Uncertainty set size α	$\{0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$
Group DRO ♣	1620	Group weights step size	$\{0.001, 0.01, 0.1, 0.2\}$
MWLD ♣	6075	L_2 penalty	$\{0, 0.1, 1\}$
		Loss variance penalty	$\{1e^{-3}, 1e^{-2}, 1e^{-1}, 1, 10, \}$
Fairness-Enhancing Methods			
Preprocessing ♢	2500	Ax	0.01
		Ay	$0.001, 0.01, 0.1, 1, 10$
		Az	$0.001, 0.01, 0.1, 1, 10$
Inprocessing ♢	1000	Constraint slack ϵ	$\{1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1\}$
		Constraint Type	$\{$DP, EO$\}$
		Max Iter.	200
Postprocessing ♢	100	No tunable hyperparameters	

Table 2: Hyperparameter grids used in all experiments. ♣: all MLP parameters also tuned. ♢: all GBM parameters also tuned.

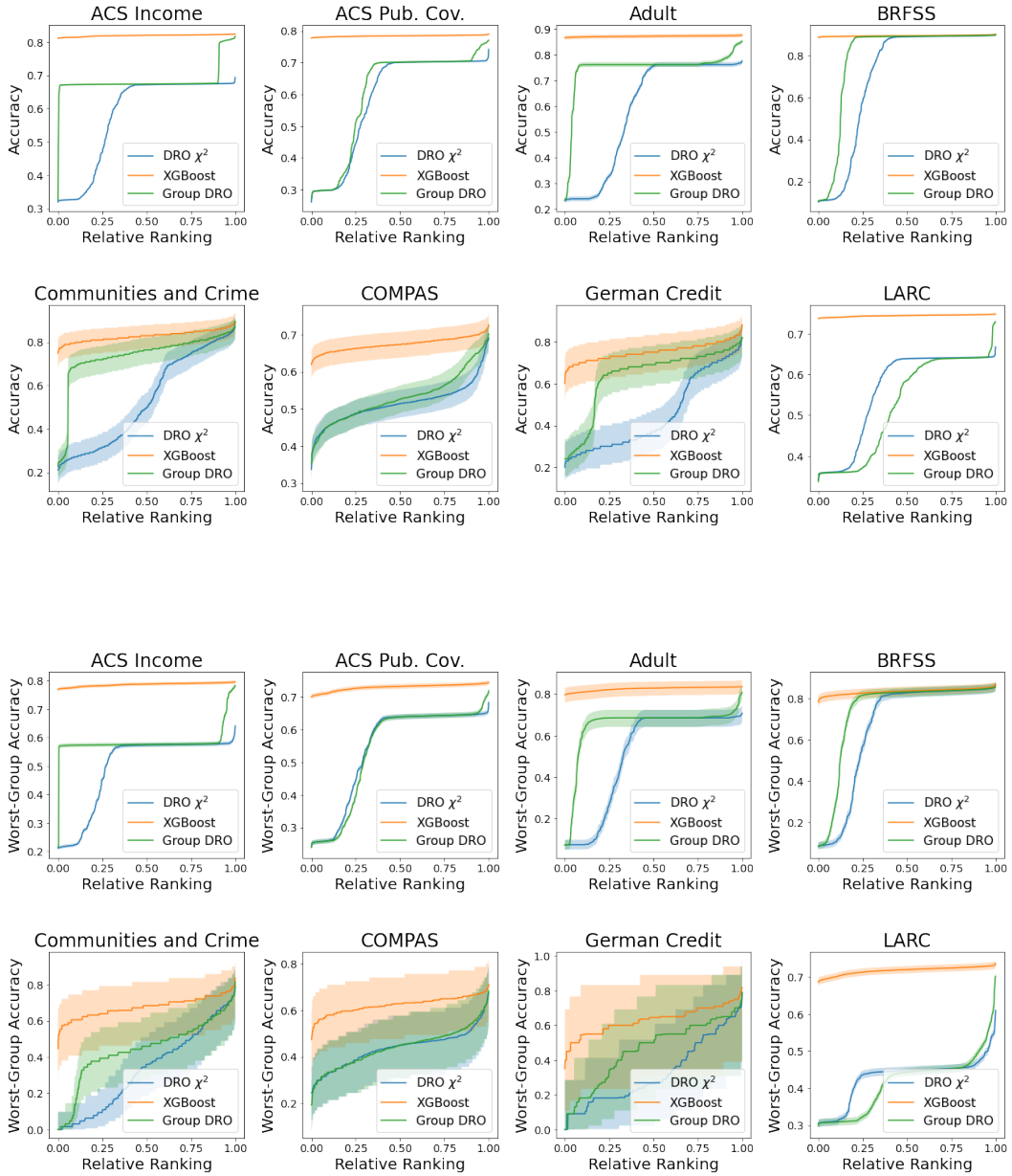

Figure 8: Hyperparameter sensitivity plots for χ^2 DRO, Group DRO, and XGBoost models. The top 8 panels show Accuracy; the lower 8 panels show worst-group accuracy (this is identical to Figure 7, reproduced here for clarity). XGBoost shows considerably lower sensitivity to hyperparameter tuning.

G.2 Peak Performance Summary

We provide the best performance per model, in terms of both overall accuracy and worst-group accuracy, in Tables 3 and 4, respectively.

G.3 Performance of Default Tree Hyperparameters

For the tree-based models in our summary, we give the performance of the default hyperparameters in Table 5.

	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 3: 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 4: 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 5: Performance of default hyperparameters for tree-based models.

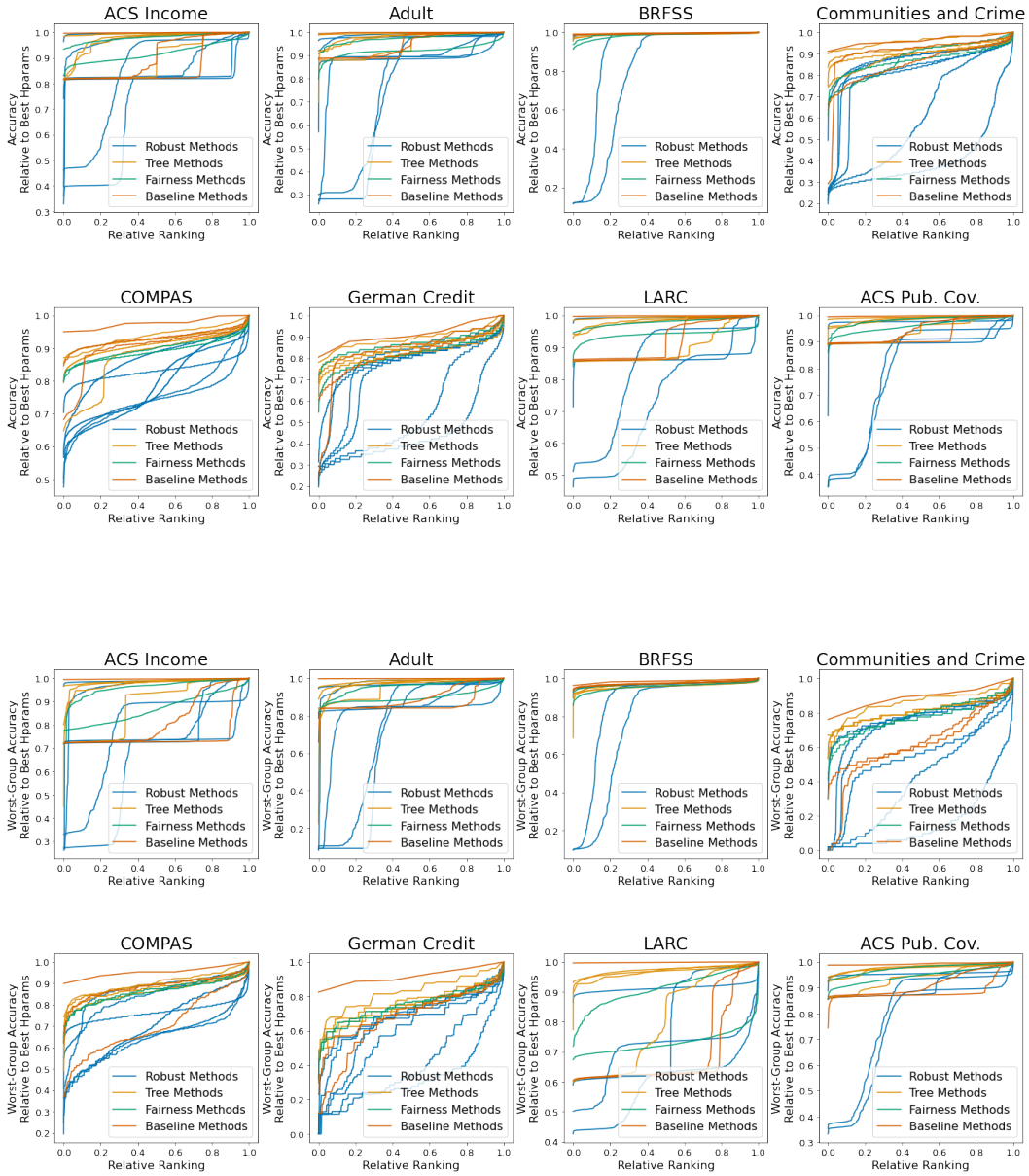

Figure 9: Hyperparameter sensitivity plots for all models evaluated. (Clopper-Pearson CIs omitted due to space).

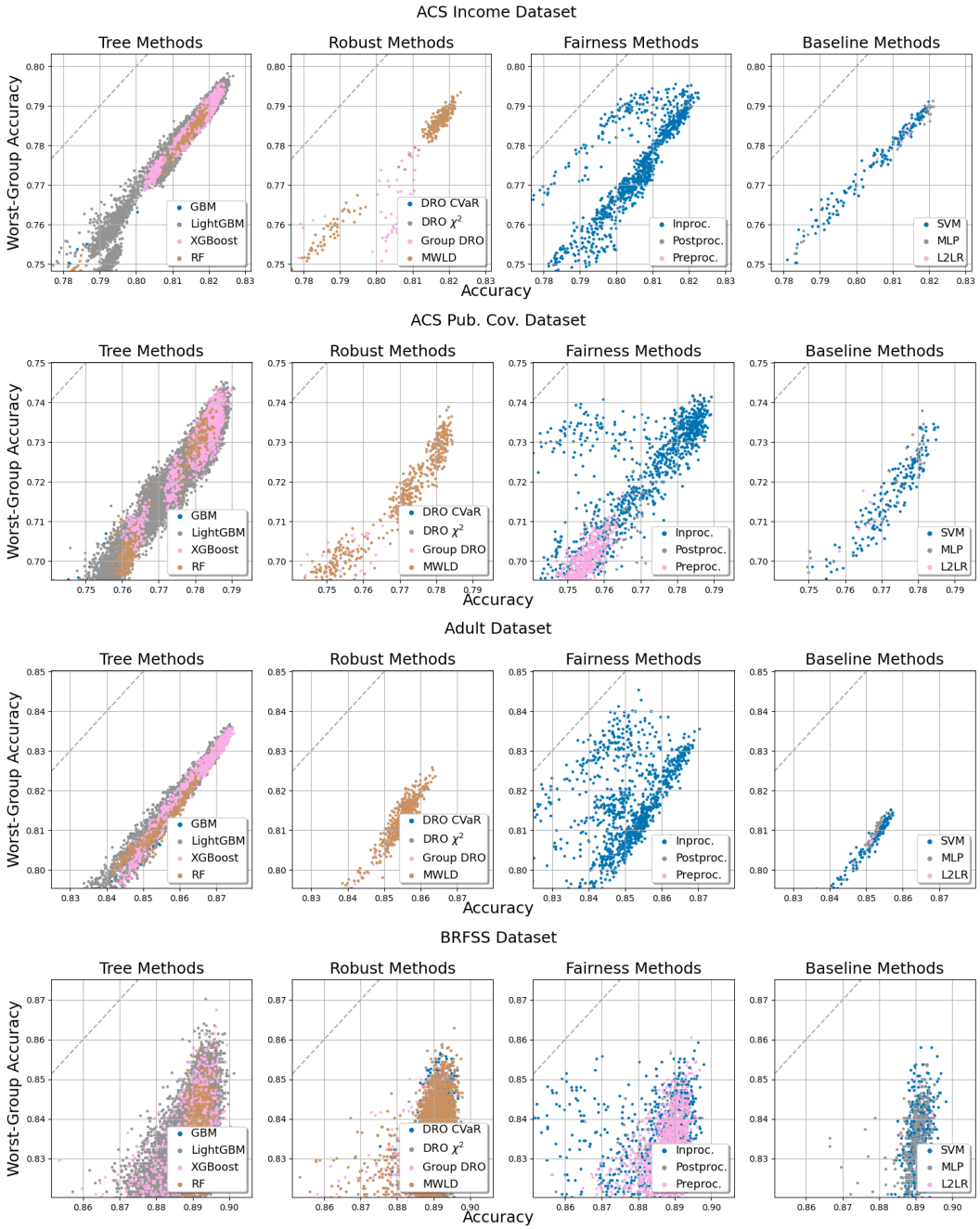

Figure 10: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and baseline models over eight tabular datasets (ACS Income, ACS Public Coverage, Adult, BRFSS). For results by individual algorithm, see Figures 12, 13, 14.

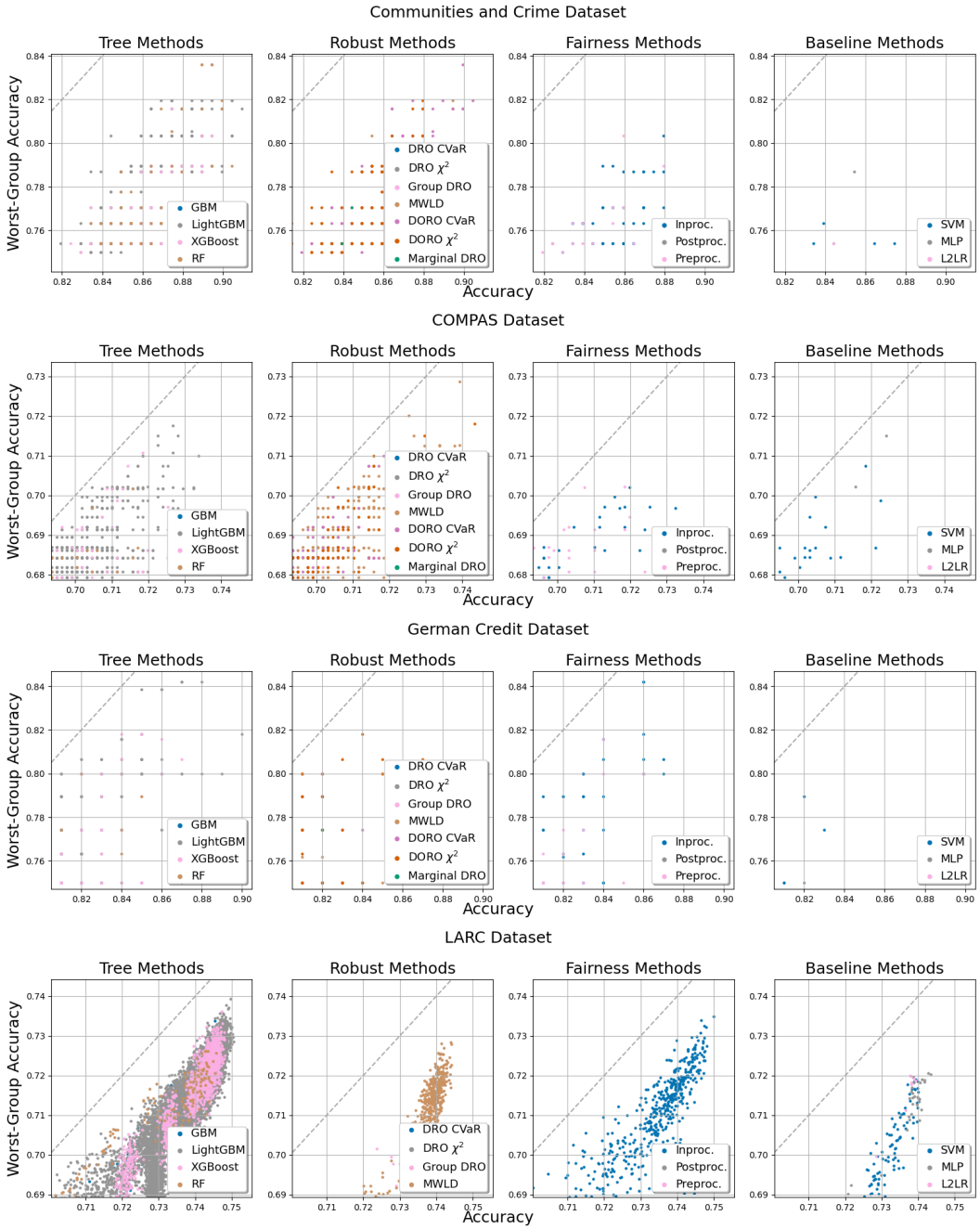

Figure 11: Overall Accuracy vs. Worst-Group Accuracy of robust, fairness-enhancing, tree-based, and baseline models over eight tabular datasets (Communities and Crime, COMPAS, German Credit, LARC). For results by individual algorithm, see Figures 12, 13, 14.

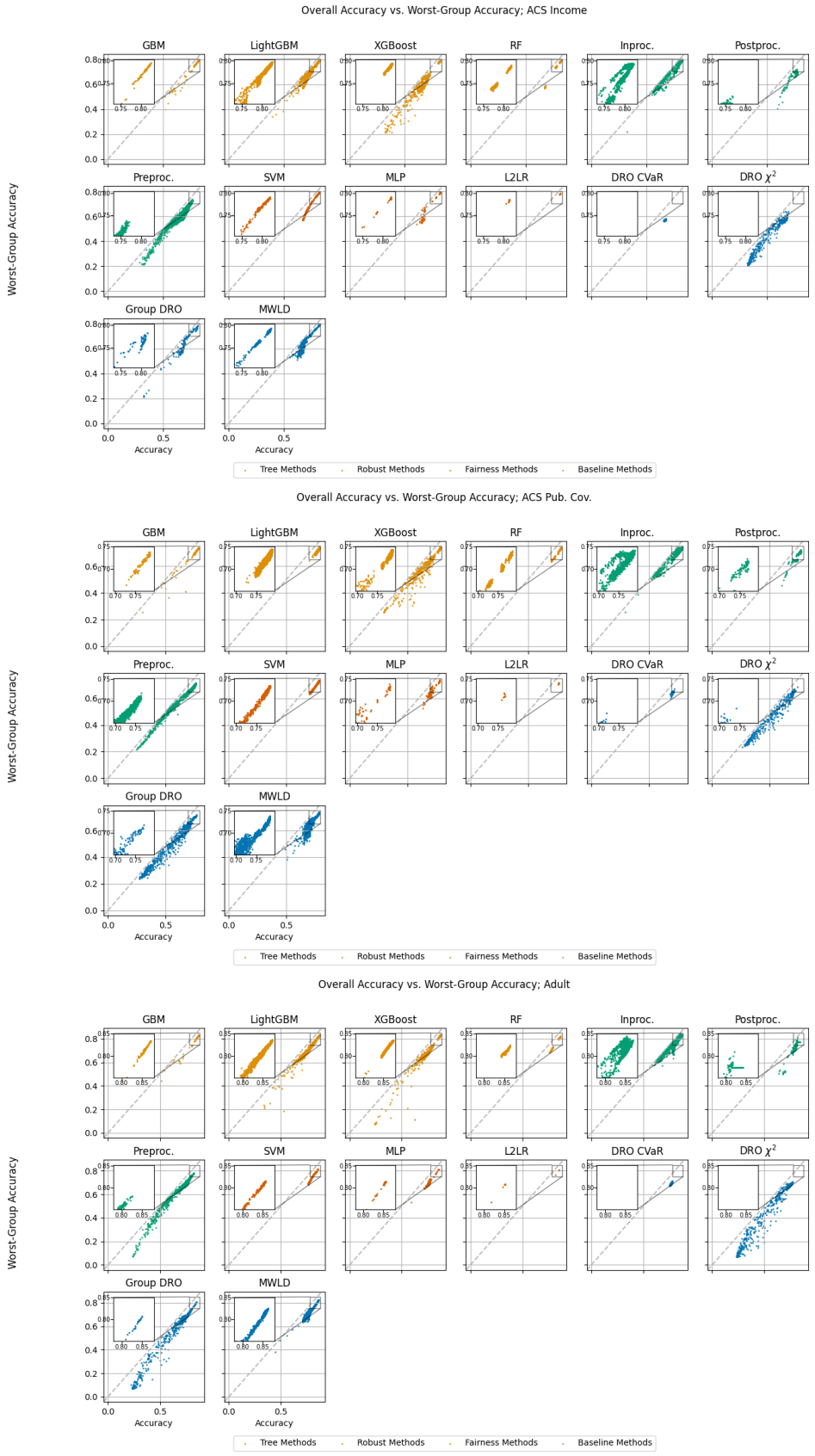

Figure 12: Detailed accuracy vs. worst-group accuracy for ACS Income, ACS Public Coverage, and Adult datasets.

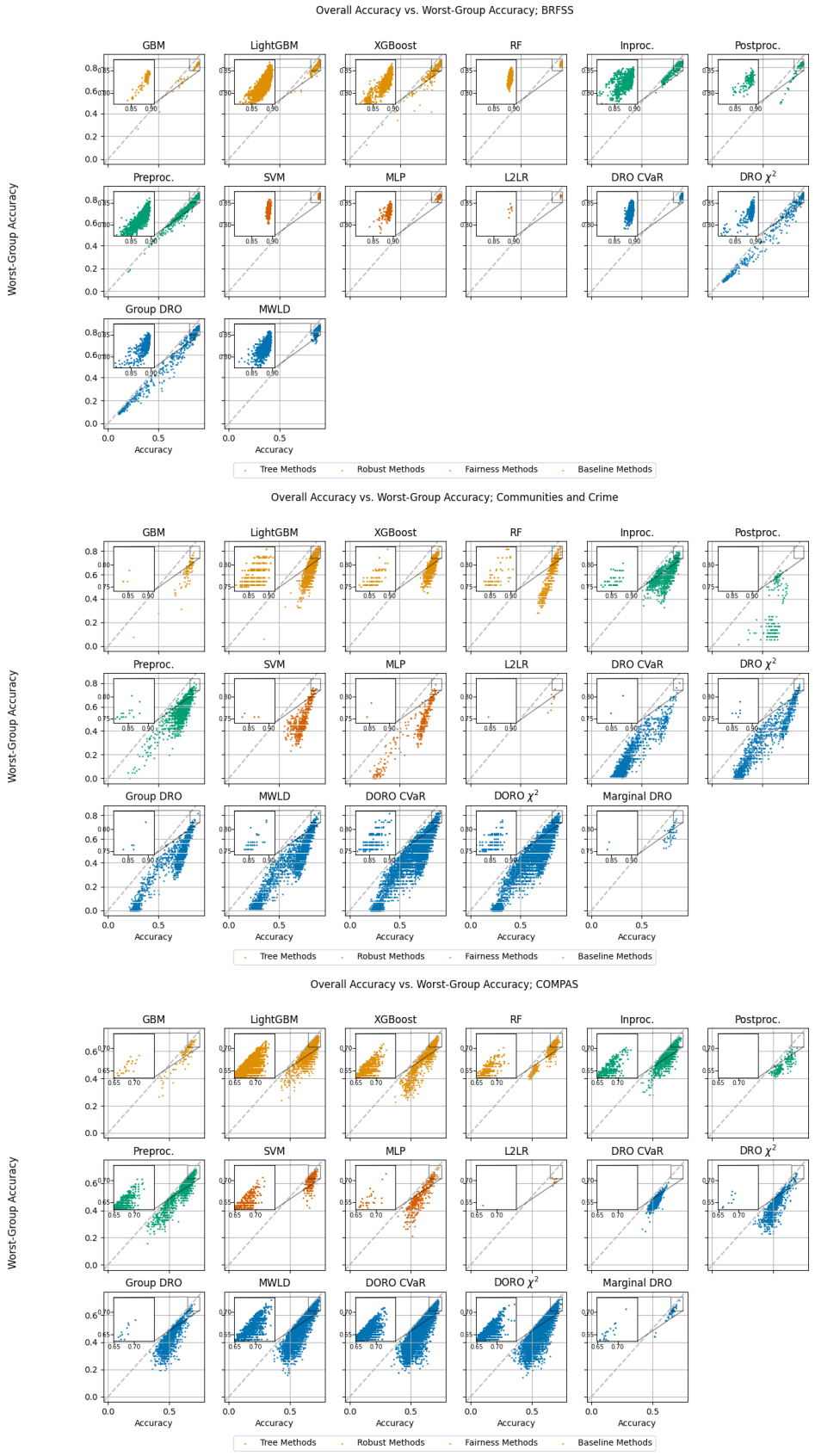

Figure 13: Detailed accuracy vs. worst-group accuracy for BRFSS, Communities and Crime, and COMPAS datasets.

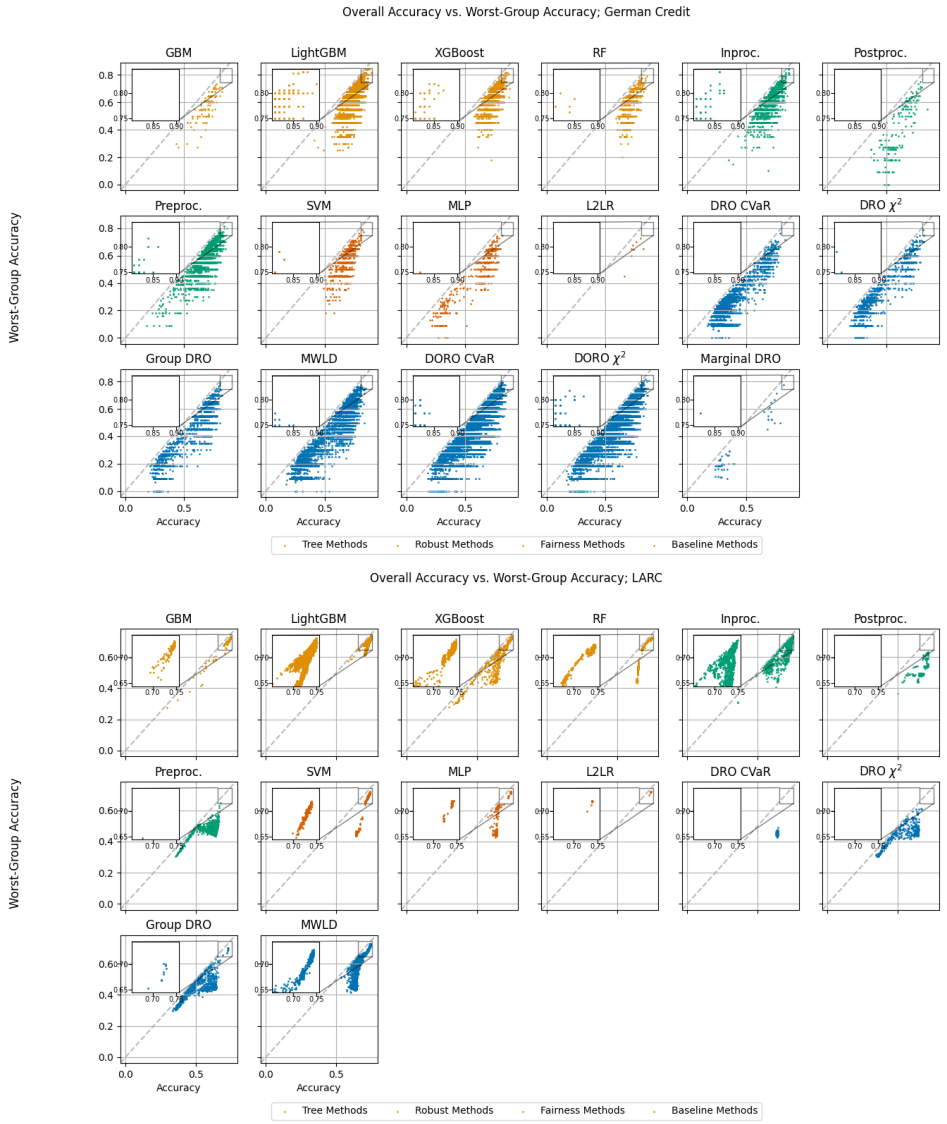

Figure 14: Detailed accuracy vs. worst-group accuracy for German and LARC datasets.

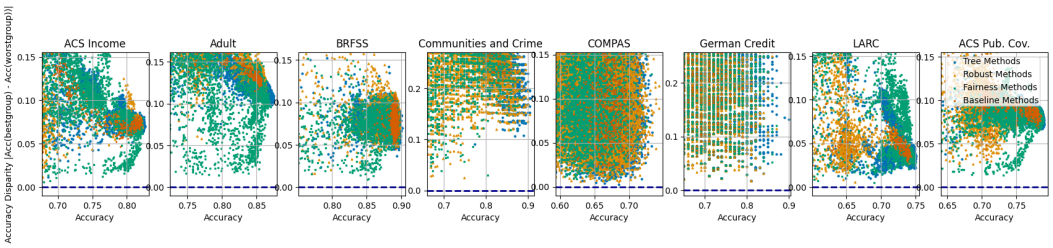

Figure 15: Overall Accuracy vs. Accuracy Disparity of robust, fairness-enhancing, tree-based, and baseline models over eight tabular datasets. Dashed lines indicates zero accuracy disparity. See Supplementary figure 10 for more detailed results by algorithm.

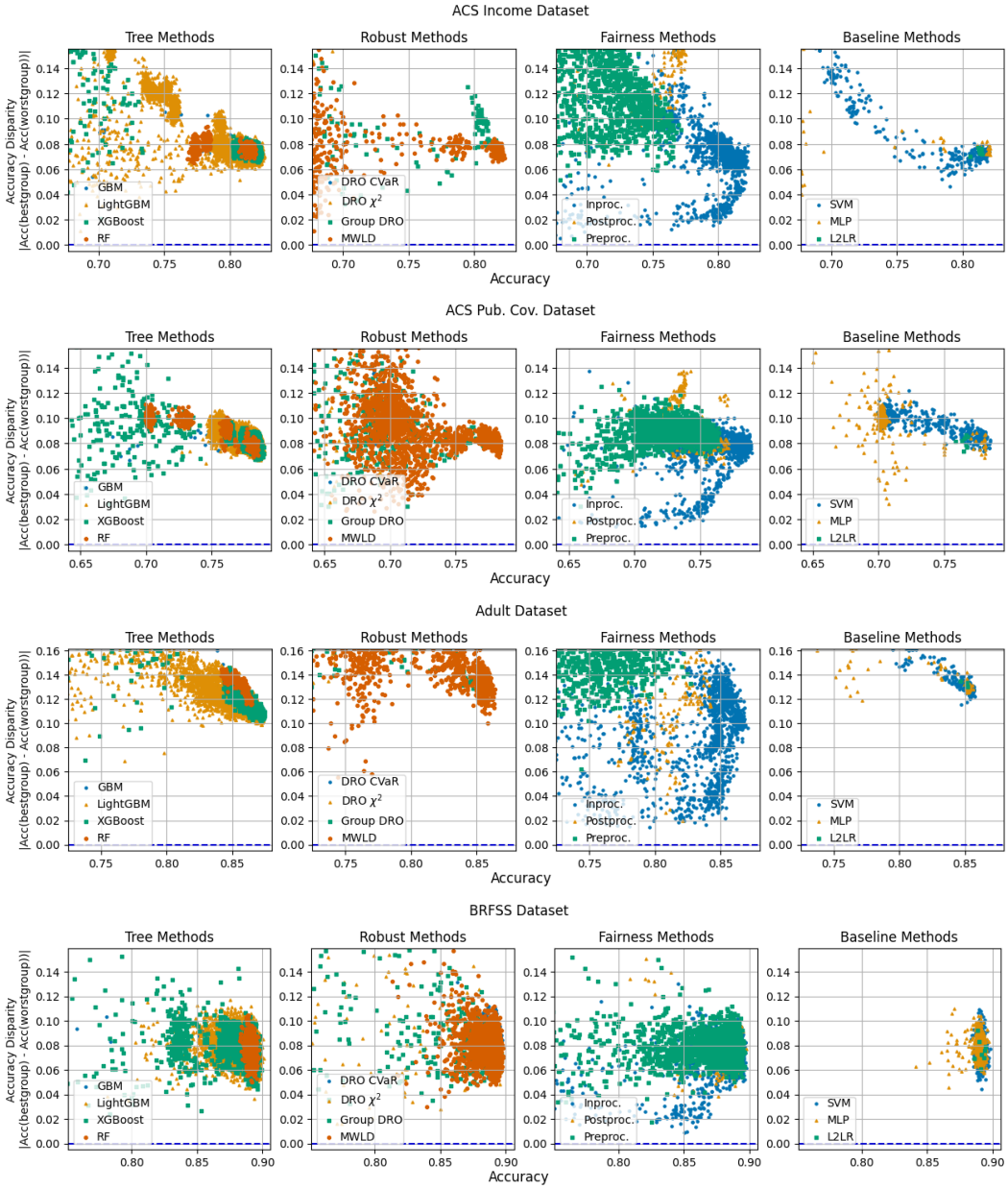

Figure 16: Overall Accuracy vs. Accuracy Disparity of robust, fairness-enhancing, tree-based, and baseline models over datasets ACS Income, ACS Public Coverage, Adult, BRFSS. See also Figure 17.

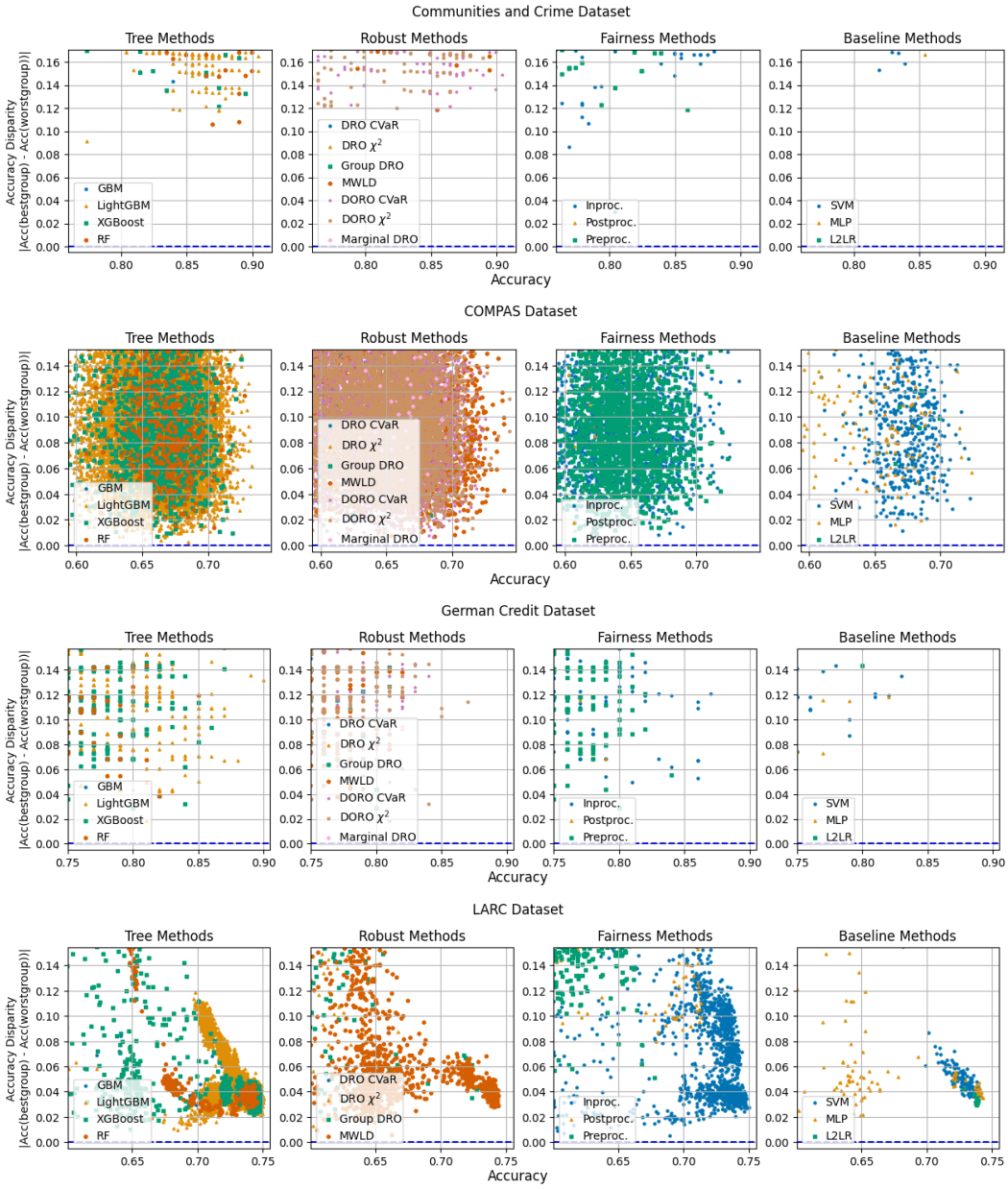

Figure 17: Overall Accuracy vs. Accuracy Disparity of robust, fairness-enhancing, tree-based, and baseline models over datasets Communities and Crime, COMPAS, German, LARC. See also Figure 16.

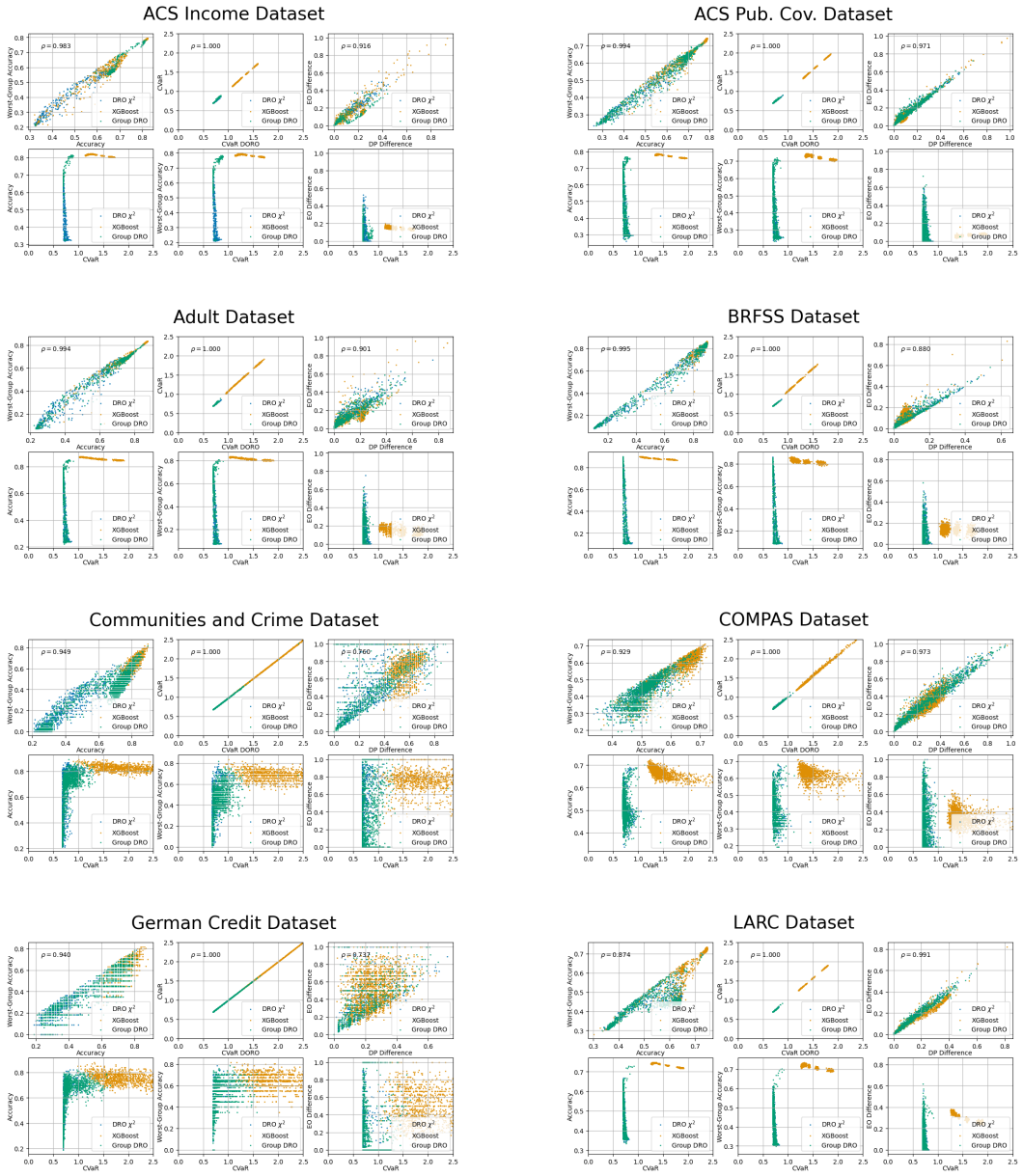

Figure 18: Correlation between complementary metrics (top row) and non-complementary metrics (bottom row) for each dataset, for DORO, XGBoost, and Group DRO models. Complementary metrics show strong correlations for all models, while non-complementary metrics do not. Pearson's r correlation coefficient for each pair of complementary metrics shown in the upper-left of each plot.

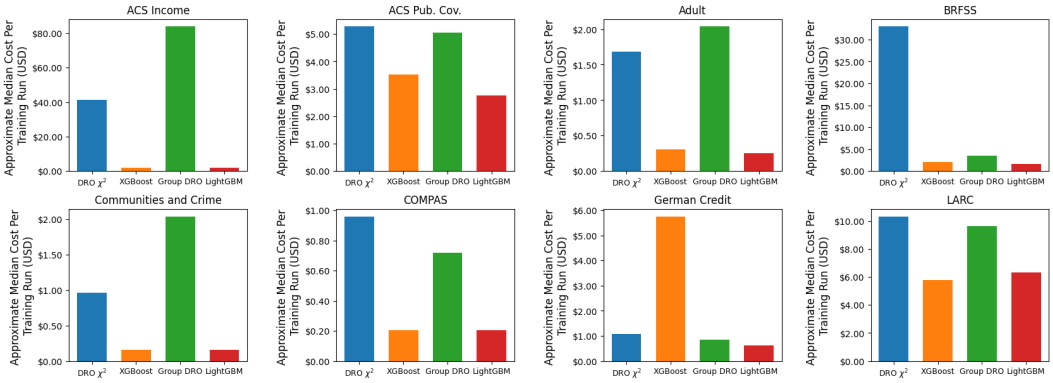

Figure 19: Estimated cost per training run, based on the median train time over the iterations in our study and the price of cloud-based computing infrastructure.