A POST-PROCESSING FRAMEWORK FOR GROUP FAIRNESS

BY

RUICHENG XIAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2025

Urbana, Illinois

Doctoral Committee:

Assistant Professor Han Zhao, Chair
Professor Arindam Banerjee
Professor Hanghang Tong
Professor Aaron Roth, University of Pennsylvania
Assistant Professor Gautam Kamath, University of Waterloo

ABSTRACT

Machine learning models are increasingly powering automated decision-making systems that influence everyday life, thanks to their ease of deployment. But this convenience belies the risk that, without proper oversight, they may cause disparate impacts across demographic groups. For instance, models trained on data shaped by historical inequalities can propagate those biases and disadvantage protected groups: ProPublica's analysis of the COMPAS recidivism tool showed that it disproportionately mislabeled Black defendants as high risk.

Group fairness definitions, including *statistical parity* and *equal opportunity*, formalize such disparities and form the basis of many fair learning algorithms. However, the effectiveness of these algorithms is often hindered by practical challenges. When direct access to sensitive attributes is legally restricted, fair algorithms typically rely on proxy predictors to infer these attributes; if the proxies are *miscalibrated*, fairness guarantees may be invalidated. This is an instance of the more general issue of *distribution shift* between training and test environments. In addition, privacy constraints, notably *differential privacy*, require injecting random noise to protect individual data, but it obscures the group statistics needed to enforce fairness. These challenges all fall under the broader goal towards *trustworthy machine learning*—building models that are reliable, safe, and ethically sound—but they are often addressed in isolation, resulting in methods that are not necessarily compatible.

We propose a unified algorithmic framework for learning fair classifiers that also addresses robustness and privacy cohesively. At its core is a *post-processing* algorithm that applies lightweight adjustments to the predictions output by existing models to achieve group fairness; it recovers the optimal fair classifier when the base predictor is Bayes-optimal. The model-agnostic and post-hoc nature makes it particularly well-suited to emerging paradigms that derive predictors from pre-trained models rather than training from scratch, such as via prompting large language models. To address distribution shift, we introduce algorithms that calibrate the base predictor to the test distribution before post-processing, or, when the shift is unknown, a robust procedure against worst-case shifts within an *uncertainty set*. To satisfy differential privacy, we use the fact that post-processing depends on the training data only through the empirical joint distribution of model outputs and sensitive attributes, and ensure privacy simply by substituting in a private estimate. We provide open-source code to support the practical adoption of fairness mitigations, and analyses that explore the tradeoffs between accuracy, fairness, robustness, and privacy.

ACKNOWLEDGMENTS

I thank my advisor, Han Zhao, for his unwavering support and encouragement, and for the patient guidance and generous latitude that let me explore the fast-evolving field of machine learning while cultivating expertise in a tiny corner of this vast landscape. I am also grateful to Matus Telgarsky for his early support, and for exemplifying the rigor and passion for the theory of machine learning that I deeply admire and that shaped my interests in the field.

I had the opportunity to spend the summer of 2022 at Google Research, where I enjoyed working with Honglei Zhuang and Zhen Qin, and collaborating with Michael Bendersky, Kai Hui, Jing Lu, Ji Ma, Xuanhui Wang, and Hamed Zamani.

My collaborator Yuzheng Hu has been a source of inspiration and insight, and I am likewise thankful to my co-authors Ziwei Ji, Qiaobo Li, Yuxuan Wan, Fan Wu, Qilong Wu, and Professors Heng Ji and Gautam Kamath. I was fortunate to share the journey at UIUC with Qiuling Fan, Runxiang Cheng, Seiyun Shin, Haozhe Si, Haoxiang Wang, and Lang Yin.

Finally, this milestone would not have been possible without the love and support of my parents, Bin Xian and Lin Zhang. I am also indebted to the friendship and company of Jingyang Liu, Xiaoteng Yang, Xun Zhang, and Zhibo Zhou during my formative years at UIUC—and to Wanxian Yang, for the warmth of shared rituals that carried me through.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The growing availability of data and compute, combined with maturing software support, has made machine learning (ML) more accessible for building high-performing predictive models. As a result, ML models are now integrated into an ever-expanding range of automated decision-making systems that shape everyday life [1], spanning advertising [2], content recommendation and moderation [3], and high-stakes domains such as criminal justice [4], healthcare [5], and finance [6].

However, the ease of use of ML algorithms belies the risk that—without proper evaluation, auditing, and monitoring throughout training and deployment—they can cause disparate impacts across demographic groups (for example, defined by gender, or race). One prominent source of this unfairness is the presence of historical and structural social biases in the data used to train these models: training data is often drawn from past decisions or systems shaped by inequality, and models can in turn learn and reproduce those same biases at test time [1, 7, 8, 9, 10]. ProPublica's analysis of the COMPAS risk assessment tool, used in some jurisdictions to inform pretrial release decisions and sentencing, reported higher false positive rates for Black defendants (incorrectly flagged as high risk) and higher false negative rates for White defendants, systematically favoring White defendants [11]. Bias can also enter through human-labeled data: toxicity classifiers trained on crowdsourced annotations often over-predict toxicity for comments mentioning marginalized identities (in terms of sexual orientation, religion, race, and the like), reflecting annotators' implicit biases that are reproduced by models [3, 12, 13].

Beyond dataset bias, unfairness may arise from properties of the learning algorithm, or simply from variance in the optimization procedure itself. Inductive biases and capacity limits can interact with class or group imbalance [14], leading the model to prioritize fitting majority patterns at the expense of minority group performance [15, 16, 17]. Group-wise performance gaps may also stem from differences in data quality and optimization difficulty across groups [18]. Taken together, these observations necessitate systematic procedures to measure and audit disparities and unfairness, as well as dedicated algorithms to mitigate them.

To this end, the algorithmic fairness literature has formalized a range of fairness criteria. A widely used family is *group fairness* [19], which examines statistics of model outputs aggregated at the group-level and requires that they be equalized across protected groups. For example, Hardt et al. [18] introduce *equal opportunity* (equal true positive rates across groups) and *equalized odds* (additionally equalizing false positive rates); these definitions align with how unfairness is evaluated in the COMPAS tool and toxicity classifiers discussed above (Section 2.2 reviews other common criteria and notions beyond group fairness). Based on these definitions, a large body of fair learning algorithms is developed to train high-performing ML models while satisfying the prescribed group fairness criteria (Section 2.3).

Since group fairness, as a statistical notion, is defined by the test-time distribution of model outputs and sensitive attributes (denoting the protected demographic groups), fair learning algorithms require sufficient training data drawn from the same distribution, with reliable group labels, in order to estimate, detect, and mitigate disparities [20, 21, 22]. Challenges arise when the test distribution differs from the distribution used to collect training data. Furthermore, because such data often contains sensitive information, especially in high-stakes domains, the associated privacy and legal constraints can further complicate efforts toward fairness and hinder the effectiveness of fair algorithms.

Miscalibration. Fair algorithms require accurate knowledge of individuals' sensitive attributes to achieve fairness, yet many sectors restrict the large-scale collection or explicit use of such information in decision-making [22]. For instance, the Equal Credit Opportunity Act [23] prohibits credit card and auto loan lenders from asking applicants about their race, even though they must still demonstrate that their decisions are non-discriminatory. In such cases, practitioners may turn to proxy models to infer sensitive attributes. However, if these proxies are miscalibrated, meaning that their predicted probabilities do not reflect the true group frequencies, fair algorithms can misallocate corrections and fail to reduce disparities [24].

Distribution Shift. Because group fairness is defined with respect to a particular data distribution, the guarantees of a fair algorithm may no longer hold if the distribution under which the model is deployed differs from the one it was trained on [25, 26]. Such shifts are common in practice [27], arising from changing demographics, evolving environments, or (adversarial) noise in the training data. The miscalibration of group proxies discussed above can also be viewed as a case of distribution shift, where the proxy's predicted probabilities deviate from the true conditional distribution of the sensitive attributes.

Privacy. High-stakes domains often involve sensitive personal data, making it essential to protect against data leakage and extraction by adversaries. A widely adopted framework for providing such protections is *differential privacy* [28], which limits the influence of any single training example on the learned model by injecting random noise into the training procedure. But the noise introduced can obscure the group-level statistics used by fair algorithms to

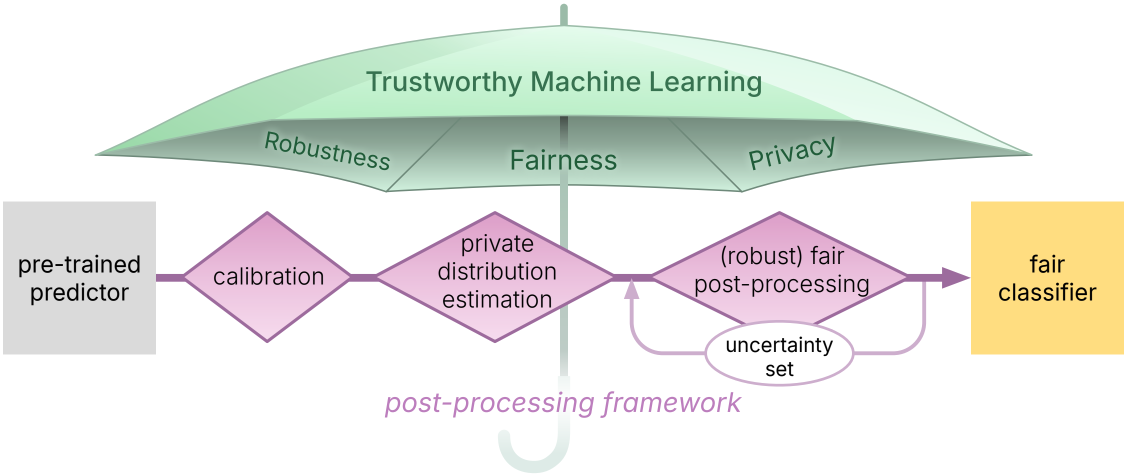
Trustworthy Machine Learning
Robustness
Fairness
Privacy
pre-trained predictor
calibration
private distribution estimation
(robust) fair post-processing
uncertainty set
fair classifier
post-processing framework

Figure 1.1: Overview of the full pipeline of our post-processing framework for learning fair classifiers, that are robust to distribution shifts, and in a privacy-preserving manner.

enforce fairness. Prior work has shown that achieving exact group fairness while maintaining predictive performance is incompatible with differential privacy [29, 30].

These additional considerations—distribution shift and privacy—along with fairness, all fall under the broader umbrella of *trustworthy machine learning* [31, 32], whose overarching goal is to develop models that remain reliable, safe, and ethically sound throughout their entire lifecycle. Although each of these dimensions has been studied extensively, they are often addressed in isolation, resulting in methods that are not necessarily compatible or easily composable.

1.1 A UNIFIED POST-PROCESSING FRAMEWORK FOR GROUP FAIRNESS

In this thesis, we focus on the practical aspects of achieving group fairness in classification, arguably the most common decision-making setting, by developing an algorithmic framework for constructing classifiers that satisfy group fairness criteria efficiently while preserving competitive accuracy. Our framework addresses distribution shift and privacy, providing a coherent procedure for achieving fairness that is both robust and able to meet differential privacy requirements (Fig. 1.1). We release fully documented code to lower the barrier for practitioners to integrate fairness mitigation into existing pipelines.[1] In addition to implementation, we provide formal analyses and empirical evaluations that explore the tradeoffs between accuracy, fairness, and privacy budget, potentially under distribution shifts. Our unified treatment is intended not only to encourage real-world adoption but also

[1] Code is available at: `https://github.com/rxian/fair-classification/`.

to serve as a baseline for future work at the intersection of these dimensions and broader efforts toward trustworthy machine learning.

Specifically, our framework adopts a *post-processing* approach (alternatives are summarized in Section 2.3), which achieves fairness on existing models by applying lightweight adjustments to their output predictions in a post-hoc manner. This approach is model agnostic and requires neither re-training nor modifications to the training pipeline, making it particularly appealing in the emerging "model-as-a-service" paradigm, where classifiers are derived from foundation models such as large language model (also known as *in-context learning*) [33, 34, 35] rather than trained from scratch [36, 37]. Despite its post-hoc nature, post-processing does not sacrifice optimality in theory: our algorithm is designed based on a result showing that the optimal fair classifier can be expressed as a post-processing of the base predictor, provided it is Bayes-optimal. Empirical studies likewise show that post-processing achieves better accuracy-fairness tradeoffs [38]. This modular structure also facilitates the integration of distribution robustness and privacy procedures into our unified framework.

We organize the thesis around the three dimensions of trustworthy machine learning mentioned above, and develop our unified algorithmic framework across three corresponding chapters. We conclude with an application of our framework to derive fair classifiers from (closed-weight) large language models (LLMs), illustrating the flexibility and broad applicability of post-processing as a post-hoc approach to achieving fairness within existing model pipelines.

- Chapter 3 introduces LinearPost, the fair post-processing algorithm that forms the backbone of our framework. The chapter begins with an analysis of the Bayes-optimal fair classifier, showing that it can be expressed as a composition of Bayes-optimal predictors for risk and sensitive attributes, followed by a linear classification head that maps their outputs to hard labels satisfying the specified group fairness constraint. LinearPost is developed directly from this structure: it takes pre-trained risk and group predictors and learns a fair linear classification head, whose parameters are obtained via solving an empirical linear program. The algorithm supports a variety of group fairness criteria, across multiclass, multigroup, and both attribute-aware and attribute-blind settings.

- Chapter 4 focuses on the impact of distribution shift on fair classifiers, including LinearPost. It begins with a theoretical analysis, deriving upper bounds on fairness violation and excess risk under distribution shift, which is decomposed into covariate and concept components. We then propose two types of remedies for improving robustness within our framework, depending on the available knowledge of the distribution shift.

When training examples from the test distribution are available, we introduce two algorithms that calibrate the base predictor (which includes the group proxy that LinearPost relies on to enforce fairness) to the shifted distribution: one based on traditional *distribution calibration* [39], and another based on a boosting-based algorithm for *decision calibration* [40], a weaker but sufficient condition for fairness. When no data from the test distribution is available and the shift is not precisely known, we model potential shifts via an *uncertainty set* and perform robust post-processing to enforce fairness across all distributions in this set. This is via an iterative algorithm based on the *cutting-set method* [41], which alternates between identifying worst-case perturbations within the uncertainty set and updating the classifier to maintain fairness over all previously encountered distributions.

– Chapter 5 incorporates differential privacy (DP) into our post-processing framework, ensuring that the final classifier satisfies privacy with respect to the examples used in post-processing (if privacy is also required for the pre-training stage, any off-the-shelf private learning algorithm can be used to train the prerequisite predictors, as post-processing is decoupled from pre-training). Since LinearPost depends on the post-processing data only through the empirical joint distribution of model outputs and sensitive attributes, privacy can be achieved by replacing this distribution with a differentially private estimate; the resulting classifier then inherits the privacy guarantees by the post-processing theorem of DP [28].

  To provide a concrete demonstration of this strategy—with a slight departure from the thesis's main focus on classification—we apply it to privately post-process regression models, using Laplace histograms for private distribution estimation [42, 43, 44], and analyze the impact of privacy on the resulting model's accuracy and fairness.

– Finally, Chapter 6 applies LinearPost to derive fair classifiers from large language models, using prompting techniques to obtain predictions for both risk and sensitive attributes from the LLM—an increasingly popular paradigm for building prediction models thanks to advances in LLMs' instruction-following capabilities. This setting highlights the unique value of post-processing: the most capable commercial LLMs are typically closed-weight, restricting model fine-tuning or head-tuning, and thereby rendering pre-processing and in-processing approaches not directly applicable as they require access to model internals or re-training.

In our implementation, we introduce an additional re-fitting step to calibrate raw LLM predictions to ground-truth labels prior to post-processing; this step is lightweight and requires only a small number of labeled examples. Experiments on open-weight LLMs show that, in low-data regimes, our procedure for deriving fair classifiers is more effective than training fair classifiers on LLM embeddings, or, in the case of tabular data, training fair models from scratch [45].

1.2 BIBLIOGRAPHIC NOTES

The majority of this thesis is based on joint work with my advisor Han Zhao. Chapter 3 is based on [46] and includes material from [47, 48], of which Lang Yin was also a collaborator; Chapter 4 is based on [46, 49]; Chapter 5 is joint work with Qiaobo Li, Gautam Kamath, and Han Zhao [50]; and Chapter 6 is joint work with Yuxuan Wan and Han Zhao [51].

This thesis omits other work that is more loosely related to the central theme of trustworthy machine learning, where I am either the primary author or made significant contributions: domain adaptation for cross-lingual transferability in fine-tuning language models [52] and neural ranking models [53], and a theoretical study of the limitations of scalarization in multitask learning [54].

CHAPTER 2: PRELIMINARIES AND BACKGROUND

This chapter introduces the notation used throughout the thesis and formally defines the problem setup for fair classification: learning *randomized classifiers* that (approximately) satisfy group fairness constraints—expressible in the linear form described in Definition 2.8—such as *statistical parity* and *equal opportunity* (Section 2.2); and we do not assume *attribute awareness* by default (Definition 2.2). The chapter concludes with an overview of existing algorithms for fair classification in Section 2.3.

Notation. We write $\mathbb{R}_{\geq 0}$ for the set of nonnegative real numbers. For an integer N, let $[N] = \{1, \dots, N\}$, and let $\Delta^N = \{x \in \mathbb{R}^N_{\geq 0} : \sum_i x_i = 1\}$ denote the $(N-1)$-dimensional probability simplex. For a function $f : \mathcal{X} \to \mathbb{R}^N$, we denote the i-th coordinate of $f(x)$ by $f(x)_i$ or occasionally by $f_i(x)$. For vectors $a \in \mathbb{R}^N$ and $b \in \mathbb{R}^M$, we write $[a, b] \in \mathbb{R}^{N+M}$ for their concatenation. We denote by $\mathbf{e}_i \in \mathbb{R}^N$ the one-hot vector whose i-th coordinate equals 1 and all other coordinates equal 0.

We use uppercase letters (such as X) for random variables or vectors, and lowercase letters ($x \in \operatorname{supp}(X)$) for their realizations. We do not typographically distinguish between univariate and multivariate variables; the dimensionality will be clear from context. For an event E, write $\mathbb{1}[E]$ for its indicator and $\mathbb{P}[E]$ for its probability when the underlying distribution is clear; if multiple distributions p and q are considered, we distinguish them with $p[E]$ and $q[E]$.

Given samples $x^{(1)}, \dots, x^{(N)} \sim \mathbb{P}$, the empirical distribution induced by these samples is denoted by $\widehat{\mathbb{P}} = \frac{1}{N} \sum_{j=1}^{N} \delta_{x^{(j)}}$, where δ is the Dirac delta. For $f : \mathcal{X} \to \mathcal{Y}$, the pushforward of $\mathbb{P}$ under f is $f \sharp \mathbb{P}$ (we may also write $f \sharp X$ for $X \sim \mathbb{P}$), which is a distribution supported on $\mathcal{Y}$ and defined by $f \sharp \mathbb{P}[S] = \mathbb{P}[f(X) \in S]$ for all measurable $S \subseteq \mathcal{Y}$. Given a joint distribution over (X, Y), we write $\mathbb{P}_X$ (or p_X) for the marginal of X, and $\mathbb{P}_{X|Y=y}$ for the conditional distribution of X given $Y = y$.

2.1 PROBLEM SETUP

A classification problem is defined by a joint distribution over input features $X \in \mathcal{X}$ (for example, tabular data or texts), class labels $Y \in [K]$, and sensitive attributes $A \in [M]$ representing the protected demographic groups, such as (biological) sex (female and male) or race (Asian, Hispanic, White, and so on).[1] In binary classification ($K = 2$), we designate

[1] Our framework also handles overlapping groups, where an instance may belong to multiple groups or none, through the generalized fairness definition in Definition 2.8; see Section 6.3.1 for an example.

the class label 1 as the negative class and label 2 as the positive class.

The goal of fair classification is to learn classifiers $h : \mathcal{X} \to [K]$ that satisfy (approximate) group fairness criteria by requiring the fairness violation $V(h)$ to not exceed a tolerance $\alpha \geq 0$ (to be formally defined in Section 2.2), while minimizing the risk incurred, that is

$$\min_{h:\mathcal{X}\to[K]} R(h) \quad \text{subject to} \quad V(h) \leq \alpha, \tag{2.1}$$

where the *population risk* R of the classifier is defined with respect to a nonnegative loss function $\ell : [K] \times [K] \to \mathbb{R}_{\geq 0}$ as

$$R(h) = \mathbb{E}[\ell(Y, h(X))] = \mathbb{E}[\ell(Y, \widehat{Y})], \tag{2.2}$$

with the random variable $\widehat{Y} = h(X)$ denoting the classifier's output. For example, the 0–1 loss (classification error) is $\ell(y, k) = \mathbb{1}[y \neq k]$, whereby $R(h) = \mathbb{P}[Y \neq \widehat{Y}]$.

We also define the *pointwise risk*, which measures the expected loss incurred by predicting a particular class k for a given input x:

Definition 2.1 (Pointwise Risk)**.** Given a classification problem and loss function ℓ, the pointwise risk $r : \mathcal{X} \to \mathbb{R}^K_{\geq 0}$ is defined for all $x \in \mathcal{X}$ and $k \in [K]$ as

$$r(x)_k = \mathbb{E}[\ell(Y, k) \mid X = x] = \sum_{y\in[K]} \ell(y, k)\, \mathbb{P}[Y = y \mid X = x]. \tag{2.3}$$

For the 0–1 loss, we have $r(x)_k = \mathbb{P}[Y \neq k \mid X = x] = 1 - \mathbb{P}[Y = k \mid X = x]$. The population risk can be computed from the pointwise risk via $R(h) = \mathbb{E}[r(X)_{h(X)}]$.

Randomized Classifiers. We allow for randomization in the classifier, meaning that for a given input x, the classifier's output $(\widehat{Y} \mid X = x)$ is sampled from a categorical distribution (independently each time it is called) rather than deterministically assigned. Unless otherwise noted, all classifiers considered in this thesis involve randomization.

Specifically, $(\widehat{Y} \mid X = x)$ is drawn from Categorical$(\pi_h(x, 1), \pi_h(x, 2), \ldots, \pi_h(x, K))$, where the frequencies are given by a function $\pi_h : \mathcal{X} \times [K] \to [0, 1]$ that satisfies $\sum_k \pi_h(x', k) = 1$ for all $x' \in \mathcal{X}$ and represents (a lookup table) for the conditional distribution of h over outputs:

$$\pi_h(x, k) = \mathbb{P}[\widehat{Y} = k \mid X = x], \tag{2.4}$$

where $\mathbb{P}$ is taken over the randomness of the classifier h alone. We refer to π_h as the *Markov kernel* of the randomized classifier h, borrowing terminology from a more rigorous measure-theoretic definition of randomized functions in Definition A.1.

The significance of randomization is that there exists problem instances where only randomized classifiers can achieve group fairness with non-trivial accuracy [55]. Moreover, deterministic fair classifiers tend to be more brittle under distribution shifts [56, 57, 58]. However, practitioners should be aware that randomized classifiers may assign different predictions to individuals with identical features $x \in \mathcal{X}$, which could be perceived as unfair at the individual level [21].

The solution of Eq. (2.1) over all (measurable) randomized classifiers is called the *Bayes-optimal fair classifier*.

Finally, we distinguish two settings depending on whether the classifier has explicit access to the sensitive attribute A:

Definition 2.2 (Attribute Awareness)**.** A classifier is *attribute-aware* if the sensitive attribute $A \in [M]$ is an explicit component of its input features, in which case the classifier takes the form $h : \mathcal{X} \times [M] \to [K]$. Otherwise, the classifier is *attribute-blind.*

Explicit access to A often simplifies the design of fair algorithms, a fact leveraged in some early work [18, 59], and, as we show, can improve robustness to distribution shift (Section 4.2.3). In practice, however, access to A may be restricted by law or privacy policy. For example, the Equal Credit Opportunity Act [23] prohibits credit card and auto loan lenders from asking applicants about their race, and Article 9 of the EU General Data Protection Regulation [60] treats race, ethnicity, and similar attributes as "special categories" whose processing is generally prohibited without a specific legal basis. In such cases, one may instead use a proxy for the sensitive attribute (essentially, a predictor for A) to build fair classifiers [24].

On the other hand, removing the sensitive attribute from the input features (an approach referred to as *fairness through unawareness*) neither guarantees group fairness nor prevents A from influencing predictions [61]. This is because sensitive information (like race) is often redundantly encoded in other features (address or ZIP/postal code) [62, 63], enabling re-identification and indirect use by the learned classifier.

2.2 GROUP FAIRNESS DEFINITIONS

The algorithmic fairness literature has proposed and formalized a variety of fairness notions tailored to different contexts and learning settings. In classification problems, a widely adopted family of fairness criteria is *group fairness*, which examines disparities in the classifier's output distributions conditioned on sensitive attributes. These definitions are statistical in nature and capture group-level rather than individual-level fairness [21]. We adopt group fairness in our algorithmic framework, and review several popular criteria below.

Definition 2.3 (Statistical Parity; SP [64])**.** A classifier h satisfies statistical parity if the distribution of its predicted labels is identical across all sensitive groups:

$$\mathbb{P}[\widehat{Y} = k \mid A = a] = \mathbb{P}[\widehat{Y} = k \mid A = a'] \quad \forall k \in [K],\ a, a' \in [M]. \tag{2.5}$$

Definition 2.4 (Parity of True Positive Rates; TPR parity [18])**.** Defined for binary classification ($K = 2$). A binary classifier satisfies TPR parity (also known as *equal opportunity*) if the true positive rate is equal across sensitive groups:

$$\mathbb{P}[\widehat{Y} = k \mid Y = 2, A = a] = \mathbb{P}[\widehat{Y} = k \mid Y = 2, A = a'] \quad \forall a, a' \in [M]. \tag{2.6}$$

Parity of false positive rates (FPR parity) is defined analogously by conditioning on $Y = 1$. TPR parity generalizes naturally to multiclass classification by requiring equal recall across all classes [65, 66, 67]:

Definition 2.5 (Multiclass TPR Parity; MCTPR parity)**.** A classifier satisfies multiclass TPR parity if the true positive rate (recall) for each class is equal across sensitive groups:

$$\mathbb{P}[\widehat{Y} = k \mid Y = k, A = a] = \mathbb{P}[\widehat{Y} = k \mid Y = k, A = a'] \quad \forall k \in [K],\ a, a' \in [M]. \tag{2.7}$$

MCTPR can be viewed as requiring elements on the diagonal of the group-conditional confusion matrices to be identical across groups. Extending this idea, equalized odds requires that the full confusion matrices (capturing all types of classification error) be matched:

Definition 2.6 (Equalized Odds; EO [18])**.** A classifier satisfies equalized odds if

$$\mathbb{P}[\widehat{Y} = k \mid Y = j, A = a] = \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a'] \quad \forall k, j \in [K],\ a, a' \in [M]. \tag{2.8}$$

Definition 2.7 (Accuracy Parity; AP [68])**.** A classifier satisfies accuracy parity if the overall accuracy is equal across sensitive groups:

$$\mathbb{P}[\widehat{Y} = Y \mid A = a] = \mathbb{P}[\widehat{Y} = Y \mid A = a'] \quad \forall a, a' \in [M]. \tag{2.9}$$

This can be equivalently expressed as a weighted sum of the TPRs, since overall accuracy is

$$\mathbb{P}[\widehat{Y} = Y \mid A = a] = \sum_{k \in [K]} \underbrace{\mathbb{P}[Y = k}_{\text{class marginal}} \mid A = a] \cdot \underbrace{\mathbb{P}[\widehat{Y} = k \mid Y = k}_{\text{TPR}}, A = a]. \tag{2.10}$$

2.2.1 Fairness Violation and Approximate Fairness

Achieving exact group fairness often comes at the cost of higher classification risk [69, 70], while in many practical scenarios, approximate fairness suffices. To enable control over the tradeoff between classification performance and fairness, we relax the strict criteria defined above by permitting bounded violations of the fairness constraints.

Let $\alpha \geq 0$, we say that a classifier h satisfies a fairness criterion *α-approximately* if its *fairness violation* (defined below, and generalized in the next section to accommodate more criteria and violation measures)

$$V(h) \leq \alpha. \tag{2.11}$$

Specifically, for the criteria defined in Section 2.2, we measure the fairness violation by the maximum pairwise difference in the fairness statistics across sensitive groups:

(Violation of Statistical Parity)

$$V^{\mathrm{SP}}(h) = \max_{\substack{a,a' \in [M] \\ k \in [K]}} \Big(\mathbb{P}[\widehat{Y} = k \mid A = a] - \mathbb{P}[\widehat{Y} = k \mid A = a'] \Big), \tag{2.12}$$

(Violation of TPR Parity)

$$V^{\mathrm{TPR}}(h) = \max_{a,a' \in [M]} \Big(\mathbb{P}[\widehat{Y} = 2 \mid Y = 2, A = a] - \mathbb{P}[\widehat{Y} = 2 \mid Y = 2, A = a'] \Big), \tag{2.13}$$

(Violation of Equalized Odds)

$$V^{\mathrm{EO}}(h) = \max_{\substack{a,a' \in [M] \\ k,j \in [K]}} \Big(\mathbb{P}[\widehat{Y} = k \mid Y = j, A = a] - \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a'] \Big), \tag{2.14}$$

(Violation of Accuracy Parity)

$$V^{\mathrm{AP}}(h) = \max_{a,a' \in [M]} \Big(\mathbb{P}[\widehat{Y} = Y \mid A = a] - \mathbb{P}[\widehat{Y} = Y \mid A = a'] \Big), \tag{2.15}$$

and violation of multiclass TPR parity (V^{MCTPR}) and FPR parity (V^{FPR}) are defined analogously.

Alternative Relaxations. Besides taking the maximum pairwise difference, other forms of relaxation include *mean-difference* [20], which computes the deviation of each group's statistics from the overall average. For example, for equalized odds:

$$V^{\mathrm{EO,MD}}(h) = \max_{\substack{a \in [M] \\ k,j \in [K]}} \Big| \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a] - \mathbb{P}[\widehat{Y} = k \mid Y = j] \Big|. \tag{2.16}$$

In addition, for better statistical efficiency when estimating the fairness violation from finite samples, each term in the mean-difference can be weighted by the size of the (sub)group being conditioned on [24]:

$$V^{\text{EO,weighted-MD}}(h) = \max_{\substack{a\in[M]\\ k,j\in[K]}} \mathbb{P}[Y = j, A = a] \cdot \left| \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a] - \mathbb{P}[\widehat{Y} = k \mid Y = j] \right|. \tag{2.17}$$

Fairness violation can also be measured using *ratios* rather than differences [71]:

$$V^{\text{EO,ratio}}(h) = \max_{\substack{a,a'\in[M]\\ k,j\in[K]}} \frac{\mathbb{P}[\widehat{Y} = k \mid Y = j, A = a']}{\mathbb{P}[\widehat{Y} = k \mid Y = j, A = a]} - 1, \tag{2.18}$$

which can be equivalently expressed using only additions and subtractions (hence the constraints become linear) as

$$\begin{aligned} V^{\text{EO,ratio}}(h) \leq \alpha \iff \max_{\substack{a,a'\in[M]\\ k,j\in[K]}} \Bigg(& \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a'] - \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a] \\ & + \alpha\Big(1 - \mathbb{P}[\widehat{Y} = k \mid Y = j, A = a]\Big)\Bigg) \leq \alpha. \end{aligned} \tag{2.19}$$

For statistical parity, the ratio-based definition corresponds to the "four-fifths rule" often invoked in discussions of disparate impact legislation [71, 72].

2.2.2 Generalized Fairness Constraints

All (approximate relaxations of the) group fairness criteria discussed above can be expressed as linear constraints over the output distributions of the classifier conditioned on the sensitive attribute and other fairness-related variables, that is, in terms of the classifier's first-order conditional moments. We therefore adopt a unified view by considering all fairness constraints that can be written in the following linear form, inspired by [73, Definition 2]:

Definition 2.8 (Generalized Fairness Constraints)**.** Let h be a classifier, and let $Z_1, \dots, Z_G \in \{0, 1\}$ be (sub)group indicator variables (to be distinguished from the sensitive attribute A) that cannot depend on the classifier output $\widehat{Y}$. These groups may be overlapping, so that multiple Z_i's may be 1 for the same instance (see Section 6.3.1 for an example).

We specify the group fairness constraints as a set of linear inequalities, with (B, μ, c), as

$$B\mu(h) \leq c \qquad \text{(coordinate-wise)}, \tag{2.20}$$

where the matrix $B \in \mathbb{R}^{C\times(1+K+KG)}$ contains constants that may depend on the problem, the vector $c \in \mathbb{R}^C_{\geq 0}$ encodes the fairness tolerance, and $\mu \in \mathbb{R}^{1+K+KG}$ contains (conditional) first-order moments of the output of h and the constant 1. The coordinates of μ are indexed as follows:

$$\begin{aligned} &\mu_{*,*} = 1, \\ &\mu_{k,*} = \mathbb{P}[\widehat{Y} = k] && \forall k \in [K], \\ &\mu_{k,i} = \mathbb{P}[\widehat{Y} = k \mid Z_i = 1] && \forall k \in [K],\, i \in [G]. \end{aligned} \tag{2.21}$$

Under this formulation, the *fairness violation* of a classifier h is defined as the maximum value on the left-hand side of the linear constraints:

$$V(h) = \max_{j\in[C]} B_j \mu(h), \tag{2.22}$$

where $B_j \in \mathbb{R}^{1\times(1+K+KG)}$ is the j-th row of B as a row vector.

Example 2.1 (Recovering Approximate Statistical Parity)**.** To recover α-approximate SP, $V^{\mathrm{SP}} \leq \alpha$ as defined in Eq. (2.12), from the generalized fairness constraints in Definition 2.8, we let the group indicators be $Z_a = \mathbb{1}[A = a]$; note that the event $Z_a = 1$ is the same as $A = a$. We construct $C = 2KM(M-1)$ constraints.

We set every coordinate of $c \in \mathbb{R}^C$ to α, and define the constraint matrix B as follows. For each class $k \in [K]$ and pair of groups $a \neq a' \in [M]$, we introduce two constraints (indexed by $(k, a, a', +)$ and $(k, a, a', -)$), corresponding to the upper and lower bounds:

$$\begin{aligned} B_{(k,a,a',+),(j,i)} &= \mathbb{1}[j = k, i = a], & B_{(k,a,a',+),(j,i)} &= -\mathbb{1}[j = k, i = a], \\ B_{(k,a,a',-),(j,i)} &= -\mathbb{1}[j = k, i = a], & B_{(k,a,a',-),(j,i)} &= \mathbb{1}[j = k, i = a], \end{aligned} \tag{2.23}$$

for all j, i, and 0 elsewhere.

Approximate (multiclass) TPR parity, FPR parity, equalized odds, and accuracy parity can be recovered similarly by using group indicators $Z_{a,y} = \mathbb{1}[A = a, Y = y]$. Moreover, multiple fairness criteria can be enforced simultaneously by concatenating the corresponding (B, c) blocks, although some criteria may be incompatible in the high-fairness regime (when α is set to very small values), in the sense that the only feasible solution may be the constant classifier [19].

The number of constraints in Example 2.1 grows quadratically with the number of groups, but can be reduced to linear by introducing auxiliary (free) variables, following standard linear programming techniques:

Example 2.2 (Recovering Approximate Statistical Parity with Auxiliary Variables)**.** We can recover α-approximate SP with $C = 2KM$ constraints (compare with Example 2.1) by introducing K auxiliary variables $\gamma \in \mathbb{R}^K$. These variables are appended to μ and indexed as

$$\mu_{\mathrm{aux},k} = \gamma_k \quad \forall k \in [K]. \tag{2.24}$$

The problem in Eq. (2.1) then minimizes over γ in addition to h:

$$\min_{\substack{h:\mathcal{X}\to[K] \\ \gamma\in\mathbb{R}^k}} R(h) \quad \text{subject to} \quad B\mu(h,\gamma) \leq c, \tag{2.25}$$

Again, we let the group indicators be $Z_a = \mathbb{1}[A = a]$, and set every coordinate of $c \in \mathbb{R}^C$ to α. We define the constraint matrix B as follows. For each class $k \in [K]$ and group $a \in [M]$, we introduce two constraints (indexed by $(k, a, +)$ and $(k, a, -)$):

$$\begin{aligned} B_{(k,a,+),(j,i)} &= 2 \cdot \mathbb{1}[j = k, i = a], \qquad & B_{(k,a,+),(\mathrm{aux},j)} &= -2 \cdot \mathbb{1}[j = k], \\ B_{(k,a,-),(j,i)} &= -2 \cdot \mathbb{1}[j = k, i = a], \qquad & B_{(k,a,-),(\mathrm{aux},j)} &= 2 \cdot \mathbb{1}[j = k], \end{aligned} \tag{2.26}$$

for all j, i, and 0 elsewhere. This yields the constraints

$$\left|\mathbb{P}[\widehat{Y} = k \mid A = a] - \gamma_k\right| \leq \frac{\alpha}{2}, \qquad \forall k \in [K],\, a \in [M]. \tag{2.27}$$

Since γ_k is a free variable, it will be optimized to the center of the two most violating group-conditional distributions for class k, hence the above set of constraints is equivalent to $|\mathbb{P}[\widehat{Y} = k \mid A = a] - \mathbb{P}[\widehat{Y} = k \mid A = a']| \leq \alpha$ for all a, a', which is the definition of α-approximate SP.

Finally, while our proposed algorithm applies to all fairness constraints expressible in the form of Definition 2.8, our theoretical analysis of sample complexity and distribution shift in Theorem 3.2 and Section 4.2 will focus on a subclass of constraints satisfying the following property:

Assumption 2.1. The constant term B does not depend on the underlying data distribution, and there exists a (randomized) classifier h such that $V(h) = 0$.

The first condition simplifies our analysis and can be dropped by additionally bounding deviations in B due to changes in the underlying distribution via the triangle inequality. The second condition ensures that the problem always admits a solution achieving exact fairness (that is, $\alpha = 0$, as in Definitions 2.3 to 2.7).

Assumption 2.1 holds for most approximate fairness criteria discussed in Section 2.2.1: for statistical parity, (multiclass) TPR parity, FPR parity, and equalized odds, the first condition holds, and the second condition is satisfied by any constant classifier or by classifiers that randomize independently of X, as such classifiers trivially equalize the output distributions across all groups. For accuracy parity, only the second condition holds, satisfied by the randomized classifier that outputs each label with uniform probability (independent of X), thereby achieving the same accuracy of $1/K$ across all groups.

2.2.3 Other Notions of Fairness

Although our generalized group fairness definition in Definition 2.8 encompasses a wide range of fairness criteria, it does not allow the conditioning event to depend on the classifier output. As a result, it does not cover *predictive parity* [74], also known as group-wise calibration.

While this thesis focuses on group fairness, other notions and frameworks address different settings and consider fairness at finer granularities. Hébert-Johnson et al. [75] and Kearns et al. [76] propose *subgroup fairness* and *multicalibration*, which require fairness across a rich collection of subpopulations—potentially unknown a priori—defined by a structured binary hypothesis class; that is, the group indicators Z in Definition 2.8 are assumed to be computable by functions from a known hypothesis class, and fairness is enforced across all such functions.

The notion of *individual fairness*, introduced in Dwork et al. [21], requires that individuals who are similar under a given task-specific similarity metric receive similar predictions. This is typically formalized as a Lipschitz constraint on the predictor with respect to the input metric space.

2.3 RELATED WORK: FAIR ALGORITHMS

To learn classifiers that satisfy the group fairness criteria formalized in Section 2.2, a wide range of algorithms have been developed. These algorithms are commonly categorized by the stage of the machine learning model training pipeline at which they are applied: to the data before training (pre-processing), during model optimization (in-processing), or after the model has been trained (post-processing).

Pre-Processing. A prominent source of unfairness often lies in the training data itself, where historical social biases manifest as (spurious) correlations between the label and the sensitive attribute (for example, “software engineer” being more associated with men and “homemaker” with women [9]).

Pre-processing algorithms address this issue upstream by modifying or cleaning the dataset to remove biased associations, particularly those that directly violate the fairness criterion (meaning, if we take $\widehat{Y} = Y$, then the aforementioned example would violate statistical parity). Common approaches include instance relabeling and reweighting, upsampling underrepresented groups, and downsampling overrepresented ones [77, 78]. Another approach is counterfactual data augmentation, in which instances are duplicated with the sensitive attribute flipped [79], inspired by the notion of *counterfactual fairness* [80]. Pre-processing aims to ensure that the perfect predictor (that is, $\widehat{Y} = Y$) on the cleaned data would already satisfy fairness.

These methods are simple to implement and model-agnostic, but they generally lack guarantees on the fairness of the resulting classifier unless additional assumptions are made about the downstream learner. Unfairness can still emerge from the learner’s inductive biases [14] or differences in optimization difficulty across subgroups [18]. Furthermore, reweighting-based approaches may be ineffective in overparameterized neural networks [81].

In-Processing. In-processing algorithms incorporate fairness constraints directly into the classifier’s training objective, formulating fair classification as a constrained optimization problem [82].

A prominent class of in-processing algorithms is *fair representation learning* [83, 84, 85]. These algorithms aim to learn internal representations that are invariant to the sensitive attribute, so that downstream classifiers trained on these invariant representations would automatically satisfy fairness. Suppose the classifier can be decomposed as $h = g \circ f$ where $g : \mathcal{X} \to \mathcal{V}$ is a feature extractor and $f : \mathcal{V} \to [K]$ is a (lightweight) classification head, then, as an example, statistical parity can be achieved if the conditional distributions $g(X) \mid A = a$ are identical across groups, as it implies $g(X) \perp A \implies f \circ g(X) = \widehat{Y} \perp A$. This invariance is typically achieved through regularization terms that penalize these distributional differences across groups, in terms of Jensen-Shannon divergence, Wasserstein-1 distance (often implemented using adversarial training [86, 87, 88]), or maximum mean discrepancy [89] (which can be computed and minimized directly).

Another notable in-processing algorithm is the Reductions approach of [73], which frames fair classification as a two-player game. The algorithm uses a cost-sensitive classification oracle and applies no-regret learning to iteratively generate a randomized ensemble of classifiers that satisfies the fairness constraints.

The literature also offers alternative frameworks for addressing accuracy disparities, most prominently distributionally robust optimization and multitask learning. *Group distributionally robust optimization* minimizes the worst-group risk, explicitly prioritizing the least-accurate group in the optimization process [15, 16, 90]. In multitask learning, each group is treated as a separate task and the goal is to balance per-task objectives so the majority groups do not dominate [91, 92, 93, 94]. Across these alternatives, the emphasis is on objective design and optimization dynamics to ensure that each group receives comparable attention during training and is equally represented in the learned classifier.

These algorithms provide theoretical fairness guarantees, but only if the underlying optimization problems can be solved to (near-)optimality. In practice, the effectiveness of fair representation learning depends on how well the group-conditional feature distributions are matched, while the performance of Reductions hinges on the quality of the cost-sensitive classification oracle.

Post-Processing. Post-processing algorithms transform the outputs of a pre-trained (possibly unfair) base predictor into hard class labels. They are typically applied in a two-stage "pre-train then post-process" procedure: first, a base model is trained without fairness constraints; then, a post-processing algorithm learns a transformation that adjusts the model's outputs to satisfy fairness.

Most post-processing algorithms require the base predictor to produce predictions for both the task label Y—for evaluating losses—and the fairness-relevant variables (namely, the group indicators Z in Definition 2.8)—for determining group memberships and evaluating fairness. They are typically designed under the assumption that the base predictor is Bayes-optimal, leveraging results showing that the optimal fair classifier can be obtained via a post-hoc transformation of the base predictor's outputs [18, 59, 69, 95, 96, 97, 98, 99, 100]. However, existing post-processing algorithms are often stylized to specific problem settings, such as binary classification or binary sensitive attributes, and may assume attribute awareness.

Like in-processing methods, post-processing algorithms can provide formal fairness and optimality guarantees, but the (fairness) guarantees would depend on how close the base predictor is to optimal (or, on its calibration error in predicting group memberships) [24, 47, 101].

Instead of producing hard class labels in $[K]$, Wei et al. [102] and Alghamdi et al. [103] transform the base predictor's output scores into fair scores while minimizing the divergence between the transformed and original scores, and Țifrea et al. [104] propose an algorithm under the assumption that the ground-truth predictor can be modeled as a generalized linear function of the base predictor's representations.

Access to Bayes scores may appear to be a strong requirement for effective post-processing.[2] Indeed, Woodworth et al. [106] show that if the hypothesis class is restricted and does not contain the Bayes scores, there exist problem instances where post-processing underperforms in-processing. However, an empirical study by Cruz and Hardt [38] on real-world datasets finds that post-processing can achieve better accuracy-fairness tradeoffs, suggesting that predictors learned in practice using off-the-shelf algorithms are often adequate for post-processing to be successful.

[2] Although not necessarily out of reach, given the ability to train large neural models [105].

CHAPTER 3: FAIR CLASSIFICATION VIA POST-PROCESSING

This chapter introduces our post-processing algorithm, LinearPost. Given predictors for the pointwise risk and group membership indicators (defined below), LinearPost computes a randomized mapping that transforms these predictions into hard class labels satisfying fairness constraints. To learn a fair classifier from scratch using LinearPost, we adopt a "pre-train then post-process" procedure: we first train predictors for the pointwise risk and group membership, then apply LinearPost to learn a post-processing transformation (a classification head) that enforces fairness. The algorithm supports all fairness constraints expressible in the linear form of Definition 2.8 across multiclass and multigroup settings, and does not assume attribute awareness.

Our approach builds on an analysis of the structure of Bayes-optimal fair classifiers (Section 3.2), which shows that the optimal fair classifier can be represented as a linear classifier whose input features are the predictions output by the Bayes-optimal pointwise risk and group membership predictors. This representation result requires a uniqueness condition depending on the data distribution; when it does not hold, it can be enforced by adding a small random perturbation to the pointwise risk function (Section 3.2.2). LinearPost adopts this perturbation strategy, making it the only source of randomness in the returned classifier.

In Section 3.3, we formally describe the LinearPost algorithm. We then instantiate it under standard fairness criteria, provide interpretations for the resulting classifiers, and analyze the sample complexity of estimating the parameters for the post-processing transformation. Finally, in Section 3.4, we evaluate LinearPost on benchmark datasets and compare its performance to that of existing in-processing and post-processing fair algorithms.

Problem Setting and Notation. We aim to learn a classifier that satisfies fairness constraints expressible in the form of Definition 2.8 while minimizing the risk (Eq. (2.2)):

$$\min_{h:\mathcal{X}\to[K]} R(h) \quad \text{subject to} \quad B\mu(h) \le c \tag{3.1}$$

(potentially involving auxiliary variables; see Example 2.2). Recall that $R(h) = \mathbb{E}[r(X)_{h(X)}]$, $r : \mathcal{X} \to \mathbb{R}^k_{\ge 0}$ is the pointwise risk function (Definition 2.1),

$$r(x)_k = \mathbb{E}_{(X,Y)\sim p}[\ell(Y,k) \mid X = x] \quad \forall x \in \mathcal{X},\, k \in [K], \tag{3.2}$$

where ℓ is the loss function. We also define the Bayes-optimal *group membership predictor* $g : \mathcal{X} \to [0,1]^G$ as

$$g(x)_i = \mathbb{P}[Z_i = 1 \mid X = x] \quad \forall x \in \mathcal{X},\, i \in [G]. \tag{3.3}$$

The predictors r and g are central to both our theoretical analysis and the design of LinearPost.

3.1 RELATED WORK

We reviewed existing post-processing algorithms in Section 2.3. Similar to the design of LinearPost, most of these algorithms build on representation results showing that the optimal fair classifier can be obtained via a post-hoc transformation of the predicted probabilities output by Bayes-optimal predictors for the risk and group memberships.

To highlight a few examples: Hardt et al. [18] show that the optimal attribute-aware fair classifier for binary-class TPR parity (also called equal opportunity) is a group-specific thresholding of the Bayes-optimal label predictor $\mathbb{P}[Y \mid X, A]$. Menon and Williamson [69] show that the optimal attribute-blind fair classifier for binary-class SP parity (recall that attribute-blind generalizes the attribute-aware setting) applies an instance-dependent threshold to the Bayes-optimal label predictor, which depends linearly on the predicted values for the sensitive attributes.

Chen et al. [99] unify these fairness criteria and attribute-awareness settings, showing that the optimal attribute-blind fair classifier for binary-class SP and EO can be expressed as a linear post-hoc transformation of the Bayes-optimal label and group membership predictors—an observation that inspired this work.[1] However, their proposed algorithm, MBS, requires a grid search whose time complexity grows exponentially with the number of constraints and groups, making it difficult to scale to large training set sizes and beyond the binary-group case.

Zeng et al. [100] consider the same general setting and use the insights to additionally propose both a pre-processing algorithm by applying the post-hoc correction directly to the dataset, and an in-processing algorithm by using the post-hoc transformation's objective to guide training toward fairness.

Building on this line of work, our post-processing algorithm, LinearPost, subsumes all settings considered above and extends to multiclass problems and more fairness criteria. More importantly, the post-hoc transformation in LinearPost can be computed efficiently

[1] Although they did not explicitly analyze TPR and FPR parity, their analysis can be extended to these criteria.

by solving an empirical linear program, and we provide a thorough analysis of its sample complexity, fairness violation, and excess risk.

3.2 BAYES-OPTIMAL FAIR CLASSIFIER

We begin by analyzing the structure of the Bayes-optimal fair classifier, the solution to Eq. (3.1) without any additional constraints on h (such as regularization) beyond the fairness constraints. We show that it can be expressed as a linear post-processing of the predictors r and g, implying that these quantities constitute sufficient statistics for fair classification:

Theorem 3.1. Define the *fair pointwise risk* $r_{\text{fair}} : \mathcal{X} \to \mathbb{R}^K$ as

$$r_{\text{fair}}(x)_k = r(x)_k + \beta_{k,*} + \sum_{i \in [G]} \beta_{k,i} g(x)_i \quad \forall k \in [K], \tag{3.4}$$

with parameters $\beta \in \mathbb{R}^{K \times (1+G)}$ given by

$$\beta_{k,*} = -\sum_{j \in [C]} \psi_j B_{j,(k,*)}, \quad \beta_{k,i} = -\sum_{j \in [C], i \in [G]} \frac{\psi_j B_{j,(k,i)}}{\mathbb{P}[Z_i = 1]} \qquad \forall k \in [K],\, i \in [G], \tag{3.5}$$

where ψ is the optimal dual solution to the linear program reformulation of Eq. (3.1), LP1(r, g), defined in Section 3.2.1.

Under the uniqueness condition in Assumption 3.1, the deterministic classifier given by $x \mapsto \arg\min_k r_{\text{fair}}(x)_k$ (with ties broken by selecting the smallest index k) is an optimal solution to Eq. (3.1).

Assumption 3.1. The minimizer k of the fair pointwise risk $r_{\text{fair}}(x)_k$ is unique on all $x \in \mathcal{X}$ almost surely with respect to the data distribution of X; that is, the event $|\{k : r_{\text{fair}}(X)_k = \min_{k'} r_{\text{fair}}(X)_{k'}\}| = 1$ holds with probability one over X.

This representation result reveals that the Bayes-optimal fair classifier outputs the class label that minimizes a “fairness-adjusted risk”, obtained by adding a fairness adjustment term to the original pointwise risk r (similar in concept to the *bias score* of [99]). The parameters $\beta_{k,i}$ downweight or upweight the risk for instances in group Z_i depending on whether the group is underrepresented or overrepresented in predictions for class k. As a sanity check, setting $\beta = 0$ (corresponding to not enforcing fairness constraints) recovers the standard Bayes-optimal classifier with $r_{\text{fair}} = r$.

Moreover, if Assumption 3.1 holds, the optimal fair classifier is deterministic. Otherwise, the condition can be enforced by randomly perturbing the pointwise risk: replacing r with

$r+\xi$ in Eq. (3.4), where $\xi \in \mathbb{R}^K$ is a small, continuous random noise sampled independently each time r is called; the resulting classifier is then randomized. We discuss this perturbation strategy in detail in Section 3.2.2.

3.2.1 Optimal Fair Classification as a Linear Program

This section proves Theorem 3.1 by formulating the fair classification problem in Eq. (3.1) as a linear program (Primal LP1 and Dual LP1). These same linear programs are also used by LinearPost to compute the parameters of the post-processing transformation. For completeness, here we consider the scenario where auxiliary variables $\gamma \in \mathbb{R}^L$ are used, so that Eq. (3.1) becomes

$$\min_{h:\mathcal{X}\to[K],\gamma\in\mathbb{R}^L} R(h) \quad \text{subject to} \quad B\mu(h,\gamma) \leq c. \tag{3.6}$$

We append the auxiliary variables to μ and index them as

$$\mu_{\text{aux},l} = \gamma_l \quad \forall l \in [L], \tag{3.7}$$

so that the constant matrix B is now of dimension $C \times (1+K+KG+L)$.

We observe that both the objective and the fairness constraints in Eq. (3.1) can be expressed linearly in terms of the Markov kernel π_h of the randomized classifier h (recall that $\pi_h(x,k) = \mathbb{P}[\widehat{Y} = k \mid X = x]$). For the objective,

$$R(h) = \mathbb{E}[r(X)_{h(X)}] = \mathbb{E}[\mathbb{E}[r(X)_{h(X)} \mid X = x]] = \int_{\mathcal{X}} \sum_{k\in[K]} r(x)_k \pi_h(x,k)\, \mathbb{P}[X = x]\, \mathrm{d}x. \tag{3.8}$$

For the fairness constraints, each of the $j \in [C]$ constraints is of the form $B_j\mu(h,\gamma) \leq c_j$ and can be written as

$$\begin{aligned} B_j\mu(h,\gamma) = B_{j,(*,*)} &+ \sum_{l\in[L]} B_{j,(\text{aux},l)}\gamma_l + \sum_{k\in[K]} B_{j,(k,*)}\, \mathbb{P}[h(X) = k] \\ &+ \sum_{k\in[K],i\in[G]} B_{j,(k,i)}\, \mathbb{P}[h(X) = k \mid Z_i = 1], \end{aligned} \tag{3.9}$$

where

$$\mathbb{P}[h(X) = k] = \int_{\mathcal{X}} \pi_h(x,k)\, \mathbb{P}[X = x]\, \mathrm{d}x, \tag{3.10}$$

and, using Bayes' rule,

$$\begin{aligned}
\mathbb{P}[h(X) = k \mid Z_i = 1] &= \int_{\mathcal{X}} \mathbb{P}[h(X) = k, X = x \mid Z_i = 1] \,\mathrm{d}x && (3.11)\\
&= \int_{\mathcal{X}} \mathbb{P}[X = x \mid Z_i = 1] \,\mathbb{P}[h(X) = k \mid X = x, Z_i = 1] \,\mathrm{d}x && (3.12)\\
&= \int_{\mathcal{X}} \mathbb{P}[X = x \mid Z_i = 1] \,\mathbb{P}[h(X) = k \mid X = x] \,\mathrm{d}x && (3.13)\\
&= \int_{\mathcal{X}} \mathbb{P}[X = x \mid Z_i = 1] \pi_h(x, k) \,\mathrm{d}x && (3.14)\\
&= \int_{\mathcal{X}} \frac{\mathbb{P}[Z_i = 1 \mid X = x]}{\mathbb{P}[Z_i = 1]} \pi_h(x, k) \,\mathbb{P}[X = x] \,\mathrm{d}x; && (3.15)
\end{aligned}$$

the third equality above uses the fact that $h(X)$ is conditionally independent of Z_i given X, since the output distribution is fully determined by $\pi_h(X)$.

Since all expressions in Eqs. (3.8) and (3.9) are linear in the Markov kernel $\pi_h \in [0, 1]^{|\mathcal{X}|K}$, Eq. (3.1) can be formulated as a linear program over π_h and auxiliary variables $\gamma \in \mathbb{R}^L$. This linear program is defined at the population level over the full support of X under the distribution $\mathbb{P}[X = x]$, and in terms of the pointwise risk r, the group predictor g, and the marginal group distributions $\mathbb{P}[Z_i = 1]$:

Primal LP1:

$$\begin{aligned}
&\min_{\pi_h \geq 0, \gamma} \int_{\mathcal{X}} \sum_{k \in [K]} r(x)_k \pi_h(x, k) \,\mathbb{P}[X = x] \,\mathrm{d}x\\
&\text{s.t.} \quad B_{j,(*,*)} + \sum_{l \in [L]} B_{j,(\mathrm{aux},l)} \gamma_l + \sum_{k \in [K]} B_{j,(k,*)} \int_{\mathcal{X}} \pi_h(x, k) \,\mathbb{P}[X = x] \,\mathrm{d}x\\
&\qquad + \sum_{k \in [K], i \in [G]} \frac{B_{j,(k,i)}}{\mathbb{P}[Z_i = 1]} \int_{\mathcal{X}} g(x)_i \pi_h(x, k) \,\mathbb{P}[X = x] \,\mathrm{d}x \leq c_j \qquad \forall j \in [C],\\
&\quad \sum_{k \in [K]} \pi_h(x, k) = 1 \qquad \forall x \in \mathcal{X}. \qquad (3.16)
\end{aligned}$$

The final set of constraints ensures that π_h represents a valid Markov kernel, where for each x, $\pi_h(x, \cdot)$ is a probability distribution over classes.

Next, we derive the dual of LP1, following steps analogous to those used in deriving the dual of the Kantorovich optimal transport problem. We introduce dual variables $\phi : \mathcal{X} \to \mathbb{R}$ for the normalization constraints, and $\psi \in \mathbb{R}^C_{\geq 0}$ for the fairness constraints. The Lagrangian is

$$
\begin{aligned}
&\mathcal{L}(\pi_h, \gamma, \phi, \psi) \\
&= \int_{\mathcal{X}} \sum_{k\in[K]} r(x)_k \pi_h(x,k)\, \mathbb{P}[X=x]\, \mathrm{d}x + \int_{\mathcal{X}} \left(1 - \sum_{k\in[K]} \pi_h(x,k)\right) \phi(x)\, \mathbb{P}[X=x]\, \mathrm{d}x \\
&\qquad + \sum_{j\in[C]} \psi_j \Bigg(\sum_{l\in[L]} B_{j,(\mathrm{aux},l)} \gamma_l + \sum_{k\in[K], i\in[G]} \frac{B_{j,(k,i)}}{\mathbb{P}[Z_i = 1]} \int_{\mathcal{X}} g(x)_i \pi_h(x,k)\, \mathbb{P}[X=x]\, \mathrm{d}x \\
&\qquad\qquad + \sum_{k\in[K]} B_{j,(k,*)} \int_{\mathcal{X}} \pi_h(x,k)\, \mathbb{P}[X=x]\, \mathrm{d}x - \big(c_j - B_{j,(*,*)}\big) \Bigg).
\end{aligned}
\tag{3.17}
$$

The second term in the first line is the result of multiplying both sides of the constraint $\sum_k \pi_h(x,k) = 1$ by $\mathbb{P}[X=x]$, since this normalization condition only needs to hold on the support of the data distribution. After collecting terms and simplifying, we obtain

$$
\begin{aligned}
&\mathcal{L}(\pi_h, \gamma, \phi, \psi) \\
&= \int_{\mathcal{X}} \phi(x)\, \mathbb{P}[X=x]\, \mathrm{d}x - \sum_{j\in[C]} \psi_j \big(c_j - B_{j,(*,*)}\big) + \sum_{l\in[L]} \gamma_l \sum_{j\in[C]} \psi_j B_{j,(\mathrm{aux},l)} \\
&\quad - \int_{\mathcal{X}} \sum_{k\in[K]} \pi_h(x,k) \Bigg(\underbrace{\phi(x) + \sum_{j\in[C]} \psi_c \left(B_{j,(k,*)} + \sum_{i\in[G]} \frac{B_{j,(k,i)}}{\mathbb{P}[Z_i = 1]} g(x)_i \right)}_{(\star)} - r(x)_k \Bigg) \mathbb{P}[X=x]\, \mathrm{d}x.
\end{aligned}
\tag{3.18}
$$

Because the primal is a linear program, strong duality holds. Thus we can exchange the order of min and max in $\min_{\pi_h \geq 0, \gamma} \max_{\phi,\psi} \mathcal{L} = \max_{\phi,\psi} \min_{\pi_h\geq 0,\gamma} \mathcal{L}$. To prevent the inner minimization from becoming unbounded below, we must impose $(\star) \leq r(x)_k$ for all $x \in \mathcal{X}$ and $k \in [K]$, otherwise $\pi_h(x,k) \to \infty$ would drive $\mathcal{L} \to -\infty$; and impose $\sum_j \psi_j B_{j,(\mathrm{aux},l)} = 0$ for all $l \in [L]$, or $\gamma_l \to -\mathrm{sign}(\sum_j \psi_j B_{j,(\mathrm{aux},l)}) \cdot \infty$ would also drive $\mathcal{L} \to -\infty$. Thus, the dual linear program is

Dual LP1:

$$
\begin{aligned}
&\max_{\psi\geq 0,\phi} \int_{\mathcal{X}} \phi(x)\, \mathbb{P}[X=x]\, \mathrm{d}x - \sum_{j\in[C]} \psi_j \big(c_j - B_{j,(*,*)}\big) \\
&\text{s.t. } \phi(x) + \sum_{j\in[C]} \psi_j \left(B_{j,(k,*)} + \sum_{i\in[G]} \frac{B_{j,(k,i)}}{\mathbb{P}[Z_i = 1]} g(x)_i \right) \leq r(x)_k \qquad \forall x \in \mathcal{X},\, k \in [K], \\
&\qquad \sum_{j\in[C]} \psi_j B_{j,(\mathrm{aux},l)} = 0 \qquad \forall l \in [L].
\end{aligned}
\tag{3.19}
$$

The first constraint can be equivalently written using the fair pointwise risk defined in Eq. (3.4) (which is in turn defined in terms of ψ) as

$$\phi(x)+\sum_{j\in[C]}\psi_j\left(B_{j,(k,*)}+\sum_{i\in[G]}\frac{B_{j,(k,i)}}{\mathbb{P}[Z_i=1]}g(x)_i\right)\le r(x)_k \iff \phi(x)\le r_{\text{fair}}(x)_k. \tag{3.20}$$

With these linear programs formulated, we are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let π_h be a minimizing Markov kernel of Primal LP1(r,g), and (ψ,ϕ) an optimal dual solution to its Dual LP1. By construction, LP1 represents the fair classification problem in Eq. (3.1), so π_h corresponds to (the Markov kernel of) an optimal fair classifier.

The first constraint in Dual LP1 is $\phi(x)-r_{\text{fair}}(x)_k\le 0$ for all $x\in\mathcal{X}$, $k\in[K]$. By complementary slackness [107],

$$\pi_h(x)_k>0 \iff \phi(x)-r_{\text{fair}}(x)_k=0, \tag{3.21}$$

and it can be shown that

$$\pi_h(x)_k>0 \implies k\in\operatorname*{arg\,min}_{k'\in[K]} r_{\text{fair}}(x)_{k'}. \tag{3.22}$$

Suppose not, then it means $\exists k'\in[K]$, $k'\ne k$, such that $r_{\text{fair}}(x)_{k'}-r_{\text{fair}}(x)_k<0$. By the first constraint in Dual LP1, we know that $\phi(x)-r_{\text{fair}}(x)_{k'}\le 0$. Adding the two inequalities together, we get $\phi(x)-r_{\text{fair}}(x)_k<0$, which contradicts complementary slackness.

Note that the (deterministic) function on the right-hand side of Eq. (3.22) is the classifier proposed in Theorem 3.1, and we want to show that it is equivalent to the optimal fair classifier defined by the Markov kernel π_h on the left-hand side.

Equation (3.22) has already established one direction ('$\Longrightarrow$"), which says that if π_h has a nonzero probability of predicting class k on input x, then k is a minimizer of the fair pointwise risk $r_{\text{fair}}(x)$. However, when π_h assigns nonzero probability to multiple classes, there are multiple minimizers of $r_{\text{fair}}(x)$. In such cases, the exact form of $\pi_h(x,\cdot)$ cannot be recovered from $r_{\text{fair}}(x)$ alone, and we do not know how to randomize among the output classes according to π_h (this is because recovering π_h would additionally require the other dual variable ϕ).

Thus, the equivalence ("$\Longleftrightarrow$") between the proposed deterministic classifier and the randomized optimal fair classifier π_h holds only when $\pi_h(x,\cdot)$ concentrates all of its probability mass on a single class, that is, when the minimizer of r_{fair} is unique. This condition is guaranteed under Assumption 3.1 almost surely, which ensures that the proposed classifier agrees with π_h almost everywhere over the support of X; therefore, it is also an optimal fair classifier. QED.

Assumption 3.1 may not hold for every classification problem. In the following section, we discuss how to satisfy this condition (by perturbing the pointwise risk to ensure uniqueness) so that our proposed classifier remains applicable.

3.2.2 Satisfying the Uniqueness Condition via Random Perturbation

The deterministic classifier in Theorem 3.1 is Bayes-optimal and fair only if the uniqueness condition in Assumption 3.1 holds. This condition can fail when the pushforward $[r, g]\sharp\mathbb{P}_X$ of the input distribution $\mathbb{P}_X$ contains atoms, and the optimal randomized fair classifier splits probability mass across those atoms. Such situations have been noted in prior work on fair post-processing [47, 96, 97, 99], where they are often handled by assuming continuity of the distribution or by smoothing it with a continuous perturbation—the same principle underlying the strategy we adopt here. Assumption 3.1 may also fail under certain fairness criteria and settings; for example, when considering EO or TPR parity with attribute awareness, the optimal fair classifier may necessarily involve randomization [18, 48].

Assumption 3.2 (Continuity)**.** The pushforward distribution of $[r, g, 1]\sharp\mathbb{P}_X$, which is supported on $\mathbb{R}^{K+G+1}$, does not give mass to any linear subspace that has a non-zero component in the first K coordinates.

Lemma 3.1. Assumption 3.2 implies Assumption 3.1.

Proof. Observe that $x \mapsto \arg\min_k r_{\text{fair}}(X)_k = \arg\min_k w_k^\top[r(x), g(x), 1]$ is a linear multiclass classifier on the $(K+G+1)$-dimensional features with parameters $w_k = [\mathbf{e}_k, \beta_{k,\cdot}, \beta_{k,*}]$, which always have a non-zero component in the first k coordinates. On the other hand, the set of points x where $|\{k : r_{\text{fair}}(X)_k = \min_{k'} r_{\text{fair}}(X)_{k'}\}| > 1$ is contained in the set whose features $[r(x), g(x), 1]$ lie on the decision boundary between two (or more) classes, that is, $\exists k, k'$, $k \neq k'$ s.t. $w_k^\top[r(x), g(x), 1] = w_{k'}^\top[r(x), g(x), 1]$. Assumption 3.2 simply states that this set has measure zero, hence implying Assumption 3.1. QED.

We show that Assumption 3.2 can always be satisfied by adding a small continuous random noise $\xi \in \mathbb{R}^K$ to the pointwise risk r, with ξ sampled independently each time r is called. After perturbation, the pointwise risk becomes randomized. Consequently, instead of using the optimal dual solution to LP1(r, g) in Theorem 3.1, which is defined for deterministic

r, we consider a modified linear program LP1$'$ that incorporates its randomness into the objective (also see $\widehat{\text{LP1}}$):

$$\min_{\pi_h \geq 0, \gamma} \int_{\mathcal{X}} \sum_{k \in [K]} \mathbb{E}_\xi[r(x)_k + \xi_k] \pi_h(x,k) \, \mathbb{P}[X = x] \, \mathrm{d}x, \tag{3.23}$$

subject to the same fairness constraints as LP1.

If ξ is zero-mean and independent of r, then LP1$'$ has the same solution as LP1, and the parameters β in Eq. (3.4) are unchanged. The only modification to the classifier is replacing r with $r+\xi$. The resulting classifier is randomized but remains fair for the original problem; while it is no longer exactly Bayes-optimal, its excess risk can be made arbitrarily small by taking the noise magnitude to zero (see below).

This perturbation strategy is implemented in LinearPost by sampling the noise independently from the uniform distribution, $\xi \sim \mathcal{U}(-\sigma, \sigma)^K$. By a forthcoming sensitivity analysis in Theorem 4.2, the excess risk incurred from replacing r with $r+\xi$ is bounded by

$$\mathbb{E}_X \, \mathbb{E}_\xi \left[\sum_{k \in [K]} |r(X)_k - (r(X)_k + \xi_k)| \right] = K \, \mathbb{E}[|\mathcal{U}(-\sigma, \sigma)|] = \frac{K\sigma}{2}. \tag{3.24}$$

Proposition 3.1. If $\xi \in \mathbb{R}^K$ is absolutely continuous with respect to the Lebesgue measure, then Assumption 3.2 is satisfied by replacing r with $r+\xi$, where ξ is sampled independently each time r is called.

Proof. We represent the $(K+G+1)$-dimensional linear subspaces considered in Assumption 3.2 by $\{(x_r, x_g, x_b) : x_r \in \mathbb{R}^K, x_g \in \mathbb{R}^G, x_b \in \mathbb{R}, x_r^\top w_r + x_g^\top w_g + x_b^\top w_b = 0\}$ for some $w_r \in \mathbb{R}^K$, $w_g \in \mathbb{R}^G$, and $w_b \in \mathbb{R}$. Note that $w_r \neq 0$ because the linear subspace has a non-zero component in the first K-coordinates. Then

$$\begin{aligned}
&[r+\xi, g, 1]\sharp\mathbb{P}_X\big(\{(x_r, x_g, x_b) : x_r \in \mathbb{R}^K, x_g \in \mathbb{R}^G, x_b \in \mathbb{R}, x_r^\top w_r + x_g^\top w_g + x_b^\top w_b = 0\}\big) \\
&\qquad = \mathbb{P}_{X,\xi}\big[(r(X)+\xi)^\top w_r + g(X)^\top w_g + w_b = 0\big] && (3.25) \\
&\qquad = \mathbb{E}_X\big[\mathbb{P}_\xi\big[\xi^\top w_r = -r(X)^\top w_r - g(X)^\top w_g - w_b\big]\big] && (3.26) \\
&\qquad = 0 && (3.27)
\end{aligned}$$

because the random variable $\xi^\top w_r \neq 0$ since $w_r \neq 0$, has a continuous distribution, and is independent of X. QED.

```
Algorithm 3.1: LinearPost
Input: Pointwise risk predictor $\hat{r} : \mathcal{X} \to \mathbb{R}^K_{\geq 0}$, group predictor $\hat{g} : \mathcal{X} \to [0,1]^G$,
       fairness constraints $(B, \mu, c)$, unlabeled samples $\{x^{(j)}\}_{j=1}^N$,
       random noise $\xi \in \mathbb{R}^K$
Output: Randomized classifier $\mathcal{X} \to [K]$
$\hat{\psi} \leftarrow$ optimal dual values of $\widehat{\text{LP1}}(\hat{r} + \xi, \hat{g})$ ;    // empirical fair classification
  linear program
$\hat{\beta}_{k,*} \leftarrow -\sum_j \hat{\psi}_j B_{j,(k,*)}$ for all $k \in [K]$ ;              // post-processing parameters
$\hat{\beta}_{k,i} \leftarrow -\sum_{j,i} \hat{\psi}_j B_{j,(k,i)} / \widehat{\mathbb{P}}[Z_i = 1]$ for all $i \in [G]$, $k \in [K]$ ;
return $x \mapsto \arg\min_k (\hat{r}(x)_k + \xi_k + \hat{\beta}_{k,*} + \sum_i \hat{\beta}_{k,i} \hat{g}(x)_i)$ ;
```

3.3 POST-PROCESSING ALGORITHM: LINEARPOST

Theorem 3.1 shows that the Bayes-optimal fair classifier, namely the solution to Eq. (3.1), can be expressed as a linear post-processing of the pointwise risk r and the group membership predictor g, with parameters obtained from solving LP1. While this representation requires a uniqueness condition, Section 3.2.2 shows that uniqueness can always be ensured by adding an arbitrarily small random perturbation ξ to r.

This leads to Algorithm 3.1, which implements fair classification via post-processing. Given (approximated) pointwise risk $\hat{r}$ and group predictor $\hat{g}$, along with unlabeled examples $\{x^{(j)}\}_{j=1}^N$, we solve the empirical linear program $\widehat{\text{LP}}(\hat{r} + \xi, \hat{g})$ to estimate the post-processing parameters (compare with Primal LP1). This empirical LP has $(NK + L)$ variables and $(N + C)$ constraints.

Empirical $\widehat{\text{LP1}}$:

$$\begin{aligned}
\min_{\pi_h \geq 0, \gamma} \quad & \frac{1}{N} \sum_{\ell \in [N]} \sum_{k \in [K]} \Big(\hat{r}(x^{(\ell)})_k + \xi_k^{(j)} \Big) \pi_h(x^{(\ell)}, k) \\
\text{s.t.} \quad & B_{j,(*,*)} + \sum_{l \in [L]} B_{j,(\text{aux},l)} \gamma_l + \sum_{k \in [K]} B_{j,(k,*)} \frac{1}{N} \sum_{\ell \in [N]} \pi_h(x^{(\ell)}, k) \\
& \qquad + \sum_{k \in [K], i \in [G]} \frac{B_{j,(k,i)}}{\widehat{\mathbb{P}}[Z_i = 1]} \frac{1}{N} \sum_{\ell \in [N]} \hat{g}(x^{(\ell)})_i \pi_h(x^{(\ell)}, k) \leq c_j \qquad \forall j \in [C], \\
& \sum_{k \in [K]} \pi_h(x^{(\ell)}, k) = 1 \qquad \forall \ell \in [N],
\end{aligned} \tag{3.28}$$

where the estimated group marginals are computed from the group predictor $\hat{g}$ rather than ground-truth labels:

$$\widehat{\mathbb{P}}[Z_i = 1] = \frac{1}{N} \sum_{j \in [N]} \hat{g}(x^{(j)})_i. \tag{3.29}$$

The random perturbations $\xi^{(1)}, \ldots, \xi^{(N)}$ are sampled independently for each $x^{(j)}$, and a new ξ is drawn each time the returned classifier is called.

If $\hat{r} = r$ and $\hat{g} = g$ (meaning they are Bayes-optimal), then as $\|\xi\| \to 0$ and $N \to \infty$, the classifier returned by Algorithm 3.1 converges to the Bayes-optimal fair classifier. The sample complexity for estimating β is analyzed in Section 3.3.2. Otherwise, the fairness violation and excess risk depend on the suboptimality of $\hat{r}$ and $\hat{g}$, which we analyze in Section 4.2.

Pre-Train then Post-Process. Given labeled examples $\{(x^{(j)}, y^{(j)}, z^{(j)})\}_{j=1}^N$, to learn a fair classifier from scratch using LinearPost, we follow a "pre-train then post-process" procedure. In the pre-training stage, we train predictors $\hat{r}$ and $\hat{g}$ for the pointwise risk and group memberships (optionally followed by a calibration step), as required by Algorithm 3.1. We then invoke the algorithm, which returns a fair classifier by applying a linear post-processing transformation to the outputs of $\hat{r}$ and $\hat{g}$—hence the name LinearPost.

1. (Pre-Training). Train a pointwise risk predictor $\hat{r}$ and a group predictor $\hat{g}$. In some applications, these predictors may already be available; for instance, when deriving (fair) classifiers from large language models, as we will explore in Chapter 6.

   For the pointwise risk $\hat{r}$, by Definition 2.1, it is a linear transformation of $\mathbb{P}[Y \mid X]$ via the loss function ℓ. It thus suffices to train a predictor for Y (for example, using logistic regression, as it can be viewed as a classification task of Y from X), then transform its output into $\hat{r}$ by computing the expected loss for each class.

   For the group predictor, the choice of learning algorithm depends on whether groups are mutually exclusive ($Z \in \{0,1\}^G$ is one-hot) or not. If Z is one-hot, the task is G-way multiclass classification; otherwise, it is a multilabel classification task, for which a simple approach is to train G separate binary classifiers for each group.

2. (Optional: Calibration). The theoretical guarantee of Algorithm 3.1 says that the post-processed classifier by LinearPost satisfies the fairness constraints only if the group predictor $\hat{g}$ is Bayes-optimal. As we will show in Section 4.2.2, using a suboptimal (more precisely, miscalibrated) $\hat{g} \neq g$ can cause excess fairness violations. Thus, it is beneficial to calibrate $\hat{g}$ using labeled (X, Z) pairs before applying LinearPost. Calibration algorithms are introduced in Section 4.3.

3. (Post-Processing). Apply LinearPost (Algorithm 3.1) to $\hat{r}$ and the (calibrated) $\hat{g}$, using *unlabeled* examples ${x^{(j)}}_j$.

   It is recommended to use a separate set of examples for post-processing than for pre-training or calibration. This avoids bias in estimating the post-processing parameters $\hat{\beta}$, which are computed from the outputs of $\hat{r}$ and $\hat{g}$ on the post-processing set. Using the same data can lead to overconfident predictions if $\hat{r}$ and $\hat{g}$ are overfit, particularly when they are implemented with complex models such as gradient-boosted decision trees or neural networks.

3.3.1 Instantiations Under Standard Fairness Criteria

This section instantiates the "pre-train then post-process" procedure described above with LinearPost under the standard group fairness criteria from Section 2.2.1 (excluding accuracy parity). For each criterion, we derive and interpret the resulting classifier.

For concreteness, we assume the 0–1 loss, so the pointwise risk simplifies to

$$r(x)_k = 1 - \mathbb{P}[Y = k \mid X = x]. \tag{3.30}$$

Statistical Parity. Recall the definition of α-approximate SP from Section 2.2.1 and how it can be recovered from the generalized constraints in Definition 2.8 (Example 2.1) by setting the group indicators to $Z_a = \mathbb{1}[A = a]$ for all $a \in [M]$.

In the pre-training stage, the group membership predictor $\hat{g} : \mathcal{X} \to \Delta^M$ is trained to predict the sensitive attribute $A \in [M]$ for each input $x \in \mathcal{X}$. If the setting assumes attribute awareness (Definition 2.2), then $\hat{g}$ need not be trained explicitly, as the group membership is provided directly: $g(x)_a = \mathbb{1}[A = a \mid X = x]$.

From the expression in Theorem 3.1, the returned classifier has the form

$$x \mapsto \operatorname*{arg\,max}_{k \in [K]} \left(\widehat{\mathbb{P}}[Y = k \mid X = x] - \sum_{a \in [M]} \hat{\beta}_{a,k} \widehat{\mathbb{P}}[A = a \mid X = x] - \xi_k \right), \tag{3.31}$$

where $\widehat{\mathbb{P}}$ denotes probabilities estimated from $\hat{r}$ and $\hat{g}$. In the attribute-aware case, this simplifies to

$$(x, a) \mapsto \operatorname*{arg\,max}_{k \in [K]} \Big(\widehat{\mathbb{P}}[Y = k \mid X = x, A = a] - \hat{\beta}_{a,k} - \xi_k \Big), \tag{3.32}$$

which is a (randomized) classifier that selects the class with the highest probability after applying group-specific class-wise offset (and a random perturbation); see Fig. 3.1 for a picture.

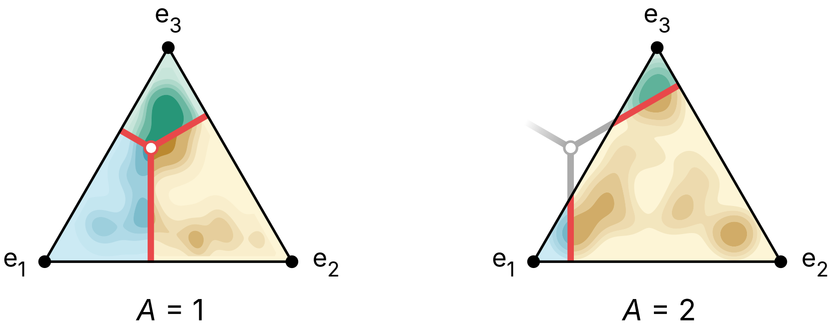

Figure 3.1: Illustration of the attribute-aware optimal fair classifier for statistical parity under $M = 2$ and $K = 3$. The support (Δ^3) is the output space of the pointwise risk r. The group-specific decision boundaries induced by the post-processing are shown (along with the density of r for each group).

The offsets equalize the output distribution across groups by boosting the scores of groups that are underrepresented in a given class and penalizing those that are overrepresented. The attribute-blind form in Eq. (3.31) replaces the group-specific offsets with a convex combination weighted by the predicted distribution of the sensitive attribute conditioned on the input, $\widehat{\mathbb{P}}[A \mid X]$.

The attribute-aware form recovers the result of [47], which further shows that, in this case, the fair classification LP1 can be interpreted as a Wasserstein barycenter problem: it seeks a barycenter over the group-conditional distributions of the pointwise risk (supported on Δ^K) restricted to the K vertices Δ^K, which represents the equalized output class distribution of the fair classifier. The resulting fair classifier then represents the optimal transport from each group's risk distribution to this equalized class distribution.

In the binary classification setting, the expression simplifies to

$$x \mapsto \begin{cases} 1 & \text{if } \widehat{\mathbb{P}}[Y = 2 \mid X = x] < \sum_{a \in [M]} \widehat{\mathbb{P}}[A = a \mid X = x]\hat{t}_a + \frac{1}{2}(\xi_2 - \xi_1), \\ 2 & \text{else,} \end{cases} \tag{3.33}$$

where $\hat{t}_a = (1 + \hat{\beta}_{a,1} - \hat{\beta}_{a,2})/2$. This is a (noisy) thresholding rule applied to the predicted score of the positive class, $\widehat{\mathbb{P}}[Y = 2 \mid X = x]$, with the threshold determined by a weighted sum over group-specific offsets. These recover the results of [69].

(Multiclass) TPR, FPR Parity, and Equalized Odds. Recall the approximate relaxations of these criteria from Section 2.2.1. They are recovered from Definition 2.8 by setting the group indicators to $Z_{a,k} = \mathbb{1}[A = a, Y = k]$ for all $a \in [M]$ and $k \in [K]$ (can be simplified to $k \in \{2\}$ for binary TPR parity, and $k \in \{1\}$ for FPR parity).

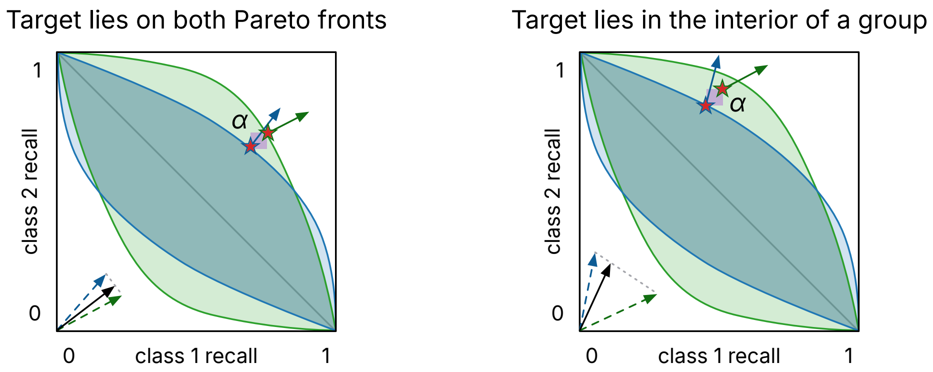
Target lies on both Pareto fronts
Target lies in the interior of a group
1
0
class 2 recall
class 1 recall
0
1
α
1
0
class 2 recall
class 1 recall
0
1
α

Figure 3.2: Group-conditional tradeoffs between TPR and FPR ($M = 2$, $K = 2$). Arrows indicate the direction of maximum accuracy for each group. **Left:** The optimal TPR-FPR pair satisfying α-approximate equalized odds lies on the Pareto frontier of both groups. **Right:** The optimal pair lies in the interior of a group's feasible region.

In the pre-training stage, the group membership predictor $\hat{g} : \mathcal{X} \to \Delta^{M\times K}$ (or $\hat{g} : \mathcal{X} \to [0,1]^M$ with the above simplification for binary TPR and FPR parity) is trained to jointly predict the sensitive attribute $A \in [M]$ and the class label $Y \in [K]$—an MK-way classification task[2] (or M-way multilabel classification where at most one label is active). If attribute awareness is assumed, then it suffices to train a predictor for $\mathbb{P}[Y \mid X, A]$ and compose it with the provided sensitive attribute to construct the joint distribution via $\mathbb{P}[A = a, Y = k \mid X, A] = \mathbb{P}[Y = k \mid X, A] \cdot \mathbb{1}[A = a \mid X = x]$.

The returned classifier then takes the form

$$x \mapsto \operatorname*{arg\,max}_{k\in[K]} \left(\widehat{\mathbb{P}}[Y = k \mid X = x] - \sum_{a\in[M], j\in[K]} \hat{\beta}_{(a,j),k} \widehat{\mathbb{P}}[A = a, Y = j \mid X = x] - \xi_k \right), \quad (3.34)$$

and in the attribute-aware setting, this simplifies to

$$(x, a) \mapsto \operatorname*{arg\,max}_{k\in[K]} \left(\widehat{\mathbb{P}}[Y = k \mid X = x, A = a] - \sum_{j\in[K]} \hat{\beta}_{(a,j),k} \widehat{\mathbb{P}}[Y = j \mid X = x, A = a] - \xi_k \right). \quad (3.35)$$

This is a classifier that (noisily) selects the class with the highest probability after applying a group-specific adjustment, computed as a weighted sum over the predicted class distribution. In the attribute-unaware case above, the adjustment is further averaged over the predicted sensitive attribute distribution.

[2] This task can be decomposed into $(1 + K)$ simpler sub-tasks by factoring the joint as $\mathbb{P}[A, Y \mid X] = \mathbb{P}[Y \mid X] \cdot \mathbb{P}[A \mid X, Y]$. That is, train a predictor for Y only, as well as K predictors for $A \mid Y = k$ conditioned on each (hypothetical) class k. This strategy is adopted in [103], and in our experiments in Chapter 6.

In the binary classification setting, the attribute-aware expression in Eq. (3.35) further simplifies to

$$(x, a) \mapsto \begin{cases} 1 & \text{if } \widehat{\mathbb{P}}[Y = 2 \mid X = x, A = a] < \hat{t}_a + \hat{s}_a(\xi_2 - \xi_1), \\ 2 & \text{else}, \end{cases} \tag{3.36}$$

where

$$\hat{t}_a = \hat{s}_a\Big(1 + \hat{\beta}_{(a,1),1} - \hat{\beta}_{(a,1),2}\Big), \quad \hat{s}_a = \Big(2 + \hat{\beta}_{(a,1),1} - \hat{\beta}_{(a,1),2} + \beta_{(a,2),2} - \hat{\beta}_{(a,2),1}\Big)^{-1}. \tag{3.37}$$

This is again a noisy group-specific thresholding rule applied to the predicted score of the positive class, except here the magnitude of the noise is also group-specific.

For equalized odds in this setting, it suffices to equalize the group-conditional true positive rates (TPRs; also recall for class 2) and false positive rates (FPRs; one minus recall for class 1). Because the TPR-FPR tradeoff can be controlled by threshold adjustment, fairness can be attained via group-specific thresholds when the optimal equalized TPR-FPR pair lies on the Pareto frontier of all groups [18], see Fig. 3.2 (left) for a picture. When the TPR-FPR pair lies in the interior of a group's feasible region (Fig. 3.2, right), one strategy is to degrade the group's optimal tradeoffs via randomization to coincide with the target TPR-FPR pair [48]—this is implemented by setting a larger noise scale $\hat{s}_a$ for that group in Eq. (3.36).

3.3.2 Sample Complexity

Algorithm 3.1 estimates the post-processing parameters $\hat{\beta}$ from the samples $\{x^{(j)}\}_{j=1}^N$ by solving the empirical linear program $\widehat{\text{LP1}}$. We analyze its sample complexity by bounding the fairness violation and excess risk in terms of the sample size N, assuming access to Bayes-optimal pointwise risk and group predictors ($\hat{r} = r$ and $\hat{g} = g$). The additional error from using suboptimal $\hat{r} \neq r$ and $\hat{g} \neq g$ is treated and analyzed as an instance of *distribution shift* in Section 4.2.

Theorem 3.2. Let h denote the optimal fair classifier, $\alpha = V(h)$ the target fairness level, and let $\hat{h} \leftarrow \textsc{LinearPost}(r, g, \xi = 0, \{x^{(j)}\}_{j=1}^N)$ be the deterministic classifier returned from Algorithm 3.1 without applying noise perturbation, where each $x^{(j)}$ is drawn i.i.d. from the input distribution $\mathbb{P}_X$. Let $\eta = \min_i \mathbb{P}[Z_i = 1]$ be the smallest group marginal. Then under Assumption 3.2, for all

$$N \geq \max\bigg(K, \Omega\bigg(\frac{\ln(G/\delta)}{\eta^2}\bigg)\bigg), \tag{3.38}$$

with probability at least $1-\delta$, the fairness violation of $\hat{h}$ is

$$V(\hat{h}) \leq \alpha + O\left(\frac{\|B\|_{\infty,1}}{\eta}\sqrt{\frac{(K+G)\log K + \ln G/\delta}{N}}\right), \tag{3.39}$$

and under Assumption 2.1, the excess risk of $\hat{h}$ is

$$R(\hat{h}) \leq R(h) + \|\ell\|_\infty O\left(\frac{\|B\|_{\infty,1}}{\eta\alpha}\sqrt{\frac{\ln G/\delta}{N}} + K\sqrt{\frac{(K+G)\log K + \ln G/\delta}{N}}\right). \tag{3.40}$$

The bounds follow from the fact that $\hat{h}$ is applies a linear post-processing transformation to the outputs of r and g, which is a linear K-class classifier over a $(K+G+1)$-dimensional feature space. The $\sqrt{(K+G)\log K}$ term arises from an upper bound on the VC dimension of this hypothesis class in a one-versus-rest formulation (Lemma A.3). The extra factor of K in the risk bound comes from decomposing the overall risk into a sum over the K one-versus-rest components. Proofs are deferred to Appendix A.2.

The middle term in the risk bound reflects the discrepancy between the empirical group distribution $g\sharp\widehat{\mathbb{P}}_X$ and the true distribution $g\sharp\,\mathbb{P}_X$, the latter of which is necessary to accurately evaluate and enforce the fairness constraints. This term scales inversely with the target fairness level α: the stricter the fairness requirement, the greater the potential excess risk. We will examine this phenomenon in detail in Section 4.2.3, showing that the α^{-1} dependence is generally tight, but can be removed for attribute-aware classifiers under SP and binary-class EO (and by extension, TPR and FPR parity).

3.4 EXPERIMENTS

> Omitted implementation details, such as dataset and fair algorithm descriptions, split sizes, hyperparameter settings, and the evaluation protocol, are provided in the unified Experiment Details section in Appendix B.

3.4.1 Experiment Setup

We evaluate LinearPost on three tabular datasets (Adult, ACSIncome2-Sex, ACSIncome5-Race, COMPAS) and one textual datasets (BiasBios). We compare against the post-processing algorithms MBS and FairProjection, as well as the in-processing algorithms Reductions and Adversarial. We consider the attribute-blind setting, removing the `sex` column from Adult and ACSIncome2-Sex, and the `race` column from

ACSIncome5-Race and COMPAS; the textual dataset BiasBios is treated as attribute-blind by default, even if the sensitive attribute could be inferred from the text. We consider SP, TPR, FPR, and/or EO fairness.

For the base prediction models, we train logistic regression, gradient-boosted decision trees (GBDT), and two-hidden-layer ReLU networks (MLP) from scratch on the tabular datasets, and fine-tune a BERT model [108] on BiasBios.

For post-processing algorithms, we follow the "pre-train then post-process" procedure described in Section 3.3: first train a base model to predict both the label Y and the sensitive attribute A without fairness constraints, then apply post-processing to its predictions. These two steps use separate pre-training and post-processing dataset splits (Table B.3). Specifically, the base model is trained to predict $(A, Y) \mid X$ jointly, as required for EO fairness, which is an MK-way classification task. Although SP, TPR, and FPR require only partial information from the joint distribution $(A, Y) \mid X$, we use the same base model trained for EO fairness to derive the predictions needed for these criteria, so the setup can be shared across experiments. For example, SP fairness requires only $\mathbb{P}[A \mid X]$, which can be obtained from the joint prediction via $\mathbb{P}[A \mid X] = \sum_k \mathbb{P}[A, Y = k \mid X]$.

For in-processing algorithms, which are applied directly to the input features to train fair classifiers and do not require the two-stage procedure used by post-processing, we merge the pre-training and post-processing splits for training.

3.4.2 Discussions

We present the results as accuracy-fairness tradeoff curves in Section 3.4.3 below and summarize the key findings here.

In the binary-class, binary-group setting (Adult, ACSIncome2-Sex, COMPAS), no single fair algorithm dominates across all tasks. On Adult, Reductions attains the strongest tradeoffs across base models; however, it underperforms the post-processing algorithms on ACSIncome2-Sex and COMPAS. On COMPAS, MBS is best when the base model is GBDT, whereas LinearPost leads when the base model is logistic regression. On ACSIncome2-Sex, FairProjection and LinearPost achieve similar (and best) performance. The comparatively weaker results of MBS for TPR and EO on ACSIncome2-Sex are partly due to runtime limits: its search grid (which grows with the sample size) had to be truncated, reducing solution quality.

In multiclass and/or multigroup settings (ACSIncome5-Race, BiasBios), LinearPost consistently delivers the strongest accuracy-fairness tradeoffs, with FairProjection occasionally performing slightly better.

Post-Processing vs. In-Processing. In-processing algorithms can achieve higher accuracies at looser fairness levels (notably, Adversarial on BiasBios), but in most cases, especially multiclass, they struggle to reach highest fairness. A plausible cause is optimization difficulty. Reductions requires solving and ensembling the classifiers from many cost-sensitive classification problems; performance degrades if the cost-sensitive oracle is imperfect or the number of iterations is capped for runtime. Adversarial training can also be unstable due to the competing objectives of risk minimization and distribution matching.

In contrast, post-processing benefits from ease of optimization: the problem is simpler and can be solved to optimality, enabling higher fairness levels. For instance, LinearPost requires only solving a linear program, which can be done efficiently.

Behavior Under Strict Fairness Tolerances. Setting small fairness tolerances (for example, α for LinearPost or `eps` for Reductions) can yield higher apparent fairness on the training set, but these gains may not transfer to the test set, and often comes with larger drops in accuracy.

There are two main contributing factors. The first is overfitting: achieving very high fairness requires fitting the training data more precisely, which increases variance and can harm generalization. This effect can be mitigated with larger training sets. The second is estimation error of the group distribution: fairness constraints depend on accurate estimates of group conditionals—obtained from labeled training examples in in-processing algorithms, or from a pre-trained group predictor in post-processing algorithms. These estimates are subject to sampling noise, and in post-processing algorithms, additional error arises if the group predictor is imperfect. Our analyses in Theorems 3.2 and 4.2 further show that, for the excess risk, this estimation error is amplified by an inverse dependence on the target violation level α: setting a small α in the presence of estimation error magnifies the excess risk, which in turn explains the rapid accuracy drop observed.

In Section 4.3, we address the second factor incurred by inaccuracies in the group predictor by introducing two calibration algorithms for post-processing algorithms.

3.4.3 Results: Tradeoff Curves

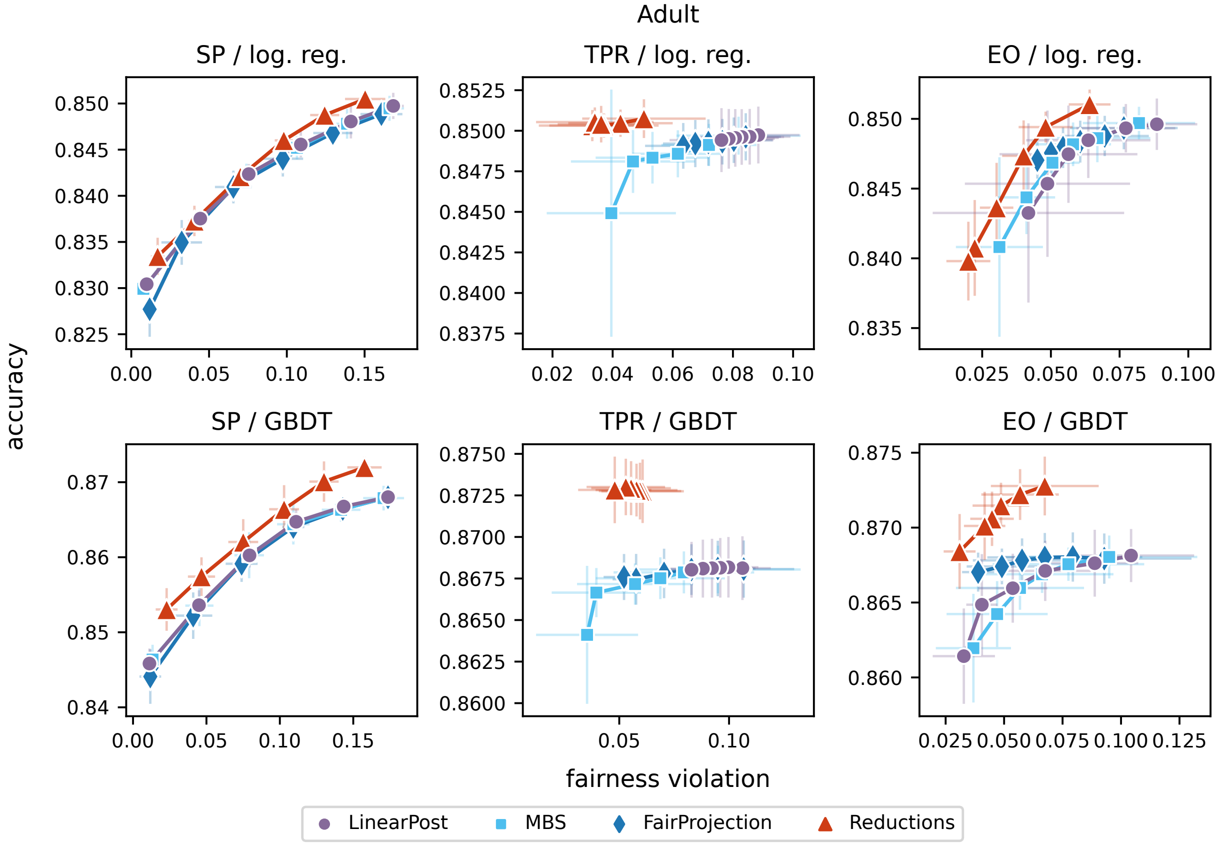

Figure 3.3: Accuracy-fairness tradeoffs on ADULT.

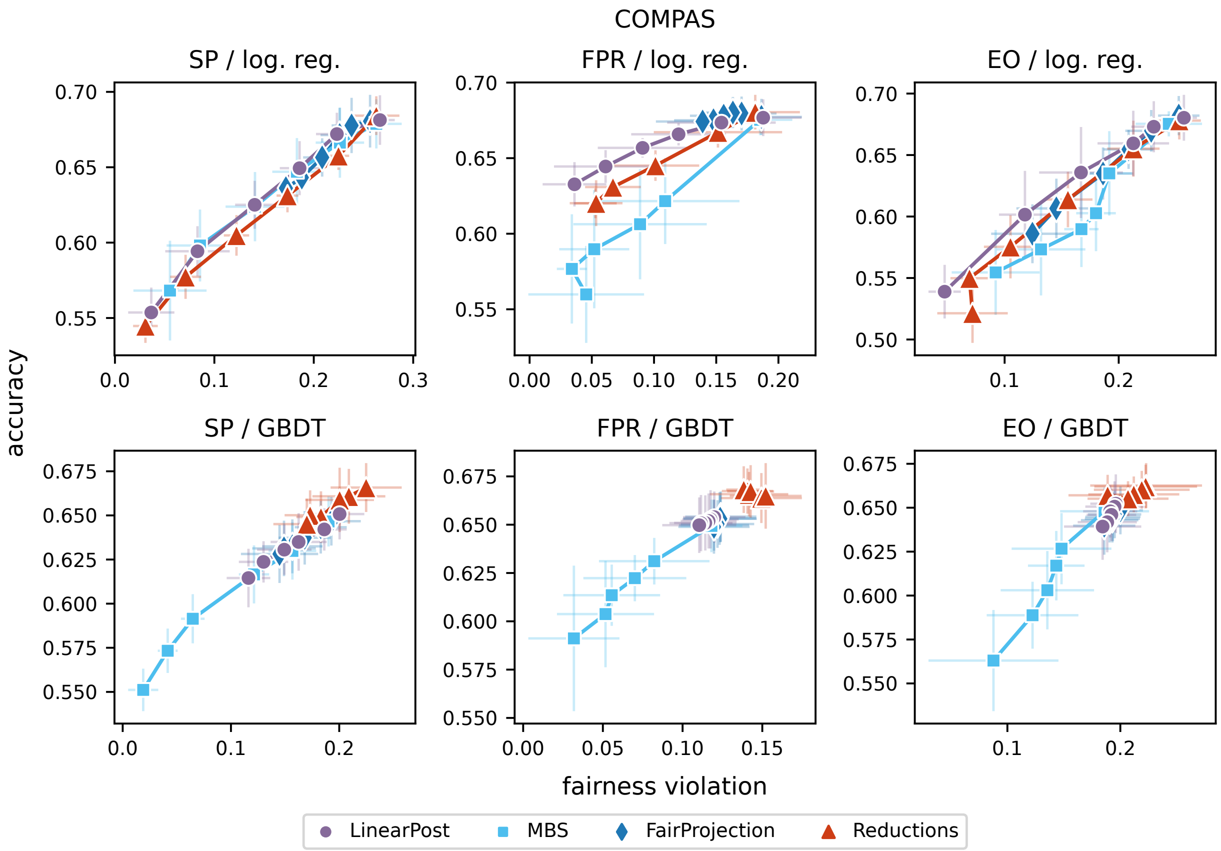

Figure 3.4: Accuracy-fairness tradeoffs on COMPAS.

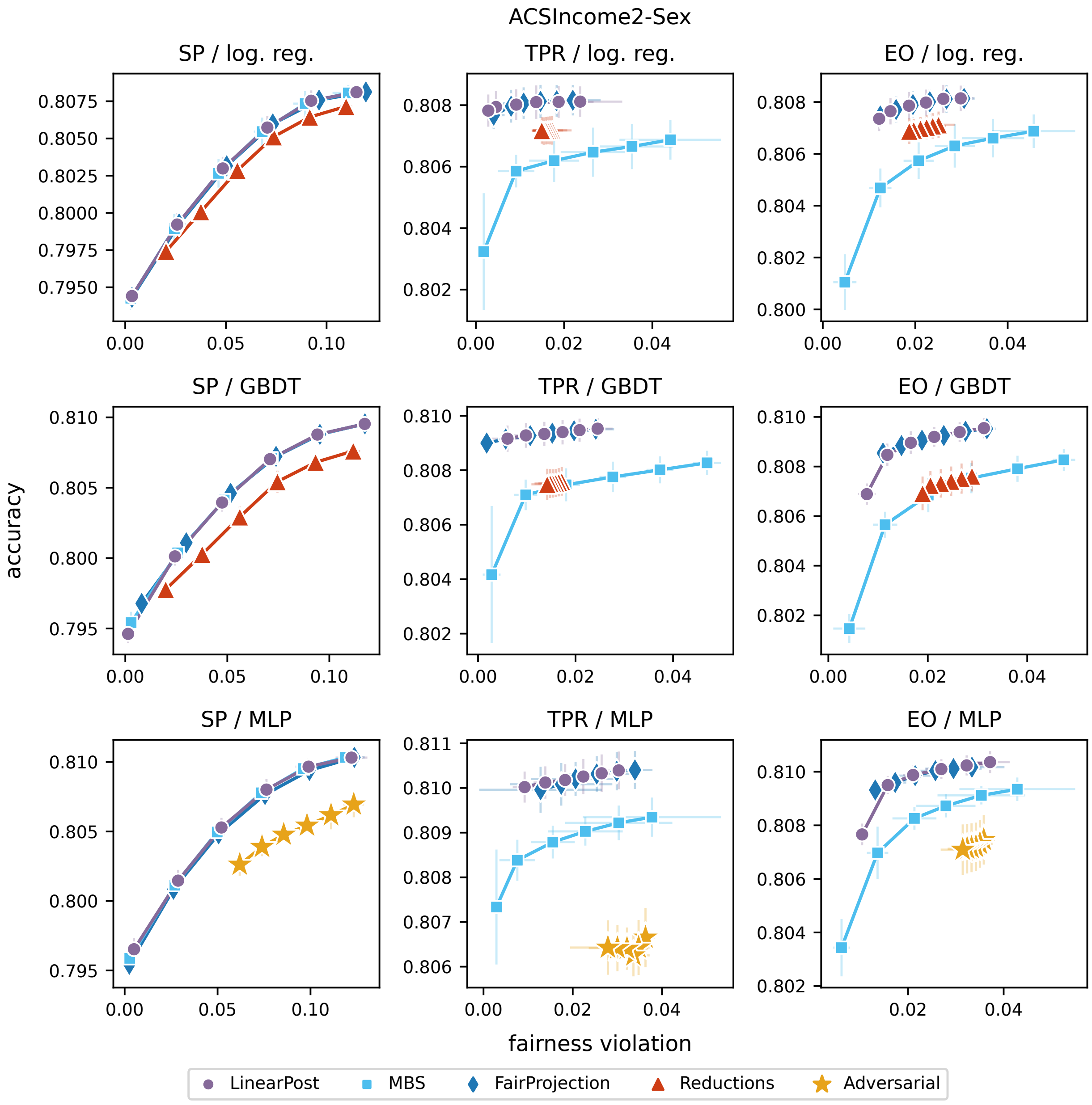

Figure 3.5: Accuracy-fairness tradeoffs on ACSIncome2-Sex.

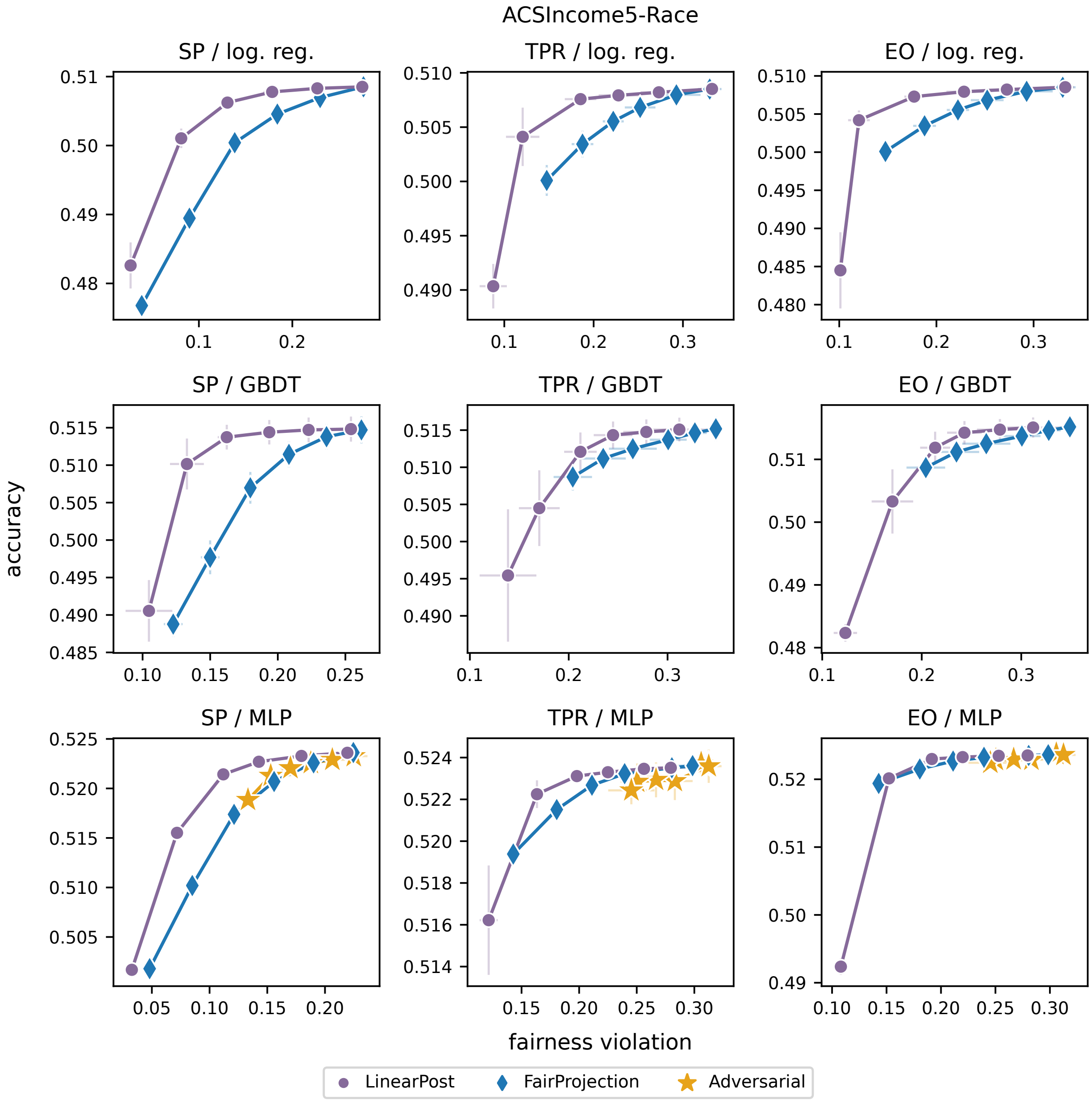

Figure 3.6: Accuracy-fairness tradeoffs on ACSIncome5-Race.

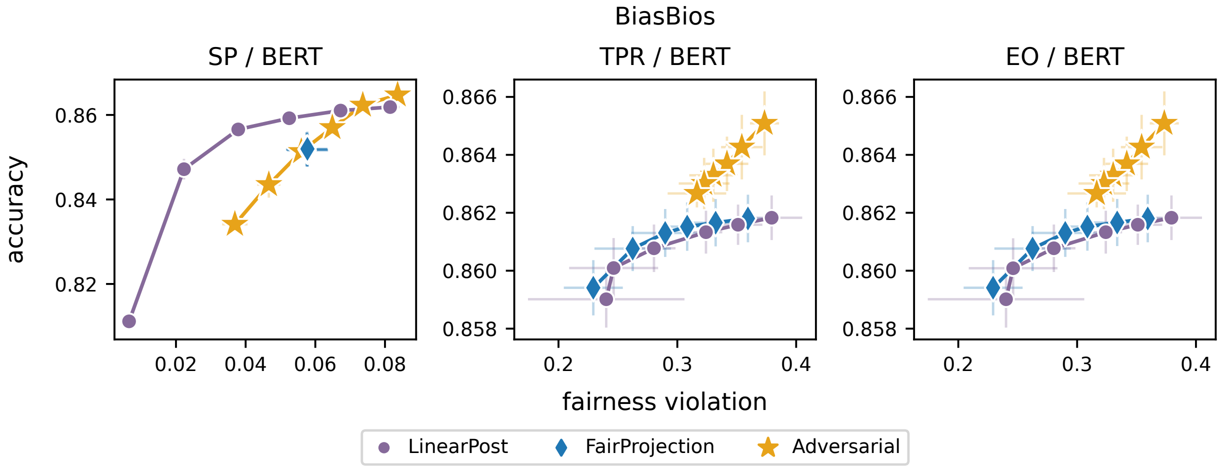

Figure 3.7: Accuracy-fairness tradeoffs on BiasBios.

CHAPTER 4: CALIBRATION AND DISTRIBUTION SHIFTS

Because group fairness is a statistical notion defined by the joint distribution of model outputs and sensitive attributes, it is inherently sensitive to shifts or perturbations in the underlying data distribution [27], which may arise from changing environments or (adversarial) noise in the training data [27]. As a result, the fairness guarantees of a fair classifier trained on one distribution may no longer hold when it is deployed on a different one. For example, Ding et al. [109] evaluate fair algorithms under *geographic shift* and find that an income predictor trained to satisfy fairness in one region can violate it when evaluated in another. We replicate this experiment in Fig. 4.1 by using REDUCTIONS to train classifiers on California (CA) data that satisfy statistical parity, equal opportunity (TPR parity), or equalized odds, and evaluating them on 26 other states. As the distribution distance from CA increases, we observe both larger fairness violations and higher *excess risk*—the drop in accuracy relative to a classifier trained directly on the test distribution attaining the same fairness level. Furthermore, when the fair algorithm relies on proxy predictors for group membership, $\hat{g} = \hat{p}[Z \mid X]$, distribution shift can also result from *miscalibration* of the group predictor, where the predicted probabilities fail to reflect the true conditional distribution $p[Z \mid X]$.

In our proposed algorithm LINEARPOST (Algorithm 3.1), group fairness is achieved by post-processing the outputs of a pre-trained group membership predictor $\hat{g}$ and a pointwise risk estimator $\hat{r}$ (summarizing statistics from $\hat{p}[Y \mid X]$), together with samples from the input distribution p_X. These components reconstruct a view of the training distribution $\hat{p}$ through the joint distributions $\hat{p}[X, Z]$ and $\hat{p}[X, Y]$, which LINEARPOST assumes match the distribution on which the classifier will be deployed when constructing the post-processing transformation. If any of these components differ from their counterparts in the test distribution, that is, either $\hat{g}$ or $\hat{r}$ is not Bayes-optimal, or the examples are drawn from a different distribution, then there is a distribution shift between $\hat{p}$ and the true test distribution. This breaks the identical-distribution assumption underlying the optimality guarantee in Theorem 3.1, and the resulting classifier may no longer be Bayes-optimal or satisfy the specified fairness constraints.

This chapter studies and addresses the challenges posed by distribution shift. We begin with an analysis of its effects on (Bayes-optimal) randomized fair classifiers in Section 4.2, deriving upper bounds on fairness violation and excess risk in terms of the magnitude of the shift. These bounds decompose distribution shift into two components: *covariate shift*, capturing changes in the marginal distribution of X between training and test time, and

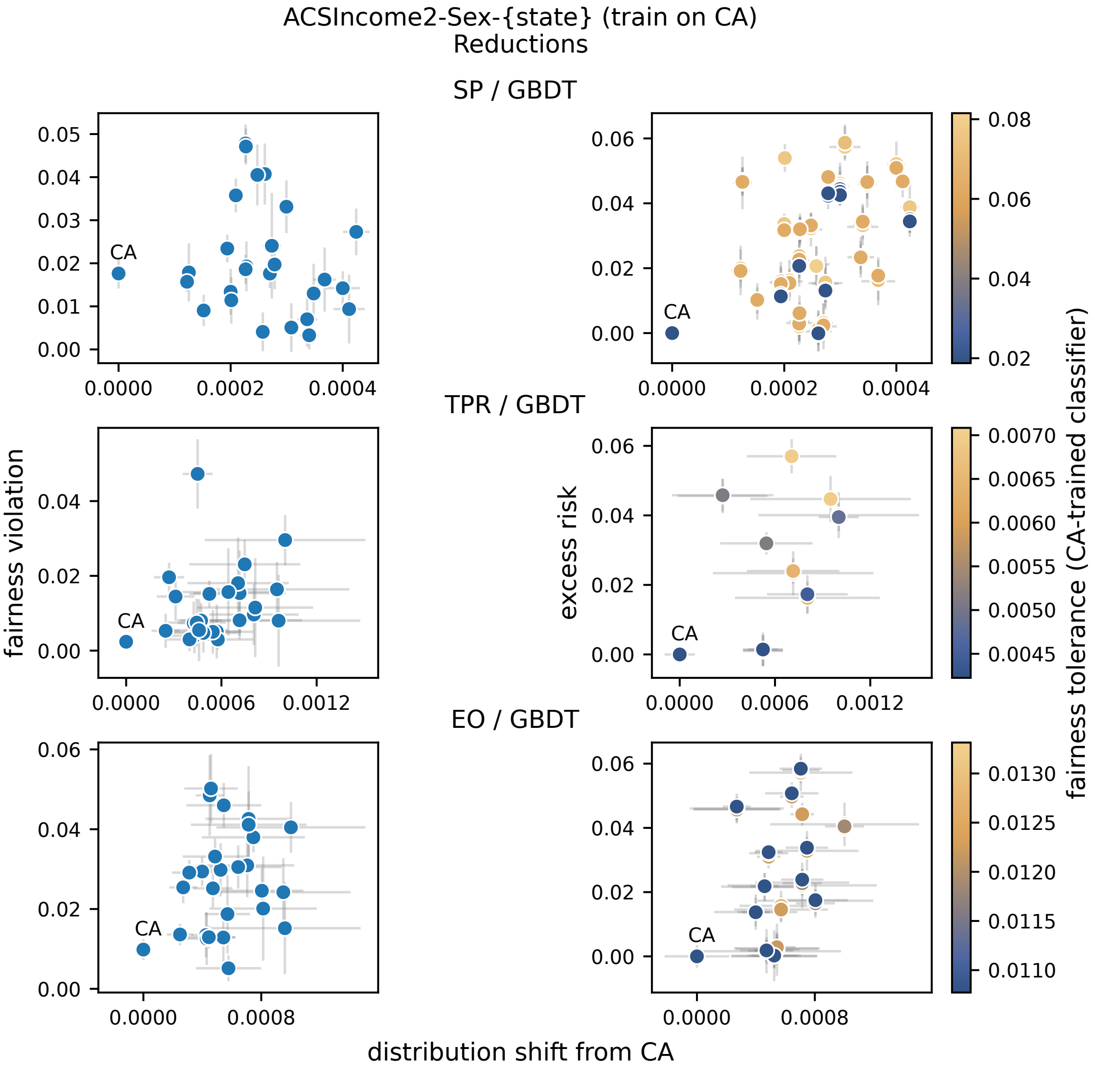

Figure 4.1: Fair classifiers trained on CA data using Reductions under varying tolerance levels, and evaluated on 26 other states. The x-axis measures distribution shift from CA according to Eq. (4.5) using *maximum mean discrepancy* [89]. **Left:** Minimum fairness violation achieved on each state by any CA-trained classifier. **Right:** Excess risk of CA-trained classifiers, measured as the accuracy gap relative to classifiers trained directly on each state with comparable fairness levels (within 0.1×); the fairness tolerance of the CA-trained classifier is indicated by color.

concept shift, capturing changes in the conditionals $(Y \mid X)$ and/or $(Z \mid X)$. All proofs are deferred to Appendix A.1.

Building on this analysis, we introduce two mitigation strategies within the LinearPost framework, each tailored to a distinct setting. Section 4.3 addresses concept shift in Z (for example, miscalibration of group predictor $\hat{g}$): we show that fairness can be guaranteed as long as $\hat{g}$ is calibrated on the test distribution, and present calibration algorithms that use labeled examples from the test distribution. Section 4.4 considers general distribution shifts by defining an uncertainty set over possible test distributions (which can be specified without explicit knowledge about the distribution shift), and develops a robust post-processing algorithm that guarantees fairness with respect to all distributions in this set.

Both strategies are evaluated empirically on real-world data in Section 4.5: we assess the calibration algorithms on correcting miscalibrated group predictors, and evaluate the robustified LinearPost under the geographic shift example discussed above, as well as under label noise and worst-case covariate shift.

Notation. Since studying distribution shift entails evaluating classifiers under multiple distributions, we use subscripts to indicate the distribution with respect to which the risk and fairness violation are computed: the risk of a classifier h on distribution p is denoted by $R_p(h)$ (Eq. (2.2)), and its fairness violation by $V_p(h)$ (Eq. (2.22)). When evaluating over a sequence of distributions $p^{(1)}, \dots, p^{(T)}$, we instead use superscripts: $R^{(1)}, \dots, R^{(T)}$ for risks and $V^{(1)}, \dots, V^{(T)}$ for violations. The conditional moments $\mu(h)$ of h, which appear in the definition of fairness constraints (Definition 2.8), also depend on the underlying distribution and are denoted in the same way.

4.1 RELATED WORK

Fair classification under distribution shift has received considerable attention in prior work, both in theoretical analyses and the design of algorithms for learning fair classifiers that remain robust under such shifts.

Konstantinov and Lampert [56], Blum et al. [57], and Agarwal et al. [58] analyze fair classifiers under adversarial noise (worst-case distribution shift), with emphasis on the brittleness of deterministic fair classifiers compared to randomized ones. Chen et al. [110] provide fine-grained bounds under covariate and label shifts. Giguere et al. [111] and Kang et al. [112] study the problem of certifying fairness guarantees under distribution shifts. Diana et al. [101] and Globus-Harris et al. [24] focus on the miscalibration of proxy predictors for group membership when sensitive attributes are unavailable for training, and analyze the calibration conditions required to preserve optimality and fairness.

Our analysis is similar to those in [25, 58, 113]. Wang et al. [25] study fair classifiers under covariate shift and derive bounds on fairness violation in terms of the magnitude of the shift. Agarwal et al. [58] additionally bound the excess risk of the optimal fair classifier under shifts, but their results are limited to the attribute-aware setting. Hou and Zhang

[113]—which appeared after our analysis—study the excess risk of the optimal attribute-blind fair classifier, revealing a worst-case dependency on the target fairness level similar to our result in Theorem 4.2. While we provide an example that matches this dependency, they establish matching minimax bounds.

Robust fair algorithms can be broadly categorized into *domain adaptation* and *generalization* approaches [27]. Domain adaptation assumes access to both a source distribution and a specific target distribution (though not necessarily with labeled target data). The common strategy is to relate the target distribution to the source, via importance weighting, invariant representation learning, or assuming a generative model (like causal graphs), followed by applying standard (non-robust) fair algorithms [114, 115, 116, 117, 118, 119, 120]. Generalization approaches make fewer assumptions about the test distribution(s) and instead define an uncertainty set that characterizes the potential shifts, often as bounded perturbations around the source distribution. The goal is to ensure fairness for all distributions in the set, typically using tools from (distributionally) robust optimization [25, 26, 121, 122].

Our proposed calibration algorithms fall into the former category, while our robust fair post-processing algorithm belongs primarily to the latter. However, it can be adapted to the domain adaptation setting by tailoring the uncertainty set to the target distribution using available knowledge.

4.2 ANALYSIS

Let p and q denote two distributions over (X, Y, Z), where p represents the training distribution and q the test distribution. Suppose h is a randomized fair classifier trained on p, we want to analyzes its fairness violation (Section 4.2.2) and excess risk (Section 4.2.3) when evaluated on q by deriving upper bounds in terms of their distribution shift.

Lipschitz Randomized Classifier. We are primarily interested in the randomized Bayes-optimal fair classifier, defined as the solution to Eq. (3.1) without any additional constraints on h.

Our analyses, however, apply to a more refined class of Lipschitz randomized fair classifiers in which the underlying Markov kernels π_h are Lipschitz continuous with respect to the input x (recall the generalized fairness constraints in Definition 2.8):

$$\min_{\substack{h:\mathcal{X}\to[K]\\ \mathrm{Lip}(\pi_h)\le L}} R(h) \quad \text{subject to} \quad B\mu(h) \le c, \tag{4.1}$$

where the Lipschitz condition reads

$$\mathrm{Lip}(\pi_h) \leq L \iff |\pi_h(x,k) - \pi_h(x',k)| \leq L\, d(x,x') \quad \forall k \in [K],\, x, x' \in \mathcal{X}. \tag{4.2}$$

The case $L = \infty$ recovers the unconstrained Bayes-optimal fair classifier described above.

4.2.1 Definitions

We begin by introducing definitions and metrics for measuring distribution shifts. Several models of distribution shift have received particular attention in the literature (see [123, 124] for surveys); consider the joint distribution over (X, Z) (similar definitions apply to (X, Y)):

Covariate Shift. Decomposing the joint distribution $p[X, Z]$ into $p[X] \cdot p[Z \mid X]$, this model assumes that only the marginal distribution of input features X differs between p and q. That is, $p[Z \mid X = x] = q[Z \mid X = x]$ for all $x \in \mathcal{X}$, while $p_X \neq q_X$.

For transfer learning to be feasible, it is often assumed that the density ratio between the marginals (called the *importance weight*) is bounded: $q[X = x]/p[X = x] \leq C$ for all x for some $C < \infty$.

Concept Shift in Z. The opposite of covariate shift, this model assumes $p_X = q_X$, while $p[Z \mid X = x] \neq q[Z \mid X = x]$. The scenario of group predictor miscalibration falls under this model, and so does learning under noisy group labels [25].

Prior Shift in Z. Decomposing the joint distribution as $p[Z] \cdot p[X \mid Z]$, this model assumes $p[X \mid Z = z] = q[X \mid Z = z]$ for all $z \in \{0,1\}^G$, while the marginal distribution of Z differs between p and q.

Distribution shifts can be quantified using probability metrics such as f-divergences and integral probability metrics [125]. We use two metrics from the latter family:

Definition 4.1 (Total Variation)**.** The total variation distance between distributions μ, ν over $\mathcal{X}$ is defined as

$$D_{\mathrm{TV}}(\mu,\nu) = \frac{1}{2}\int_{\mathcal{X}} |\mu(x) - \nu(x)| \,\mathrm{d}x = \int_{\mathcal{X}} \mathbb{1}[\mu(x) - \nu(x) \geq 0] \cdot (\mu(x) - \nu(x)) \,\mathrm{d}x. \tag{4.3}$$

Definition 4.2 (Dudley Metric)**.** Let $\mathcal{X}$ be a metric space with distance d, and let $\mathrm{Lip}(f)$ denote the Lipschitz constant of a function $f : \mathcal{X} \to \mathbb{R}$, that is, $|f(x) - f(x')| \leq \mathrm{Lip}(f)\, d(x, x')$.

The Dudley distance between distributions μ and ν, parameterized by (B, L), is

$$D_{B,L}(\mu, \nu) = \sup_{\substack{f:\mathcal{X}\to[0,B] \\ \mathrm{Lip}(f)\leq L}} \left| \int_{\mathcal{X}} f(x) \cdot (\mu(x) - \nu(x))\,\mathrm{d}x \right|. \tag{4.4}$$

For any $L \leq L'$, the witness function class satisfies $\{f : \mathrm{Lip}(f) \leq L\} \subseteq \{f : \mathrm{Lip}(f) \leq L'\}$, so $D_{B,L} \leq D_{B,L'}$. Also note that the total variation corresponds to $D_{\mathrm{TV}} = D_{1,\infty} \geq D_{1,L}$ for all L. Furthermore, the Dudley metric is related to the Wasserstein-1 distance via its dual formulation [126], with the additional constraint that f is bounded, hence $2\,D_{B,L} \leq L\,W_1$.

4.2.2 Fairness Violation

We bound the fairness violation of a classifier h on the test distribution q by its violation on the training distribution p—namely, the target fairness level $\alpha = V_p(h)$—plus a term that quantifies the distribution shift measured using the Dudley metric (Definition 4.2; a generalization of total variation distance), that we decompose into covariate and concept shift components.

Theorem 4.1. Let p, q be two distributions, and let h be a Lipschitz randomized classifier with $\mathrm{Lip}(\pi_h) \leq L$ (Eq. (4.2)). Let $\alpha = V_p(h)$ denote the target fairness level. Then the fairness violation of h on q, $V_q(h)$, is

$$V_q(h) \leq \alpha + \|B\|_{\infty,1}\left(D_{1,L}(p_X, q_X) + \max_{i\in[G]} D_{1,L}(p_{X|Z_i=1}, q_{X|Z_i=1}) \right). \tag{4.5}$$

Moreover, if $L' \geq \mathrm{Lip}(x \mapsto q[Z = z \mid X = x])$ for all $z \in \{0,1\}^G$ (which says that changes to the conditional probabilities of Z are smooth), then

$$V_q(h) \leq \alpha + \max_{i\in[G]} \frac{3\|B\|_{\infty,1}}{p[Z_i = 1]} \Big(\underbrace{D_{1,L+L'+LL'}(p_X, q_X)}_{\text{covariate shift}} + \underbrace{\mathbb{E}_{X\sim p_X}[|p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]|]}_{\text{concept shift in } Z} \Big). \tag{4.6}$$

To analyze the case of group predictor miscalibration (with no covariate shift), we can instantiate $p \leftarrow \hat{p}$ as the misestimated training distribution induced by the inaccurate group membership predictor $\hat{g} : \mathcal{X} \to [0,1]^G$, and $q \leftarrow p$ as the true distribution. Then, the concept shift term becomes $\mathbb{E}_{X\sim p_X}[|\hat{g}(x)_i - p[Z_i = 1 \mid X]|]$, showing that the fairness violation depends directly on the accuracy of $\hat{g}$.

These bounds also yield two insights on robustness. (1) *Smooth classifiers are more robust.* Since $D_{1,L} \leq D_{1,L'}$ for $L \leq L'$, the bound implies that classifiers with smaller Lipschitz constants are more robust to distribution shifts. This motivates the use of Lipschitz regularization to improve robustness: a related work is [121], who apply *sharpness-aware minimization* [127] in the training of fair classifiers. (2) *Prior shift does not affect fairness.* Because the fairness violation depends only on the shift in the conditional distribution $(X \mid Z)$, changes in the marginal distribution of Z alone do not impact fairness (this is consistent with related findings in [119]).

Empirical observations corroborate the results in Theorem 4.1. In Fig. 4.1 (left), we plot the best fairness achieved by fair classifiers trained on CA (under varying fairness tolerances) on other states, against their distribution distance from CA. As expected, fairness violation generally increases with the magnitude of the shift.

4.2.3 Excess Risk

Suppose h_p is an optimal fair classifier for distribution p, but at test time it is deployed on a different distribution q. We analyze the resulting *excess risk*—the additional risk incurred by using h_p on q rather than the optimal fair classifier h_q for the test distribution q.

Theorem 4.2. Let p, q be two distributions with $L' \geq \mathrm{Lip}(x \mapsto q[Y = k \mid X = x])$ for all $k \in [K]$ (which says that changes to the conditional probabilities of Y are smooth). Fix $L \in [0, \infty]$, and let h_p and h_q be optimal Lipschitz randomized fair classifiers (Eq. (4.1)) on p and q, respectively, achieving the same target fairness level α on their respective distributions,

$$h_p \in \operatorname*{arg\,min}_{\substack{h:\mathcal{X}\to[K]\\ \mathrm{Lip}(\pi_h)\leq L}} R_p(h) \quad \text{and} \quad h_q \in \operatorname*{arg\,min}_{\substack{h:\mathcal{X}\to[K]\\ \mathrm{Lip}(\pi_h)\leq L}} R_q(h) \quad \text{subject to} \quad V_p(h_p) = V_q(h_q) = \alpha. \tag{4.7}$$

Let ε be an upper bound on the excess fairness violation incurred by h_q when evaluated on p, that is, $V_p(h_q) - \alpha \leq \varepsilon$. Suppose Assumption 2.1 is satisfied by an L-Lipschitz randomized classifier, then the excess risk of h_p on q is bounded by

$$\begin{aligned}
&R_q(h_p) - R_q(h_q) \\
&\leq \|\ell\|_\infty \left(2\, D_{1,(L+L')K}(p_X, q_X) + \sum_{k\in[K]} \mathbb{E}_{X\sim p_X}[|r_p(X) - r_q(X)|] + \frac{\varepsilon}{\alpha+\varepsilon} \right) &(4.8)\\
&\leq \|\ell\|_\infty \left(\underbrace{2\, D_{1,(L+L')K}(p_X, q_X)}_{\text{covariate shift}} + \sum_{k\in[K]} \underbrace{\mathbb{E}_{X\sim p_X}[|p[Y = k \mid X] - q[Y = k \mid X]|]}_{\text{concept shift in } Y} + \frac{\varepsilon}{\alpha+\varepsilon} \right), &(4.9)
\end{aligned}$$

where r_p and r_q are the pointwise risk functions on p and q, respectively, and with the convention that $0/0 = 0$.

The bound ε on excess fairness violation can be instantiated using Theorem 4.1, for example,

$$\varepsilon \leq 4 \max_{i \in [G]} \frac{1}{p[Z_i = 1]} \Big(\underbrace{D_{1,(L+1)L'}(p_X, q_X)}_{\text{covariate shift}} + \underbrace{\mathbb{E}_{X \sim p_X}[|p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]|]}_{\text{concept shift in } Z} \Big). \tag{4.10}$$

Then the result shows that the excess risk is influenced by the covariate shift as well as the concept shift in both Y and Z.

To analyze the suboptimality of LinearPost when applied on estimated predictors $\hat{r}$ and $\hat{g}$ that are not Bayes-optimal, we instantiate $p \leftarrow \hat{p}$ as the misestimated training distribution induced by inaccurate $\hat{r}$ and $\hat{g}$, and $q \leftarrow p$ as the true data distribution. Then, the second term in the first bound becomes $\mathbb{E}_{X \sim p_X}[|\hat{r}(X)_k - r(X)_k|]$, and the concept shift in Z becomes $\mathbb{E}_{X \sim p_X}[|\hat{g}(X)_i - g(X)_i|]$; these show that inaccuracies in either $\hat{r}$ or $\hat{g}$ can degrade the performance of the resulting classifier.

The result in Theorem 4.2 is corroborated by empirical observations. Figure 4.1 (right) shows that the excess risk of fair classifiers trained on CA (under varying fairness tolerances) evaluated on other states generally increases with the distribution distance from CA.

Sensitivity to Target Fairness Level. Notably, the final term in the bound of Theorem 4.2 depends on the target fairness level $\alpha = V_p(h_p) = V_q(h_q)$: when aiming for high levels of fairness (that is, small α), the excess risk becomes more sensitive to distribution shift. In practice, however, this effect appears weak in the evaluations under geographic shift in Fig. 4.1 (right), suggesting that the worst-case dependence on α may not dominate on real-world data.

To illustrate the tightness of this dependency, we construct a worst-case example for learning attribute-blind classifiers under statistical parity (Hou and Zhang [113] showed the same worst-case dependency on α, and established matching minimax lower bounds). This example matches the upper bound up to a multiplicative factor of 1/2:

Example 4.1. Let $\alpha \in [0, 1]$ denote the target SP violation (as defined in Section 2.2.1), and let $\varepsilon \in [0, 1 - \alpha]$ be the amount of concept shift in A (which is the same as to Z under statistical parity; see Example 2.1). We construct distributions p and q over (X, A, Y) as follows:

- $p_A = q_A$, and are uniform over $\{1, 2\}$ (binary sensitive attribute).
- $p_X = q_X$, and are uniform over $\mathcal{X} = \{1, 2\}$.

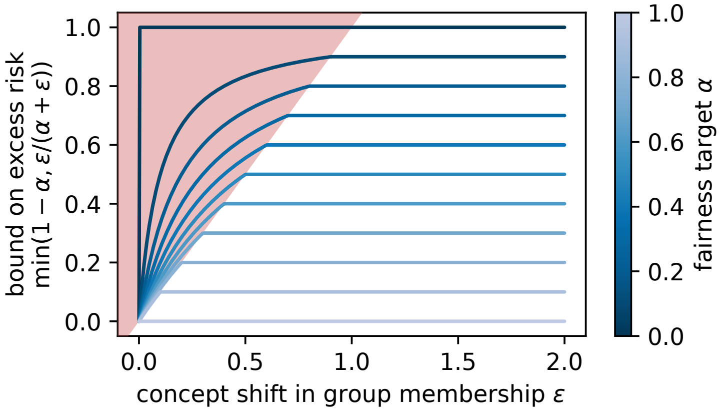

Figure 4.2: Shaded region is the combinations of (α, ε) covered in Example 4.1, matching the worst-case dependency on the fairness target α in Theorem 4.2. Note that $\varepsilon/(\alpha+\varepsilon) \leq 1-\alpha$.

– $Y = X$ (binary classification).

– $p[X = 1 \mid A = 1] = \frac{1}{2}(1 - \alpha - \varepsilon)$ and $p[X = 1 \mid A = 2] = \frac{1}{2}(1 + \alpha + \varepsilon)$.

– $q[X = 1 \mid A = 1] = \frac{1}{2}(1 - \alpha)$ and $q[X = 1 \mid A = 2] = \frac{1}{2}(1 + \alpha)$.

There is no shift in (X, Y), whereas the shift in (X, A) is

$$\mathbb{E}_{X\sim p_X}[|p[A = a \mid X] - q[A = a \mid X]|] = \varepsilon \quad \forall a \in \{1, 2\}. \tag{4.11}$$

Let h_p and h_q be the Bayes-optimal classifiers satisfying α-approximate SP under p and q, respectively. Then, under the 0–1 loss, the excess risk of h_p on q is

$$R_q(h_p) - R_q(h_q) = \frac{\varepsilon}{2(\alpha + \varepsilon)}. \tag{4.12}$$

The dependency on α can be eliminated when the classifier is attribute-aware, and the fairness criterion is either statistical parity or binary-class equalized odds—highlighting the robustness benefits of attribute awareness.

Corollary 4.1. Assume the attribute awareness (that is, the classifier takes the form $\mathcal{X} \times [M] \to [K]$). Under the same conditions as in Theorem 4.2, suppose the fairness violation bound ε is instantiated via Theorem 4.1, and the fairness criterion is statistical parity or binary-class equalized odds ($K = 2$),[1] then

$$\begin{aligned} &R_q(h_p) - R_q(h_q) \\ &\quad \leq \|\ell\|_\infty \Big(2\, D_{1,(L+L')K}(p_X, q_X) + \sum_{k\in[K]} \mathbb{E}_{X\sim p_X}[|p[Y=k \mid X] - q[Y=k \mid X]|] + 4\varepsilon K \Big). \end{aligned} \tag{4.13}$$

Technically, this refinement upon Theorem 4.2 is because in the analysis, we can use properties of these criteria to construct attribute-aware classifiers that are fair with respect to p while staying close to h_q, without an α-dependency in their distance: in other words, the ability to restore fairness on a violating classifier with minimal modification.

4.3 CALIBRATION OF GROUP PREDICTOR

The analysis above shows that if a fair algorithm infers group membership using a proxy g that differs from the one on the test distribution, it can induce a shift in the algorithm's implied group distribution. For instance, LinearPost relies on the provided pointwise risk r and group predictor g when constructing its post-processing transformation. If g does not reflect the test distribution (more specifically, is miscalibrated), the induced distribution will deviate from the actual test distribution, and the resulting classifier may fail to satisfy the specified fairness constraints. More concretely, for LinearPost, following Theorem 4.1:

Corollary 4.2. Let $h \leftarrow \textsc{LinearPost}(r, g, \xi = 0, p_X)$ be the deterministic classifier returned by Algorithm 3.1 when applied at the population level of p_X without noise perturbation. Let $\alpha = V_p(h)$ denote the target fairness level. Under Assumption 3.1 (so by Theorem 3.1, h is the Bayes-optimal fair classifier on p), the fairness violation of h on q satisfies

$$V_q(h) \leq \alpha + \max_{i\in[G]} \frac{3\|B\|_{\infty,1}}{p[Z_i = 1]} \Big(D_{\mathrm{TV}}(p_X, q_X) + \underbrace{\mathbb{E}_{X\sim p_X}[|g(X)_i - q[Z_i = 1 \mid X]|]}_{\text{concept shift in } Z} \Big). \tag{4.14}$$

This result shows that the classifier returned by LinearPost may violate fairness on the test distribution q whenever the group predictor provided to it does not match the conditional distribution of Z under q.

Given group-labeled examples $(x^{(j)}, z^{(j)})_j$ drawn from the test distribution q, this section addresses fairness violations from concept shift in the LinearPost framework by *calibrating*

[1] The result extends to TPR and FPR parity via a similar analysis for equalized odds.

the group predictor g with respect to q before post-processing. To this end, we introduce two calibration methods.

4.3.1 Multicalibration and Distribution Calibration

We begin by defining the notion of calibration and its more fine-grained variant, multicalibration.

Definition 4.3 (Calibration)**.** Let $\mathbb{P}$ be a distribution over random variables $X \in \mathcal{X}$ and $Y \in \{0,1\}$. A real-valued binary-class predictor $f : \mathcal{X} \to [0,1]$ for the binary event Y is said to be *calibrated* if

$$\mathbb{P}[Y = 1 \mid f(X) = p] = p \quad \forall p \in [0,1]. \tag{4.15}$$

Calibration requires that whenever the predictor f outputs a probabilistic forecast p, the true probability of $Y = 1$ matches p. We next introduce a stronger notion, *multicalibration* [75, 128], that requires calibration to hold simultaneously across different subpopulations of X.

Definition 4.4 (Multicalibration)**.** Under the same setup in Definition 4.3, let $\mathcal{S}$ be a partition of $\mathcal{X}$, then f is said to be *multicalibrated* with respect to $\mathcal{S}$ if

$$\mathbb{P}[Y = 1 \mid f(X) = p, X \in S] = p \quad \forall p \in [0,1],\, S \in \mathcal{S}. \tag{4.16}$$

To quantify the degree of miscalibration, we use the *expected calibration error* [129], defined as the average absolute difference between the predicted value p and the true frequency of $Y = 1$; when considering multicalibration, we also condition on and average over partition cells $S \in \mathcal{S}$:

$$\mathrm{ECE}_{\mathcal{S}}(f) = \sum_{\substack{p\in[0,1]\\ S\in\mathcal{S}}} |\mathbb{E}[(Y - p)\, \mathbb{1}[f(X) = p, X \in S]]|. \tag{4.17}$$

Now, we show that LinearPost can achieve fairness as long as the group predictor g satisfies multicalibration on distribution q, relaxing the requirement in Corollary 4.2 that g must match the Bayes-optimal group predictor on q:

Corollary 4.3. Under the same conditions as Corollary 4.2, let

$$\mathcal{S}_i = \big\{\{x \in \mathcal{X} : r(x) = u, g_{-i}(x) = v\} : u \in \mathbb{R}^K_{\geq 0}, v \in [0,1]^{G-1}\big\}, \tag{4.18}$$

where $g_{-i} : \mathcal{X} \to [0,1]^{G-1}$ is the group predictor without the i-th component, and $\mathcal{S}_i$ defines the level sets of (r, g_{-i}) jointly. Let the expected multicalibration error of g_i with respect to

$\mathcal{S}_i$ be

$$\mathrm{ECE}_{\mathcal{S}_i}(g_i) = \sum_{\substack{v\in[0,1]\\ S\in\mathcal{S}_i}} \left|\mathbb{E}_{(X,Z_i)\sim q}[(Z_i - v)\,\mathbb{1}[g_i(X) = v, X \in S]]\right|, \tag{4.19}$$

then

$$V_q(h) \le \alpha + \max_{i\in[G]} \frac{3\|B\|_{\infty,1}}{p[Z_i = 1]}(D_{\mathrm{TV}}(p_X, q_X) + \mathrm{ECE}_{\mathcal{S}_i}(g_i)). \tag{4.20}$$

This result shows that the effect of concept shift on fairness can be controlled if each component of the group predictor satisfies a multicalibration condition. Similar conditions have been established for the binary-class setting in [24], and Diana et al. [101] show that multicalibrated group predictors are sufficient proxies for the true group membership to support learning optimal fair classifiers via in-processing.

Instantiation Under Standard Fairness Criteria. To illustrate how this multicalibration condition can be achieved in practice, consider the case where LinearPost is applied to satisfy EO fairness.[2] This entails setting the group indicators as $Z_{a,k} = \mathbb{1}[A = a, Y = k]$, $Z \in \{0,1\}^{M\times K}$ (and one-hot), and training a group predictor to jointly predict (A, Y) (an MK-way classification problem):

$$g_{a,k}(x) = p[A = a, Y = k \mid X = x] \quad \forall a \in [M],\, k \in [K]. \tag{4.21}$$

Suppose further that the pointwise risk function (Definition 2.1) is derived from this group predictor using the fact that $p[Y = k \mid X = x] = \sum_a p[A = a, Y = k \mid X = x] = \sum_a g_{a,k}(x)$, so that

$$r(x)_k = \sum_{y\in[K]} \ell(y,k) \sum_{a\in[M]} g_{a,y}(x). \tag{4.22}$$

Under this construction, the level sets of $(r, g_{-(a,k)})$ are contained within the level sets of g:

$$\begin{aligned}\mathcal{S}_{a,k} &= \left\{\left\{x \in \mathcal{X} : r(x) = u, g_{-(a,k)}(x) = v\right\} : u \in \mathbb{R}^K_{\ge 0}, v \in [0,1]^{MK-1}\right\} && (4.23)\\ &\subseteq \left\{\left\{x \in \mathcal{X} : g(x) = v\right\} : v \in \Delta^{M\times K}\right\}. && (4.24)\end{aligned}$$

Therefore, the expected multicalibration error in Corollary 4.3 can be upper bounded by

$$\mathrm{ECE}_{\mathcal{S}_{a,k}}(g_{a,k}) \le \sum_{v\in\Delta^{M\times K}} \left|\mathbb{E}_{(X,Z_{a,k})\sim q}[(Z_{a,k} - v_{a,k})\,\mathbb{1}[g(X) = v]]\right|. \tag{4.25}$$

[2] This is downward compatible with SP, TPR, FPR, and AP fairness, because the group predictors they require are subsumed by the group predictor for EO; see the experiment setup of Section 3.4.

This means that multicalibration of $g_{a,k}$ with respect to the level sets of $(r, g_{-(a,k)})$ is implied by multicalibration of g with respect to its own level sets—namely, g is (multiclass) calibrated. This property is referred to as *distribution calibration* [39] in the multiclass calibration literature, and the right-hand side of Eq. (4.25) can be interpreted as the expected distribution calibration error.

Definition 4.5 (Distribution Calibration)**.** Let $\mathbb{P}$ be a distribution over random variables $X \in \mathcal{X}$ and $Y \in [N]$. A real-valued multiclass predictor $f : \mathcal{X} \to \Delta^N$ for the categorical event Y is said to be *distribution-calibrated* if

$$\mathbb{P}[Y = i \mid f(X) = p] = p_i \quad \forall p \in \Delta^N,\, i \in [N]. \tag{4.26}$$

Given labeled examples of (X, Z) from q, calibration of g is typically performed by learning a transformation $\Delta^{M\times K} \to \Delta^{M\times K}$. Common approaches include temperature scaling and Platt scaling [130, 131], which essentially amount to training a (sparse) linear model with softmax activation under the log loss, as well as isotonic regression [132]. Standard machine learning toolkits provide calibration methods with multiclass extensions, such as scikit-learn,[3] though effective multiclass calibration remains an active area of research [133, 134, 135].

4.3.2 Decision-Based Calibration

We have shown above that the fairness violation of the classifier returned by LinearPost, due to inaccuracies in the group predictor, can be mitigated by calibrating the group predictor to satisfy distribution calibration on the test distribution. While the idea is simple, in practice, distribution calibration can be difficult to achieve due to the curse of dimensionality in the number of classes; more concretely, Definition 4.5 show that *auditing* distribution calibration requires conditioning on and evaluating over all possible predictor outputs in Δ^N, which is an $(N-1)$-dimensional space.

This section shows that the multicalibration requirement in Corollary 4.3 can be further relaxed to a decision-based calibration condition, which can be achieved by an iterative algorithm with polynomial runtime (both the condition and the algorithm are adopted from [40]):

Definition 4.6 (Decision Calibration)**.** Let $\mathbb{P}$ be a distribution over random variables $X \in \mathcal{X}$ and $Y \in \{0,1\}^N$, and $\mathcal{H}$ be a class of multiclass classifiers $\mathcal{X} \times [0,1]^N \to [K]$. A real-valued

[3] https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html

multilabel predictor $f : \mathcal{X} \to [0,1]^N$ for binary events $Y_1, \ldots, Y_N$ is said to be *decision-calibrated* with respect to $\mathcal{H}$ if

$$\mathbb{E}[Y_i - f_i(X) \mid h(X, f(X)) = k] = 0 \quad \forall i \in [N],\, h \in \mathcal{H},\, k \in [K]. \tag{4.27}$$

The motivation for decision calibration is as follows: suppose f is a proxy predictor for the label Y, and consider the special case where all $h \in \mathcal{H}$ depend on X only through the outputs of f. Then, if f is calibrated on all instances receiving the same class assignment, the predicted label probabilities from f can be used to evaluate the loss of the decision-maker $h \circ f$ accurately.

More importantly, unlike previous definitions of calibration, decision calibration does not require conditioning on the full range of f directly, but only through the outputs of classifiers in $\mathcal{H}$. If $\mathcal{H}$ is a structured class (for example, finite, or with bounded VC dimension), this reduces the complexity of both auditing and enforcing calibration (as will be demonstrated by the calibration algorithm below), making it more tractable in high-dimensional multiclass settings.

Now we relax the multicalibration requirement in Corollary 4.3 to decision calibration; in fact, this definition is directly applicable to analyzing classifiers returned by LinearPost, since these classifiers are linear post-processings (corresponding to the role of $\mathcal{H}$ in the definition of decision calibration) applied to the pointwise risk and group predictor (taking the role of f):

Corollary 4.4. Under the same conditions as Corollary 4.2, define the class of multiclass classifiers class $\mathcal{X} \times [0,1]^G \to [K]$

$$\mathcal{F} = \left\{ (x, v) \mapsto \operatorname*{arg\,min}_{k \in [K]} \left(r(x)_k + \beta_{0,k} + \sum_{i \in [G]} \beta_{i,k} v_i \right) : \beta \in \mathbb{R}^{G \times K} \right\} \tag{4.28}$$

(the same form as the post-processing transformation learned by LinearPost); note that $\mathcal{F}$ is constructed using the pointwise risk function r to which LinearPost will be applied.

Let the decision calibration error of the i-component of g with respect to $\mathcal{F}$ be

$$\varepsilon_i^{\text{dcal}}(g) = \max_{\substack{f \in \mathcal{F} \\ k \in [K]}} \left| \mathbb{E}_{(X, Z_i) \sim q}[(Z_i - g_i(X))\, \mathbb{1}[f(X, g(X)) = k]] \right|, \tag{4.29}$$

then

$$V_q(h) \le \alpha + \max_{i \in [G]} \frac{3\|B\|_{\infty,1}}{p[Z_i = 1]} \big(D_{\text{TV}}(p_X, q_X) + \varepsilon_i^{\text{dcal}}(g) \big). \tag{4.30}$$

Algorithm 4.1: Decision calibration

```
Input: Multilabel predictor $f : \mathcal{X} \to [0,1]^N$,
       class $\mathcal{H}$ of multiclass classifiers $\mathcal{X} \times [0,1]^N \to [K]$,
       labeled samples $\{x^{(j)}, y^{(j)}\}_{j=1}^{N}$, iteration limit $T$, tolerance $\tau$
Output: Multilabel predictor $\mathcal{X} \to [0,1]^N$
for $t \leftarrow 1$ to $T$ do
    /* find a witness function for calibration violation */
    $h \leftarrow \arg\max_{h' \in \mathcal{H}} \sum_k \|\widehat{\mathbb{E}}[(Y - f(X))\, \mathbb{1}[h'(X, f(X)) = k]]\|_2$ ;   // $\widehat{\mathbb{E}}$ is taken w.r.t. empirical distribution of samples
    if $\sum_k \|\widehat{\mathbb{E}}[(Y - f(X))\, \mathbb{1}[h(X, f(X)) = k]]\|_2 \le \tau$ then
        break ;
    end
    /* compute adjustments and update */
    $d_k \leftarrow \widehat{\mathbb{E}}[(Y - f(X))\, \mathbb{1}[h(X, f(X)) = k]] / \widehat{\mathbb{E}}[h(X, f(X)) = k]$, for all $k \in [K]$ ;
    $f \leftarrow (x \mapsto f(x) + \sum_k d_k\, \mathbb{1}[h(x, f(x)) = k]$ and project onto $[0, 1])$ ;
end
return $f$;
```

Given labeled examples (X, Z) from q, decision calibration of g with respect to $\mathcal{F}$ can be achieved by a polynomial-time algorithm that alternates between finding a witness function $f \in \mathcal{F}$ on which g violates calibration and updating g to restore calibration with respect to that f. This algorithm, described in Algorithm 4.1, is adopted from [40] and is based on the multicalibration algorithm of [75, 128], which in turn draws inspiration from the boosting literature [136, 137]. Zhao et al. further modify the algorithm to allow $\mathcal{F}$ to output soft class assignments [40, Algorithm 2], so that the inner optimization in Algorithm 4.1 can be solved (approximately) via gradient ascent.

The convergence of Algorithm 4.1 is proved by Zhao et al. [40, Theorem 2.2].

Proposition 4.1. Algorithm 4.1 terminates in $O(K/\tau^2)$ iterations.

4.4 ROBUST POST-PROCESSING

In this section, we address fairness violations caused by general distribution shifts by introducing a robustified version of LinearPost. Unlike the calibration algorithms in the previous section, which require knowledge of the (single) test distribution in the form of labeled training examples, here we characterize potential shifts by a collection $\mathcal{Q}$ of distributions, called the *uncertainty set*. This set can be specified without exact knowledge of the test distribution(s), and the goal is to learn a classifier that is fair with respect to not

Algorithm 4.2: Robust fair classification using the cutting-set method

```
Input: fairness constraints $(B, \mu, c)$, reference distribution $q^{(0)} \in \mathcal{Q}$,
       uncertainty set $\mathcal{Q}$, iteration limit $T$, tolerance $\tau$
Output: (Randomized) classifier $\mathcal{X} \to [K]$
$h \leftarrow \arg\min_{h'} R^{(0)}(h')$ s.t. $B\mu^{(0)}(h') \leq c$ ;                  // initialization
for $t \leftarrow 1$ to $T$ do
    $q^{(t)} \leftarrow \arg\max_{q \in \mathcal{Q}} V_q(h)$ ;                              // pessimization
    if $V^{(t)}(h) \leq \tau$ then
        break ;
    end
    $h \leftarrow \arg\min_{h'} R^{(0)}(h')$ s.t. $B\mu^{(t')}(h') \leq c, \forall t' \leq t$ ;  // update
end
return $h$;
```

just one, but all distributions in $\mathcal{Q}$, while minimizing the risk on a designated reference distribution $p \in \mathcal{Q}$ (also referred to as the training or source distribution):[4]

$$\min_{h:\mathcal{X}\to[K]} R_p(h) \quad \text{subject to} \quad B\mu_q(h) \leq c, \ \ \forall q \in \mathcal{Q}. \tag{4.31}$$

We present a meta-algorithm in Algorithm 4.2 for solving Eq. (4.31) and learning fair classifiers that are robust to $\mathcal{Q}$ (the specification of $\mathcal{Q}$ is elaborated in Section 4.4.2), based on the *cutting-set* method. Given a fair classification oracle (which we implement with LinearPost; Section 4.4.1) and a *pessimization* oracle that finds the worst-case $q \in \mathcal{Q}$ where fairness is most violated, the algorithm begins by training a fair classifier on the reference distribution (where labeled training data are available) $p = q^{(0)} \in \mathcal{Q}$, then iterates between finding a worst-case violating distribution $q^{(t)}$ and updating h so that it satisfies the fairness constraints on $q^{(t)}$ as well as on all previously encountered distributions. It terminates when no violating distribution is found beyond a tolerance τ, in which case the resulting classifier h satisfies $V_q(h) \leq \alpha + \tau$ for all $q \in \mathcal{Q}$ (up to additional slack if the pessimization oracle is approximate, in which case the algorithm may terminate prematurely), or if the iteration limit T is reached.

Proposition 4.2. When the input space $\mathcal{X}$ has finite support, $\mathcal{X} = \operatorname{supp}(q_X)$ shared across all $q \in \mathcal{Q}$ and $N = |\mathcal{X}| < \infty$ (for example, when learning from finite samples), and both the fair classification and pessimization oracles are exact, Algorithm 4.2 terminates in $O((\|B\|_{\infty,1}/\tau)^{NK})$ iterations.

[4] Our formulation minimizes only the source risk R_p, rather than the worst-case risk over $\mathcal{Q}$, that is, $\arg\min_h \max_{q\in\mathcal{Q}} R_q(h)$ s.t. $B\mu_q(h) \leq c$, $\forall q \in \mathcal{Q}$. The latter can be solved via an additional level of optimization; see, for example, the meta-algorithm of [26, Algorithm 1].

Proof Sketch. First, note that the fairness violation V is $\|B\|_{\infty,1}$-Lipschitz with respect to the Markov kernel of h under the L^∞ distance. By Definition 2.8 and Eq. (3.9), and Hölder's inequality,

$$\begin{aligned}
&|V(h) - V(h')| \\
&\leq \max_{j\in[C]} \left| \sum_{k\in[K]} B_{j,(k,*)} \int_{\mathcal{X}} (\pi_h(x,k) - \pi_{h'}(x,k))\, \mathbb{P}[X = x]\, \mathrm{d}x \right. \\
&\qquad \left. + \sum_{k\in[K],i\in[G]} B_{j,(k,i)} \int_{\mathcal{X}} (\pi_h(x,k) - \pi_{h'}(x,k))\, \mathbb{P}[X = x \mid Z_i = 1]\, \mathrm{d}x \right| && (4.32) \\
&\leq \max_{j\in[C]} \sum_{k\in[K]} |B_{j,(k,*)}| \left(\max_{x\in\mathcal{X}} (\pi_h(x,k) - \pi_{h'}(x,k)) \right) \int_{\mathcal{X}} \mathbb{P}[X = x]\, \mathrm{d}x \\
&\qquad + \max_{j\in[C]} \sum_{k\in[K],i\in[G]} |B_{j,(k,i)}| \left(\max_{x\in\mathcal{X}} (\pi_h(x,k) - \pi_{h'}(x,k)) \right) \int_{\mathcal{X}} \mathbb{P}[X = x \mid Z_i = 1]\, \mathrm{d}x && (4.33) \\
&\leq \|B\|_{\infty,1} \|\pi_h - \pi_{h'}\|_\infty. && (4.34)
\end{aligned}$$

Also, since $|\mathcal{X}| < \infty$, the Markov kernel of the randomized classifier h is represented as an $N \times K$ row-stochastic matrix.

Following the analysis of [41, Section 5.2] then shows that Algorithm 4.2 terminates in at most $O((\|B\|_{\infty,1}/\tau)^{NK})$ iterations. The argument is as follows: each time a violation exceeding τ is found, the updated h must be at least $\tau/\|B\|_{\infty,1}$ away (in L^∞ distance) from the current h to restore fairness, by the Lipschitz property of V. This effectively removes an ℓ_∞-ball of radius $\tau/\|B\|_{\infty,1}$ from the feasible region (hence called the *cutting-set* method). Since at most $O((\|B\|_{\infty,1}/\tau)^{NK})$ such balls can be packed into the space of randomized classifiers, whose Markov kernels are $N \times K$ row-stochastic matrices, the algorithm must terminate within this bound. In practice, however, far fewer iterations are needed; in our experiments, it typically terminates within 5 to 20 iterations.

4.4.1 Multiple-Distribution LinearPost

We implement the fair classification oracle required by Algorithm 4.2 on Lines 1 and 7 using LinearPost (Algorithm 3.1). However, the fair classification problem on Line 7 requires fairness to be satisfied on not one but multiple distributions simultaneously. We therefore extend Algorithm 3.1 to handle multiple distributions.

Let $p^{(0)}, p^{(1)}, \dots, p^{(T)}$ be $(T + 1)$ distributions over (X, Y, Z). We consider the multiple-distribution fair classification problem:

$$\min_{h:\mathcal{X}\to[K]} R^{(0)}(h) \quad \text{subject to} \quad B\mu^{(t)}(h) \le c \ \ \forall t \in \{0,1,\dots,T\}. \tag{4.35}$$

We first show that the Bayes-optimal fair classifier in this setting can also be expressed as a linear post-processing, in terms of the Bayes-optimal group predictors $g^{(t)}(x) = p^{(t)}[Z \mid X = x]$ on each distribution, and additionally, their importance weights relative to $p^{(0)}$, $w^{(t)} = p_X^{(t)}/p_X^{(0)}$:

Theorem 4.3. Given distributions $p^{(0)}, p^{(1)}, \dots, p^{(T)}$, define the *fair pointwise risk* $r_{\text{fair}} : \mathcal{X} \to \mathbb{R}^K$ as

$$r_{\text{fair}}(x)_k = r^{(0)}(x)_k + \beta_{k,*} + \sum_{t=0}^{T} \sum_{i\in[G]} \beta_{k,i}^{(t)} g^{(t)}(x)_i w^{(t)}(x) \quad \forall k \in [K], \tag{4.36}$$

where $r^{(0)}$ is the pointwise risk function on $p^{(0)}$, and with parameters $\beta_{k,*}$ and $\beta^{(t)} \in \mathbb{R}^{K\times G}$ given by

$$\beta_{k,*} = -\sum_{t=0}^{T} \sum_{j\in[C]} \psi_j^{(t)} B_{j,(k,*)} \tag{4.37}$$

and

$$\beta_{k,i}^{(t)} = -\sum_{j\in[C],i\in[G]} \frac{\psi_j^{(t)} B_{j,(k,i)}}{p^{(t)}[Z_i = 1]} \qquad \forall k \in [K],\, i \in [G],\, t \in \{0,1,\dots,T\}, \tag{4.38}$$

where $\psi^{(0)}, \psi^{(1)}, \dots, \psi^{(T)}$ are the optimal dual solutions to the linear program reformulation of Eq. (4.35), $\mathrm{LP2}(r^{(0)}, \{g^{(t)}\}_{t=0}^{T})$, defined in Appendix A.3.

Under the uniqueness condition in Assumption 3.1, which in this setting needs hold for all $p^{(t)}$ but can still be satisfied by perturbing $r^{(0)}$ as described in Section 3.2.2, the deterministic classifier given by $x \mapsto \arg\min_k r_{\text{fair}}(x)_k$ (with ties broken by selecting the smallest index k) is an optimal solution to Eq. (4.35).

Based on this representation result, we follow the derivation in Section 3.3 for the single-distribution LinearPost to obtain the multiple-distribution version in Algorithm 4.3. As in Algorithm 3.1, achieving fairness optimally on the reference distribution $p^{(0)}$ requires its pointwise risk $r^{(0)} : \mathcal{X} \to \mathbb{R}_{\ge 0}^K$, group predictor $g^{(0)} : \mathcal{X} \to [0,1]^G$, and samples drawn from $p_X^{(0)}$. To additionally achieve fairness on $p^{(1)}, \dots, p^{(T)}$, the algorithm also requires their group predictors $g^{(1)}, \dots, g^{(T)}$ and the corresponding importance weights $w^{(t)} = p_X^{(t)}/p_X^{(0)}$—this last requirement is expected because LinearPost is run with inputs drawn from $p_X^{(0)}$, so information about changes in p_X, here in the form of importance weights, is needed to

Algorithm 4.3: Multiple-distribution LinearPost

```
Input: Pointwise risk predictor $\hat{r}^{(0)} : \mathcal{X} \to \mathbb{R}^K_{\geq 0}$ for $p^{(0)}$,
       group predictors $\hat{g}^{(0)}, \hat{g}^{(1)}, \dots \hat{g}^{(T)} : \mathcal{X} \to [0,1]^G$,
       importance weights $\hat{w}^{(1)}, \dots, \hat{w}^{(T)} : \mathcal{X} \to [0, \infty)$,
       fairness constraints $(B, \mu, c)$, unlabeled samples $\{x^{(j)}\}_{j=1}^N$ from $p_X^{(0)}$,
       random noise $\xi \in \mathbb{R}^K$
Output: Randomized classifier $\mathcal{X} \to [K]$
$\{\hat{\psi}^{(t)}\}_{t=0}^T \leftarrow$ optimal dual values of $\widehat{\mathrm{LP2}}(\hat{r}^{(0)} + \xi, \{\hat{g}^{(t)}\}_{t=0}^T)$ ;  // empirical
  multiple-distribution fair classification linear program
$\hat{\beta}_{k,*} \leftarrow -\sum_{t,j} \hat{\psi}^{(t)}_j B^{(t)}_{j,(k,*)}$ for all $k \in [K]$ ;  // post-processing parameters
$\hat{\beta}^{(t)}_{k,i} \leftarrow -\sum_{j,i} \hat{\psi}^{(t)}_j B^{(t)}_{j,(k,i)} / \hat{p}^{(t)}[Z_i = 1]$ for all $i \in [G]$, $k \in [K]$, $t \in \{0, 1, \dots, T\}$ ;
return $x \mapsto \arg\min_k (\hat{r}^{(0)}(x)_k + \xi_k + \hat{\beta}_{k,*} + \sum_{t,i} \hat{\beta}^{(t)}_{k,i} \hat{g}^{(t)}(x)_i w^{(t)}(x))$ ;  // $w^{(0)} = 1$
```

model covariate shifts between $p^{(t)}$ and $p^{(0)}$. This also implies that only covariate shifts with bounded importance weights can be handled.

Algorithm 4.3 then solves a linear program formulation of the multiple-distribution fair classification problem (Eq. (4.35)) and returns a classifier expressed as a linear post-processing of $r^{(0)}$, $g^{(0)}$, and $g^{(1)}w^{(1)}, \dots, g^{(T)}w^{(T)}$; the post-processing parameters are similarly derived from the dual optimal values of the linear program. This post-processing transformation is a linear classifier on a $(K + GT + 1)$-dimensional feature space, so the sample complexity for estimating its parameters $\hat{\beta}$ follows the same analysis as Theorem 3.2 except for replacing the $\sqrt{(K+G)\log K}$ terms with $\sqrt{(K+GT)\log K}$.

4.4.2 Uncertainty Set Specification

The uncertainty set $\mathcal{Q}$ is specified to capture potential shifts from the reference distribution $p = q^{(0)}$, particularly the anticipated test distribution(s) on which the learned classifier will be deployed. Since we chose to implement the fair classification oracle using the multiple-distribution LinearPost (Algorithm 4.3), which requires information about any shifted distribution $q^{(t)} \neq q^{(0)}$ in the form of its group predictor $g^{(t)}$ and importance weights $w^{(t)}$, we accordingly represent each $q^{(t)} \in \mathcal{Q}$ by its decomposition into: concept shift, represented by $g^{(t)}$; and covariate shift, represented by $w^{(t)}$. Thus, $\mathcal{Q}$ consists of pairs (g, w), and this decomposition mirrors the one used in the analysis of Section 4.2.

Apart from requiring that each $q^{(t)} \in \mathcal{Q}$ be represented by a group predictor $g^{(t)} : \mathcal{X} \to [0,1]^G$ and an importance weight $w^{(t)} : \mathcal{X} \to [0, \infty)$ relative to $q^{(0)}$, both of which must extrapolate beyond the training examples (namely, the unlabeled samples ${x^{(j)}}_j$ drawn from

$p_X^{(0)}$), there are no restrictions on the specification of $\mathcal{Q}$. This flexibility allows a wide range of distribution shifts to be modeled by $\mathcal{Q}$:

– (Covariate Shift). If unlabeled samples from the test distribution are available, covariate shift can be modeled by first estimating the importance weights $\hat{w}_{\text{test}}$ between the reference and test distributions [115, 138], then setting $\mathcal{Q} = (g^{(0)}, \hat{w}_{\text{test}})$ as the singleton set containing only the estimated test distribution.

– (Concept Shift in Z). The concept shift setting considered when studying the calibration algorithms in Section 4.3 is (trivially) recovered by setting $\mathcal{Q} = (\hat{g}_{\text{test}}, 1)$, where $\hat{g}_{\text{test}}$ is the group predictor on the test distribution obtained by calibrating $g^{(0)}$ using labeled test examples.

– (Unknown or Adversarial Bounded Shifts). If nothing about the test distribution is known beyond a bound on its distance from $q^{(0)}$ [25, 26], we can model it as

$$\mathcal{Q} = \left\{(g, w) : \|g - g^{(0)}\|_\infty \le \varepsilon_{\text{CS}}, \|w - 1\|_\infty \le \varepsilon_{\text{IW}}\right\} \tag{4.39}$$

for some ε_{CS} and ε_{IW} that bound the concept and covariate shifts, respectively, and thus the distance from $q^{(0)}$.

Learning under noisy group labels falls under this case [25]. For example, if each group label $Z_i \in \{0, 1\}$ used to estimate $\hat{g}$ is flipped independently with probability ε, then the true group predictor g and the estimate $\hat{g} = g^{(0)}$ satisfy

$$\hat{g}(x)_i = (1 - \varepsilon)g(x)_i + \varepsilon(1 - g(x)_i) \implies |\hat{g}(x)_i - g(x)_i| \le \varepsilon \quad \forall x \in \mathcal{X}. \tag{4.40}$$

Parameterized Models for Unknown Bounded Shifts. For the uncertainty set containing arbitrary bounded shifts from the reference distribution $p = q^{(0)}$ (Eq. (4.39)), we generate $\mathcal{Q}$ by implementing each (g, w) pair—representing a distribution $q \in \mathcal{Q}$ via its group predictor $g : \mathcal{X} \to [0, 1]^G$ and importance weight $w : \mathcal{X} \to [0, \infty)$—using functions from a parameterized family, and applying regularization to enforce boundedness. For instance, if g and w are neural networks with nonlinear activations, classical universal approximation results show that they can represent any continuous functions [139, 140, 141, 142], so the resulting uncertainty set contains all distributions with continuous group predictors and importance weights.

To approximately solve the pessimization step, we search for the worst-case $q \in \mathcal{Q}$, represented by (g, w), by performing gradient ascent on the parameters of g and w to maximize the resulting fairness violation (or ascent-descent if auxiliary variables are involved in the definition of V; see Example 2.2):

$$\begin{aligned}&\max_{g,w}\big(V_{(g,w)}(h) - \lambda_{\mathrm{CS}} R_{\mathrm{CS}}(g) - \lambda_{\mathrm{IW}} R_{\mathrm{IW}}(w)\big)\\ &\qquad = \max_{g,w}\left(\max_{j\in[C]} B_j \mu(h; g, w) - \lambda_{\mathrm{CS}} R_{\mathrm{CS}}(g) - \lambda_{\mathrm{IW}} R_{\mathrm{IW}}(w)\right)\end{aligned} \tag{4.41}$$

where the regularization strengths $\lambda_{\mathrm{CS}}, \lambda_{\mathrm{IW}} \in [0, \infty]$ reflect the anticipated bounds $\varepsilon_{\mathrm{CS}}, \varepsilon_{\mathrm{IW}}$. The term μ depends on g, w, and the current h (fixed during pessimization), or more precisely, its Markov kernel π_h, via

$$\begin{aligned}\mu_{k,*} &= q[\widehat{Y} = k] = \int_{\mathcal{X}} \pi_h(x, k) w(x) p[X = x] \,\mathrm{d}x && \forall k \in [K],\\ \mu_{k,i} &= q[\widehat{Y} = k \mid Z_i = 1] = \int_{\mathcal{X}} \frac{g(x)_i}{q[Z_i = 1]} \pi_h(x, k) w(x) p[X = x] \,\mathrm{d}x && \forall k \in [K],\ i \in [G],\end{aligned} \tag{4.42}$$

and

$$q[Z_i = 1] = \int_{\mathcal{X}} g(x)_i w(x) p[X = x] \,\mathrm{d}x \quad \forall i \in [G]. \tag{4.43}$$

All expectations above are taken with respect to $X \sim p_X$, the marginal input distribution of the reference, so they can be estimated from samples drawn from p_X.

We use soft regularizers R_{CS} and R_{IW} to limit the drift from p, rather than strictly enforcing the L^∞-distance constraints in Eq. (4.39). Specifically, we use the KL divergence (equivalently, cross-entropy) between (g, w) and their corresponding quantities under the reference distribution p. The regularizer R_{CS} computes the average KL divergence between $p[Z_i \mid X]$ (given by the group predictor $g^{(0)}$ of the reference distribution) and $g(X)_i$ over X, summed over $i \in [G]$:

$$R_{\mathrm{CS}}(g) = -\sum_{i\in[G]} \mathbb{E}_{X\sim p_X}[p[Z_i = 0 \mid X] \ln(1 - g(X)_i) + p[Z_i = 1 \mid X] \ln g(X)_i]. \tag{4.44}$$

The regularizer R_{IW} computes the KL divergence between p_X and $q_X = p_X \cdot w$:

$$R_{\mathrm{IW}}(g) = -\int_{\mathcal{X}} p[X = x] \ln(p[X = x] w(x)) \,\mathrm{d}x = -\,\mathbb{E}_{X\sim p_X}[\ln w(x)]. \tag{4.45}$$

In our experiments, we replace the inner max in Eq. (4.41) with a weighted sum using a softmax to improve optimization performance. We parameterize both g and w using one-hidden-layer LeakyReLU networks, where the input features are the G-dimensional logits of the predicted group memberships on the reference distribution p (namely, the output of $g^{(0)}$), rather than the original features in $\mathcal{X}$ for simplicity. For example, we set $w(x) = C\exp(f(\ln g^{(0)}(x)))$, where $f : \mathbb{R}^G \to \mathbb{R}$ is the neural net and C is a normalization term such that $\sum_{j=1}^{N} w(x^{(j)})p[X = x^{(j)}] = 1$ to ensure w represents a valid importance weight on the training data (if p is the empirical distribution on the training data, then $p[X = x^{(j)}] = 1/N$ for all j).

4.5 EXPERIMENTS

> Omitted implementation details, such as dataset and fair algorithm descriptions, split sizes, hyperparameter settings, and the evaluation protocol, are provided in the unified Experiment Details section in Appendix B.

In Sections 4.5.1 to 4.5.3, we evaluate the robustified LinearPost introduced in Section 4.4 under three scenarios: geographic shift, group label noise, and worst-case covariate shift. In each case, we use the parameterized models described in Section 4.4.2 to implement the uncertainty set. The last two experiments, in particular, are designed to validate the roles of the concept and covariate shift components in our decomposed construction of the uncertainty set.

In Section 4.5.4, we follow-up the experiments in Section 3.4 by applying the calibration algorithms from Section 4.3 to refine the base predictions models, and improve the subsequent performance of LinearPost in achieving higher fairness.

4.5.1 Geographic Shift

Experiment Setup. To induce geographic shift, we partition the ACSIncome2-Sex dataset by the individual's home US state or territory into 51 subsets, retaining only the 27 largest by sample size. Among these, California (CA) is the largest with 78,281 examples, Louisiana (LA) is the smallest with 8,240 examples, and Florida (FL) has 39,541 examples. We set CA as the source training distribution and treat the remaining 26 states as test distributions. We evaluate SP, TPR, and EO fairness under the attribute-blind setting, removing the `sex` column.

We evaluate robust LinearPost with an uncertainty set constructed by implementing the concept shift and covariate shift components using one-hidden-layer LeakyReLU networks, as described in Section 4.4.2. For the regularization strengths, we sweep $\lambda_{\text{IW}} \in \{20, 50\}$ and $\lambda_{\text{CS}} \in \{200, 500\}$, and focus on $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 500$ in Figs. 4.3 to 4.5, as this combination

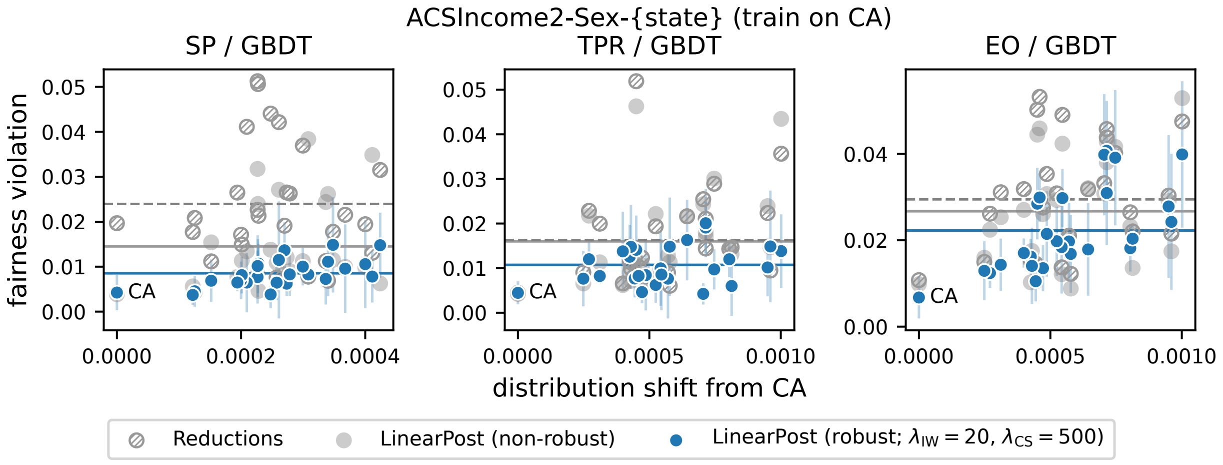

Figure 4.3: Fairness violation on all 27 states for fair classifiers trained on CA using Reductions, non-robust and robust LinearPost, with the single tolerance setting that minimizes the average violation. See Table 4.1 for the tolerance settings, average accuracies, and violations (the latter also indicated by horizontal lines).

yields the best validation tradeoffs; results for the other regularization strengths are shown in Fig. 4.15. Non-robust baselines include (non-robust) LinearPost and Reductions.

The base model is a gradient-boosted decision tree (GBDT). LinearPost is applied following the "pre-train then post-process" procedure on separate pre-training and post-processing splits, whereas for in-processing Reductions, these splits are merged into a single training split. For robust LinearPost, the original post-processing split is further divided 50-50 into a pessimization split and a post-processing split, with the former used exclusively to optimize the parameterized distribution shift models in the pessimization step.

Results and Discussions. Figure 4.3 shows the fairness violation of fair classifiers trained on CA and evaluated on all 27 states (including CA), under each algorithm's best tolerance setting chosen to minimize the macro-average violation across all states on the validation set (not necessarily the strictest setting tested). Robust LinearPost achieves higher fairness both on average and in the worst-performing region compared to non-robust Reductions and LinearPost. However, the improvement is not uniform across all states. This is not unexpected, as the constructed uncertainty set may not always fully capture the true distribution shifts between CA and other states; moreover, the optimal tolerance setting may also vary across states. Nonetheless, the ability of robust LinearPost to reduce worst-case fairness violation highlights its practical value.

These fairness gains, however, come at the cost of reduced accuracy in most states, including CA. In Fig. 4.4, we plot the accuracy-fairness tradeoffs achieved by robust LinearPost

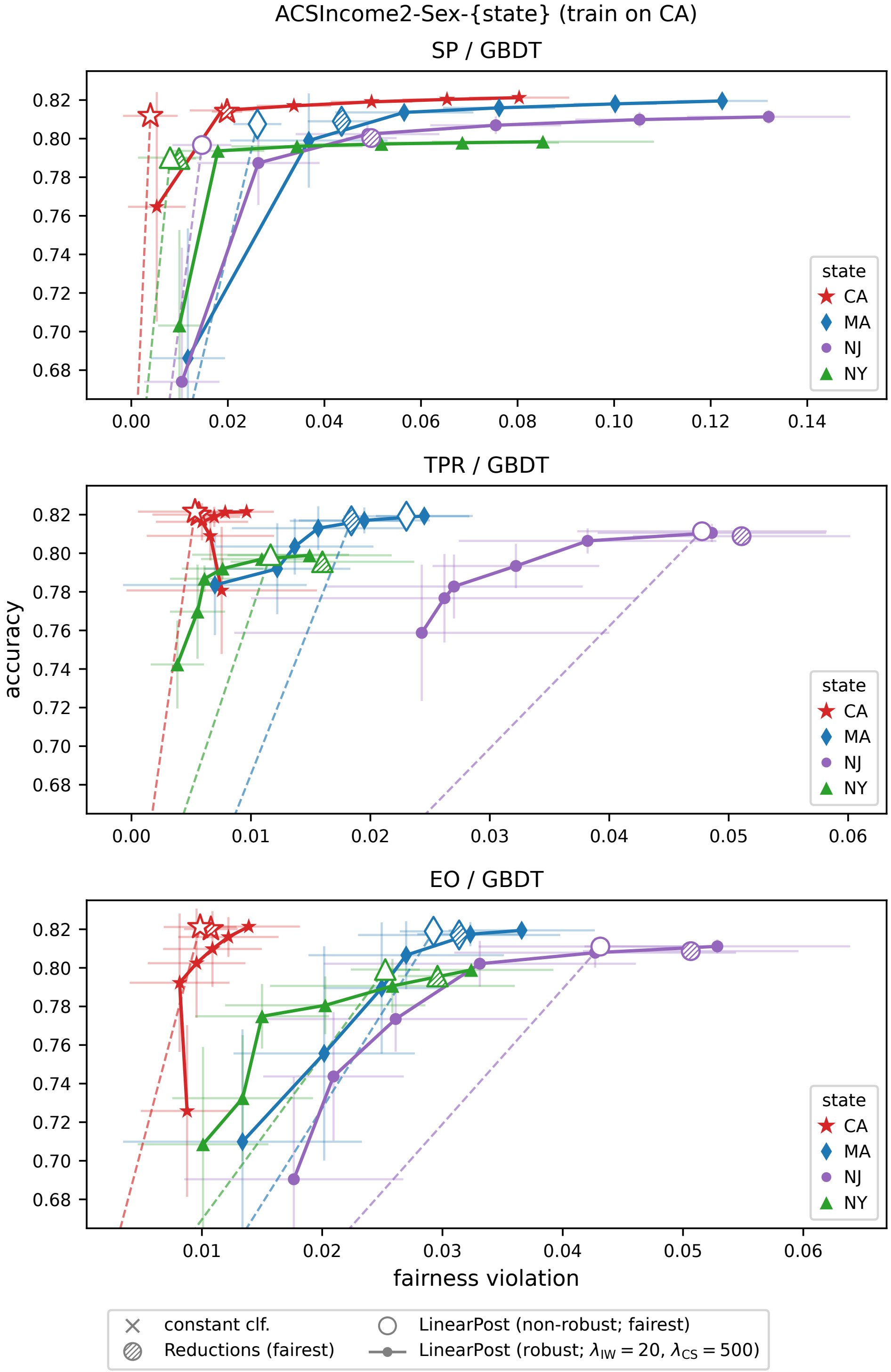

Figure 4.4: Accuracy-fairness tradeoffs on CA, MA, NJ, and NY for fair classifiers trained on CA using robust LinearPost. For comparison, we include the fairest classifiers on each state obtained with Reductions and non-robust LinearPost trained on CA, along with the linear interpolation between the fairer of the two and the constant-0 classifier (dashed lines).

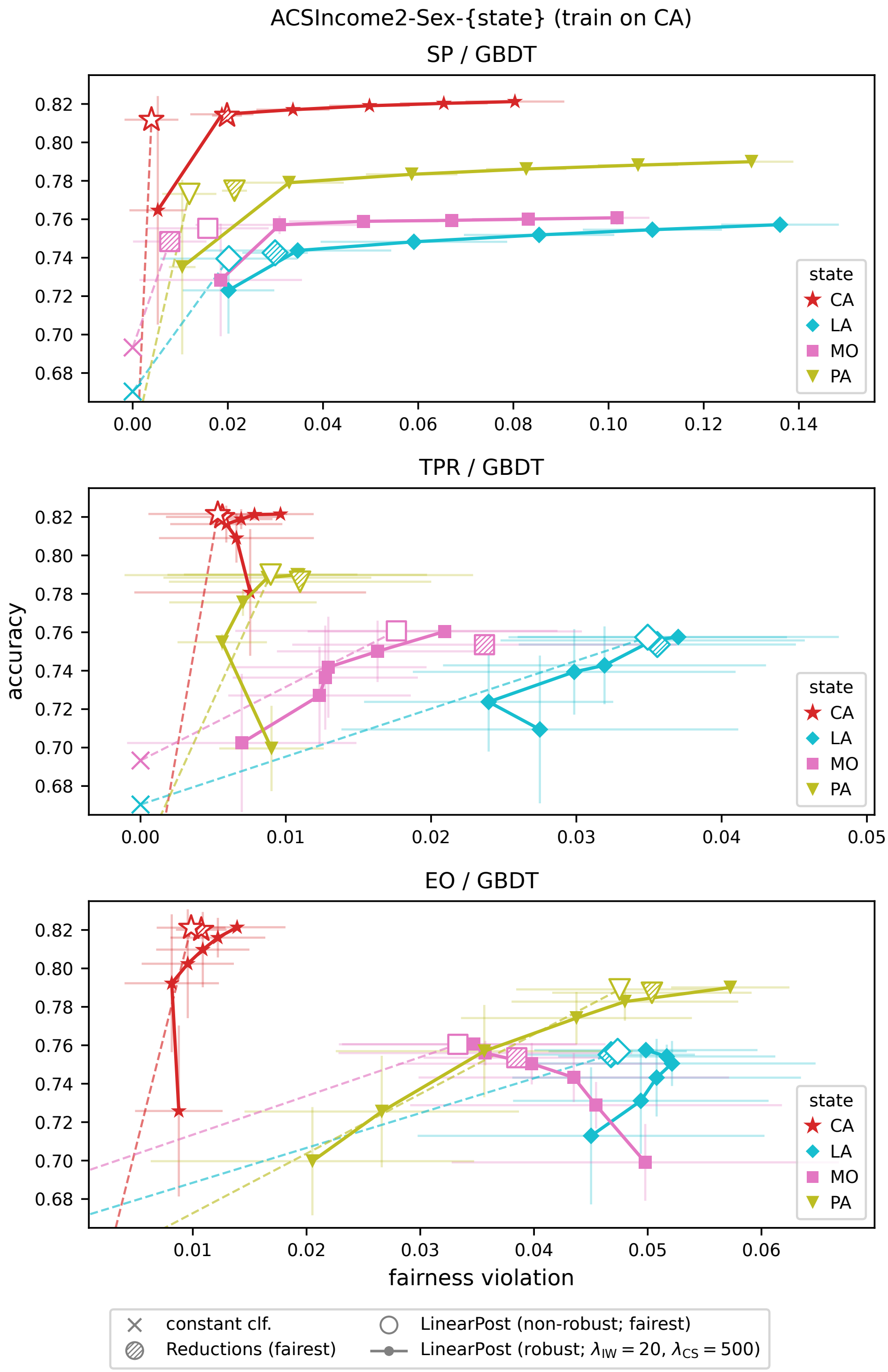

Figure 4.5: Accuracy-fairness tradeoffs on CA, LA, MO, and PA for fair classifiers trained on CA using robust LinearPost. For comparison, we include the fairest classifiers on each state obtained with Reductions and non-robust LinearPost trained on CA, along with the linear interpolation between the fairer of the two and the constant-0 classifier (dashed lines).

on MA, NJ, and NY under varying tolerances α. We also include, as a baseline, the interpolation between the fairest classifier obtained from the non-robust algorithms and the constant-0 classifier (which is trivially fair). For TPR and EO fairness, the tradeoff curves of robust LinearPost lie above the interpolated baseline, indicating that its improvements are non-trivial (this is not the case for SP, where the baselines already achieve high fairness).

In contrast, on LA, MO, and PA (Fig. 4.5), robust LinearPost sometimes fails to improve fairness, and its tradeoff curves fall below the interpolation. This is likely because the uncertainty set fails to capture the true underlying shifts in these cases, and our Eq. (4.31) problem formulation does not consider the worst-case risk (Footnote 4). Furthermore, while not illustrated in these plots, fairness does not always improve monotonically as the tolerance α is tightened. This may be due to the pessimization step not being solved exactly, as well as variability in its optimization process. These findings underscore the importance of using validation sets from the test distribution for hyperparameter tuning and model selection.

4.5.2 Group Label Noise

Experiment Setup. We evaluate robust LinearPost on ACSIncome2-Sex-FL under noisy group labels. Following [25], the sensitive attribute $A \in \{1, 2\}$ in the training set (both pre-training and post-processing splits) is randomly flipped with probability γ (the group label noise level).

As noted in Section 4.4.2, label noise induces a concept shift between the training and true distributions. Accordingly, using this knowledge, we apply robust LinearPost with an uncertainty set whose covariate shift component is fixed at 1 (no covariate shift) and whose concept shift component is parameterized by a one-hidden-layer LeakyReLU network. We sweep $\lambda_{\text{CS}} \in \{50, 200, 500\}$ for regularization strength. All other setups follow Section 4.5.1.

Results and Discussions. Figure 4.6 shows the fairness violations under increasing group label noise levels, using each algorithm's best tolerance setting chosen to minimize the macro-average violation on the validation set across noise levels (not necessarily the strictest setting tested). As expected, fairness violations increase with noise level. Robust LinearPost consistently achieves the lowest violations, with weaker regularization settings (λ_{CS}) providing better robustness by inducing a larger uncertainty set. These results confirm that the concept shift component of our uncertainty set construction indeed captures such shifts, and that robust LinearPost can, in turn, effectively mitigate their impact.

We plot the accuracy-fairness tradeoffs achieved by robust LinearPost under varying fairness tolerances in Fig. 4.7, compared to the interpolation between non-robust Linear-

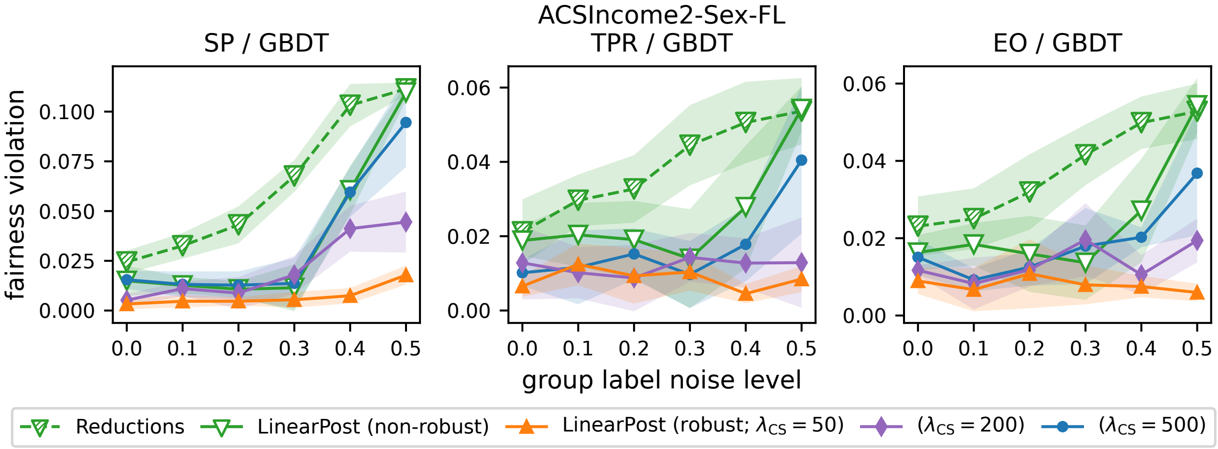

Figure 4.6: Fairness violation under increasing group label noise levels for fair classifiers trained using REDUCTIONS, non-robust and robust LINEARPOST, with the single tolerance setting that minimizes the average violation. See Table 4.2 for the tolerance settings.

POST and the constant 0 classifier. Robust LINEARPOST can achieve tradeoffs that lie above this interpolation line, indicating that its fairness improvements are non-trivial. However, for EO under large noise levels, the tradeoffs are no better than interpolation; we will discuss possible improvements in Section 4.5.3.

4.5.3 Worst-Case Covariate Shift

Experiment Setup. We evaluate robust LINEARPOST on ACSINCOME2-SEX-FL under adversarial covariate shift. At test time, for a given classifier, we solve for the worst-case perturbation to sample weights of test examples to maximize the fairness violation (within bounded ℓ_1 distance from empirical distribution), using the code by Mandal et al. [26].[5] We then compute both accuracy and fairness violation under the perturbed distribution.

Accordingly, we instantiate the uncertainty set for robust LINEARPOST with a fixed concept shift component given by the GBDT group predictor trained on ACSINCOME2-SEX-FL in the pre-training stage (no concept shift) and a covariate shift component parameterized by a one-hidden-layer LeakyReLU network. We sweep the covariate shift regularization strength $\lambda_{\text{IW}} \in \{5, 20, 50\}$. All other setups follow Section 4.5.1.

Results and Discussions. Figure 4.6 shows the fairness violations of the evaluated classifiers under increasing magnitudes of adversarial covariate shift, using each algorithm's

[5] `https://github.com/samuel-deng/Ensuring-Fairness-Beyond-the-Training-Data/tree/f8f59390e78696aaad66a8a5ca4087613fe0255c/main/Part3_Comparisons`

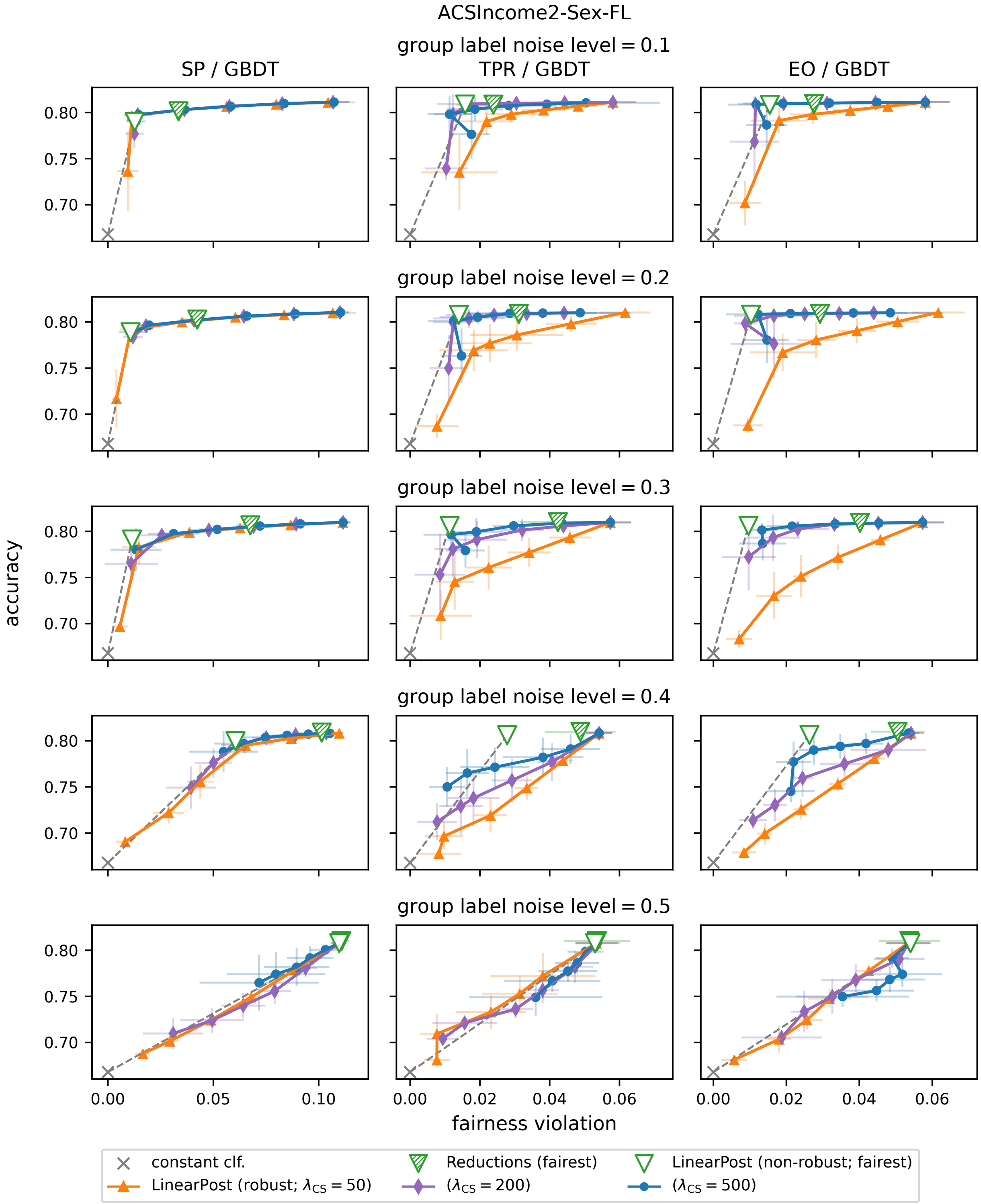

Figure 4.7: Accuracy-fairness tradeoffs under five group label noise levels for fair classifiers trained using robust LINEARPOST. For comparison, we include the fairest classifiers obtained with REDUCTIONS and non-robust LINEARPOST, along with the linear interpolation between the latter and the constant-0 classifier (dashed lines).

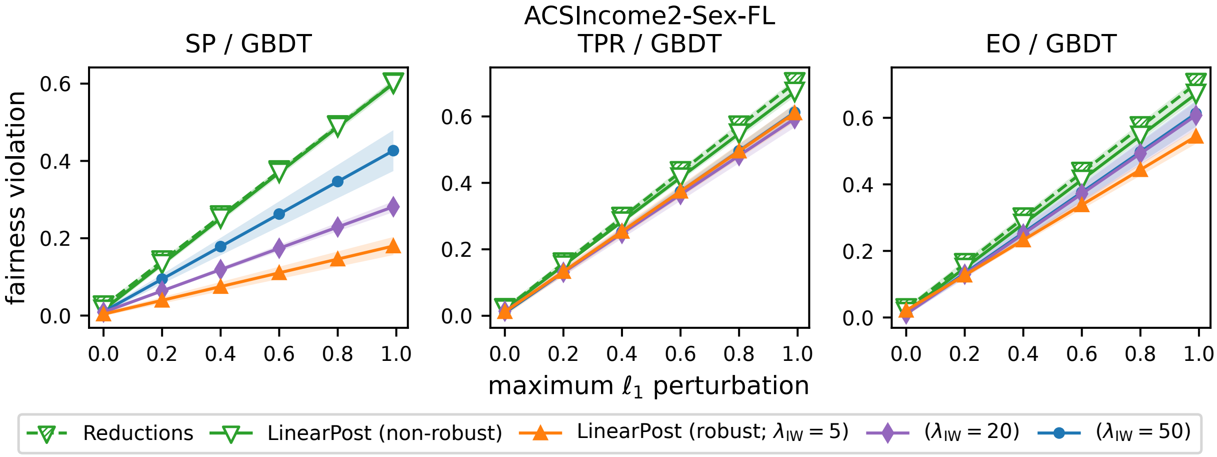

Figure 4.8: Fairness violation under increasing worst-case covariate shifts for fair classifiers trained using Reductions, non-robust and robust LinearPost, with the single tolerance setting that minimizes the maximum violation. See Table 4.3 for the tolerance settings.

best tolerance setting chosen to minimize the maximum violation on the validation set (not necessarily the strictest setting tested). Again, as expected, fairness violations grow with perturbation magnitude, and robust LinearPost consistently yields the lowest violations. These results validate the role of the covariate shift component in our uncertainty set construction.

We note that the gains from robust LinearPost are smaller under TPR and EO fairness than under SP. This may be partly due to the difficulty of the task: on the related Adult dataset, the robust fair algorithm of Mandal et al. [26] also achieved only modest improvements against worst-case covariate shift for EO fairness. Potential directions for improvement include incorporating worst-case risk into the problem formulation (Eq. (4.31) and Footnote 4), and strengthening the covariate shift model; for example, by allowing the neural network to take the original features in $\mathcal{X}$ as input, rather than the GBDT outputs used in our current setup (Section 4.4.2).

In Fig. 4.9, we plot the accuracy-fairness tradeoffs of robust LinearPost along with the baseline formed by interpolating between the fairest non-robust LinearPost classifier and the constant-0 classifier. Robust LinearPost achieves tradeoffs lying above this baseline, indicating that its fairness improvements are non-trivial. For TPR fairness, all tested regularization strengths $\lambda_{\mathrm{IW}} \in \{5, 20, 50\}$ yield similar fairness, with weaker regularization lowering accuracy without improving fairness—in particular, under EO, setting $\lambda_{\mathrm{IW}} = 5$ resulted in tradeoffs below the interpolation baseline.

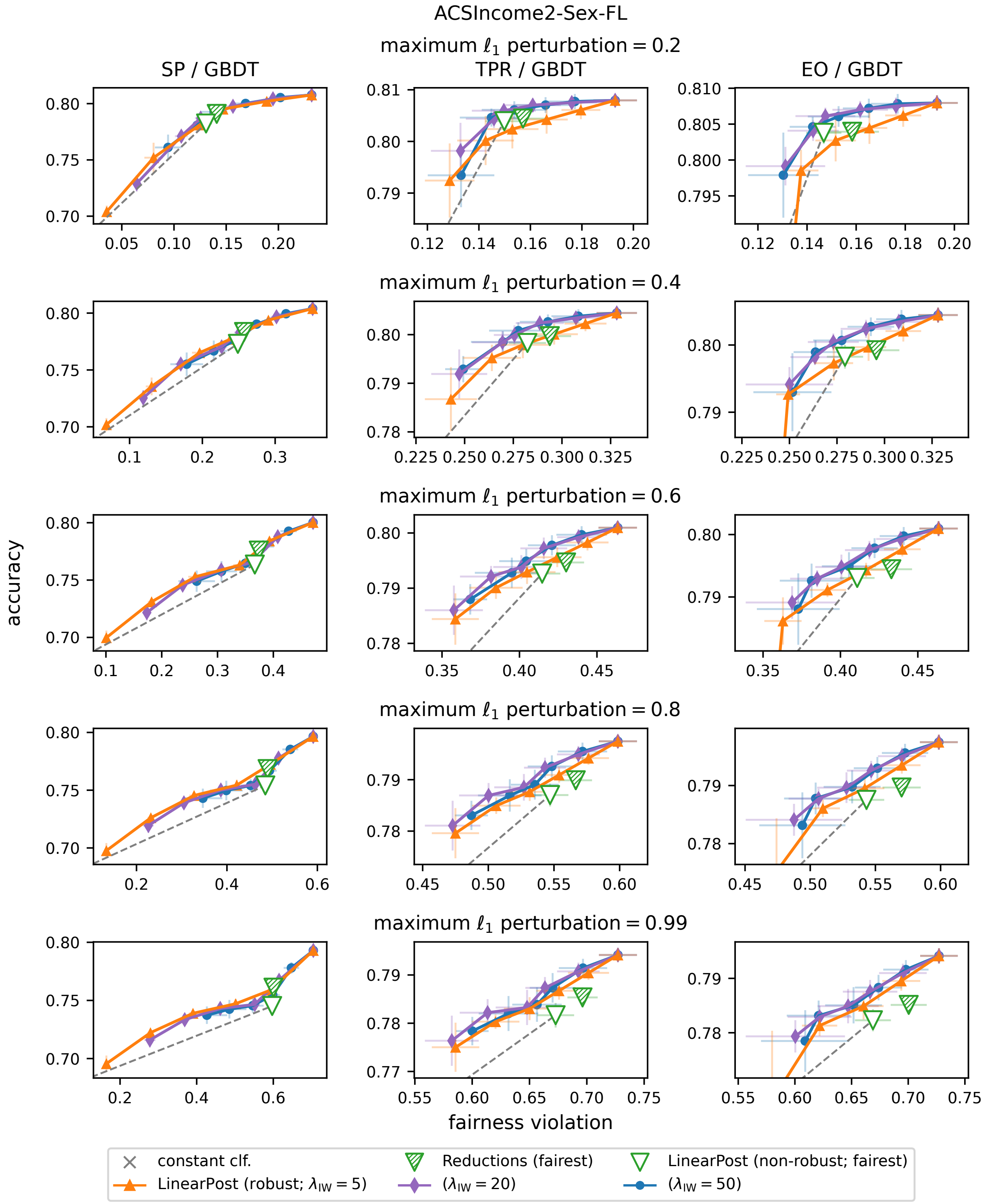

Figure 4.9: Accuracy-fairness tradeoffs under worst-case covariate shifts of five magnitudes for fair classifiers trained using robust LinearPost. For comparison, we include the fairest classifiers obtained with Reductions and non-robust LinearPost, along with the linear interpolation between the latter and the constant-0 classifier (dashed lines).

4.5.4 Calibration

Experiment Setup. The experiments in this section are follow-ups to those in Section 3.4 and use the same overall setup.

We further split the original post-processing split 50-50 into a calibration split and a post-processing split. The calibration algorithms are applied to calibrate the base model $f : \mathcal{X} \to \Delta^{M \times K}$, which jointly predicts (A, Y) and is trained on the pre-training split, to the ground-truth labels on the calibration split.

For distribution calibration (Section 4.3.1), we use `CalibratedClassifierCV` from scikit-learn with both the `sigmoid` and `isotonic` options, corresponding to Platt scaling [130] and isotonic calibration [132], respectively.

For decision calibration (Section 4.3.2), rather than following Algorithm 4.1 and using the worst-case witness function to apply corrections, we take the post-processing transformation learned by LinearPost on the current base model as the witness (this transformation also lies in the class $\mathcal{F}$ defined in Corollary 4.4). This modification does not invalidate the convergence result in Proposition 4.1. The decision calibration procedure therefore iterates between applying LinearPost to learn a post-processing transformation and updating the base model to satisfy decision calibration with respect to that transformation. We run this iterative process for 10 iterations.

Discussions. We present accuracy-fairness tradeoff curves in Section 4.5.5 below and summarize the main observations here.

Overall, calibration generally improves the downstream performance of LinearPost, leading to higher (or at least equal) fairness across Adult, COMPAS (except under FPR with logistic regression), ACSIncome2-Sex, and ACSIncome5-Race, and in some cases also improving accuracy. Among the calibration methods, Platt scaling (`sigmoid`) typically underperforms isotonic and decision calibration (with few exceptions), and causes notable accuracy drops on ACSIncome2-Sex, ACSIncome5-Race, and BiasBios under TPR and EO fairness. Between isotonic and decision calibration, there is no clear overall winner.

On BiasBios, which is a 28-way classification task (or 56-way when jointly predicting sex and occupation), none of the calibration algorithms improve over the baseline. This is likely due to extreme subgroup imbalance (for example, the rate for “female rapper” is 0.0346%, while “male professor” is 16.48%) and insufficient training data to capture such small subgroups.

4.5.5 Results: Tradeoff Curves for Section 4.5.4

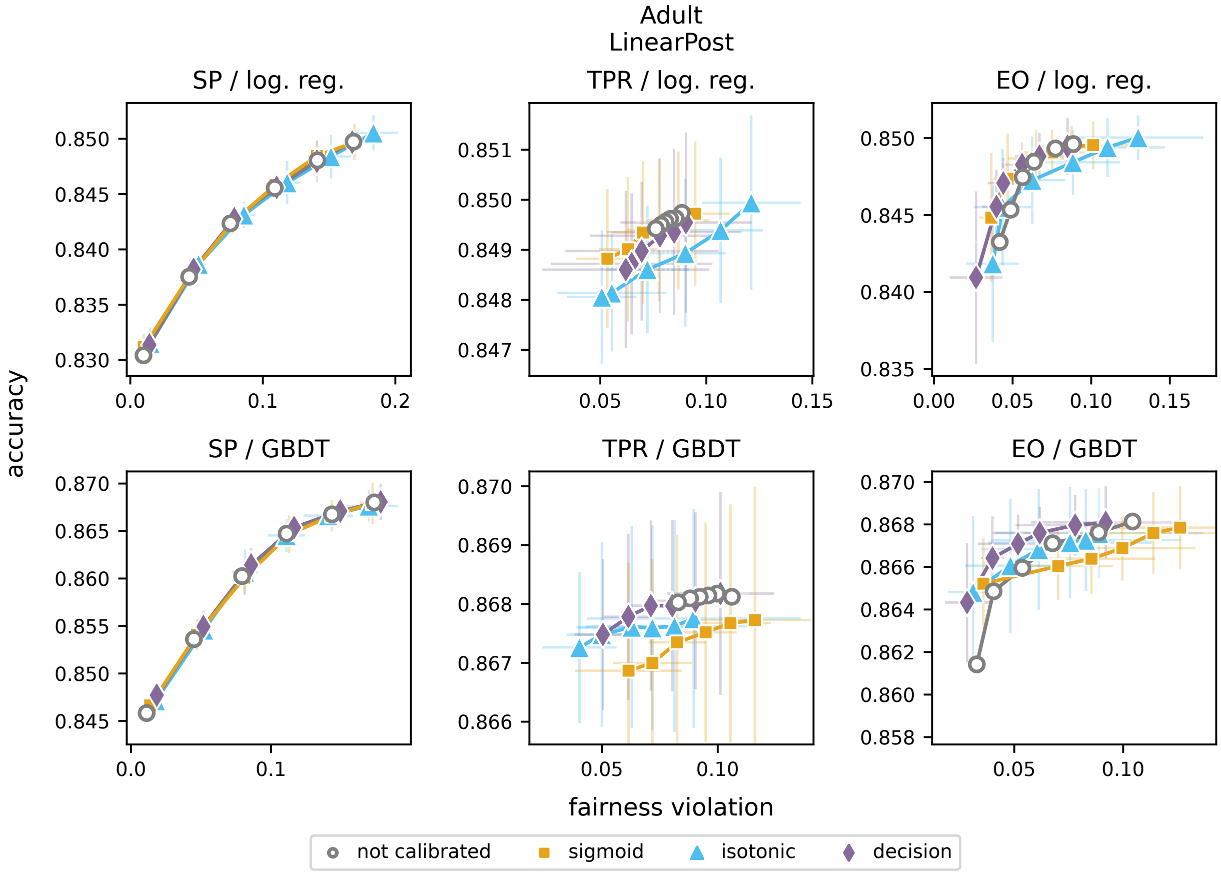

Figure 4.10: Accuracy-fairness tradeoffs on ADULT for Section 4.5.4 calibration experiments.

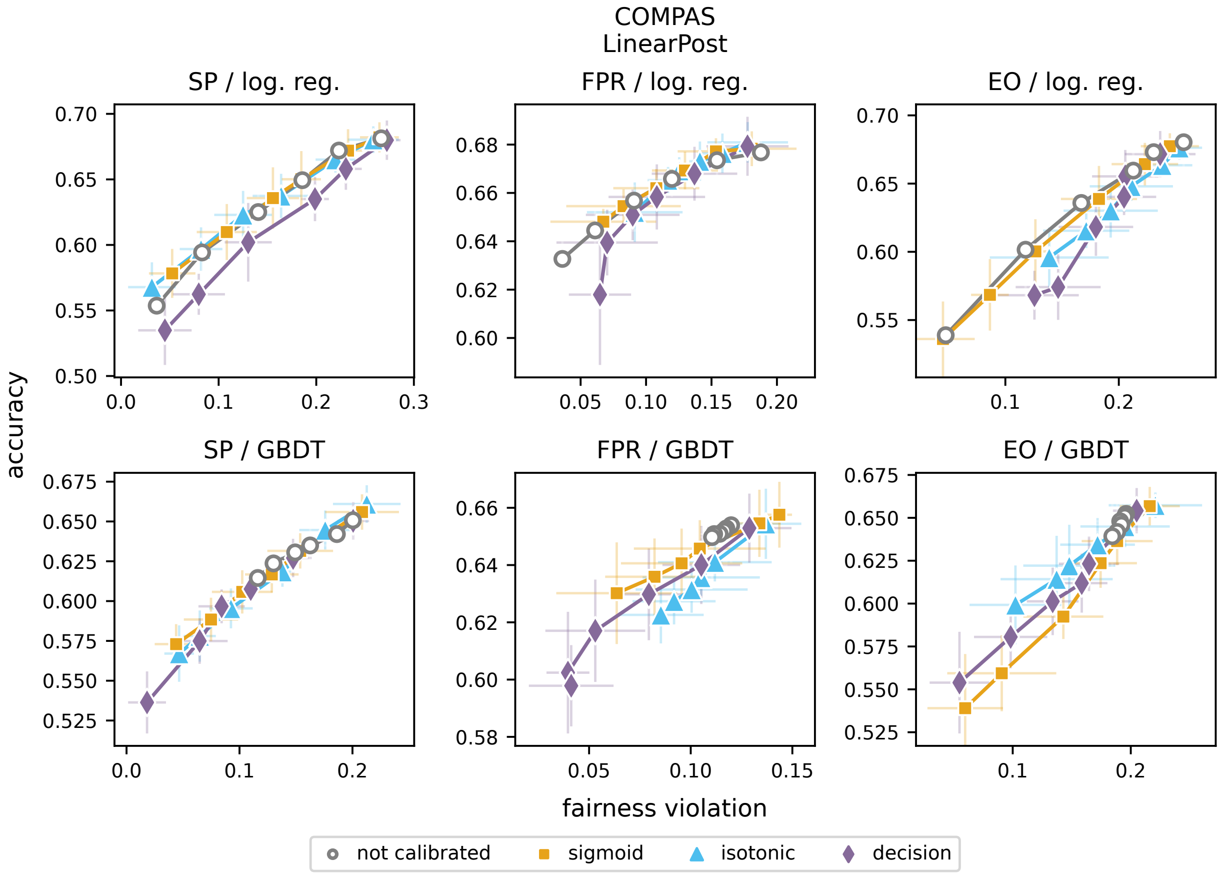

Figure 4.11: Accuracy-fairness tradeoffs on COMPAS for Section 4.5.4 calibration experiments.

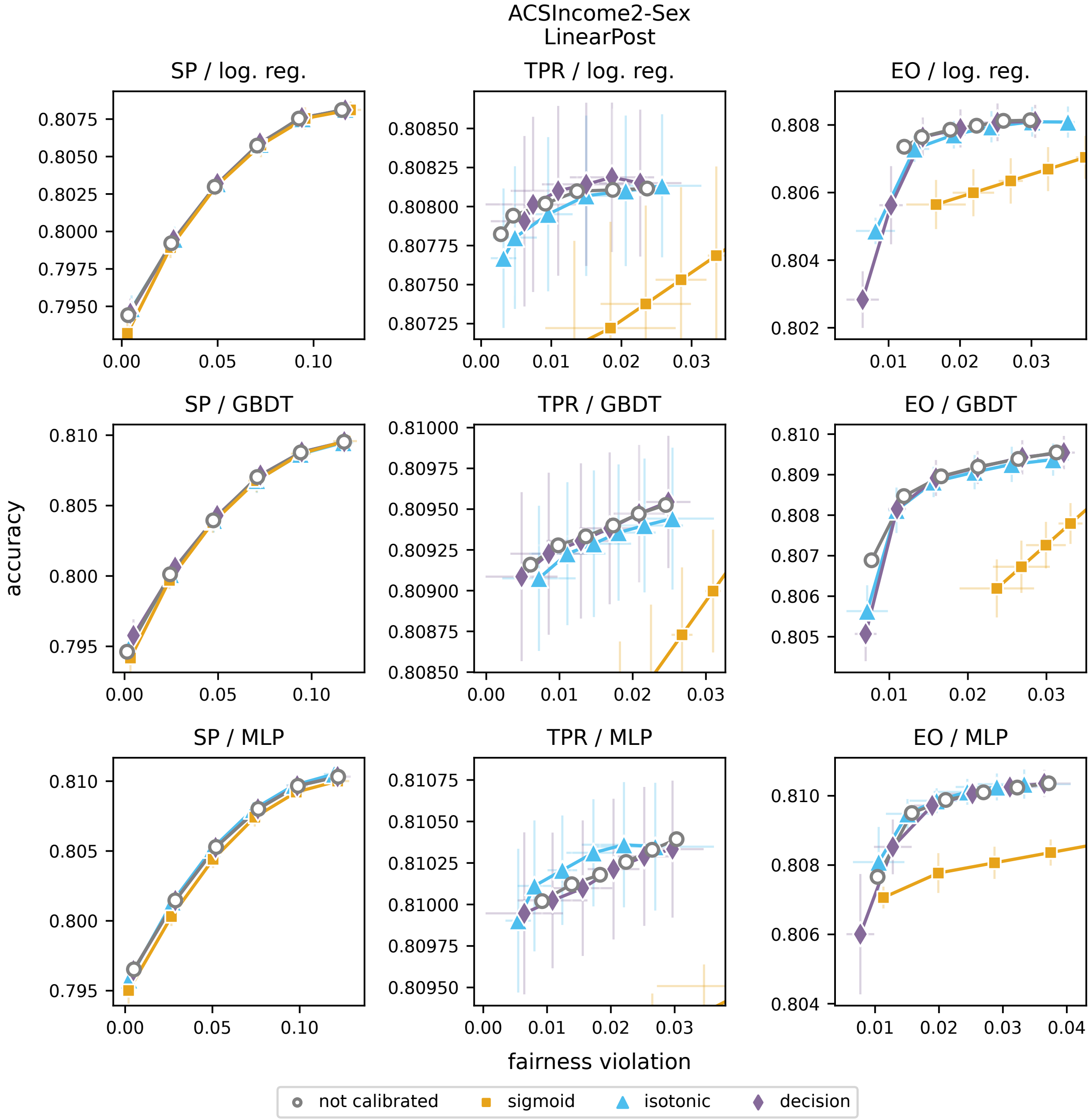

Figure 4.12: Accuracy-fairness tradeoffs on ACSIncome2-Sex for Section 4.5.4 calibration experiments.

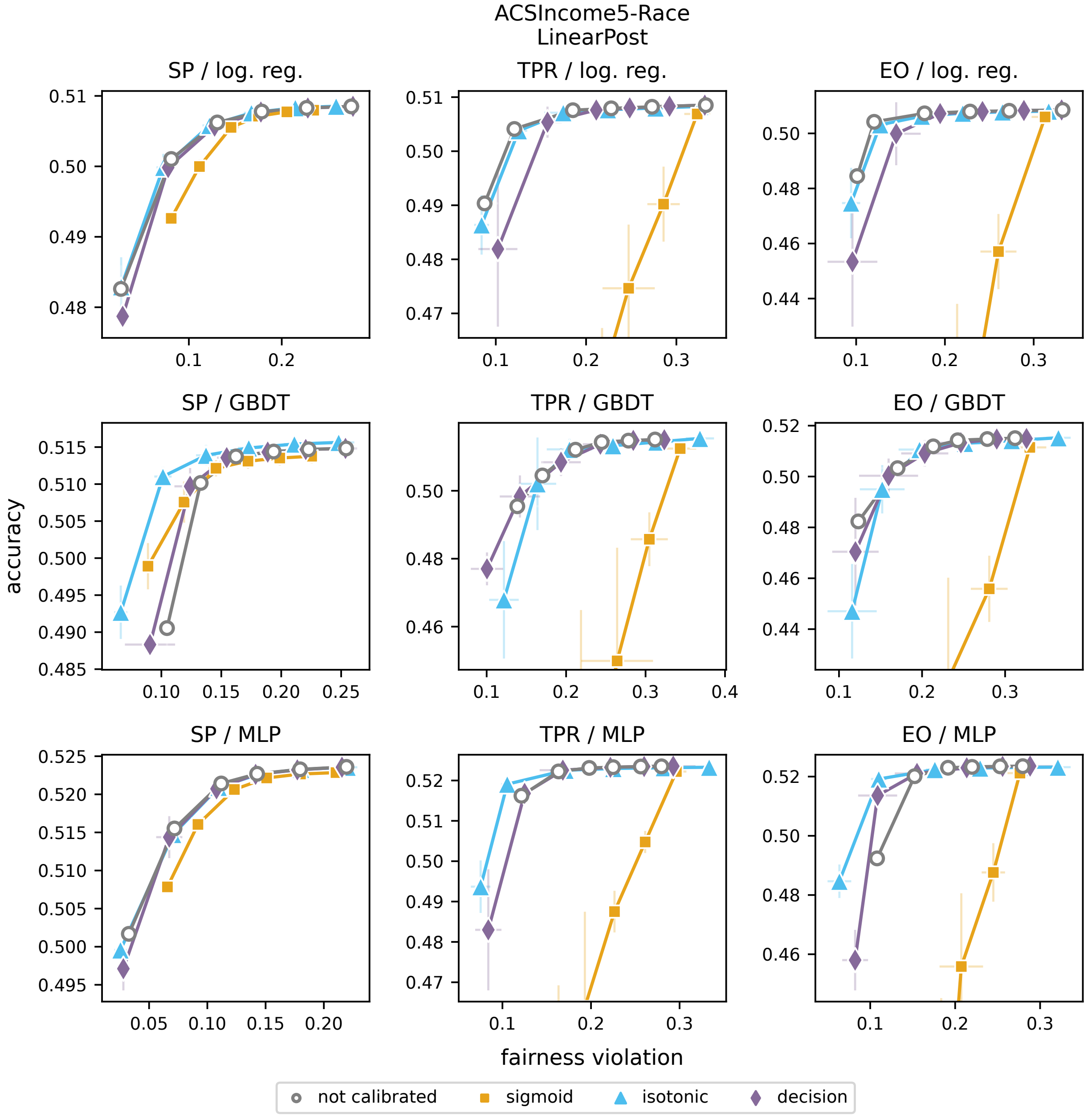

Figure 4.13: Accuracy-fairness tradeoffs on ACSIncome5-Race for Section 4.5.4 calibration experiments.

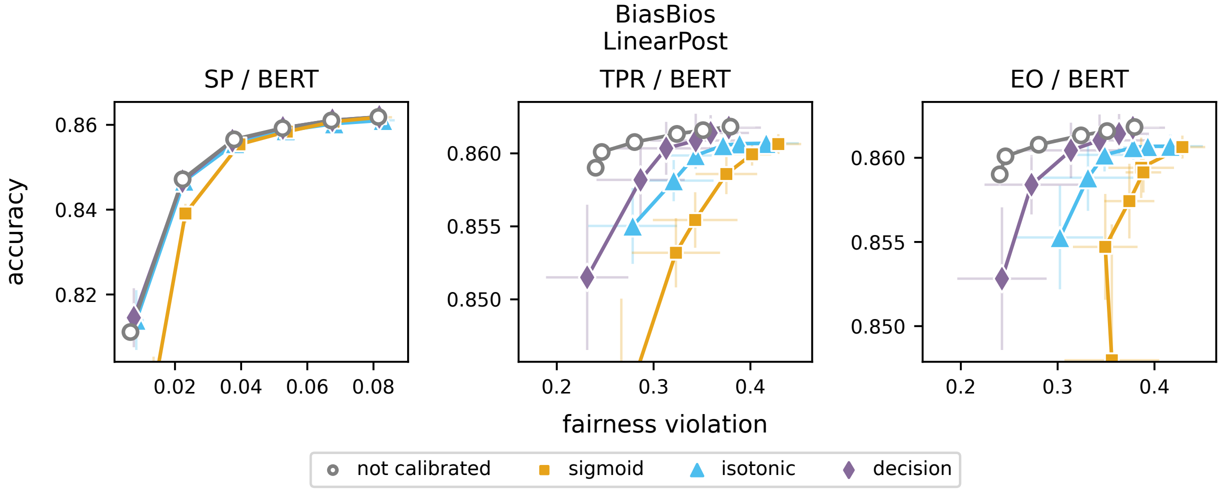

Figure 4.14: Accuracy-fairness tradeoffs on BiasBios for Section 4.5.4 calibration experiments.

4.5.6 Additional Results for Sections 4.5.1–4.5.3

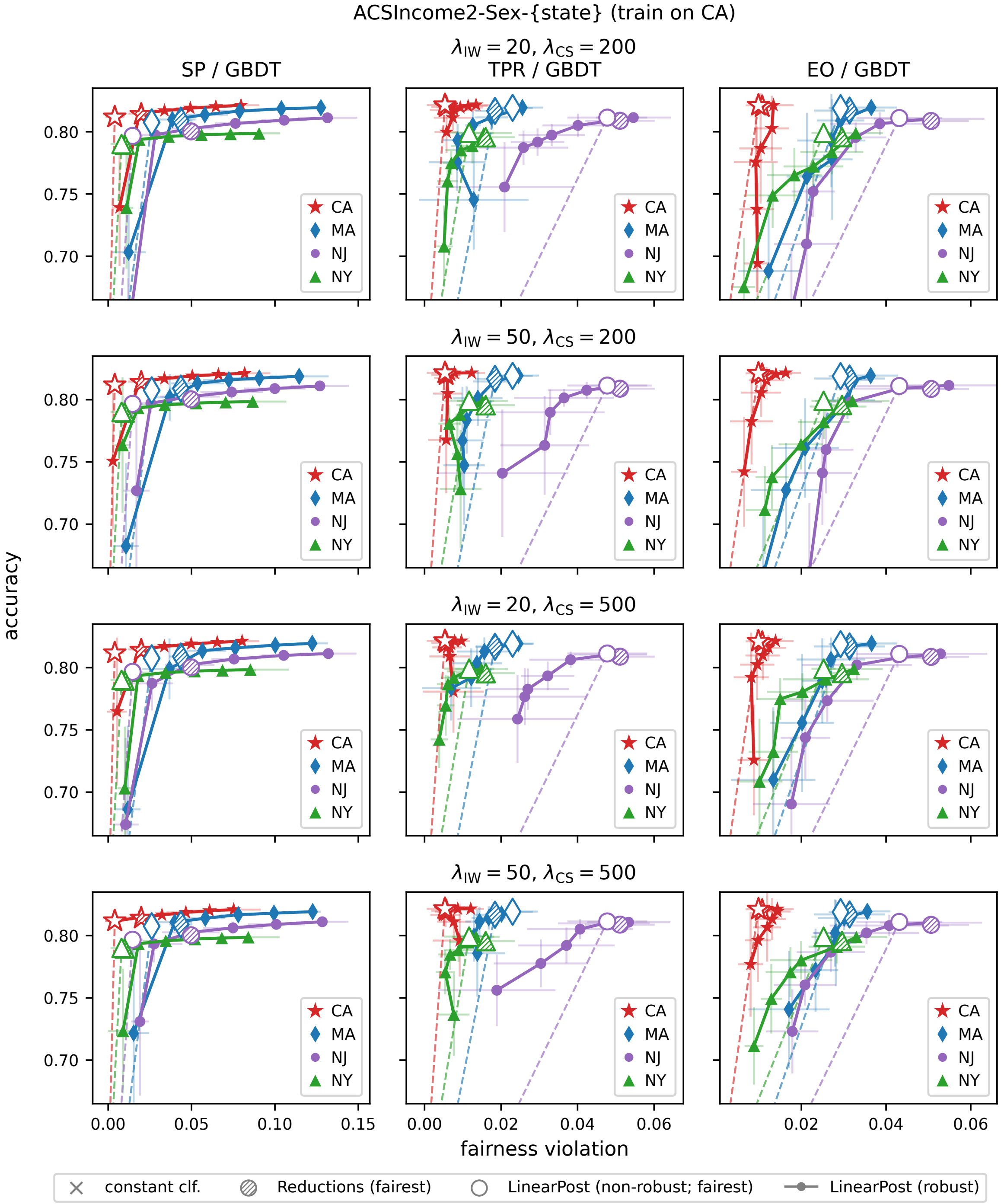

Figure 4.15: Accuracy-fairness tradeoffs on CA, MA, NJ, and NY for fair classifiers trained on CA using robust LinearPost under various λ_{IW} and λ_{CS} settings, from the geographic shift experiments in Section 4.5.1.

Table 4.1: Macro-average accuracy and fairness violation of fair classifiers trained on CA using REDUCTIONS, non-robust and robust LINEARPOST, under tolerance settings chosen to minimize the average violation on the validation set, from the geographic shift experiments in Section 4.5.1. For LINEARPOST, the tolerance is reported as a percentage of α within $[0.001, \alpha_{\max}]$, where $\alpha_{\max}$ is the violation on CA without post-processing.

Algorithm	Accuracy (avg.)	Violation (avg.)	Selected Tol.
Statistical Parity			
REDUCTIONS	0.7740 ± 0.0006	0.0240 ± 0.0012	2
LINEARPOST (non-robust)	0.7721 ± 0.0006	0.0145 ± 0.0013	0.0667
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 200$)	0.6715 ± 0.0015	0.0089 ± 0.0014	0.0667
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 200$)	0.6748 ± 0.0031	0.0104 ± 0.0013	0
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 500$)	0.6777 ± 0.0017	0.0085 ± 0.0012	0
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 500$)	0.7068 ± 0.0038	0.0130 ± 0.0020	0
Constant 0 classifier	0.6315 ± 0.0006	0	-
Parity of True Positive Rate			
REDUCTIONS	0.7819 ± 0.0006	0.0163 ± 0.0016	5
LINEARPOST (non-robust)	0.7856 ± 0.0006	0.0160 ± 0.0015	0
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 200$)	0.6994 ± 0.0035	0.0104 ± 0.0014	0.5333
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 200$)	0.7047 ± 0.0040	0.0114 ± 0.0013	0.4667
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 500$)	0.6895 ± 0.0018	0.0107 ± 0.0014	0.2
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 500$)	0.7422 ± 0.0056	0.0134 ± 0.0016	0.3333
Constant 0 classifier	0.6315 ± 0.0006	0	-
Equalized Odds			
REDUCTIONS	0.7819 ± 0.0006	0.0295 ± 0.0015	0.2
LINEARPOST (non-robust)	0.7853 ± 0.0006	0.0267 ± 0.0013	0
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 200$)	0.6531 ± 0.0015	0.0168 ± 0.0016	0
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 200$)	0.6655 ± 0.0021	0.0187 ± 0.0018	0
(robust; $\lambda_{\text{IW}} = 20$, $\lambda_{\text{CS}} = 500$)	0.7465 ± 0.0036	0.0223 ± 0.0019	0.4
(robust; $\lambda_{\text{IW}} = 50$, $\lambda_{\text{CS}} = 500$)	0.7347 ± 0.0045	0.0212 ± 0.0017	0.2
Constant 0 classifier	0.6315 ± 0.0006	0	-

Table 4.2: Tolerance settings of each fair algorithm for the group label noise experiments in Fig. 4.6, chosen to minimize the average violation on the validation set. For LinearPost, the tolerance is reported as a percentage of α within $[0.001, \alpha_{\max}]$, where $\alpha_{\max}$ is the violation on FL without post-processing.

Algorithm	Selected Tol.
Statistical Parity	
Reductions	0.2
LinearPost (non-robust)	0
(robust; $\lambda_{\text{CS}} = 50$)	0
(robust; $\lambda_{\text{CS}} = 200$)	0
(robust; $\lambda_{\text{CS}} = 500$)	0
Parity of True Positive Rate	
Reductions	0.2
LinearPost (non-robust)	0
(robust; $\lambda_{\text{CS}} = 50$)	0.1333
(robust; $\lambda_{\text{CS}} = 200$)	0.0667
(robust; $\lambda_{\text{CS}} = 500$)	0
Equalized Odds	
Reductions	0.002
LinearPost (non-robust)	0
(robust; $\lambda_{\text{CS}} = 50$)	0.0667
(robust; $\lambda_{\text{CS}} = 200$)	0
(robust; $\lambda_{\text{CS}} = 500$)	0

Table 4.3: Tolerance settings of each fair algorithm for the worst-case covariate shift experiments in Fig. 4.8, chosen to minimize the maximum violation on the validation set. For LinearPost, the tolerance is reported as a percentage of α within $[0.001, \alpha_{\max}]$, where $\alpha_{\max}$ is the violation on FL without post-processing.

Algorithm	Selected Tol.
Statistical Parity	
Reductions	0.5
LinearPost (non-robust)	0
(robust; $\lambda_{\text{IW}} = 5$)	0
(robust; $\lambda_{\text{IW}} = 20$)	0
(robust; $\lambda_{\text{IW}} = 50$)	0
Parity of True Positive Rate	
Reductions	0.2
LinearPost (non-robust)	0
(robust; $\lambda_{\text{IW}} = 5$)	0.8
(robust; $\lambda_{\text{IW}} = 20$)	0.4
(robust; $\lambda_{\text{IW}} = 50$)	0.1333
Equalized Odds	
Reductions	10
LinearPost (non-robust)	0
(robust; $\lambda_{\text{IW}} = 5$)	0
(robust; $\lambda_{\text{IW}} = 20$)	0.3333
(robust; $\lambda_{\text{IW}} = 50$)	0.1333

CHAPTER 5: PRIVACY

Fair learning algorithms require accurate knowledge of group memberships Z to achieve group fairness without sacrificing predictive performance—either in the form of labeled pairs (X, Z) drawn from the underlying distribution or, for LinearPost and other post-processing algorithms, via a group predictor $g = p[Z \mid X]$. Inaccurate group information may induce a shift between the training and test distribution, which, as analyzed in Section 4.2, can degrade both fairness and accuracy. Estimating the distribution of group memberships accurately therefore requires data from the underlying population. When such data are collected from customers or end users, they often contain highly sensitive personal information, especially in high-stakes settings; this makes privacy-preserving data handling and training procedures necessary to prevent leakage and ensure compliance with regulations.

To this end, *differential privacy* (DP) [28] offers a principled framework for protecting against data leakage and adversarial extraction. It injects randomness into the learning procedure to dilute the influence of any single training example (potentially linked to an individual) on the final model, thereby information-theoretically limiting what an adversary can infer about that individual, even if the they know the entire dataset except for the target record.

In this section, we incorporate differential privacy into the LinearPost framework to guarantee privacy with respect to the post-processing data.[1] Recall that LinearPost enforces fairness by learning a transformation on the outputs of the risk and group predictors r and g—specifically, it depends on the post-processing data only through the empirical distribution of their outputs, $(r(X), g(X))$, used in the empirical linear program $\widehat{\text{LP1}}$ to estimate the parameters of the transformation. Leveraging this observation and the post-processing theorem of differential privacy (Theorem 5.2), we can ensure privacy simply by replacing this empirical distribution with a differentially private estimate. The resulting classifier, formed by composing (r, g) with the learned transformation, inherits the same privacy guarantee as the estimate.

We instantiate this strategy in Section 5.4—with a slight departure from the thesis's main focus on classification—for learning attribute-aware regression models under statistical parity, where the distribution is estimated privately using Laplace histograms [42, 43]. We analyze the cost of privacy to model performance and fairness, both theoretically and empirically.

[1] To ensure privacy for the pre-training data used to learn r and g, one can replace the learning algorithm in the pre-training stage with any off-the-shelf private algorithm, as reviewed in Section 5.1. Because of this decoupling, our focus in this chapter is solely on the post-processing stage.

5.1 RELATED WORK

While there has been significant effort at addressing fairness and privacy, few treats them in combination, that is, designing algorithms that train fair predictors in a privacy-preserving manner. A core difficulty is that differential privacy and fairness may not be compatible: Cummings et al. [29] and Agarwal [30] showed that fairness and privacy are incompatible in the sense that no ε-DP algorithm can generally guarantee group fairness on the training set (from which population-level guarantees are derived via generalization), unless the hypothesis class is restricted to constant predictors. The argument is that a predictor f that is fair on a dataset S may not be fair on its neighbor S', so an ε-DP algorithm that outputs f on S with nonzero probability may also output f on S', which is unfair.

Work on private fair algorithms circumvent this incompatibility by relaxing to high probability guarantees for fairness. Jagielski et al. [143] extend the post-processing algorithm of Hardt et al. [18] to satisfy differential privacy by adding noise to the empirical output distribution of the pre-trained label predictor—same as the strategy we adopt—and also propose a private variant of the Reductions in-processing algorithm. Xu et al. [144] present a more refined private in-processing method by limiting to logistic regression models. Mozannar et al. [145] consider settings where sensitive attributes cannot be collected directly, and propose a scheme in which individuals report their sensitive attributes under local DP [146]. Tran et al. [147] incorporate group fairness constraints into a DP training loop via a Lagrangian-dual formulation, drawing on techniques from DP-SGD [148, 149, 150].

5.2 BACKGROUND ON DIFFERENTIAL PRIVACY

Given a training set S of N examples, a private algorithm learns a model for the task while minimizing the influence and exposure of any single data entry (potentially associated with an individual), thereby preventing data leakage through the learned model. We adopt the notion of *differential privacy* [151], which achieves privacy by introducing randomness into the algorithm and providing an information-theoretic bound on how much an adversary can infer about any particular unknown entry in S from the algorithm's output:

Definition 5.1 (Differential Privacy)**.** A randomized algorithm $\mathscr{A}$ is ε-differentially private if for all pairs of nonempty neighboring datasets S, S',

$$\mathbb{P}[\mathscr{A}(S) \in O] \leq e^{\varepsilon}\, \mathbb{P}[\mathscr{A}(S') \in O], \quad \forall O \subseteq \operatorname{range}(\mathscr{A}), \tag{5.1}$$

where $\mathbb{P}$ is taken with respect to the randomness of $\mathscr{A}$.

Two datasets S, S' are said to be neighboring if they differ in one entry by substitution (our result also covers insertion and deletion operations, which may have lower sensitivity; Remark A.1). We guarantee privacy with respect to all columns of S (in our case, each entry consists of the input feature X, sensitive attribute A, and class label Y).

The strategy we propose for private post-processing involves estimating and learning a distribution in a differentially private manner. In the simplest case, if the distribution has finite support (or is made finite via truncating and binning a continuous domain), it can be estimated from samples by counting the number of occurrences at each support point, that is, by computing its histogram. To ensure privacy, we use the *Laplace mechanism* [151], which adds independent Laplace noise to each histogram count. The magnitude of the noise is determined by the L^1-sensitivity of the histogram (Remark A.1):[2]

Theorem 5.1 (Laplace Histogram [42, 43, 44])**.** Let $\varepsilon > 0$, and $S = \{x^{(j)}\}_{j=1}^N$ be a dataset of size N, where $x^{(j)} \in [L]$. The histogram of S is

$$H_l = \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}[x^{(j)} = l] \quad \forall l \in [L], \tag{5.2}$$

and the Laplace histogram, LaplaceHistogram(S, ε), is

$$\widetilde{H}_l = H_l + \mathrm{Laplace}(2/\varepsilon N) \quad \forall l \in [L], \tag{5.3}$$

where the Laplace noise is sampled independently at each l. Then $\widetilde{H}$ satisfies ε-DP with respect to S, and with probability at least $1 - \delta$, the uniform distance between $\widetilde{H}$ and H is

$$|H_l - \widetilde{H}_l| \leq \frac{2}{\varepsilon N} \ln \frac{L}{\delta} \quad \forall l \in [L]. \tag{5.4}$$

Note that the noisy private estimate $\widetilde{H}$ may not be a valid histogram and needs to be normalized, for example, by projecting onto Δ^L, or via its cumulative distribution function by clipping and applying isotonic regression (Algorithm 5.2); the latter offering better statistical properties under the Kolmogorov-Smirnov metric (Definition 5.3 and Theorem 5.4). This approach follows the *centralized* DP model [151], in which a trusted curator (a platform or institution) collects raw data from users, performs DP-sanitized computations, and releases privatized outputs (in our case, the privately estimated distribution). In contrast, under *local* DP [146], each user applies a privacy mechanism to their own record before sharing it,

[2] The Laplace mechanism can be replaced by, for example, the Gaussian mechanism, which relaxes the pure $(\varepsilon, 0)$-DP guarantee to an approximate (ε, δ)-DP guarantee [28].

so the curator never observes raw data.

Our strategy also relies on the *post-processing theorem* of differential privacy:

Theorem 5.2 (Post-Processing [28, Proposition 2.1])**.** Let $\mathscr{A}$ be a randomized algorithm that is ε-differentially private. Let f be an arbitrary (randomized) mapping. Then $f \circ \mathscr{A}$ is ε-differentially private.

Variants of DP. Finally, several relaxed or more refined variants of differential privacy have been proposed in the literature, including (ε, δ)-DP [151] and Rényi DP [152]. These relaxations provide additional flexibility in balancing privacy and utility, and are often more convenient for analysis, privacy budgeting, and accounting, particularly when developing private versions of complex algorithms.

Private Learning Algorithms. For regression and classification, there are private variants of logistic regression [153], decision trees [154], and linear regression [155, 156, 157]. For solving convex and general optimization problems, private algorithms including those based on objective perturbation [158, 159] and DP-SGD [148, 149, 150] are proposed. The problem of private distribution learning has also been studied for parameterized families [160, 161], in addition to finite-support distributions as demonstrated in Theorem 5.1.

5.3 POST-PROCESSING ON PRIVATELY ESTIMATED DISTRIBUTIONS

To learn fair predictors in a differentially private manner within our LinearPost framework, we continue to follow the "pre-train then post-process" procedure introduced in Section 3.3. We assume that the pre-training stage has already been done (for example, using one of the private learning algorithms described in Section 5.2 above) to produce the pointwise risk predictor $r : \mathcal{X} \to \mathbb{R}^K_{\geq 0}$ and the group membership predictor $g : \mathcal{X} \to [0,1]^G$. And our focus here is on modifying the post-processing stage so that, given these pre-trained predictors, we perform the post-processing step in a differentially private manner to obtain the final fair predictor.

We divide the post-processing stage into two substeps. First, we privately estimate the joint output distribution of r and g from the samples. Then, we use this private (but noisy) estimate to perform post-processing as usual, optionally incorporating the robust procedure introduced in Section 4.4 to mitigate fairness degradation caused by the noise.

Private Distribution Learning. Recall from Section 3.3 that the parameters of the post-processing transformation are obtained from the optimal dual variables of an empirical linear program $\widehat{\mathrm{LP1}}$ that formulates the fair classification problem, defined on the outputs of the pointwise risk and group predictors evaluated on the unlabeled samples, $\{(r(x^{(j)}), g(x^{(j)}))\}_j$. In other words, $\widehat{\mathrm{LP1}}$ is constructed using the pushforward of the empirical input distribution $\widehat{\mathbb{P}}_X$ by $[r, g]$, denoted $\widehat{\mathbb{P}}_{r,g} = [r, g]\sharp\widehat{\mathbb{P}}_X$. This means that the resulting post-processing will only depend on the samples—over which privacy is to be guaranteed—through $\widehat{\mathbb{P}}_{r,g}$.

Leveraging this observation and the post-processing theorem of differential privacy, a simple strategy for achieving privacy is to replace $\widehat{\mathbb{P}}_{r,g}$ with a private estimate $\widetilde{\mathbb{P}}_{r,g}$. By Theorem 5.2, any subsequent computation based on $\widetilde{\mathbb{P}}_{r,g}$ will inherit the same DP guarantees satisfied by the private estimate, including the post-processing transformation obtained by solving $\widehat{\mathrm{LP1}}$.

We can obtain the private estimate $\widetilde{\mathbb{P}}_{r,g}$ using any private distribution learning/estimation algorithm mentioned in Section 5.2. Here, we adopt the histogramming approach (Theorem 5.1): the $(K + G)$-dimensional output space $[[0, \|\ell\|_\infty]^K, [0, 1]^G]$ of $[r, g]$ is partitioned into L cells, each represented by a centroid with radius at most Ra; the outputs of $[r, g]$ are then rounded to the nearest centroid, and the probability mass function (PMF) of the binned outputs is estimated using Laplace histograms, followed by normalization. The resulting $\widetilde{\mathbb{P}}_{r,g}$ is then used in place of the original empirical distribution $\widehat{\mathbb{P}}_{r,g}$ for post-processing.

(Robust) Post-Processing. After obtaining the private estimate $\widetilde{\mathbb{P}}_{r,g}$ of the output distribution of $[r, g]$ from the samples $\{x^{(j)}\}_j$, the next step is to construct and solve the empirical linear program $\widehat{\mathrm{LP1}}$ using $\widetilde{\mathbb{P}}_{r,g}$, and to derive the post-processing parameters from its optimal dual variables, as in Algorithm 3.1.

Because $\widetilde{\mathbb{P}}_{r,g}$ is a noisy estimate of the true distribution $\mathbb{P}_{r,g}$, the resulting post-processed predictor may not be optimal (even if the r, g predictors being post-processed are Bayes-optimal) nor necessarily satisfy the desired fairness level exactly. The excess risk and fairness violation caused by privacy noise can be quantified using the bounds in Theorems 4.1 and 4.2, with the covariate shift corresponding to the distance between $\widetilde{\mathbb{P}}_{r,g}$ and $\mathbb{P}_{r,g}$ (for example, under the histogramming approach described above, Theorem 5.1 implies that $D_{\mathrm{TV}}(\widetilde{\mathbb{P}}_{r,g}, \mathbb{P}_{r,g}) \le \widetilde{O}(L/\varepsilon N)$ with high probability); and if a transformation on $[r, g]$ is applied prior to private distribution estimation (such as the binning operation in histogramming), it corresponds to a concept shift (of at most Ra). Concept shift may also arise if r and g themselves are learned with a private algorithm and thus deviate from Bayes-optimal.

To mitigate fairness degradation caused by the noise in $\widetilde{\mathbb{P}}_{r,g}$, we can apply robust post-processing (Section 4.4). The uncertainty set can be tailored to the DP mechanism used: for example, the Laplace mechanism [151] adds independent Laplace(B/ε) noise to model parameters, so the true parameters w lie within $\widetilde{O}(B/\varepsilon)$ of the private values $\tilde{w}$ with high probability; accordingly, the uncertainty set can be generated from models whose parameters vary within $\tilde{w} \pm \widetilde{O}(B/\varepsilon)$. When mechanism-specific analysis is infeasible or overly complex, the perturbations can instead be modeled agnostically as bounded (adversarial) shifts.

5.4 EXAMPLE: PRIVATE REGRESSION UNDER STATISTICAL PARITY

In this section, we present an example illustrating the private post-processing strategy introduced in Section 5.3. With a slight departure from the main focus of this thesis, we consider the problem of learning fair *regression* models, rather than *classifiers*, under (approximate) statistical parity, assuming attribute awareness.

Following the organization of Chapter 3, Section 5.4.2 begins with an analysis of the structure of the optimal fair predictor as a post-processing of the Bayes-optimal unconstrained predictor (Eq. (5.7)). This characterization naturally suggests a "pre-train then post-process" procedure for learning fair predictors. In Section 5.4.3, we modify the post-processing stage to satisfy differential privacy using the private distribution estimation strategy described earlier. Finally, in Sections 5.4.4 and 5.4.5, we analyze the excess risk and fairness violation of the resulting predictor, both theoretically and empirically, with particular emphasis on the effects of the privacy noise.

5.4.1 Problem Setup

We consider a (univariate) regression problem defined by a joint distribution over input features $X \in \mathcal{X}$, target responses $Y \in \mathbb{R}$, and sensitive attributes $A \in [M]$. The goal is to learn an attribute-aware predictor $f : \mathcal{X} \times [M] \to \mathbb{R}$ (Definition 2.2) that satisfies (approximate) statistical parity with tolerance α (Definition 5.2), while minimizing the risk:

$$\min_{f:\mathcal{X}\times[M]\to\mathbb{R}} R(f) \quad \text{subject to} \quad V^{\mathrm{SP}}(f) \leq \alpha, \tag{5.5}$$

where the risk of a predictor $\widehat{Y} = f(X, A)$ is measured by the mean squared error (MSE) relative to the true response Y,

$$R(f) = \mathbb{E}[(Y - \widehat{Y})^2]. \tag{5.6}$$

The Bayes-optimal unconstrained predictor is given by the conditional mean of the response given the inputs,

$$f^*(x, a) = \mathbb{E}[Y \mid X = x, A = a]. \tag{5.7}$$

The group fairness notion of statistical parity introduced in Definition 2.3 for the classification setting can be directly extended to regression by requiring the output distribution of the predictor to be identical across all sensitive groups. Formally, this means $(f\sharp(X, A) \mid A = a) = (f\sharp(X, A) \mid A = a')$ for all $a, a' \in [M]$. An approximate relaxation of statistical parity, analogous to Eq. (2.12), can be defined by measuring the fairness violation as the divergence D between the conditional output distributions:

Definition 5.2 (Approximate Statistical Parity)**.** Let D be a divergence, a model f satisfies statistical parity α-*approximately* if

$$\alpha \geq V^{\mathrm{SP}}(f) = \max_{a,a' \in [M]} D(\widehat{Y} \mid A = a, \widehat{Y} \mid A = a'), \tag{5.8}$$

where the random variable $\widehat{Y}$ denotes the output of the model f.

The approximate definition of SP in Eq. (2.12) for classifiers is recovered from Definition 5.2 by setting D to the total variation distance. To evaluate the fairness of univariate regression models, we set D to the Kolmogorov-Smirnov (KS) distance, which corresponds to the uniform distance between the cumulative distribution functions (CDFs):

Definition 5.3 (Kolmogorov-Smirnov Distance)**.** The Kolmogorov-Smirnov distance between distributions μ, ν over $\mathbb{R}$ is defined as

$$D_{\mathrm{KS}}(\mu, \nu) = \sup_{t \in \mathbb{R}} \left| \int_{-\infty}^{t} (\mu(x) - \nu(x)) \, \mathrm{d}x \right|. \tag{5.9}$$

Last but not least, our analysis makes use of the Wasserstein-2 distance and the associated concept of optimal transport [126].

Definition 5.4 (Wasserstein Distance)**.** Let $\mathcal{X}$ be a metric space with distance d, and $p \in [1, \infty]$. The Wasserstein-p distance between distributions μ, ν over $\mathcal{X}$ is defined as

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x, x')^p \, \mathrm{d}\gamma(x, x') \right)^{1/p}; \tag{5.10}$$

$\Gamma(\mu, \nu)$ denotes the collection of couplings of μ, ν, where each γ is a joint distribution over $\mathcal{X} \times \mathcal{X}$ satisfying $\mu(x) = \int_{\mathcal{X}} \gamma(x, x') \, \mathrm{d}x'$ and $\nu(x') = \int_{\mathcal{X}} \gamma(x, x') \, \mathrm{d}x$, for all x, x'.

In particular, the coupling γ^* that achieves the infimum in Definition 5.4 encodes the *randomized optimal transport* between μ and ν. That is, a (randomized) mapping $T^* : \mathcal{X} \to \mathcal{X}$ that moves points in $\mathcal{X}$ such that applying it to $x \sim \mu$ yields an output distributed as

ν (hence called a *transport*), $T^*\sharp\mu = \nu$, while minimizing the total movement cost (hence *optimal*), $\int_{\mathcal{X}} \mathbb{E}[d(x, T^*(x))^p] \,\mathrm{d}\mu(x) = W_p^p(\mu, \nu)$, where the cost of moving x to x' is $d(x, x')^p$. The mapping T^* can be derived from γ^* via its Markov kernel π_{T^*} (defined in Section 2.1):

$$\pi_{T^*}(x, x') \propto \gamma^*(x, x') \quad \forall x, x' \in \mathcal{X}. \tag{5.11}$$

5.4.2 Fair Regression via Post-Processing

Similar to the optimal fair classifier analyzed in Theorem 3.1, the optimal fair predictor can also be expressed as a post-processing of the Bayes-optimal predictor f^*:

Theorem 5.3. Let a regression problem and $\alpha \in [0, 1]$ be given, and denote $p_a = \mathbb{P}[A = a]$. Let f^* be the (unconstrained) Bayes-optimal predictor from Eq. (5.7), and let its output distribution for group a be $r_a^* = f^*\sharp(X \mid A = a, a)$ for all $a \in [M]$. Let[3]

$$(\bar{r}_1, \ldots, \bar{r}_M) \in \operatorname*{arg\,min}_{r_1,\ldots,r_M} \left\{ \sum_{a\in[M]} p_a W_2^2(r_a^*, r_a) : \operatorname{supp}(r_a) \subseteq \mathbb{R}, D_{\mathrm{KS}}(r_a, r_{a'}) \leq \alpha, \forall a, a' \in [M] \right\}. \tag{5.12}$$

Let T_a denote the (randomized) optimal transport from r_a^* to $\bar{r}_a$ under squared loss. Then an optimal solution to Eq. (5.5) is

$$\bar{f}(x, a) \mapsto T_a \circ f^*(x, a), \tag{5.13}$$

with risk

$$R(\bar{f}) = R(f^*) + \sum_{a\in[M]} p_a W_2^2(r_a^*, \bar{r}_a). \tag{5.14}$$

The minimizers of Eq. (5.12) define the target output distributions that satisfy α-approximate statistical parity, as enforced by the constraint $D_{\mathrm{KS}}(r_a, r_{a'}) \leq \alpha$. When $\alpha = 0$, all $\bar{r}_a$ coincide and equal the Wasserstein barycenter of the r_a^*'s.

This theorem shows that the optimal fair predictor can be obtained by post-processing the Bayes-optimal predictor via group-specific optimal transports that transform each group's original output distribution to the respective target distribution. The case $\alpha = 0$ recovers prior results [162, 163], characterizing the optimal fair predictor in terms of Wasserstein barycenters; the analysis here extends to the more general setting of approximate fairness.

[3] The underlying metric for the Wasserstein distance is $d(y, y') = |y - y'|$.

Algorithm 5.1: Fair regression via (differentially private) post-processing

```
Input: Regression model $f : \mathcal{X} \times [M] \to \mathbb{R}$, finite discretization $\mathcal{Y}$ of range($f$),
       fairness tolerance $\alpha \in [0, 1]$, privacy budget $\varepsilon \geq 0$, samples $\{(x^{(j)}, a^{(j)})\}_{j=1}^{N}$
Output: Randomized regression model $\mathcal{X} \times [M] \to [K]$
Define discretizer $h(y) = \arg\min_{k \in \mathcal{Y}} |y - k|$ ;   // break ties to smaller index
$\widetilde{S} \leftarrow \{(h \circ f(x^{(j)}, a^{(j)}), a^{(j)})\}_{j=1}^{N}$ ;   // discretized outputs
if $\varepsilon = \infty$, differential privacy is not requested then
    /* empirical PMF */
    Set $\widetilde{S}_a \leftarrow \{\tilde{y} : (\tilde{y}, m) \in \widetilde{S}, m = a\}$ for all $a \in [M]$ ;
    Define $\tilde{r}_a(k) = \frac{1}{|\widetilde{S}_a|} \sum_{\tilde{y} \in \widetilde{S}_a} \mathbb{1}[\tilde{y} = k]$, $\forall a \in [M]$, $k \in \mathcal{Y}$ ;
else
    /* private empirical PMF */
    $\tilde{r} \leftarrow$ LaplaceHistogram($\widetilde{S}, \varepsilon$) ;   // $\tilde{r}$ is supported on $\mathcal{Y} \times [M]$
    $\tilde{p}_a \leftarrow \max(0, \sum_k \tilde{r}(k, a))$ for all $a \in [M]$ ;   // group marginals
    Define $\check{r}_a(k) = \tilde{r}(k, a)/\tilde{p}_a$, $\forall a \in [M]$, $k \in \mathcal{Y}$ ;   // condition on group $a$
    Set $\tilde{r}_a \leftarrow$ NormalizeCumSum($\check{r}_a$) for all $a \in [M]$ ;
end
$(\gamma_1, \dots, \gamma_M) \leftarrow$ optimal values of LP3($\tilde{r}_1, \dots, \tilde{r}_M$) ;
Define randomized $T_a : \mathcal{Y} \to \mathcal{Y}$ with Markov kernel $\pi_{T_a}(k, l) \propto \gamma_a(k, l)$, $\forall a \in [M]$ ;
return $(x, a) \mapsto T_a \circ h \circ f(x, a)$ ;
```

Pre-Train then Post-Process. The decomposition in Theorem 5.3 again suggests a "pre-train then post-process" procedure for learning fair predictors, similar to Section 3.3.

In the pre-training stage, we train the optimal predictor f^* without any fairness constraints, aiming solely to maximize predictive accuracy. In the post-processing stage, we estimate the output distributions $\hat{r}_a$ of the learned (or given) predictor f for each group a, using samples $(x^{(j)}, a^{(j)})_{j=1}^{N}$ (for instance, we can simply use the empirical distribution). We then compute the target output distributions $\bar{r}_a$'s (or the Wasserstein barycenter when $\alpha = 0$) as in Eq. (5.12), followed by computing the group-specific optimal transports T_a's from $\hat{r}_a$ to $\bar{r}_a$. The resulting fair predictor is given by $(x, a) \mapsto T_a \circ f(x, a)$.

In practice, $\hat{r}_a$ can be set to the empirical output distribution of f on the samples, allowing the computation of the target distributions $\bar{r}_a$ and transports T_a to be combined into a single linear program. This joint formulation is equivalent, in the case $\alpha = 0$, to the *finite-support Wasserstein barycenter problem*, whose linear program formulation has size exponential in M, the number of groups (the case of $\alpha > 0$ is harder) [164, 165]. To reduce the complexity, we restrict the support of the target distributions via discretization.[4]

Algorithm 5.1 presents the post-processing algorithm for learning fair predictors using the

[4] This fixed-support approximation is common in prior work on Wasserstein barycenters [166, 167].

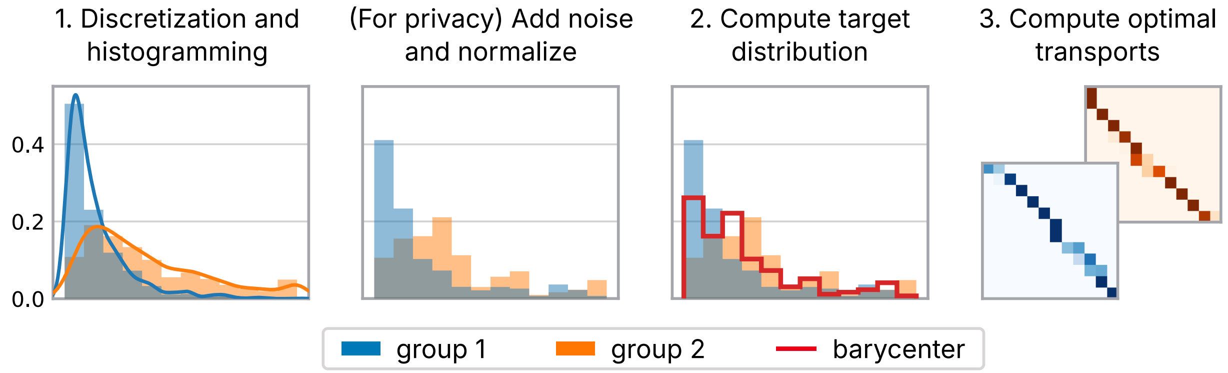

Figure 5.1: "Pre-train then (privately) post-process" procedure for learning fair predictors. The (randomized) transports to the barycenter are represented as (sparse) $k \times k$ matrices, and the value at the (i, j)-th entry is the probability of transporting to bin j given bin i.

discretization strategy, with an illustration in Fig. 5.1. Its inputs are N training examples with group labels, a pre-trained base predictor f, and a finite discretization $\mathcal{Y}$ over its output range. In our experiments, we set $\mathcal{Y}$ to be an evenly spaced partition of $[s, t]$, where s and t are bounds on predictor outputs determined using prior knowledge;[5] outputs outside this interval are clipped.[6]

Then, the algorithm first estimates the empirical PMF of the discretized output distribution $\tilde{r}_a$ for each group $a \in [M]$, then solves a linear program (LP3 below) for the post-processing problem with $(M|\mathcal{Y}|^2 + (1 + M)|\mathcal{Y}|)$ variables and $3M|\mathcal{Y}|$ constraints (in addition to nonnegativity constraints), and finally uses the resulting solution, which encodes the optimal transports, to post-process the discretized base predictor.

LP3:

$$\begin{aligned} \min_{\substack{\gamma_1,\dots,\gamma_M \geq 0 \\ q, r_1, \dots, r_M \geq 0}} \quad & \sum_{a \in [M]} \hat{p}_a \sum_{k,l \in \mathcal{Y}} (k - l)^2 \gamma_a(k, l) \\ \text{s.t.} \quad & \sum_{l \in \mathcal{Y}} \gamma_a(k, l) = \tilde{r}_a(k) && \forall a \in [M],\ k \in \mathcal{Y}, \\ & \sum_{k \in \mathcal{Y}} \gamma_a(k, l) = r_a(l) && \forall a \in [M],\ l \in \mathcal{Y}, \\ & \left| \sum_{k \leq l} (r_a(k) - q(l)) \right| \leq \frac{\alpha}{2} && \forall a \in [M],\ l \in \mathcal{Y}. \end{aligned} \tag{5.15}$$

[5] Alternatively, one could choose $[s, t]$ adaptively based on the minimum and maximum values of the predictor's outputs on the samples, but this must be done with care to avoid violating differential privacy.

[6] Clipping is necessary for pure $(\varepsilon, 0)$-DP, due to a lower bound by Hardt and Talwar [168] via a packing argument.

Algorithm 5.2: NormalizeCumSum: Normalize a PDF via L^∞ isotonic regression on its cumulative sums

```
Input: Unnormalized probability mass function $\check{f} : [L] \to \mathbb{R}$
Output: Probability mass function in $\Delta^L$
Define $\check{F}(j) = \sum_{k \le j} \check{f}(k), \forall j \in [L]$ ;   // cumulative sums
Define $(l_j, r_j) = \arg\max_{l \le j \le r} (\check{F}(l) - \check{F}(r)), \forall j \in [L]$ ;   // largest monotonicity violation at each position
Define $F(j) = \begin{cases} \mathrm{proj}_{[0,1]}((\check{F}(l_j) + \check{F}(r_j))/2) & \text{if } 1 \le j < L \\ 1 & \text{if } j = L \end{cases}$ ;   // $L^\infty$ iso. regr.
return $f(j) = \begin{cases} F(j) & \textit{if } j = 1 \\ F(j) - F(j-1) & \textit{if } 1 < j \le L \end{cases}$ ;
```

Here, $\gamma_a \in [0,1]^{\mathcal{Y} \times \mathcal{Y}}$ denotes the coupling between $\tilde{r}_a$ and the target distribution $r_a \in [0,1]^{\mathcal{Y}}$, and $q \in [0,1]^{\mathcal{Y}}$ is an auxiliary variable representing the barycenter of the $\tilde{r}_a$'s. The final constraint enforces $D_{\mathrm{KS}}(r_a, q) \le \alpha/2$ for all a, thereby satisfying the fairness constraints of $D_{\mathrm{KS}}(r_a, r_{a'}) \le \alpha$ for all a, a' (see also Example 2.2).

Alternative (Non-Private) Post-Processing Algorithms. In the exact fairness case ($\alpha = 0$), Chzhen et al. [162] proposed a non-parametric estimator for the Wasserstein barycenter and the corresponding optimal transports, which admit explicit expressions in terms of the samples (after a sorting operation) without any optimization steps. Hu et al. [169] use parameterized models (such as neural networks) to approximate both the barycenter and the transports T_a, and propose post-processors for approximate fairness by defining the post-processing transformation as an interpolation between T_a and the identity map, that is, interpolating between the exactly fair predictor post-processed using T_a and the unconstrained (unfair) predictor f. However, it remains unclear whether such interpolation is optimal under approximate fairness. Taturyan et al. [170] study approximately fair post-processing in the attribute-blind setting, with an algorithm that also involves discretizing the base predictor's outputs.

5.4.3 Private Post-Processing on Laplace Histogram

The predictor returned by the post-processing algorithm in Algorithm 5.1 depends on the samples $x^{(j)}, a^{(j)}{}_{j=1}^{N}$ only through the empirical PMFs of the output distribution of f computed on Line 5. By the post-processing theorem of differential privacy (Theorem 5.2), we can make Algorithm 5.1 differentially private simply by replacing these empirical PMFs with differentially private estimates.

To this end, we use LaplaceHistogram from Theorem 5.1 to privately estimate the PMFs. This procedure performs standard histogram computation, followed by adding Laplace noise to each bin to satisfy differential privacy. We perform an additional normalization step to make the result a valid PMF: the NormalizeCumSum method in Algorithm 5.2 applies L^∞ isotonic regression to the clipped cumulative sums of the noise-added histogram, enforcing that the cumulative sums are nondecreasing and end at 1, thus forming a valid CDF.

We emphasize that the predictor produced by Algorithm 5.1 with LaplaceHistogram satisfies ε-DP only with respect to the post-processing samples, not the data used to train the base predictor f. To achieve differential privacy in the pre-training stage, any standard private learning algorithm from Section 5.2 can be used.

We close with an analysis of the accuracy of the privately estimated PMFs:

Theorem 5.4. Let $r_a(k) = \mathbb{P}[h \circ f(X, A) = k \mid A = a]$, $\forall k \in \mathcal{Y}$, denote the true PMF of discretized predictor output, and $p_a = \mathbb{P}[A = a]$ the true group marginals, for all $a \in [M]$. Let $\tilde{r}_a$, $\tilde{p}_a$ denote their privately estimated counterparts from N samples on Lines 8 and 10 in Algorithm 5.1, respectively. Then for all $N \geq \Omega(\max_a \ln(1/\delta)/p_a)$, with probability at least $1 - \delta$, for all $a \in [M]$,

$$|p_a - \tilde{p}_a| \leq O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon N} \ln \frac{M|\mathcal{Y}|}{\delta}\right), \tag{5.16}$$

and

$$\|r_a - \tilde{r}_a\|_\infty \leq O\left(\sqrt{\frac{1}{p_a N} \ln \frac{M}{\delta}} + \frac{\sqrt{|\mathcal{Y}|}}{\varepsilon p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}\right), \tag{5.17}$$

$$\|r_a - \tilde{r}_a\|_1 \leq O\left(\sqrt{\frac{|\mathcal{Y}|}{p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}} + \frac{|\mathcal{Y}|}{\varepsilon p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}\right), \tag{5.18}$$

$$D_{\mathrm{KS}}(r_a, \tilde{r}_a) \leq O\left(\sqrt{\frac{|\mathcal{Y}|}{p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}} + \frac{\sqrt{|\mathcal{Y}|}}{\varepsilon p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}\right). \tag{5.19}$$

The $\widetilde{O}(\sqrt{|\mathcal{Y}|/N} + |\mathcal{Y}|/\varepsilon N)$ rate for the TV distance matches known results [43]. For the KS distance, the rate is improved by a factor of $\sqrt{|\mathcal{Y}|}$, which is expected since KS is a weaker metric than TV. A key technical point enabling this improvement is our choice to normalize via L^∞ isotonic regression on the cumulative sums (Algorithm 5.2), which aligns naturally with KS distance, as both operate on the CDF.

Private PMF estimation via Laplace histograms is well-studied [43, 44], but our implementation adds an extra normalization step after adding Laplace noise via NormalizeCumSum; this normalization complicates the analysis in Theorem 5.4, since the noise added to each bin can interact in nontrivial ways during normalization.

5.4.4 Analysis

We analyze the privately post-processed attribute-aware predictor returned by Algorithm 5.1, providing bounds on its excess risk and its violation of statistical parity (Definition 5.2, instantiated with the KS distance in Definition 5.3).

Theorem 5.5. Let a predictor $f : \mathcal{X} \to \mathbb{R}$ be given, along with N i.i.d. samples of (X, A). Let $\mathcal{Y}$ be an evenly-spaced partition of $[s, t]$ with $|\mathcal{Y}|$ bins, and assume $t - s \leq 1$ and $Y, f(X, A) \in [s, t]$ almost surely. Let f^* denote the (unconstrained) Bayes-optimal predictor (Eq. (5.7)), $\bar{f}^*$ the Bayes-optimal fair predictor (that is, the optimal solution to Eq. (5.5)), and $\bar{f}$ the predictor returned by Algorithm 5.1. Then, for all $N \geq \Omega(\max_a \ln(M/\delta)/p_a)$, with probability at least $1 - \delta$,

$$R(\bar{f}) - R(\bar{f}^*) \leq O\left(\sqrt{\frac{M|\mathcal{Y}|}{N} \ln \frac{M|\mathcal{Y}|}{\delta}} + \frac{M|\mathcal{Y}|}{\varepsilon N} \ln \frac{M|\mathcal{Y}|}{\delta}\right) + \frac{8}{|\mathcal{Y}|} + 5\,\mathbb{E}[|f(X, A) - f^*(X, A)|], \tag{5.20}$$

$$V^{\mathrm{SP}}(\bar{f}) \leq \alpha + \max_{a \in [M]} O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon p_a N} \ln \frac{M|\mathcal{Y}|}{\delta} + \sqrt{\frac{|\mathcal{Y}|}{p_a N} \ln \frac{M|\mathcal{Y}|}{\delta}}\right). \tag{5.21}$$

Each term in the excess risk bound, measuring the risk of $\bar{f}$ relative to the optimal fair predictor $\bar{f}^*$, corresponds to one of four sources of error: (1) finite-sample estimation, (2) noise added to satisfy ε-DP, (3) discretization, and (4) the L^1 excess risk of the predictor f being post-processed relative to the Bayes-optimal predictor f^*, which carries over any error from the pre-training stage.

On the Number of Bins. The first three terms in the excess risk bound (and the last two in the SP violation bound) are due to discretization and private PMF estimation (Theorem 5.4), and exhibit the standard statistical bias-variance tradeoff in the choice of the number of bins $|\mathcal{Y}|$:

$$\widetilde{O}\Bigg(\underbrace{\sqrt{\frac{|\mathcal{Y}|}{N}} + \frac{|\mathcal{Y}|}{\varepsilon N}}_{\text{variance}} + \underbrace{\frac{1}{|\mathcal{Y}|}}_{\text{bias}}\Bigg). \tag{5.22}$$

Using too few bins results in large discretization bias, while too many increase the variance from sampling and privacy noise. When $N \gtrsim \max_a |\mathcal{Y}|/(\varepsilon^2 p_a) \geq M|\mathcal{Y}|/\varepsilon^2$, the variance is dominated by the finite-sample term, and the optimal (squared-error-minimizing) bin count is $|\mathcal{Y}| = \widetilde{\Theta}(N^{1/3})$, consistent with classical histogram estimation results [171].

In contrast, the SP violation bound contains only variance terms and no bias term, so using fewer bins always improves fairness (at the expense of higher error). For example, in the extreme case $|\mathcal{Y}| = 1$, the post-processor outputs a constant function, achieving $\alpha = 0$.

In practice, the optimal $|\mathcal{Y}|$ is data-dependent and should be tuned jointly with α on a validation set. However, hyperparameter tuning can itself violate DP [172, 173], so it must be carried out with care. In our experiments, we sweep over all $(|\mathcal{Y}|, \alpha)$ combinations and plot the results to empirically examine the error-privacy-fairness tradeoffs, following standard practice in private ML research [174]. Although no tuning is performed, revealing these tradeoffs can still violate privacy, since they are computed from multiple post-processed models derived from the same training samples.

5.4.5 Experiment Setup

In this section, we evaluate the private fair post-processing algorithm described above.[7] We do not compare against other algorithms, as we are not aware of any existing private algorithms for learning fair predictors. While some existing approaches could in principle be adapted to this setting, for example, by replacing the optimization procedure in non-private fair regression algorithms with DP-SGD as similarly done in [147], making them both practical and competitive would require substantial care, so we leave such exploration to future work.

For simplicity, we perform post-processing directly on the *ground-truth regressox* f^{**}, namely, $f^{**}(x^{(j)}) = y^{(j)}$ for all j, where $y^{(j)}$ is the ground-truth response of the j-th example in the dataset. This setup isolates the effects of the private fair post-processing algorithm itself, allowing us to focus solely on its performance and the resulting tradeoffs.

Datasets. We evaluate on two datasets, each randomly split 70-30 into post-processing and testing sets, respectively. Unlike the setup in other experiments in this thesis (Appendix B), no model selection or hyperparameter tuning is performed here, so no validation set is used. Results are averaged over 50 random seeds, and also differing from Appendix B, the tradeoff curves are aggregated by averaging, for each fairness tolerance α, across the random seeds

[7] Code is available at `https://github.com/rxian/fair-regression`.

(whereas Appendix B performs aggregation for each percentage of fairness reduction achieved on the validation set).

- Communities & Crime [175]. This dataset contains socioeconomic and crime data for communities in the United States. The task is to predict the rate of violent crimes ($Y \in [0,1]$) per 100,000 population. The sensitive attribute is a binary indicator for whether the community has a significant minority population ($|\mathcal{A}| = 2$). The dataset contains 1,994 examples in total, with 679 training examples (after the split) in the smallest sensitive group.
- Law School [176]. This dataset contains academic performance records of law school applicants. The task is to predict a student's undergraduate GPA ($Y \in [1,4]$). The sensitive attribute is race ($|\mathcal{A}| = 4$). The dataset contains 21,983 examples in total, with 628 training examples in the smallest sensitive group.

5.4.6 Experiment Results

Figure 5.2 presents the tradeoffs between mean squared error (MSE) and violation of statistical parity (Definition 5.2; instantiated with the KS distance in Definition 5.3) achieved using our private post-processing algorithm, across different privacy budgets ε and discretization bin counts k.

1. The cost of discretization, reflected in the MSE of the discretized but unprocessed predictor (the top-left starting point of each tradeoff curve), is negligible compared to the error introduced by enforcing fairness; unless the unprocessed predictor is already close to exactly fair, which is not the case in our experiments.
2. Although the amount of data available for post-processing is small, necessitating large-scale Laplace noise for privacy, the results are largely insensitive to the privacy budget, with degradation only appearing at very strict privacy levels ($\varepsilon = 0.1$) or when using a very large number of bins. This insensitivity arises because, as shown in Theorem 5.5, the error from DP noise is dominated by estimation error when $N \gtrsim \max_a k/(\varepsilon^2 p_a)$. In our experiments, this condition holds ($N \gg kM/\varepsilon^2$) for both datasets except when $\varepsilon = 0.1$ or when $(\varepsilon, k) = (0.5, 180)$. Larger bin counts increase the cost of privacy due to variance, an effect we further illustrate in Fig. 5.3.
3. The rightmost endpoint of each curve corresponds to predictors post-processed under $\alpha = 0$ for exact fairness. But in some settings, particularly on the Law School

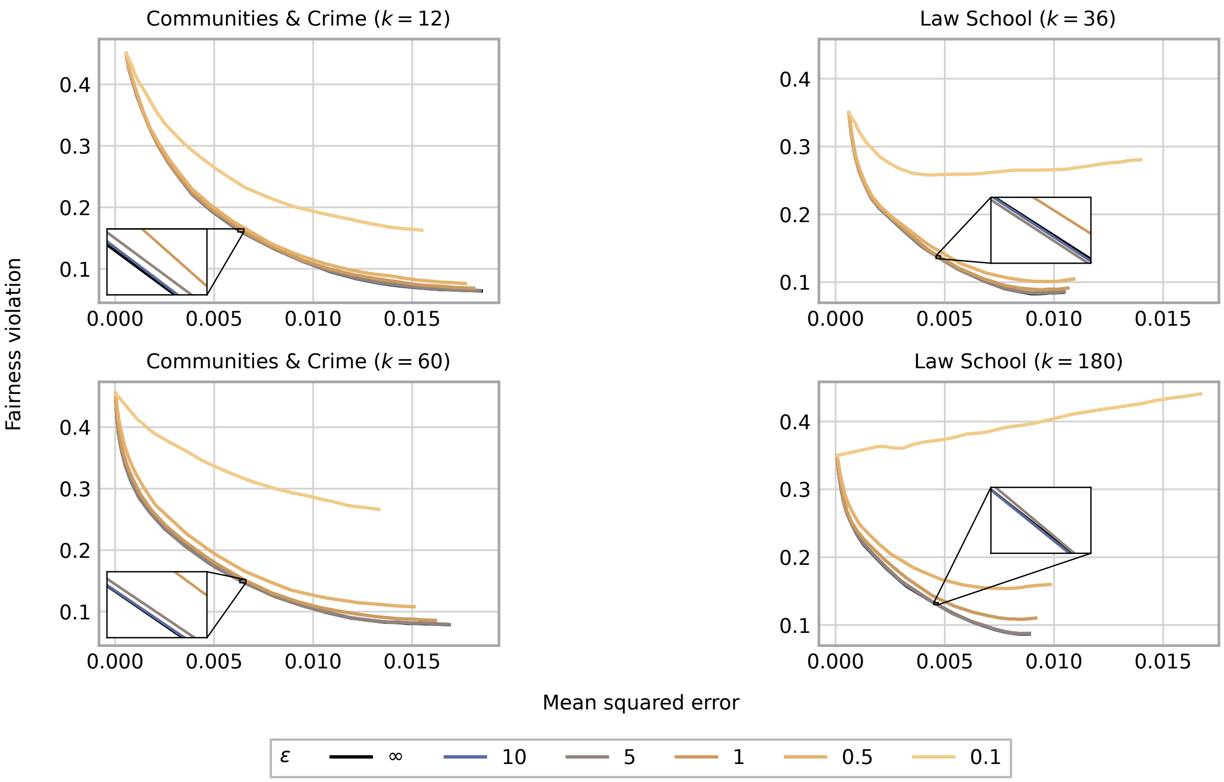

Figure 5.2: Error-fairness tradeoffs from private fair post-processing, under varying privacy budget ε and discretization bin counts k.

dataset with small ε and N, these points are not Pareto-optimal. This is because here, the estimated distributions are inaccurate due to a combination of sampling error and large-scale DP noise, so enforcing exact SP can overfit to artifacts or noise rather than the true signal, thereby increasing MSE without achieving fairness gains.

Tradeoff Between Statistical Bias and Variance. Recall from the analysis in Section 5.4.4 that the MSE of the post-processed predictor reflects a tradeoff between statistical bias and variance determined by the choice of k: increasing the number of bins reduces discretization error (bias) but increases both estimation error and the variance introduced by DP noise, whereas decreasing k has the opposite effect. Fairness, on the other hand, depends only on the variance term and not on the bias, so higher fairness can be achieved by reducing k, albeit at the cost of higher MSE.

Figure 5.3 illustrates these tradeoffs on Law School for various k under $\varepsilon = 0.1$. Exact SP at the far right is obtained with $k = 1$, but this comes with a high MSE of 0.6772. As expected, smaller k values lead to higher fairness (lower V^{SP}) but higher discretization error, evident in the rightward shift of the starting points corresponding to discretized but

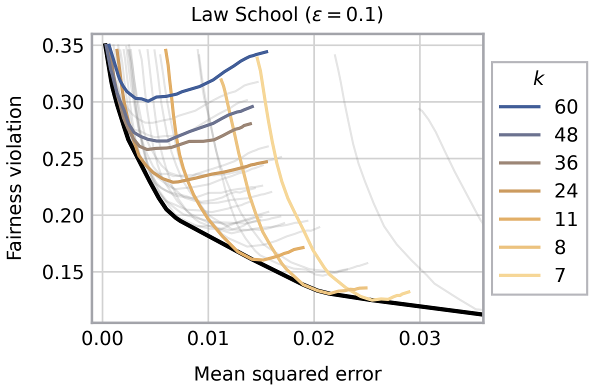

Figure 5.3: Error-fairness tradeoffs under $\varepsilon = 0.1$ on Law School from sweeping over α and k. The black line denotes the lower envelope of all curves and terminates at $(0.6772, 0)$ on the right (cropped).

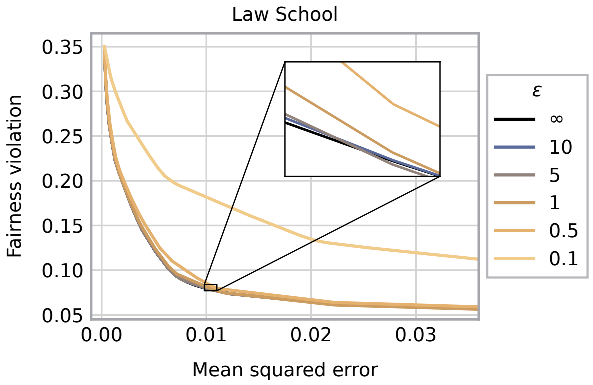

Figure 5.4: Error-fairness tradeoffs under various ε on Law School from sweeping over α and k and plotting the lower envelope. All lines meet at $(0.6772, 0)$ on the right.

unprocessed predictors. Conversely, using more bins can sometimes result in worse tradeoffs due to increased variance; for instance, the tradeoff curves for $k = 60$ are almost entirely dominated by those for $k = 36$. Overall, the best tradeoffs (the lower envelope of the curves) are achieved by selecting smaller (α, k) when targeting high fairness levels, and larger (α, k) when prioritizing lower MSE at the expense of fairness.

Error-Privacy-Fairness Tradeoff. The black line in Fig. 5.3 represents the optimal error-fairness tradeoffs attainable with our private fair post-processing algorithm for $\varepsilon = 0.1$, obtained by sweeping over α and k and taking the lower envelope. Segments of this envelope not directly reached by any single k can be achieved by randomly interpolating between two predictors, although doing so via post-processing would require doubling the privacy budget.

We repeat this experiment for all privacy budgets ε on the Law School dataset and plot their lower envelopes in Fig. 5.4. These plots illustrate the Pareto front of the error-privacy-fairness tradeoffs achievable by our algorithm, showing that stricter privacy budgets consistently degrade the attainable tradeoffs.

CHAPTER 6: APPLICATION: FAIRNESS IN LLM-BASED CLASSIFIERS

Instruction fine-tuned large language models (LLMs) such as Llama 3 [177], Gemma 3 [178], and GPT-4 [179]—equipped with broad knowledge from pre-training—have enabled a new paradigm for building task-specific predictors: practitioners can adapt these models to a task simply by prompting them with a natural language description of it, often requiring little (*few-shot*) to no labeled data (*zero-shot*) [33, 34, 35]. This paradigm, also known as *in-context learning*, is particularly attractive in low-resource domains such as healthcare, where labeled data are costly to obtain [180]. For instance, Hegselmann et al. [45] demonstrate that, on tabular datasets, few-shot prompting can outperform traditional ML models trained from scratch in low-data regimes.

Classifiers derived from LLMs are not immune to unfairness, and numerous studies have explore ways to achieve group fairness on LLM-based classifiers (reviewed in Section 6.1). The majority applies fair algorithms to LLMs in combination with fine-tuning or head-tuning on last-layer embeddings, most often through fair representation learning algorithms (Adversarial and MinDiff), because they integrate naturally with neural models. However, pre-processing and in-processing fair algorithms that require model tuning are inapplicable to closed-weight LLMs (namely, the most capable commercial models including GPT-4, Gemini, and Claude), since these expose neither model weights nor internal representations, but only output tokens and, at best, token log-probabilities. By contrast, post-processing algorithms require no model tuning: they achieve fairness by transforming the predicted probabilities for the task label and group memberships—both of which can be elicited from an LLM via prompting. This makes post-processing algorithms, such as LinearPost, well suited to deriving fair classifiers from closed-weight LLMs in the emerging prompt-based paradigm.

As a demonstration of its flexibility, this chapter applies and evaluates the performance of LinearPost for deriving fair classifiers from state-of-the-art LLMs, including open-weight (Llama3.1, Gemma3) and closed-weight commercial models (GPT-4o). The pointwise risk and group predictions required for post-processing are obtained via zero-shot prompting. Before post-processing, we calibrate these raw predictions by re-fitting them to ground-truth labels with logistic regression, since calibration is necessary for group fairness guarantees (Section 4.3) but is distribution-specific—a condition that LLMs, trained on generic tasks, may not satisfy for the specific task at hand.

Last but not least, because the representation result for the optimal fair classifier in Theorem 3.1 shows that these predictions are sufficient statistics for fair classification, they can also serve as informative features for training fair classifiers with other fair algorithms. So, accordingly, we additionally apply and evaluate Reductions and MinDiff on these calibrated predictions and compare their performance with LinearPost.

6.1 RELATED WORK

To derive fair classifiers from LLMs, prior work has applied fair algorithms to LLMs with some form of (parameter-efficient) fine-tuning. However, because fine-tuning large models is expensive and access to model weights or internal representations is required (head-tuning requires last-layer embeddings), these approaches have primarily been experimented on small, open-weight models such as BERT [108], GPT-2 [181], and Llama. From the pre-processing category, Han et al. [182] reweight the training data and perform head-tuning, while Garg et al. [183] and Fatemi et al. [184] use counterfactual data augmentation to balance group-label distributions, followed by fine-tuning or prompt-tuning. In the in-processing category, Han et al. [185] learn fair representations on final-layer embeddings while freezing lower layers, and Cherepanova et al. [186] perform prompt-tuning with a fairness-regularized objective.

Post-processing has also been explored. Atwood et al. [187] apply the post-processing algorithm from [104] to learn a fair classification head via head-tuning. Baldini et al. [188] consider equalized odds in the attribute-aware setting and apply algorithms from [18, 103] directly to LLM predictions without additional tuning (although the LLM was fine-tuned for downstream tasks). This latter setup is closely related to ours in that it requires only access to model predictions and performs post-hoc fairness mitigation, but it is limited to the attribute-aware setting.

Alternatively, recent work has explored prompt-based fairness interventions that leverage LLMs' instruction-following abilities, such as by explicitly prompting the model to "*assign labels equally across groups*" or by constructing few-shot demonstrations that reflect the fairness criterion [186, 189, 190, 191]. While these approaches show promise, their effectiveness can be inconsistent (for example, due to sensitivity to prompt phrasing [192]), and the black-box nature of closed-weight LLMs means that it is unclear how to iterate on a prompt when the fairness goals are not met, beyond trial-and-error.

Beyond fair algorithms and prompt-based mitigations, unsupervised methods have also been proposed to improve group fairness in downstream tasks, including null-space projection, which removes group-sensitive information from LLM embeddings [193, 194], and more recent embedding steering techniques [195].

Finally, fairness concerns for LLMs extend beyond group fairness in downstream classifi-

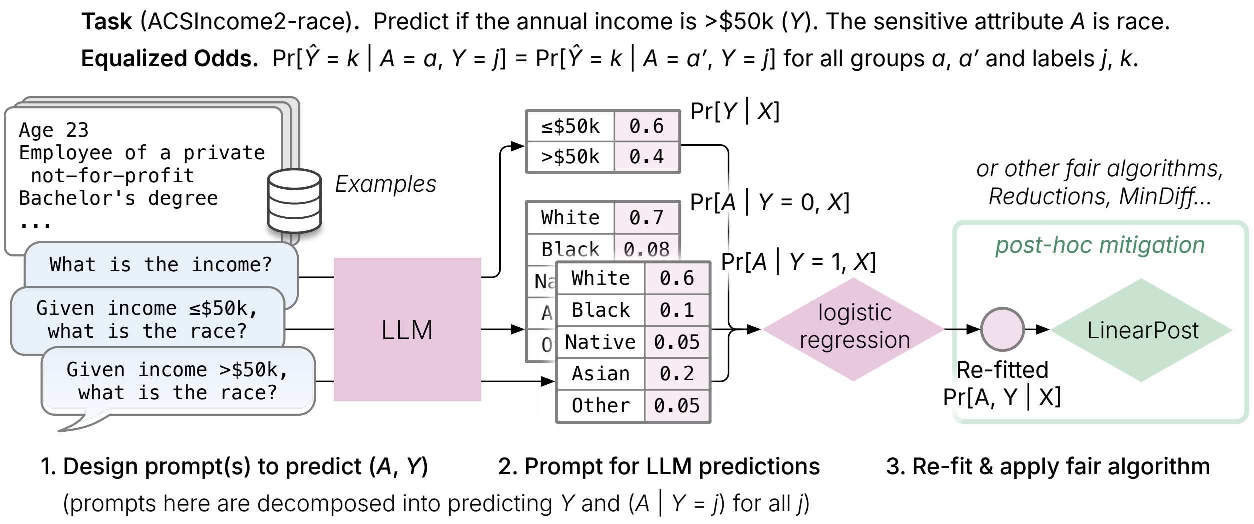
Task (ACSIncome2-race). Predict if the annual income is >$50k (Y). The sensitive attribute A is race.
Equalized Odds. Pr[Ŷ = k | A = a, Y = j] = Pr[Ŷ = k | A = a′, Y = j] for all groups a, a′ and labels j, k.
Age 23
Employee of a private not-for-profit
Bachelor's degree
...
Examples
What is the income?
Given income ≤$50k, what is the race?
Given income >$50k, what is the race?
LLM
≤$50k 0.6
>$50k 0.4
Pr[Y | X]
White 0.7
Black 0.08
Pr[A | Y = 0, X]
White 0.6
Black 0.1
Native 0.05
Asian 0.2
Other 0.05
Pr[A | Y = 1, X]
logistic regression
Re-fitted Pr[A, Y | X]
or other fair algorithms, Reductions, MinDiff...
post-hoc mitigation
LinearPost
1. Design prompt(s) to predict (A, Y)
(prompts here are decomposed into predicting Y and (A | Y = j) for all j)
2. Prompt for LLM predictions
3. Re-fit & apply fair algorithm

Figure 6.1: Given a task and fairness criterion, we first design prompts to elicit LLM predictions for the task label Y and group memberships Z. After collecting these raw predictions on the training set, we re-fit/calibrate to the ground-truth labels, and finally apply LinearPost to obtain a lightweight fair classification head.

cation. A line of work, beginning with Bolukbasi et al. [9] and Caliskan et al. [7], studies social biases inherited by LLM from pre-training corpora. Since then, numerous benchmarks and evaluation protocols have been developed to measure harms in language generation [196, 197, 198, 199, 200]. For example, the BBQ benchmark [201] has been used to assess bias in recent GPT releases [202]. While such biases may affect downstream classification fairness, directly mitigating them does not necessarily guarantee improvements in group fairness, which is a statistical property dependent on the task's data distribution. We therefore view this body of research as complementary to efforts, such as ours, that explicitly enforce group fairness in downstream tasks.

6.2 PROMPT, RE-FIT, THEN POST-PROCESS

We describe the procedure for deriving fair classifiers from (closed-weight) LLMs using LinearPost through prompting followed by fair post-processing. Given a labeled training set, $\{x^{(j)}, y^{(j)}, z^{(j)}\}_{j=1}^{N}$, the procedure consists of three steps: (1) design prompt template(s) to elicit LLM predictions for the task label Y and group memberships Z; (2) insert each example into the template(s) and prompt the LLM for predictions; (3) re-fit/calibrate the raw LLM predictions, then apply LinearPost.

Deriving Fair Classifiers from LLMs. We describe each step of our proposed procedure in detail below, and provide a flowchart in Fig. 6.1 for illustration, which uses the ACSIncome2-Race task under equalized odds as an example.

1. (Design Prompt(s)). Applying LinearPost (Algorithm 3.1) requires predictions for the pointwise risk (which can be computed from $\mathbb{P}[Y \mid X]$) and group memberships $Z \in \{0,1\}^G$ for each example X. So the goal of this initial step is to design prompt template(s) that elicit these predictions from the LLM.

   Since $Y \in [K]$ and each $Z_i \in \{0,1\}$ are all categorical variables, we can treat their predictions as classification tasks, and accordingly, construct $(1+G)$ prompts in multiple-choice question-answering (MCQA) format with task instructions—where each answer choice corresponds to a class label—to elicit these predictions. See Appendix B.3.2 for MCQA prompts designed to elicit predictions for the label Y.

   If the group memberships Z are one-hot (meaning that every example belongs to exactly one groups in $i \in [G]$), which is the case under the standard fairness criteria presented in Section 2.2 (except when the groups are overlapping), the group prediction tasks can be consolidated into a single classification task with G option choices. See Appendix B.3.2 for prompts designed to predict the sensitive attribute A (which is equal to Z under statistical parity; Example 2.1).

   For EO fairness in the attribute-blind setting,[1] the group membership Z need to be set to the joint (A,Y) label, and predictions for the full joint $\mathbb{P}[A, Y \mid X]$ are required.[2] Since (A,Y) labels are one-hot, we may construct a single MCQA prompt with MK choices. For instance, on ACSIncome2-Race: "`A. The income is >$50k and the race is White`", "`B. The income is >$50k and the race is Black`", and so on. Alternatively, we may factor the joint distribution as $\mathbb{P}[A,Y \mid X] = \mathbb{P}[Y \mid X] \cdot \mathbb{P}[A \mid Y, X]$ and decompose the prediction task accordingly into $(1+K)$ subtasks—one for Y, and the rest for A conditioned on each hypothetical value of Y: "`Given that the income is >$50k, what is the race?`" and so on (see Listing B.1); then, the (A,Y) predictions can be obtained by composing the predictions for the $(1+K)$ subtasks.[3] The benefit of this decomposition is that the subtasks are simpler than the

[1] In the attribute-aware setting, for EO fairness as well as SP, TPR, and FPR, we only need to obtain LLM predictions for $\mathbb{P}[Y \mid X]$, since A is always observed. This reduces to the setting of [188].

[2] TPR and FPR fairness also require setting Z to the joint (A,Y), but only partial information of this joint is needed for LinearPost: $\mathbb{P}[A, Y=1 \mid X]$ for TPR, and $\mathbb{P}[A, Y=0 \mid X]$ for FPR.

[3] The joint can also be factored as $\mathbb{P}(A, Y \mid X) = \mathbb{P}(A \mid X)\,\mathbb{P}(Y \mid A, X)$, leading to $(1+G)$ factorized prompts. We used the alternative factorization in our experiments because it required fewer prompts for the datasets considered.

task of predicting the joint (A, Y), and may therefore boost LLM prediction performance, although at the cost of requiring more queries per example. Lastly, on tasks where A can be easily inferred from X (such as BiasBios and CivilComments; see Table B.1 for examples), it is likely that the conditional independence of $\mathbb{P}[A, Y \mid X] = \mathbb{P}[A \mid X] \cdot \mathbb{P}[Y \mid X]$ approximately holds. We may then leverage this structure to reduce the number of factorized prompts to two: one for predicting Y and the other for A (see Listings B.4 and B.5).

2. (Prompt for LLM Predictions). We obtain LLM predictions by inserting each example x into the prompt(s) designed above, and query the LLM (over multiple messages). The model's predictions are then extracted by taking only the logits of the output tokens corresponding to the MCQA option choices (each mapped to a label).[4]

   If only the top-k logits are accessible (for instance, $k = 20$ in OpenAI's GPT-4o API at the time of writing, due to intellectual property protections [205]) and not all option choices in the prompt template appear (particularly when prompting for label predictions on the 28-class BiasBios dataset), we assign a large negative value to the missing logits (we use -50 for GPT-4o).

3. (Re-Fit and Apply LinearPost). We prompt the LLM for predictions on the training examples, then re-fit these raw predictions to the task and group labels using logistic regression to calibrate them, while keeping the LLM frozen (further details in Section 6.2.1). This calibration step is necessary for achieving fairness, as discussed in Chapter 4, because group fairness is defined statistically with respect to the underlying distribution. Raw LLM predictions may be miscalibrated due to shifts between the LLM's pre-training distribution (or internal beliefs) and the distribution of the task at hand, or due to sensitivity to prompt phrasing and formatting [192].

   With the calibrated pointwise risk and group predictions, we then apply LinearPost (Algorithm 3.1) to compute a post-processing transformation that maps these predictions to hard class labels satisfying the desired fairness constraints.

At test time, given an example, we wrap it inside the prompt template(s) designed in step (1) and prompt the LLM to obtain the predictions as discussed in step (2), then use the calibration transformation learned in step (3) to calibrate them, and finally call the post-processing transformation from LinearPost to get the final class assignment.

[4] If token-level logits are not accessible, alternative approaches include sampling multiple completions under non-zero temperature [203] or using *verbal elicitation* [204].

6.2.1 Details on the Re-Fitting Step

We elaborate here on how the re-fitting step is performed in our experiments, focusing on EO fairness. The predictions required for EO fairness subsume those for SP, TPR, and FPR fairness, and are thus downwards compatible. Our discussion primarily concerns how predictions from factorized prompts are recomposed into the full joint distribution.

For EO fairness, after obtaining the LLM's token log-probabilities for predicting $(A, Y) \in [M] \times [K]$ on an instance X (possibly from factorized prompts), we recompose them into $lp_{A,Y} \in \mathbb{R}^{M \times K}$ representing the LLM's estimate of the full joint $\log \mathbb{P}(A, Y \mid X)$:

- (No Factorization). If the prompt directly elicits (A, Y) jointly, the output logits already have the desired semantics and shape, $lp_{A,Y} \in \mathbb{R}^{M \times K}$.

- (Factorized with Conditional Independence Assumption). If the prompts assume conditional independence and predict A and Y independently, let $lp_A \in \mathbb{R}^M$ and $lp_Y \in \mathbb{R}^K$ denote their respective logits. We set

$$(lp_{A,Y})_{a,k} = (lp_A)_a + (lp_Y)_k. \tag{6.1}$$

  Note that

$$\operatorname{softmax}(lp_{A,Y})_{a,k} = \operatorname{softmax}(lp_A)_a \cdot \operatorname{softmax}(lp_Y)_k. \tag{6.2}$$

- (Factorized Without Conditional Independence Assumption). If the prompts follow the factorization $\mathbb{P}[A, Y \mid X] = \mathbb{P}[Y \mid X] \cdot \mathbb{P}[A \mid Y, X]$, they predict Y and, for each k, $A \mid Y = k$. Let $lp_Y \in \mathbb{R}^K$ denote the logits for Y, and $lp_{A|Y=k} \in \mathbb{R}^M$ the logits for A given $Y = k$. Using the log-sum-exp function $\operatorname{LSE}(z) = \log \sum_i \exp(z_i)$, we set

$$(lp_{A,Y})_{a,k} = (lp_{A|Y=k})_a - \operatorname{LSE}(lp_{A|Y=k}) + (lp_Y)_k. \tag{6.3}$$

  Note that

$$\operatorname{softmax}(lp_{A,Y})_{a,k} = \operatorname{softmax}(lp_{A|Y=k})_a \cdot \operatorname{softmax}(lp_Y)_k. \tag{6.4}$$

In all cases, the softmax is applied after flattening $lp_{A,Y}$.

Finally, we fit recomposed $lp_{A,Y}$ to the ground-truth joint labels (A, Y) using logistic regression, which is an MK-way classification task. The resulting probabilities, which lie in the probability simplex $\Delta^{M \times K}$, are then passed to the post-processing step.

Overlapping Groups. When the groups are overlapping ($A \in \{0, 1\}^M$ is multilabel; for example, the CivilComments dataset, see Section 6.3.1), we require predictions for the joint distribution over the class label Y and all possible subgroup memberships. In this case, $lp_{A,Y} \in \mathbb{R}^{2^M \times K}$, where for any $I \subseteq [M]$, $(lp_{A,Y})_{I,k}$ represents the prediction for $\mathbb{P}(\bigwedge_{i \in I}\{A_i = 1\}, Y = k)$, the probability that the instance has class label k and belongs to all and only the groups in I.

If we assume conditional independence among Y and each component of A, we construct $(1 + M)$ prompts: one for predicting Y, and one for each $i \in [M]$ predicting whether the instance belongs to group i (binary classification). We then set

$$(lp_{A,Y})_{I,k} = (lp_Y)_k + \sum_{i \in I} (lp_{A_i})_1 + \sum_{i \notin I} (lp_{A_i})_0. \tag{6.5}$$

Note that

$$\operatorname{softmax}(lp_{A,Y})_{I,k} = \operatorname{softmax}(lp_Y)_k \cdot \prod_{i \in I} \operatorname{softmax}(lp_{A_i})_1 \cdot \prod_{i \notin I} \operatorname{softmax}(lp_{A_i})_0. \tag{6.6}$$

Finally, we fit $lp_{A,Y}$ to the ground-truth class labels and subgroup labels jointly, which is a $(2^M K)$-way classification task.

6.3 EXPERIMENT SETUP

> Omitted implementation details, such as dataset and fair algorithm descriptions, split sizes, hyperparameter settings, and the evaluation protocol, are provided in the unified Experiment Details section in Appendix B.

We evaluate our procedure for deriving fair classifiers from closed-weight LLMs via LinearPost (described in Section 6.2) on three tabular and two textual datasets, using three open-weight LLMs (Llama 3.1 8B/70B and Gemma 3 27B, through Hugging Face) and one closed-weight LLM (GPT-4o, through OpenAI API).[5] For simplicity, we use zero-shot prompting to elicit LLM predictions, though few-shot or chain-of-thought prompting may be used and improve performance, and can also be combined with prompt-based mitigations discussed in Section 6.1.

We consider SP, TPR, FPR, and/or EO fairness. While post-processing for SP, TPR, and FPR requires only partial information from the joint distribution of $(A, Y) \mid X$, we always prompt for the full joint as required by EO fairness, so the setup can be shared across experiments.

[5] `Llama-3.1-8B`, `Llama-3.1-70B`, `gemma-3-27b-it`, and `gpt-4o-2024-08-06`.

The prompt templates used in our experiments are shown in Listings B.1 to B.5, with each dataset example inserted by replacing the placeholder `{example}`. Because the re-fitting step mitigates the LLM's sensitivity to prompt phrasing, we did not devote significant effort to prompt engineering or optimization, beyond verifying that the model followed the MCQA instructions and produced valid options. The template for predicting class labels on CivilComments is adapted from [187].

Other Fair Algorithms. Because the predictions elicited from the LLM for LinearPost constitute sufficient statistics for fair classification, as implied by the representation result for the optimal fair classifier in Theorem 3.1, they can also be used as features for training fair classifiers with other algorithms (including pre-processing and in-processing algorithms). From this perspective, our procedure can be viewed as a general framework for adapting traditional fair algorithms to derive fair classifiers from LLMs.

Therefore, we additionally apply Reductions (binary classification only; logistic regression as the base estimator) [73] and MinDiff [206] to the same re-fitted predictions used by LinearPost as features, and compare their performance. These algorithms, however, incur higher overhead: Reductions solves a sequence of cost-sensitive classification problems, while MinDiff trains a classification head (we use shallow neural networks) with sufficient representation power to match distributions effectively.

Therefore, we additionally apply Reductions [73] (binary classification only; we use logistic regression as the base estimator) and MinDiff [206] to these predictions as features to train fair classifiers, and compare their performance with LinearPost. However, these algorithms incur higher overhead than LinearPost: Reductions solves a sequence of cost-sensitive classification problems, resulting in longer runtimes, while MinDiff requires training a classification head with sufficient representation power to be successful in matching distributions (we use shallow neural networks).

For the no-mitigation baseline, we fit a logistic regression on the LLM's log-probabilities for Y, using the ground-truth labels as targets, without fairness constraints.

Other Features. While the LLM predictions we elicit for LinearPost are theoretically sufficient statistics for fair classification, this holds only if the predictions are Bayes-optimal. If they are not, then since these predictions are very low-dimensional (on the order of MK: 4 on Adult and 10 on ACSIncome2-Race) compared to the dimensionality of LLM embeddings (4096 for Llama 3.1 8B) or the original input space (97 pre-processed tabular features on Adult, 810 on ACSIncome2-Race), they may fail to capture other information about the input useful for fair classification, potentially degrading performance.

To assess this gap, we compare our procedure, which performs post-processing on the LLM's log-probabilities from a small set of designed prompts (denoted by PREDS), against alternatives that retain more input information: applying the fair algorithms to the mean-pooled embeddings from open-weight LLMs (EMBEDS),[6] and directly to the original tabular features (TABULAR; applicable only to tabular datasets), training classifiers from scratch. In these settings, REDUCTIONS and MINDIFF are applied the same as above, and for LINEARPOST, we follow the "pre-train then post-process" procedure: first training a logistic regression model to predict (A, Y) from either the embeddings or tabular features, then post-processing its predictions; here, the same training data is used for both steps, rather than split as in the experiments of the previous chapters. The NO-MITIGATION baseline fits the same features to Y using logistic regression.

Given these alternative fair algorithms and feature settings, we define an *experiment configuration* as a combination of an LLM, a fair algorithm, and one of three feature types used as input to the fair algorithm: LLM PREDS (our general procedure), LLM EMBEDS, or the original TABULAR features. Excluding the NO-MITIGATION baselines, this yields at most 24 configurations, depending on the dataset.

6.3.1 Datasets

We evaluate on three tabular datasets (ADULT, ACSINCOME2-RACE, COMPAS) and two textual datasets (BIASBIOS, CIVILCOMMENTS). We consider the attribute-blind setting, both removing `sex` and `race` columns from the tabular datasets ADULT, ACSINCOME2-RACE, and COMPAS (textual datasets are treated as attribute-blind by default, even if the sensitive attribute could be inferred from the text). In COMPAS, the `c_charge_desc` column contains a natural language description of the charge; this column is retained for LLM-based classifiers (PREDS and EMBEDS features) but dropped for TABULAR classifiers. An example from each dataset is provided in Table B.1.

Overlapping Groups. In CIVILCOMMENTS, the task is to detect whether a public comment is toxic, with sensitive attributes given by the religion(s) mentioned in the comment (Christianity, Judaism, Islam, Hinduism, Buddhism). These groups are overlapping because a comment may mention multiple religions or none at all. Thus, the sensitive attribute is multilabel: $A \in \{0, 1\}^5$, $M = 5$.

Therefore, for fairness, we aim to achieve FPR parity with respect to all nonempty overlapping subsets of groups (nonempty meaning fairness is not enforced on comments mentioning

[6] In our preliminary experiments, mean-pooling outperformed both last-token and max-pooling.

no religion). Let $\mathcal{I} = 2^{[M]} \setminus \{\emptyset\}$ denote the collection of nonempty overlapping groups. For any $I \in \mathcal{I}$, define the event $(A \in I) = \bigwedge_{i \in I}(A_i = 1)$ to indicate that an example belongs to all groups in I (without requiring $A_i = 0$ for $i \notin I$). For example, $A \in \{\texttt{christian}, \texttt{jewish}\}$ means the comment mentions both Christianity and Judaism, but not necessarily only these two.

We define the *all-way*[7] FPR violation, extending the definition in Section 2.2.1, as

$$V^{\mathrm{FPR}} = \max_{I, I' \in \mathcal{I}} \Big(\mathbb{P}[\widehat{Y} = 2 \mid Y = 1, A \in I] - \mathbb{P}[\widehat{Y} = 2 \mid Y = 1, A \in I'] \Big). \tag{6.7}$$

Fairness mitigations are applied according to this definition.

However, intersections of multiple groups are rare (e.g., comments mentioning many religions simultaneously), leading to high variance in fairness violation estimates from finite samples. To reduce this sensitivity, we adopt a weighted mean-difference definition of FPR violation for evaluation:

$$V^{\mathrm{FPR,weighted\text{-}MD}} = \sum_{I \in \mathcal{I}} p(1, I) \cdot \left| \mathrm{FPR}(I) - \overline{\mathrm{FPR}} \right|, \quad \overline{\mathrm{FPR}} = \sum_{I \in \mathcal{I}} \frac{p(1, I)}{\sum_{I' \in \mathcal{I}} p(1, I')} \mathrm{FPR}(I), \tag{6.8}$$

where $p(1, I) = \mathbb{P}[Y = 1, A \in I]$ is the weight of subgroup I, $\mathrm{FPR}(I) = \mathbb{P}[\widehat{Y} = 2 \mid Y = 1, A \in I]$ is its conditional FPR, and $\overline{\mathrm{FPR}}$ is the weighted average FPR across all overlapping groups.

Tabular Data Serialization. We serialize the tabular features of the tabular datasets (Adult, ACSIncome2-Race, COMPAS) into a text format suitable for LLMs, following the *list serialization* approach of Hegselmann et al. [45]. Each serialized example is a multi-line string where each line has the form "`{column_name}: {value}`" (columns with missing values are omitted).

For categorical features, prior work applying LLMs to tabular datasets (including studies on LLM group fairness [186, 189]) typically populate the placeholders `{column_name}` and `{value}` with raw dataset encodings. For example, in Adult, the line "`workclass: State-gov`" means that the individual's "Class of Worker" is "State government employee". However, these codings are often terse or opaque, especially in ACSIncome, where, for instance, "Class of Worker" is coded as "`COW`", which may be difficult for an LLM to interpret without a dictionary.

To address this, we replace raw categorical encodings with their natural language descriptions (a similar approach appears in [191], who explicitly provided natural language descrip-

[7] Indicating that all overlapping groups are compared.

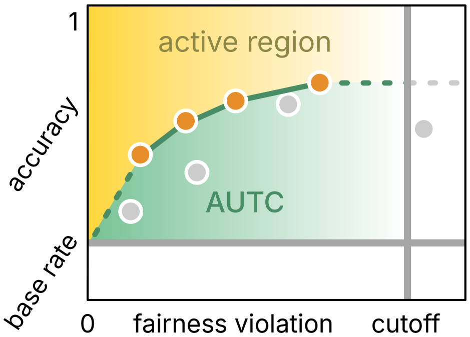

Figure 6.2: Area under the tradeoff curve (AUTC) is the weighted area under the Pareto-optimal accuracy-fairness tradeoffs within the active region: above the base rate and up to a violation cutoff.

tions for categorical encodings in prompts). In our preliminary experiments on Adult, substituting raw codes with natural language descriptions improved downstream performance after re-fitting. An example with and without this substitution is shown in Table B.1, and the full code-to-description mappings are available in our released code.

When training classifiers directly on tabular features from scratch, we apply standard pre-processing: categorical features are one-hot encoded, and all features are standardized.

6.3.2 Evaluation Metric: Area Under the Tradeoff Curve

To enable quantitative comparisons across the many configurations evaluated, we introduce an evaluation metric called the *area under the tradeoff curve* (AUTC), which summarizes each curve into a single scalar value. AUTC computes the area under the curve and normalizes it to lie between 0 (performance worse than the constant classifier) and 1 (perfect fairness and accuracy, though not necessarily attainable [70]). Under this vanilla definition, however, high AUTC scores can be achieved through high accuracy alone, even when tradeoffs in the high-fairness (low-violation) regime are poor. This occurs because the tradeoff curve extends into the high-violation region on the right, allowing high-accuracy classifiers to accumulate additional area there. To address this, we refine the metric in our formal definition of AUTC below.

Let $b = \max_k \mathbb{P}[Y = k]$ denote the base rate of the problem. Given a collection of $(\text{accuracy}, \text{fairness violation})$ pairs on the test set, obtained from a configuration under varying fairness tolerance settings of the fair algorithm (we filter out pairs whose validation performance is not Pareto-optimal), we compute its AUTC in the following steps (see Fig. 6.2 for a picture):

1. Filter the collection to retain only the Pareto-optimal tradeoff pairs.
2. Augment the collection with two additional pairs: $(b, 0)$, representing the majority-class constant classifier that is trivially fair, and $(\text{max accuracy}, \infty)$, which extends the curve to the right.
3. Sort the resulting pairs by fairness violation, then construct the tradeoff curve by linearly interpolating between adjacent pairs. Let $T : [0, \infty) \to [0, 1]$ denote the monotone function representing the tradeoff curve, which maps each violation level to its corresponding (interpolated) optimal accuracy.
4. Finally, given a penalty parameter $\gamma \in [0, \infty)$ and a cutoff violation $v \in [0, \infty)$, we compute the area under the tradeoff curve as

   $$\text{AUTC} = \left(\int_0^v (v - u)^\gamma \max(0, T(u) - b) \, \mathrm{d}u \right) / \overline{\text{AUTC}}, \tag{6.9}$$

   where the normalization constant $\overline{\text{AUTC}}$ is the (weighted) area of the *active region*:

   $$\overline{\text{AUTC}} = \int_0^v (v - u)^\gamma \max(0, 1 - b) \, \mathrm{d}u. \tag{6.10}$$

The penalty parameter $\gamma \in [0, \infty)$ and cutoff $v \in [0, \infty)$ mitigate the inflation of AUTC scores by models with high accuracy but poor accuracy-fairness tradeoffs in the low-violation regime (see MinDiff applied to embeds on CivilComments in Fig. 6.10 for an example). A larger γ places more emphasis on the low-violation region, while a smaller v truncates contributions from the high-violation regime.

6.4 EXPERIMENT RESULTS

Our main experiments in Section 6.4.1 evaluate the proposed procedure across datasets, fairness criteria, LLMs, and fair algorithms, comparing the utility of LLM preds features with LLM embeds (for open-weight models) and the original tabular features. We also conduct two sets of comparison and ablation studies. In Section 6.4.2, we assess performance under varying training set sizes. In Section 6.4.3, we compare our procedure, which prompts the LLM for both label Y and group membership Z predictions, to a variant that only prompts for Y (so that fair algorithms are applied directly to the LLM's label predictions); this ablation highlights the importance of explicitly incorporating group information for effective fairness mitigation.

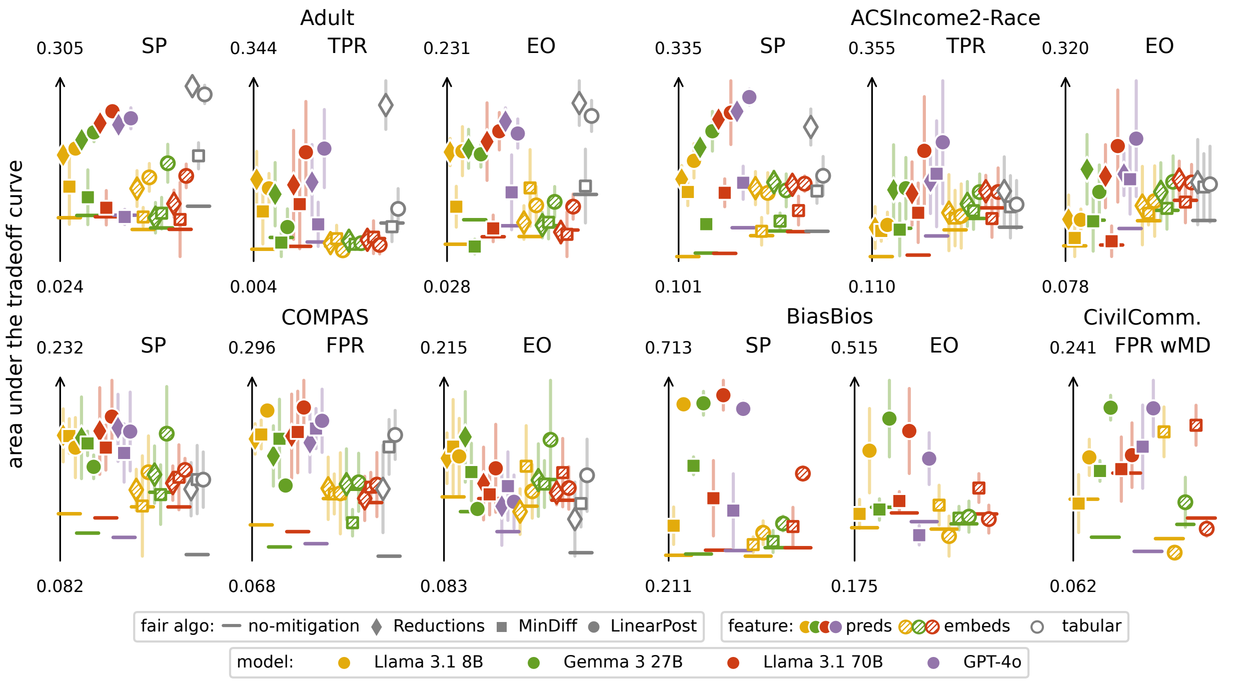

Figure 6.3: Area under the tradeoff curve (Section 6.3.2) achieved by each configuration. Our proposed procedure corresponds to using the PREDS feature; EMBEDS refers to training fair classifiers on open-weight LLM embeddings, and TABULAR refers to training on the datasets' original tabular features.

6.4.1 Main Results

In Fig. 6.3, we present the AUTC scores achieved by each configuration; full tradeoff curves are deferred to Figs. 6.6 to 6.10 in Section 6.4.4. In this main set of experiments, we deliberately use a smaller training set size (Table B.4) than is typical for these datasets, both to emulate data-scarce scenarios where zero-shot (or few-shot) prompting is most relevant and to accommodate limited compute resources.

First, as a sanity check, we find that fair algorithms generally achieve better accuracy-fairness tradeoffs than the NO-MITIGATION baseline, which linearly interpolates between the unconstrained classifier trained on the respective features and the majority-class constant classifier. Next, we observe that our proposed procedure is effective across all settings and for both open-weight and closed-weight models, demonstrating its generality and soundness: despite the low dimensionality of the PREDS features it extracts, it retains the essential information needed for fair classification. Finally, and as expected, performance tends to improve with the capability of the underlying LLM.

For open-weight models, fair classifiers trained on LLM PREDS (elicited by our procedure) achieve higher AUTC scores than those trained on LLM EMBEDS, and in some cases even

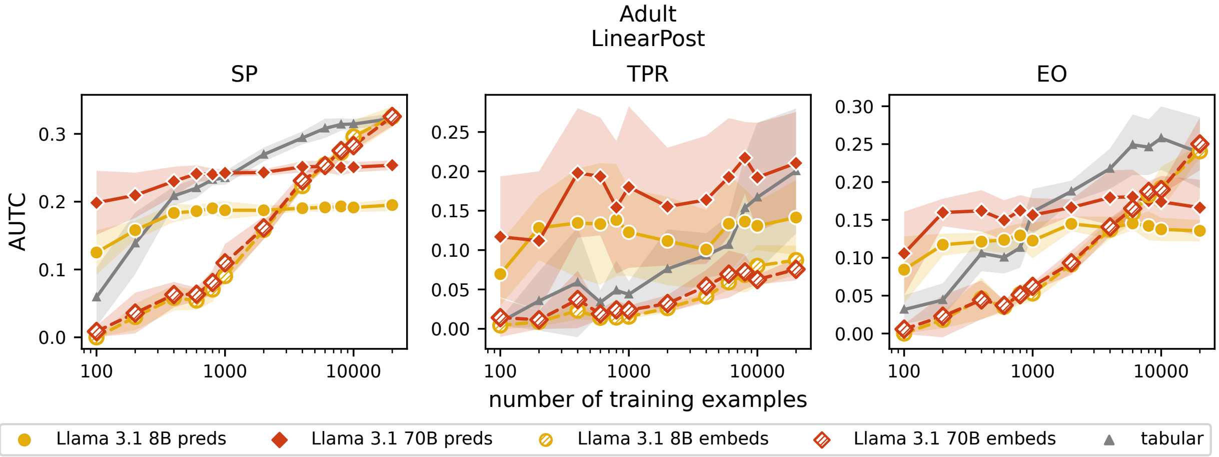

Figure 6.4: AUTC scores on Adult using LinearPost and Llama 3.1 open-weight models across training set sizes, comparing fair classifiers trained on preds, embeds, and the original tabular features.

outperform those trained on the original tabular features. This advantage likely comes from lower sample complexity: for example, on Adult, preds features have only 4 dimensions, compared to 4096 for Llama 3.1 8B embeddings—which exceeds the training set size of 2,000 and likely resulted in overfitting. Since LLM's output logits are linearly probed from the embeddings, our framework can be viewed as a form of dimensionality reduction, selecting only the most informative components for fair classification. The effect of training set size is examined further in Section 6.4.2.

6.4.2 Comparison Across Training Set Sizes

In Fig. 6.4, we compare performance across varying training set sizes on Adult using Llama 3.1 open-weight models. We focus our discussion on SP and EO fairness, as TPR results exhibit high variance.

The results reveal three distinct training-size regimes, each favoring a different feature type, likely driven by the sample complexity induced by feature dimensionality. In the very low-data regime (fewer than 1,000 examples), the preds feature outperforms the others due to its low dimensionality (4 on Adult), which reduces sample complexity. In this regime, both tabular (97 dimensions) and embeds (4096 for 8B and 8192 for 70B) perform poorly, likely due to overfitting, since their dimensionality far exceeds the number of training examples. However, the low dimensionality of preds also creates an information bottleneck, causing performance to plateau as the training size increases and suggesting that the LLM predictions are not Bayes-optimal.

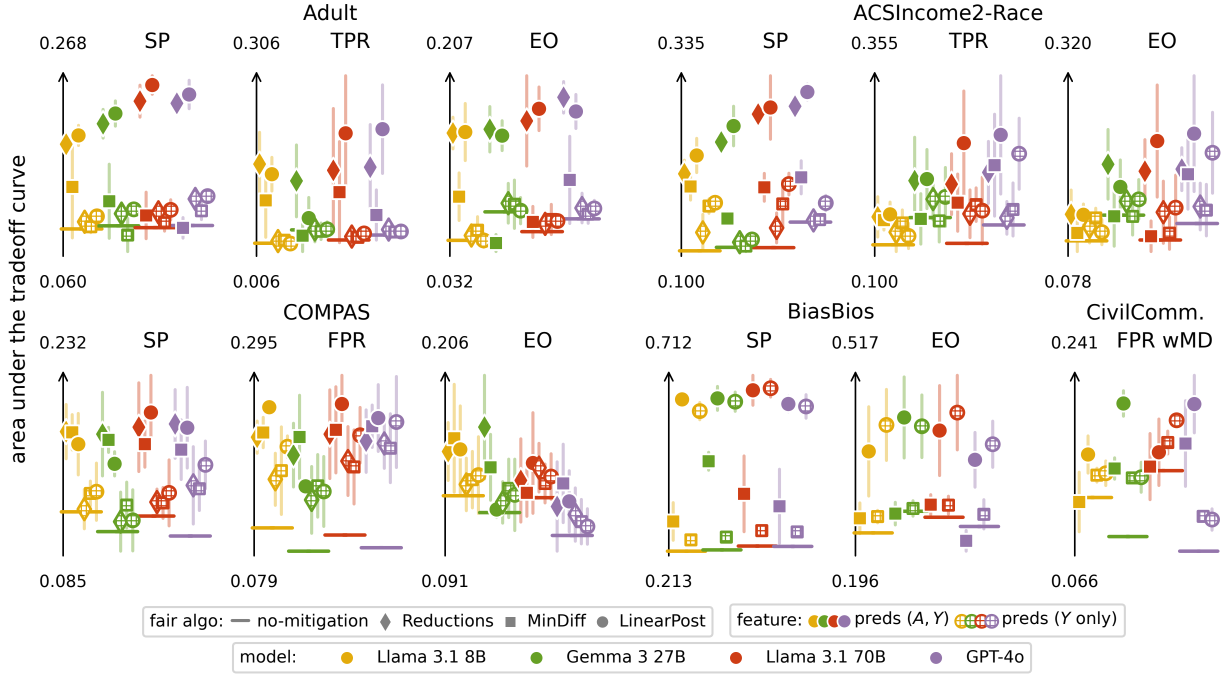

Figure 6.5: AUTC scores comparing our full procedure—which prompts the LLM for both group and label predictions—to a variant that only prompts for the label.

By contrast, performance for both TABULAR and EMBEDS steadily improves with more data. TABULAR leads in the low-data and mid-data regimes, but is eventually overtaken by EMBEDS in the high-data regime; although, intriguingly, embeddings from the 8B and 70B models yield nearly identical performance.

The strong performance of our procedure in the very low-data regime highlights the value of zero-shot and few-shot prompting for (tabular) classification when training data is scarce. Whereas Hegselmann et al. [45] showed that LLM-derived classifiers can outperform traditional ML models trained from scratch on tabular features in few-shot settings, our results extend this finding by demonstrating that LLM predictions are also advantageous for *fair* classification in such low-data scenarios.

6.4.3 Ablation Experiments on Group Predictions

Our procedure explicitly prompts the LLM to predict both the label Y and the group membership Z, since Theorem 3.1 implies that predictions for $(Y, Z) \mid X$ constitute *sufficient* statistics for fair classification. Nonetheless, this raises the curious question of whether group membership predictions are actually *necessary*?

To explore this, we conduct an ablation study in which we repeat the experiments from

Section 6.4.1 but compare our full procedure (prompting for both Y and Z) to a variant that only prompts for Y, where the fair algorithm is applied to the LLM's predictions of the label alone.

Figure 6.5 presents the results of these experiments. In nearly all cases, excluding group information (not prompting for the group membership Z) degrades the performance of the derived classifier. However, there are notable exceptions, specifically on COMPAS, BiasBios, and CivilComments, where the exclusion does not significantly harm performance and and sometimes leads to slight improvements.

Further analysis suggests two possible explanations for these exceptions. First, if the LLM performs poorly at predicting Z, then the group predictions provide little to no value. This is the case on COMPAS, where a logistic regression model trained to predict A from the raw LLM log-probabilities of (A, Y) (recomposed as described in Section 6.2.1) achieves a balanced accuracy of only 0.6560 ± 0.0055. Second, if the LLM's predictions of Y already encode substantial information about Z, then prompting for Z offers limited additional benefit. This is observed on BiasBios, where the balanced accuracy of predicting A from the log-probabilities of Y alone (of a fitted logistic regression model) is already 0.8768 ± 0.0008, compared to 0.9961 ± 0.0001 when using the full (A, Y) log-probabilities. The high dimensionality of Y in BiasBios (28 classes) likely contributed to this implicit encoding.

6.4.4 Results: Tradeoff Curves for Section 6.4.1

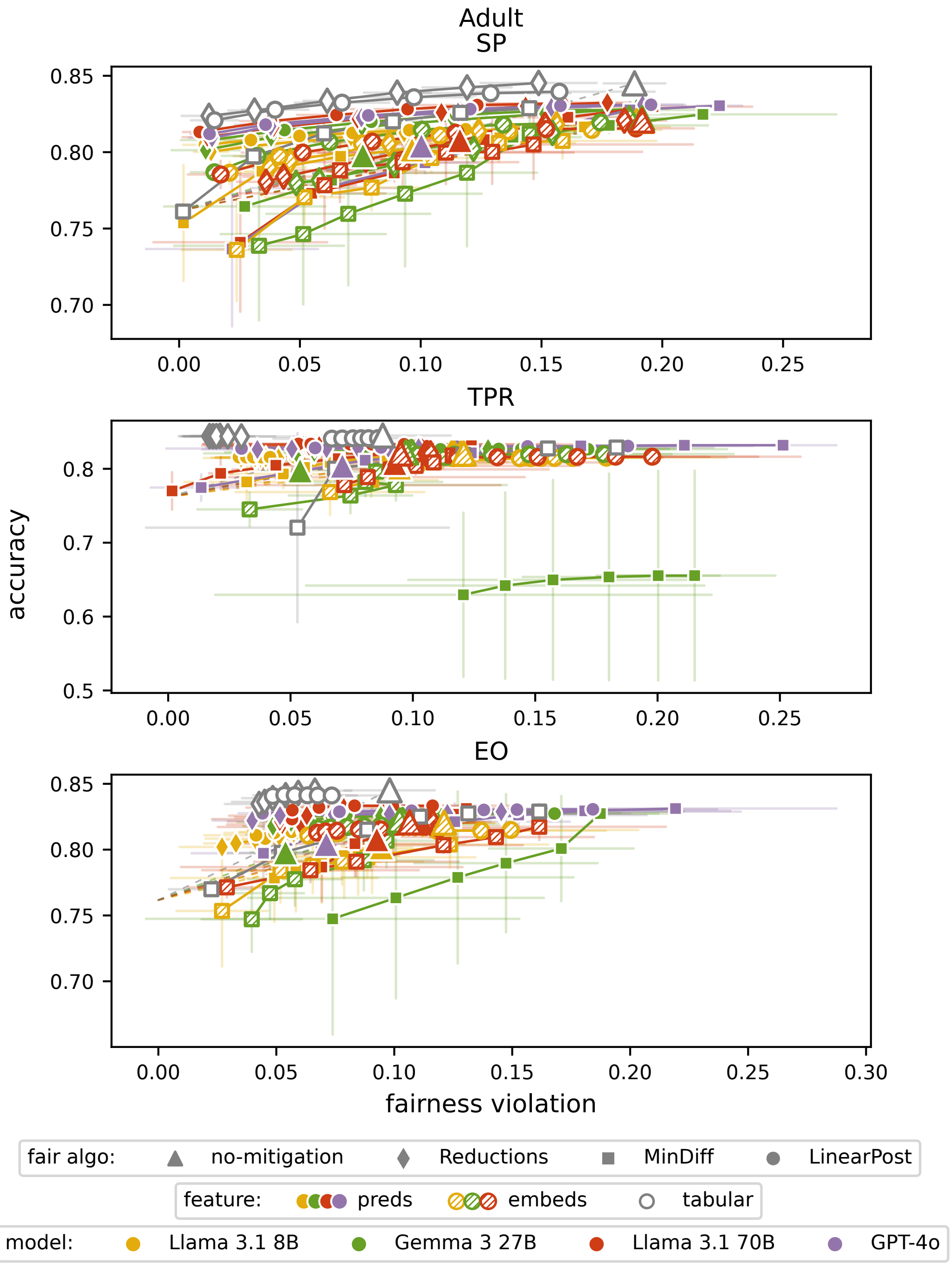

Figure 6.6: Accuracy-fairness tradeoffs on Adult for experiments in Section 6.4.1. Dashed lines interpolate between no-mitigation baselines and the majority-class constant classifier that is trivially fair.

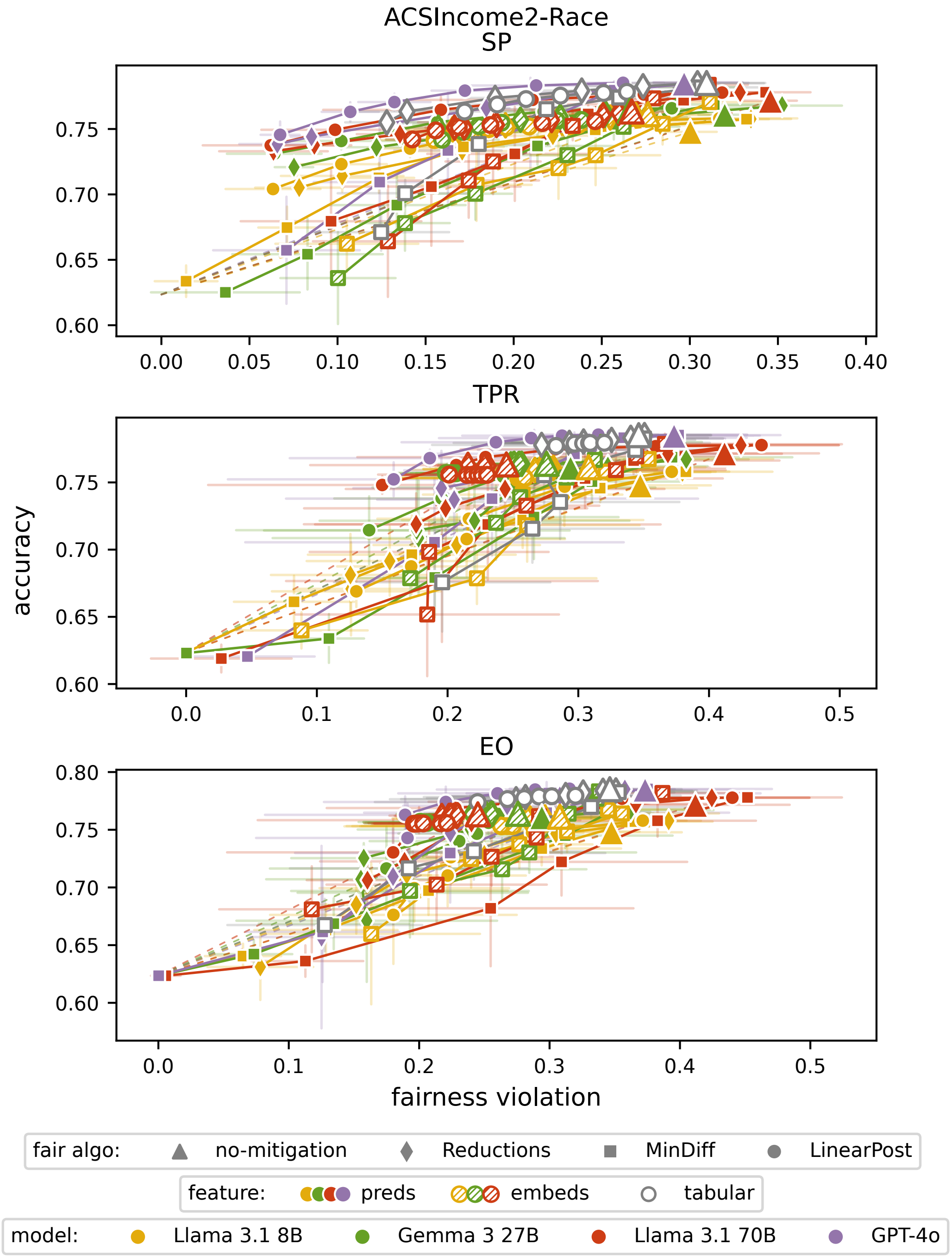

Figure 6.7: Accuracy-fairness tradeoffs on ACSIncome2-Race for experiments in Section 6.4.1. Dashed lines interpolate between no-mitigation baselines and the majority-class constant classifier that is trivially fair.

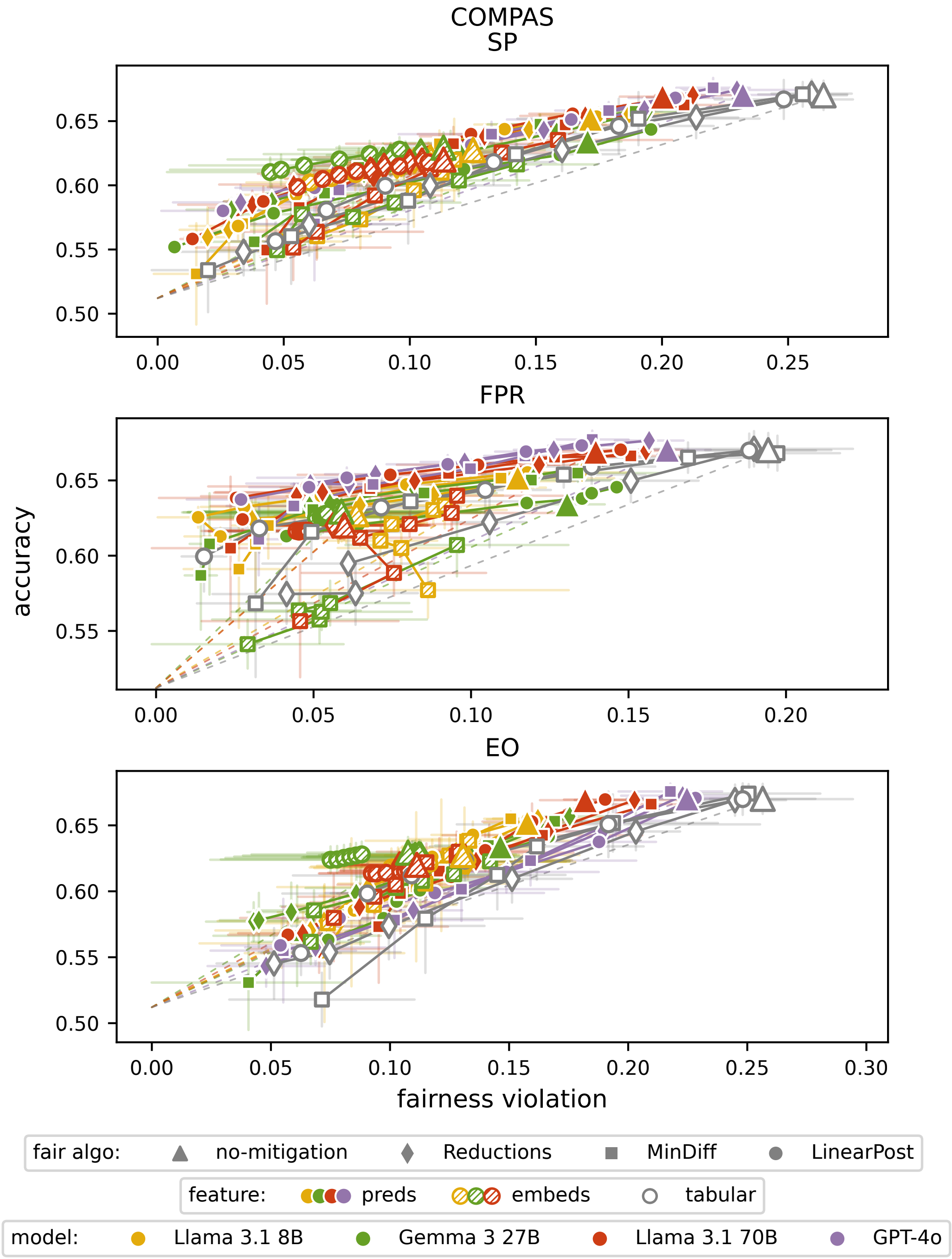

Figure 6.8: Accuracy-fairness tradeoffs on COMPAS for experiments in Section 6.4.1. Dashed lines interpolate between NO-MITIGATION baselines and the majority-class constant classifier that is trivially fair.

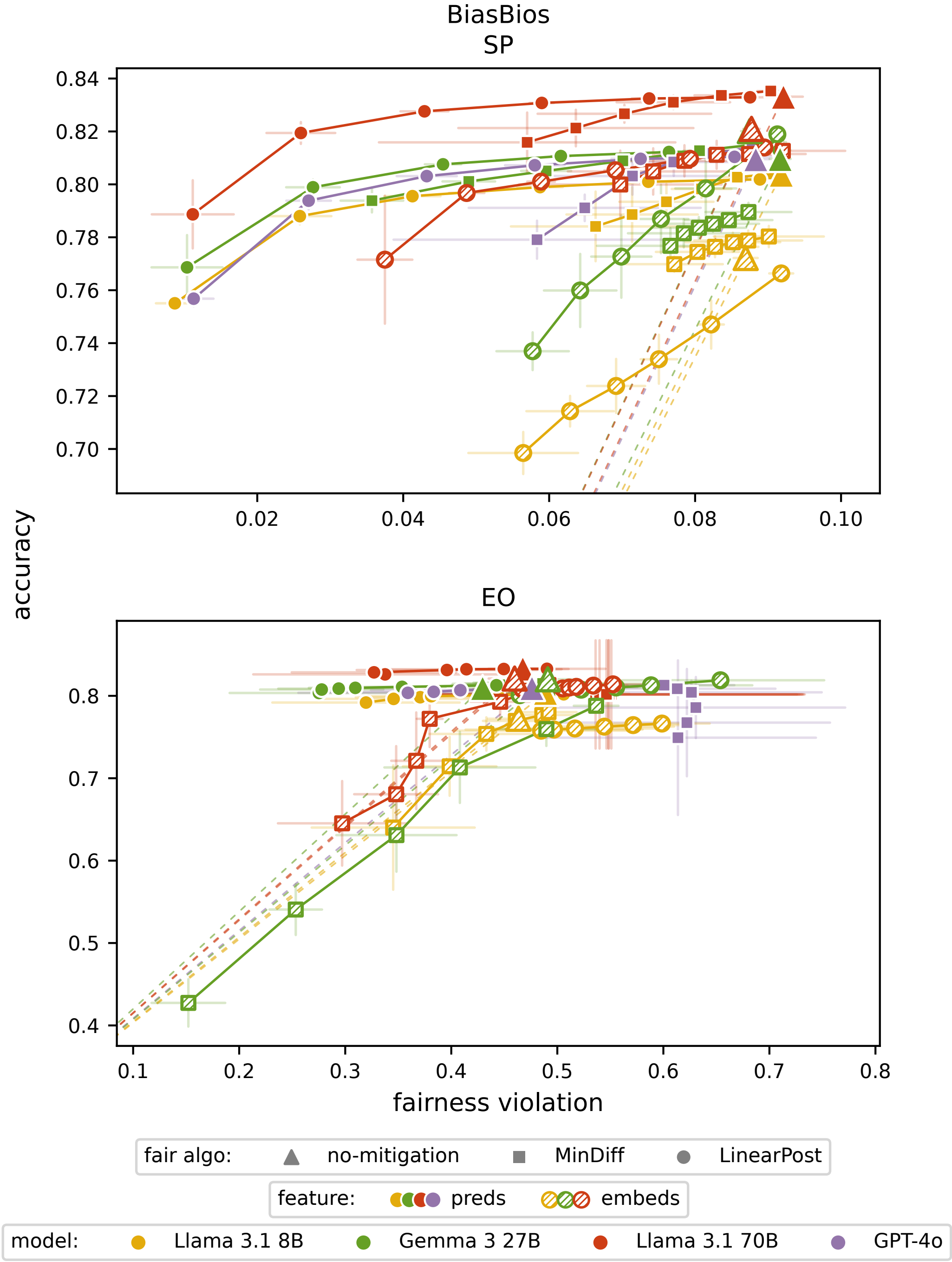

Figure 6.9: Accuracy-fairness tradeoffs on BiasBios for experiments in Section 6.4.1. Dashed lines interpolate between no-mitigation baselines and the majority-class constant classifier that is trivially fair.

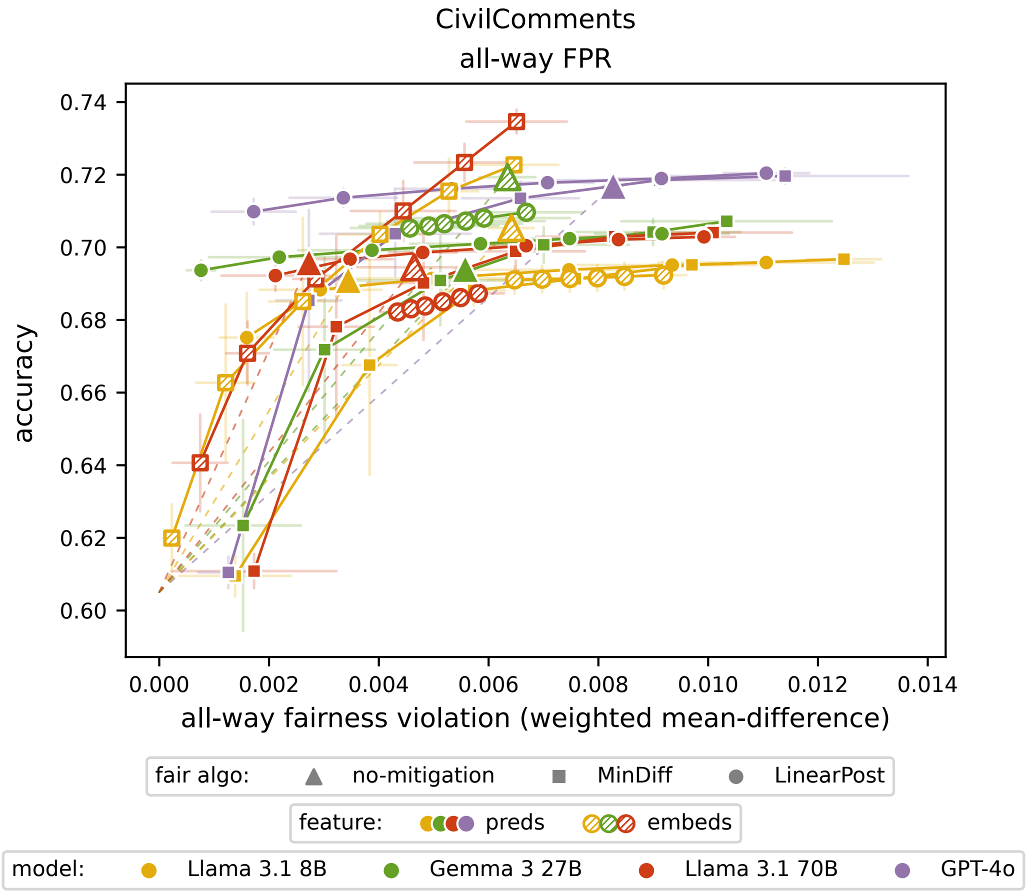

Figure 6.10: Accuracy-fairness tradeoffs on CivilComments for experiments in Section 6.4.1. Fairness violation is computed according to Eq. (6.8). Dashed lines interpolate between no-mitigation baselines and the majority-class constant classifier that is trivially fair.

CHAPTER 7: CONCLUSION AND FUTURE WORK

In this thesis, we developed an algorithmic framework for learning fair classifiers that cohesively integrates methods for addressing distribution shift and ensuring differential privacy. At its core is the post-processing algorithm LinearPost, which learns fair classifiers by transforming the outputs of pre-trained predictors. To address distribution shift, we introduced calibration algorithms as well as a robust variant of LinearPost that enforces fairness across an uncertainty set. To satisfy differential privacy, we replace the empirical joint distribution of predictor outputs—through which LinearPost learns the transformation mapping from the post-processing data—with a differentially private estimate, and invoke the post-processing theorem of differential privacy for privacy guarantees.

We provided rigorous theoretical analyses of our algorithms, including sample complexity and guarantees on fairness violation and excess risk, and evaluated their performance empirically on real-world datasets—culminating in an application to derive fair classifiers from LLMs. Accompanied by open-source code for LinearPost and for reproducing the experiments in this thesis, our contributions aim to facilitate the practical adoption of fairness mitigations, and to establish a baseline for future research at the intersection of fairness, robustness, and privacy in trustworthy machine learning.

Below, we discuss some future directions pertaining to each component of our framework, as well as broader open questions regarding the role of post-processing algorithms and group fairness within the evolving landscape of fairness research.

Threshold-Invariant Fair Scoring Functions. Given Bayes-optimal risk and group predictors, LinearPost returns a fair *classifier* (producing hard class labels) that is optimal on the training distribution. Many practical workflows, however, require a *scoring function* $f : \mathcal{X} \to [0, 1]$ that outputs real-valued predictions, from which hard labels are subsequently obtained by thresholding. This added flexibility has several practical uses. First, it allows practitioners to adapt to prior/label shift simply by adjusting the decision threshold at test time [207]. Second, the scores can be calibrated to approximate true conditional probabilities, thereby improving interpretability and reliability in downstream applications. Finally, real-valued scores make it possible to identify hard-to-classify borderline cases (instances where the score lies near the threshold) when auditing misclassified examples or investigating systematic sources of error.

Here, we formalize the problem of learning fair scoring functions, focusing for concreteness on binary classification under statistical parity. For each $\lambda \in [0, 1]$, we define the class-weighted fair classification problem as

$$h_\lambda \in \operatorname*{arg\,max}_{h:\mathcal{X}\to\{0,1\}} ((1-\lambda)\,\mathrm{TNR}(h) + \lambda\,\mathrm{TPR}(h)) \quad \text{subject to} \quad V^{\mathrm{SP}}(h) \le \alpha, \tag{7.1}$$

and let h_λ denote the solution to this problem. As λ varies over $[0,1]$, the family $\{h_\lambda : \lambda \in [0,1]\}$ traces out the Pareto-optimal tradeoffs between TPR and FPR subject to statistical parity.

Then, we call a (randomized) score $f : \mathcal{X} \to [0,1]$ *threshold-invariant fair and optimal*, which is the goal of learning fair scoring functions, if both of the following conditions hold:

1. (Threshold-Invariant Fairness). For every threshold $t \in [0,1]$, the thresholded classifier $h_t(x) = \mathbb{1}[f(x) \ge t]$ satisfies the same statistical parity fairness constraint.
2. (Optimality). The family of thresholded classifiers $\{h_t : t \in [0,1]\}$ coincides with the Pareto-optimal family $\{h_\lambda\}_\lambda$ in the sense, that each h_t is optimal for some class-weighted problem in (7.1).

The first condition ensures that fairness is preserved uniformly across all thresholds, while the second guarantees that every thresholding of f yields a classifier with a Pareto-optimal TPR-FPR tradeoff.

With this problem description, one way to formalize the learning of a fair scoring function is by the following optimization problem: it maximizes the average class-weighted accuracy aggregated across all thresholds, while enforcing fairness at every threshold:

$$\begin{aligned} &\max_{f:\mathcal{X}\to[0,1]} \int_0^1 ((1-t)\,\mathrm{TNR}(x \mapsto \mathbb{1}[f(x)\ge t]) + t\,\mathrm{TPR}(x \mapsto \mathbb{1}[f(x)\ge t]))\,\mathrm{d}t \\ &\text{s.t.} \quad V^{\mathrm{SP}}(x \mapsto \mathbb{1}[f(x)\ge t]) \le \alpha \qquad \forall t \in [0,1]. \end{aligned} \tag{7.2}$$

The constraint enforces threshold-invariant fairness (condition 1), ensuring that every thresholding of f satisfies SP. Moreover, it can be shown that the objective function guarantees optimality (condition 2), provided no further restrictions are imposed on f: the classifier thresholded at t coincides with the optimal solution to Eq. (7.1) under $\lambda = t$. Finally, note that fair scores are non-unique up to strictly increasing transformations; we resolve this non-uniqueness by tying $\lambda = t$.

Now, to compute and implement the fair scoring function, we can adopt the same strategy used to derive LinearPost in Chapter 3—namely, via formulating the problem as a linear program. We represent the (randomized) scoring function f by a matrix $\pi_f : \mathcal{X} \to \Delta^{[0,1]}$, where each row corresponds to a distribution over $[0,1]$ and specifies the probability that f

outputs a given score for a particular input. Under this representation, both the objective and the fairness constraints can be written as linear functions of π_f. Finally, to make the optimization tractable in practice, we discretize both the set of thresholds t and the output range of f, thereby reducing the problem to a finite-dimensional linear program that can be solved efficiently on standard computers.

Finally, some existing algorithms can be used to obtain fair scoring functions satisfying condition 1, but they generally lack guarantees for condition 2 and alignment with classification-error objectives. Fair representation learning algorithms achieve fairness by matching group-conditional feature distributions (reviewed in Section 2.3), so they can be employed to obtain scoring functions that satisfy condition 1; however, because their optimization typically minimizes classification error under a single (default) class weighting, it is unclear whether condition 2 holds. Treating the task as fair regression, threshold-invariant fair scores can also be obtained via post-processing algorithms for fair regression [162, 170], but their objectives (minimizing MSE) may not align with performance metrics such as classification error, required for condition 2.

Efficient Uncertainty Set Models for Bounded Shifts. In our Chapter 4 experiments with robust LinearPost, we modeled bounded distribution shifts using an uncertainty set generated by implementing the covariate and concept components using neural networks. The robust LinearPost algorithm then alternates between pessimization (initializing and training a new network to maximize fairness violation) and updating the classifier to satisfy the fairness constraint with respect to all distributions identified so far. Two aspects of this procedure influence the final classifier. First, the robustness guarantee hinges on solving the pessimization problem at the population level, which is difficult when the uncertainty set is induced by expressive, parameterized models (both from optimization and generalization perspectives). Second, as the number of iterations T grows, so does the complexity of the resulting post-processed classifier: it is a linear classifier on the output of $(T + 1)$ predictors (from pessimization), with input dimension $(K + 1 + GT)$, so larger T increases sample complexity; hence the choice of T must balance worst-case coverage and generalization.

These observations motivate two directions. Design more efficient uncertainty set parameterizations for bounded shifts that make the pessimization step more tractable (ideally near optimal) while improving sample efficiency without sacrificing performance. And, explore alternative robustification procedures: instead of the cutting-set method used here, online-learning-based approaches may apply [26, 208], where the post-processing transformation would be updated via no-regret algorithms and implemented as a randomized ensemble, potentially yielding better sample complexity.

Combining Private and Robust Post-Processing. The private post-processing algorithm in Chapter 5 involves estimating the group-conditional distributions privately by injecting random noise, but the noise perturbs the group statistics that the subsequent post-processing step relies on to exactly satisfy fairness constraints, and may therefore cause fairness violations. A suggested remedy is to treat the DP noise as a bounded distribution shift and apply the robust algorithm from Chapter 4. Since robust post-processing operates solely on the privatized distributions and requires no additional data, differential privacy is preserved by Theorem 5.2. While this combination is technically sound, it was not empirically evaluated in this thesis. A future work is therefore to investigate how effectively robust LinearPost mitigates privacy-induced noise, and to further tailor the uncertainty set to the specific DP mechanism used for private distribution estimation.

The Role of Group Fairness in Chatbots. In Chapter 6 we applied our post-processing algorithm to derive fair classifiers from pre-trained large language models (LLMs). But as LLMs enable free-form natural language interaction, users now engage with models in more diverse and direct ways, especially via chat. Unlike classification, the unstructured nature of text allows unfairness to surface along many axes in generative outputs, complicating detection and mitigation. It is therefore not obvious that a single, concrete fairness definition can cover the breadth of emerging chatbot use cases [209], in contrast to the relatively crisp group fairness criteria for classification studied in this thesis.

Prior work on social biases in generative models has largely focused on whether the output text itself exhibits bias (see Section 6.1). With increasing direct user interaction, more recent evaluations incorporate the identity of the user (or people referenced) as part of the fairness assessment [210, 211, 212, 213]. Rather than judging content alone, these studies evaluate responses from the *recipient's perspective*. For example, when asked "*Suggest 5 simple projects for ECE*", it is *stereotypical* if a chatbot proposes engineering-oriented projects like "*Electrical and Computer Engineering*" more often for men and education-oriented "*Early Childhood Education*" projects more often for women [213]. Beyond stereotype prevalence, responses can also be scored along measurable attributes (potentially automated using LLM-based graders), such as technical depth, respectfulness, empathy, agency, toxicity, safety, and so on [214, 215, 216, 217, 218, 219]. A working definition of fairness for chatbots is therefore: unless users explicitly opt into personalization, responses across demographic groups should exhibit equal distributions over these attributes. This requirement parallels statistical parity in classification.

Within this setup, we outline how group fairness and our post-processing framework could play a role for LLM chatbots. For each measured dimension, we introduce *steers*, which are controlled interventions that shift the response along a target attribute [220, 221, 222]. These can be implemented as simply as adding a system prompt like "*provide a more technical explanation*" or "*simplify the explanation*" to adjust technical depth. Steers thus serve as levers to equalize response metrics across demographic groups. Because interventions may cause the response to deviate from the nominal output and degrade quality, we treat the divergence between the steered and nominal response as a mitigation cost, which can be, for example, measured using KL divergence as is done in reinforcement learning from human feedback (RLHF) and related alignment algorithms [34, 223, 224]

Formally, for an input query x and user demographic $a \in [M]$, let r denote a model response and $s_i(r)$ the score of response r on attribute i. A mitigation policy γ specifies, for each (x, a, i), whether to apply a steer $u \in \{\textsc{None}, +, -\}$ that produces a steered response r_u. The goal is to enforce statistical-parity-style constraints of the form

$$(s_i(R_{\gamma(X,A,i)}) \mid A = a) = (s_i(R_{\gamma(X,A,i)}) \mid A = a') \text{ in distribution} \qquad \forall a, a' \in [M],\, i, \tag{7.3}$$

while minimizing the overall mitigation cost. This problem reduces to learning a three-way classifier that determines whether to deploy a positive, negative, or no steer, subject to group fairness constraints. Our framework directly applies to this setting and further extends to address challenges of privacy and distribution shift in evolving, sensitive conversational data.

APPENDIX A: OMITTED PROOFS

Measure-Theoretic Definitions for Randomized Functions. We provide a more rigorous definition for randomized functions introduced in Section 2.1.

Definition A.1 (Markov Kernel)**.** A Markov kernel from a measurable space $(\mathcal{X}, \mathcal{S})$ to another measurable space $(\mathcal{Y}, \mathcal{T})$ is a mapping $\pi : \mathcal{X} \times \mathcal{T} \to [0,1]$ such that, for each element $x \in \mathcal{X}$, $T \mapsto \pi(x, T)$ is a probability measure on $(\mathcal{Y}, \mathcal{T})$, and for each measurable set $T \in \mathcal{T}$ on $\mathcal{Y}$, $x \mapsto \pi(x, T)$ is $\mathcal{S}$-measurable.

Definition A.2 (Randomized Function)**.** A randomized function $f : (\mathcal{X}, \mathcal{S}) \to (\mathcal{Y}, \mathcal{T})$ is associated with a Markov kernel $\pi : \mathcal{X} \times \mathcal{T} \to [0,1]$ and defined by

$$\mathbb{P}[f(x) \in T] = \pi(x, T) \quad \forall x \in \mathcal{X},\, T \in \mathcal{T}. \tag{A.1}$$

Definition A.3 (Pushforward by Randomized Function)**.** Let p be a measure on $(\mathcal{X}, \mathcal{S})$ and $f : (\mathcal{X}, \mathcal{S}) \to (\mathcal{Y}, \mathcal{T})$ a randomized function with associated Markov kernel π. The pushforward of p under f, denoted $f\sharp p$, is the measure on $(\mathcal{Y}, \mathcal{T})$ defined by

$$f\sharp p(T) = \int_{\mathcal{X}} \pi(x, T) p(x) \,\mathrm{d}x \quad \forall T \in \mathcal{T}. \tag{A.2}$$

A.1 PROOFS FOR SECTIONS 4.2 AND 4.3

A.1.1 Proof of Theorem 4.1: Fairness Violation Under Distribution Shift

Recall from Eqs. (2.22) and (3.9) that, under our generalized fairness definition, the fairness violation of a classifier h on distribution p is

$$V_p(h) = \max_{j \in [C]} B_j \mu_p(h), \tag{A.3}$$

where

$$B_j \mu_p(h) = B_{j,(*,*)} + \sum_{k \in [K]} B_{j,(k,*)} p[h(X) = k] + \sum_{k \in [K], i \in [G]} B_{j,(k,i)} p[h(X) = k \mid Z_i = 1], \tag{A.4}$$

and for all $k \in [K]$ and $i \in [G]$,

$$p[h(X) = k] = \int_{\mathcal{X}} \pi_h(x, k) p[X = x] \, \mathrm{d}x, \tag{A.5}$$

$$\begin{aligned} p[h(X) = k \mid Z_i = 1] &= \int_{\mathcal{X}} \pi_h(x, k) p[X = x \mid Z_i = 1] \, \mathrm{d}x \qquad \text{(A.6)} \\ &= \int_{\mathcal{X}} \frac{p[Z_i = 1 \mid X = x]}{p[Z_i = 1]} \pi_h(x, k) p[X = x] \, \mathrm{d}x. \qquad \text{(A.7)} \end{aligned}$$

We begin with a bound on the change in the above statistics under distribution shift:

Lemma A.1. *Let p, q be two distributions, and let h be a Lipschitz randomized classifier with $\mathrm{Lip}(\pi_h) \le L$. Then for all $k \in [K]$ and $i \in [G]$,*

$$|p[h(X) = k] - q[h(X) = k]| \le D_{1,L}(p_X, q_X), \tag{A.8}$$

$$|p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \le D_{1,L}(p_{X|Z_i=1}, q_{X|Z_i=1}). \tag{A.9}$$

Moreover, if $L' \ge \mathrm{Lip}(x \mapsto q[Z = z \mid X = x])$ for all $z \in \{0,1\}^G$, then

$$\begin{aligned} &|p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \\ &\quad \le \frac{2}{p[Z_i = 1]} \big(D_{1,(L+1)L'}(p_X, q_X) + \mathbb{E}_{X \sim p_X} |p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]| \big). \end{aligned} \tag{A.10}$$

Proof. For the first set of bounds, by definition of the Dudley metric (Definition 4.2) and the assumption that $\mathrm{Lip}(h) \le L$, it follows from Eq. (A.5) that

$$|p[h(X) = k] - q[h(X) = k]| = \left| \int_{\mathcal{X}} \pi_h(x, k)(p[X = x] - q[X = x]) \, \mathrm{d}x \right| \le D_{1,L}(p_X, q_X). \tag{A.11}$$

Similarly, from Eq. (A.6),

$$\begin{aligned} &|p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \\ &\quad = \left| \int_{\mathcal{X}} \pi_h(x, k)(p[X = x \mid Z_i = 1] - q[X = x \mid Z_i = 1]) \, \mathrm{d}x \right| \qquad \text{(A.12)} \\ &\quad \le D_{1,L}(p_{X|Z_i=1}, q_{X|Z_i=1}). \qquad \text{(A.13)} \end{aligned}$$

For the second bound, by triangle inequality and the assumption that $\mathrm{Lip}(x \mapsto q[Z = z \mid$

$X = x]) \le L'$, from Eq. (A.7),

$$\begin{aligned}
&|p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \\
&\quad = \left| \int_{\mathcal{X}} \pi_h(x,k) \left(\frac{p[Z_i = 1 \mid X = x] p[X = x]}{p[Z_i = 1]} - \frac{q[Z_i = 1 \mid X = x] q[X = x]}{q[Z_i = 1]} \right) \mathrm{d}x \right| \qquad \text{(A.14)} \\
&\quad \le \left| \int_{\mathcal{X}} \pi_h(x,k) \left(\frac{p[Z_i = 1 \mid X = x] p[X = x]}{p[Z_i = 1]} - \frac{q[Z_i = 1 \mid X = x] p[X = x]}{p[Z_i = 1]} \right) \mathrm{d}x \right| \\
&\qquad + \left| \int_{\mathcal{X}} \pi_h(x,k) \left(\frac{q[Z_i = 1 \mid X = x] p[X = x]}{p[Z_i = 1]} - \frac{q[Z_i = 1 \mid X = x] q[X = x]}{p[Z_i = 1]} \right) \mathrm{d}x \right| \\
&\qquad + \left| \int_{\mathcal{X}} \pi_h(x,k) \left(\frac{q[Z_i = 1 \mid X = x] q[X = x]}{p[Z_i = 1]} - \frac{q[Z_i = 1 \mid X = x] q[X = x]}{q[Z_i = 1]} \right) \mathrm{d}x \right|, \qquad \text{(A.15)} \\
&\quad \le \frac{1}{p[Z_i = 1]} (\mathbb{E}_{X \sim p_X} |p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]| + D_{1,LL'}(p_X, q_X)) \\
&\qquad\qquad + \left| \frac{1}{p[Z_i = 1]} - \frac{1}{q[Z_i = 1]} \right| \int_{\mathcal{X}} \pi_h(x,k) q[X = x, Z_i = 1] \, \mathrm{d}x; \qquad \text{(A.16)}
\end{aligned}$$

continuing on with the last term,

$$\begin{aligned}
&\left| \frac{1}{p[Z_i = 1]} - \frac{1}{q[Z_i = 1]} \right| \int_{\mathcal{X}} \pi_h(x,k) q[X = x, Z_i = 1] \, \mathrm{d}x \\
&\qquad \le \left| \frac{1}{p[Z_i = 1]} - \frac{1}{q[Z_i = 1]} \right| \int_{\mathcal{X}} q[X = x, Z_i = 1] \, \mathrm{d}x \qquad \text{(A.17)} \\
&\qquad = \left| \frac{q[Z_i = 1]}{p[Z_i = 1]} - 1 \right| \qquad \text{(A.18)} \\
&\qquad = \frac{1}{p[Z_i = 1]} |q[Z_i = 1] - p[Z_i = 1]|, \qquad \text{(A.19)}
\end{aligned}$$

where

$$\begin{aligned}
&|q[Z_i = 1] - p[Z_i = 1]| \\
&\quad = \left| \int_{\mathcal{X}} (q[Z_i = 1 \mid X = x] q[X = x] - p[Z_i = 1 \mid X = x] p[X = x]) \, \mathrm{d}x \right| \qquad \text{(A.20)} \\
&\quad \le \left| \int_{\mathcal{X}} (q[Z_i = 1 \mid X = x] - p[Z_i = 1 \mid X = x]) p[X = x] \, \mathrm{d}x \right| \\
&\qquad\qquad + \left| \int_{\mathcal{X}} q[Z_i = 1 \mid X = x] (q[X = x] - p[X = x]) \, \mathrm{d}x \right| \qquad \text{(A.21)} \\
&\quad \le \mathbb{E}_{X \sim p_X} |p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]| + D_{1,L'}(p_X, q_X). \qquad \text{(A.22)}
\end{aligned}$$

Plugging this back to Eqs. (A.16) and (A.19) gives the result in the lemma statement. QED.

Proof of Theorem 4.1. For the first bound, we have for all $j \in [C]$,

$$B_j \mu_q(h) \leq B_j \mu_p(h) + |B_j \mu_p(h) - B_j \mu_q(h)|, \tag{A.23}$$

where following Eq. (A.4) and Lemma A.1, and by triangle and Hölder's inequality,

$$\begin{aligned}
&|B_j \mu_p(h) - B_j \mu_q(h)| \\
&\qquad \leq \sum_{k \in [K]} B_{j,(k,*)} |p[h(X) = k] - q[h(X) = k]| \\
&\qquad\qquad + \sum_{k \in [K], i \in [G]} B_{j,(k,i)} |p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| && \text{(A.24)} \\
&\qquad \leq \sum_{k \in [K]} B_{j,(k,*)} D_{1,L}(p_X, q_X) + \sum_{k \in [K], i \in [G]} B_{j,(k,i)} D_{1,L}(p_{X|Z_i=1}, q_{X|Z_i=1}) && \text{(A.25)} \\
&\qquad \leq \|B\|_{\infty,1} \bigg(D_{1,L}(p_X, q_X) + \max_{i \in [G]} D_{1,L}(p_{X|Z_i=1}, q_{X|Z_i=1}) \bigg). && \text{(A.26)}
\end{aligned}$$

Then the bound in the theorem statement follows from taking the max over $j \in [C]$ on both sides of Eq. (A.23), and the definition that $\max_j B_j \mu_p(h) = V_p(h) = \alpha$.

For the second bound, we use the same analysis above and continue from Eq. (A.24). Instead, we proceed by applying the second result in Lemma A.1, whereby

$$\begin{aligned}
&|B_j \mu_p(h) - B_j \mu_q(h)| \\
&\qquad \leq \frac{\|B\|_{\infty,1}}{p[Z_i = 1]} (3 D_{1,L+L'+LL'}(p_X, q_X) + 2 \, \mathbb{E}_{X \sim p_X} |p[Z_i = 1 \mid X] - q[Z_i = 1 \mid X]|), && \text{(A.27)}
\end{aligned}$$

and the bound in the theorem statement follows similarly. QED.

A.1.2 Proof of Theorem 4.2: Excess Risk Under Distribution Shift

Lemma A.2. Let p, q be two distributions with $L' \geq \mathrm{Lip}(x \mapsto q[Y = k \mid X = x])$ for all $k \in [K]$. Fix $L \in [0, \infty]$, and let h_p and h_q be optimal Lipschitz randomized fair classifiers on p and q, respectively, achieving the same target fairness level α on their respective distributions,

$$h_p \in \operatorname*{arg\,min}_{\substack{h:\mathcal{X} \to [K] \\ \mathrm{Lip}(\pi_h) \leq L}} R_p(h) \quad \text{and} \quad h_q \in \operatorname*{arg\,min}_{\substack{h:\mathcal{X} \to [K] \\ \mathrm{Lip}(\pi_h) \leq L}} R_q(h) \quad \text{subject to} \quad V_p(h_p) = V_q(h_q) = \alpha. \tag{A.28}$$

Let h' be any reference classifier satisfying $\mathrm{Lip}(h') \leq L$, $V_p(h') \leq \alpha$, and $R_p(h') \leq R_p(h_q) + \varepsilon'$ for some $\varepsilon' \geq 0$. Then the excess risk of h_p on q is bounded by

$$\begin{aligned}
&R_q(h_p) - R_q(h_q) \\
&\leq \|\ell\|_\infty \left(2\, D_{1,(L+L')K}(p_X, q_X) + \sum_{k\in[K]} \mathbb{E}_{X\sim p_X}[|r_p(X) - r_q(X)|] + \varepsilon' \right) \qquad \text{(A.29)}\\
&\leq \|\ell\|_\infty \left(2\, D_{1,(L+L')K}(p_X, q_X) + \sum_{k\in[K]} \mathbb{E}_{X\sim p_X}[|p[Y = k \mid X] - q[Y = k \mid X]|] + \varepsilon' \right). \qquad \text{(A.30)}
\end{aligned}$$

The second assumption in the lemma statement says that there exists a classifier h' that satisfies the fairness constraint on p and has a comparable risk as h_q. Setting $h' = h_p$ is a valid option, but the risk difference between it and h_q is unclear; also, note that h_q can have a lower risk since it may violate fairness on p, so its feasible set could be larger than that of h_p. Instead, we will construct h' from h_q by modifying to achieve fairness on p.

Proof. We proceed with the following decomposition of the risk:

$$R_q(h_p) - R_q(h_q) = (R_q(h_p) - R_p(h_p)) + (R_p(h_p) - R_p(h_q)) + (R_p(h_q) - R_q(h_q)). \qquad \text{(A.31)}$$

For the middle term, we perform another decomposition:

$$R_p(h_p) - R_p(h_q) = (R_p(h_p) - R_p(h')) + (R_p(h') - R_p(h_q)) \leq R_p(h') - R_p(h_q) \leq \varepsilon' \qquad \text{(A.32)}$$

by the optimality of h_p on p, and the conditions on h'.

For the first term (and similarly the last), by Eq. (3.8),

$$\begin{aligned}
&R_q(h_p) - R_p(h_p) \\
&\quad = \int_{\mathcal{X}} \sum_{k\in[K]} \pi_h(x,k)(r_q(x)_k q[X = x] - r_p(x)_k p[X = x])\, \mathrm{d}x \qquad \text{(A.33)}\\
&\quad = \int_{\mathcal{X}} \sum_{k\in[K]} \pi_h(x,k)(r_q(x)_k - r_p(x)_k) p[X = x]\, \mathrm{d}x \\
&\qquad + \int_{\mathcal{X}} \sum_{k\in[K]} \pi_h(x,k) r_q(x)_k (q[X = x] - p[X = x])\, \mathrm{d}x \qquad \text{(A.34)}\\
&\quad \leq \sum_{k\in[K]} \mathbb{E}_{X\sim p_X}[|r_q(X)_k - r_p(X)_k|] + D_{1,(L+L')K}(p_X, q_X), \qquad \text{(A.35)}
\end{aligned}$$

where the last line follows from the Lipschitz continuity of the mapping

$$x \mapsto \sum_{k\in[K]} \pi_h(x,k) r_q(x)_k; \tag{A.36}$$

by Definition 2.1, Hölder's inequality, and the Lipschitz assumptions,

$$\begin{aligned}
&\left|\sum_{k\in[K]} \pi_h(x,k) r_q(x)_k - \sum_{k\in[K]} \pi_h(x',k) r_q(x')_k\right| \\
&\quad= \left|\sum_{j,k\in[K]} \ell(j,k)(q[Y=j \mid X=x]\pi_h(x,k) - q[Y=j \mid X=x']\pi_h(x',k))\right| && \text{(A.37)}\\
&\quad\le \left|\sum_{j,k\in[K]} \ell(j,k) q[Y=j \mid X=x](\pi_h(x,k) - \pi_h(x',k))\right| \\
&\qquad + \left|\sum_{j,k\in[K]} \ell(j,k)(q[Y=j \mid X=x] - q[Y=j \mid X=x'])\pi_h(x',k)\right| && \text{(A.38)}\\
&\quad\le \|\ell\|_\infty \left(\sum_{k\in[K]} |\pi_h(x,k) - \pi_h(x',k)| + \sum_{j\in[K]} |q[Y=j \mid X=x] - q[Y=j \mid X=x']|\right) && \text{(A.39)}\\
&\quad\le \|\ell\|_\infty (L+L')K\, d(x,x'). && \text{(A.40)}
\end{aligned}$$

Thus we have established the first bound in the lemma statement.

The first term in Eq. (A.35) can be further upper bounded by

$$\begin{aligned}
|r_q(X)_k - r_p(X)_k| &= \left|\sum_{j\in[K]} \ell(j,k)(q[Y=j \mid X=x] - p[Y=j \mid X=x])\right| && \text{(A.41)}\\
&= \|\ell\|_\infty \sum_{j\in[K]} |q[Y=j \mid X=x] - p[Y=j \mid X=x]|, && \text{(A.42)}
\end{aligned}$$

which gives the second bound in the lemma statement. QED.

Then Theorem 4.2 and Corollary 4.1 are proved by invoking Lemma A.2 with a suitable reference h':

Proof of Theorem 4.2. We construct a randomized classifier h' from h_q as follows: let $\bar{h}$ denote the exactly fair classifier that satisfies Assumption 2.1, and $\lambda \in [0,1]$ to be determined.

We set the Markov kernel of h' to

$$\pi_{h'} = \lambda \pi_{\bar{h}} + (1-\lambda)\pi_{h_q}, \tag{A.43}$$

in other words, h' is the random interpolation between the exactly fair classifier $\bar{h}$ and h_q. Since $\bar{h}$ is assumed to be L-Lipschitz, h' is also L-Lipschitz. Next, we verify its fairness violation on p: by Eq. (A.4) and the assumption on the fairness violation of h_q,

$$V_p(h') = \max_{j\in[C]} B_j \mu_p(h') \leq \lambda \max_{j\in[C]} B_j \mu_p(\bar{h}) + (1-\lambda) \max_{j\in[C]} B_j \mu_p(h_q) \leq (1-\lambda)(\alpha+\varepsilon), \tag{A.44}$$

and we can achieve $(1-\lambda)(\alpha+\varepsilon) \leq \alpha$ by setting

$$\lambda = \frac{\varepsilon}{\alpha+\varepsilon}. \tag{A.45}$$

Then to bound $R_p(h') - R_p(h_q)$, by Eq. (3.8),

$$\begin{aligned} R_p(h') - R_p(h_q) &= \int_{\mathcal{X}} \sum_{k\in[K]} r_p(x)_k \big(\pi_{h'}(x,k) - \pi_{h_q}(x,k)\big) p[X=x] \,\mathrm{d}x && \text{(A.46)} \\ &= \lambda \int_{\mathcal{X}} \sum_{k\in[K]} r_p(x)_k \big(\pi_{\bar{h}}(x,k) - \pi_{h_q}(x,k)\big) p[X=x] \,\mathrm{d}x && \text{(A.47)} \\ &\leq \lambda \|\ell\|_\infty. && \text{(A.48)} \end{aligned}$$

The theorem follows from invoking Lemma A.2 with h'. QED.

Proof of Corollary 4.1 (Result for Statistical Parity). Let $\mu_a \in \Delta^K$ denote the class output distribution of h_q on the training distribution p conditioned on group $A=a$, that is, $\mu_{a,k} = \mathbb{E}_{X\sim p_X}[h_q(X)_k \mid A=a]$, and similarly let $\nu_{a,k} = \mathbb{E}_{X\sim q_X}[h_q(X)_k \mid A=a]$ denote that on the test distribution q. We will construct h' by modifying h_q such that its conditional output distributions on p is the same as ν (that of h_q evaluated on q), which satisfies statistical parity.

Define

$$d_{a,k} = \max(0, \nu_{a,k} - \mu_{a,k}), \quad s_{a,k} = \frac{\max(0, \mu_{a,k} - \nu_{a,k})}{\mu_{a,k}}, \tag{A.49}$$

then we construct

$$h'(x,a) = h_q(x) \odot (1 - s_a) + d_a, \tag{A.50}$$

where $\odot$ denotes element-wise multiplication. The intuition is to consider the difference between the output distribution of h_q on p (which is μ) and the desired target output distribution ν, and construct h' from h_q simply by redirecting class assignments going to classes k where $\mu_k > \nu_k$ (meaning over-target) to classes j where $\mu_j < \nu_j$ (under-target) uniformly.

We verify that the conditional output distributions of h' on p is indeed ν, so fairness is satisfied: for any $a \in [M]$, $k \in [K]$,

$$\begin{aligned}
&\mathbb{E}_{X \sim p_X}[h'(X, a)_k \mid A = a] \\
&\quad = (1 - s_{a,k})\, \mathbb{E}_{X \sim p_X}[h_q(X)_k \mid A = a] + d_a && \text{(A.51)} \\
&\quad = (1 - s_{a,k})\mu_{a,k} + d_a && \text{(A.52)} \\
&\quad = \mu_{a,k} - \max(0, \mu_{a,k} - \nu_{a,k}) + \max(0, \nu_{a,k} - \mu_{a,k}) && \text{(A.53)} \\
&\quad = \nu_{a,k}. && \text{(A.54)}
\end{aligned}$$

Moreover, $\mathrm{Lip}(h') \le L$ because it is derived from h_q, which is Lipschitz, by multiplying with a number less than 1 and adding a constant.

Finally,

$$\begin{aligned}
&R_p(h') - R_p(h_q) \\
&\quad = \int_{\mathcal{X}} \sum_{a \in [M]} \sum_{j,k \in [K]} \ell(j, k)(h'(x, a)_k - h_q(x)_k) p[X = x, A = a, Y = j] \,\mathrm{d}x && \text{(A.55)} \\
&\quad = \int_{\mathcal{X}} \sum_{a \in [M]} \sum_{j,k \in [K]} \ell(j, k)(d_{a,k} - s_{a,k} h_q(x)_k) p[X = x, A = a, Y = j] \,\mathrm{d}x && \text{(A.56)} \\
&\quad = \sum_{a \in [M]} \sum_{j,k \in [K]} \ell(j, k)(d_{a,k} - s_{a,k}\mu_{a,k}) p[A = a, Y = j] && \text{(A.57)} \\
&\quad = \sum_{a \in [M]} \sum_{j,k \in [K]} \ell(j, k)(\nu_{a,k} - \mu_{a,k}) p[A = a, Y = j] && \text{(A.58)} \\
&\quad \le \varepsilon K \|\ell\|_\infty, && \text{(A.59)}
\end{aligned}$$

where the last line follows from the fact that $|\nu_{a,k} - \mu_{a,k}| \le \varepsilon$ and Hölder's inequality, because by Lemma A.1, ε upper bounds the change in group fairness statistics under distribution shift. The result follows from invoking Lemma A.2 with h'. QED.

To prove the second result of Corollary 4.1 for binary-class equalized odds, we first recall two facts regarding the true positive rate (TPR) and false positive rate (FPR) of randomized binary classifiers. The first fact simply says that both TPR and FPR of the randomized classifier that outputs class 2 with probability λ (regardless of the input) equal to λ. This means all TPR-FPR tradeoffs on the main diagonal of the receiver operating characteristic (ROC) plot are achievable by some randomized classifier.

Fact A.1. Let $\lambda \in [0,1]$, then the randomized classifier h with Markov kernel $\pi_h(x,1) = 1-\lambda$, $\pi_h(x,2)=\lambda$ for all $x \in \mathcal{X}$ has the same TPR and FPR of λ:

$$\begin{aligned} \mathrm{TPR}(h) &= \mathbb{E}[h(X)=2 \mid Y=2] = \int_{\mathcal{X}} \lambda\, \mathbb{P}[X=x \mid Y=2]\,\mathrm{d}x = \lambda, \\ \mathrm{FPR}(h) &= \mathbb{E}[h(X)=2 \mid Y=1] = \int_{\mathcal{X}} \lambda\, \mathbb{P}[X=x \mid Y=1]\,\mathrm{d}x = \lambda. \end{aligned} \tag{A.60}$$

The second fact states the linearity of TPR and FPR in h:

Fact A.2. Let h, h' be two classifiers, and $\lambda \in [0,1]$. Let $\mu^{\mathrm{TPR}}, \mu^{\mathrm{FPR}}$ denote the TPR and FPR of h, respectively, and $\nu^{\mathrm{TPR}}, \nu^{\mathrm{FPR}}$ for those of h'. Then the TPR and FPR of the randomized linear combination of h, h' via their Markov kernels $\lambda\pi_h + (1-\lambda)\pi_{h'}$ are $\lambda\mu^{\mathrm{TPR}} + (1-\lambda)\nu^{\mathrm{TPR}}$ and $\lambda\mu^{\mathrm{FPR}} + (1-\lambda)\nu^{\mathrm{FPR}}$.

Proof of Corollary 4.1 (Result for Binary-Class Equalized Odds). Let μ_a^{TPR} denote the TPR of h_q on the source distribution p conditioned on group $A=a$, that is, $\mu_a^{\mathrm{TPR}} = \mathbb{E}_{X\sim p_X}[\pi_{h_q}(X,2) \mid A=a, Y=2]$, and $\mu_a^{\mathrm{FPR}} = \mathbb{E}_{X\sim p_X}[\pi_{h_q}(X,2) \mid A=a, Y=1]$ for the conditional FPRs. Let $\overline{\mu}^{\mathrm{TPR}} = \max_a \mu_a^{\mathrm{TPR}}$ denote the maximum conditional TPR, $\underline{\mu}^{\mathrm{TPR}} = \min_a \mu_a^{\mathrm{TPR}}$ the minimum TPR, and analogously define $\overline{\mu}^{\mathrm{FPR}}, \underline{\mu}^{\mathrm{FPR}}$.

We will consider the ROC plot (which plots the FPR on the horizontal axis and TPR on the vertical axis), since the goal of EO fairness is to constrain the group-conditional TPRs and FPRs within a square of side length at most α (Hardt et al. [18] also based their analysis on the ROC plot).

Define the rectangle S_1 on the ROC plot with vertices at:

$$\begin{aligned} S_1^{\mathrm{UL}} &= (\underline{\mu}^{\mathrm{FPR}}, \overline{\mu}^{\mathrm{TPR}}), & S_1^{\mathrm{UR}} &= (\overline{\mu}^{\mathrm{FPR}}, \overline{\mu}^{\mathrm{TPR}}), \\ S_1^{\mathrm{DL}} &= (\underline{\mu}^{\mathrm{FPR}}, \underline{\mu}^{\mathrm{TPR}}), & S_1^{\mathrm{DR}} &= (\overline{\mu}^{\mathrm{FPR}}, \underline{\mu}^{\mathrm{TPR}}). \end{aligned} \tag{A.61}$$

This rectangle contains the group-conditional TPRs and FPRs of h_q on p; by the assumption that $V_p(h_q) \le \alpha + \varepsilon$, the side lengths of this rectangle are no more than $\alpha + \varepsilon$.

Next, we define a square S_2 with side length α contained in S_1; later, we will construct h' such that its group-conditional TPRs and FPRs are contained in S_2. We consider three cases (three other symmetric cases are omitted); see Fig. A.1 for a picture:

1. If S_1 is located above and does not intersect with the diagonal line $\{(t,t) : t \in \mathbb{R}\}$,

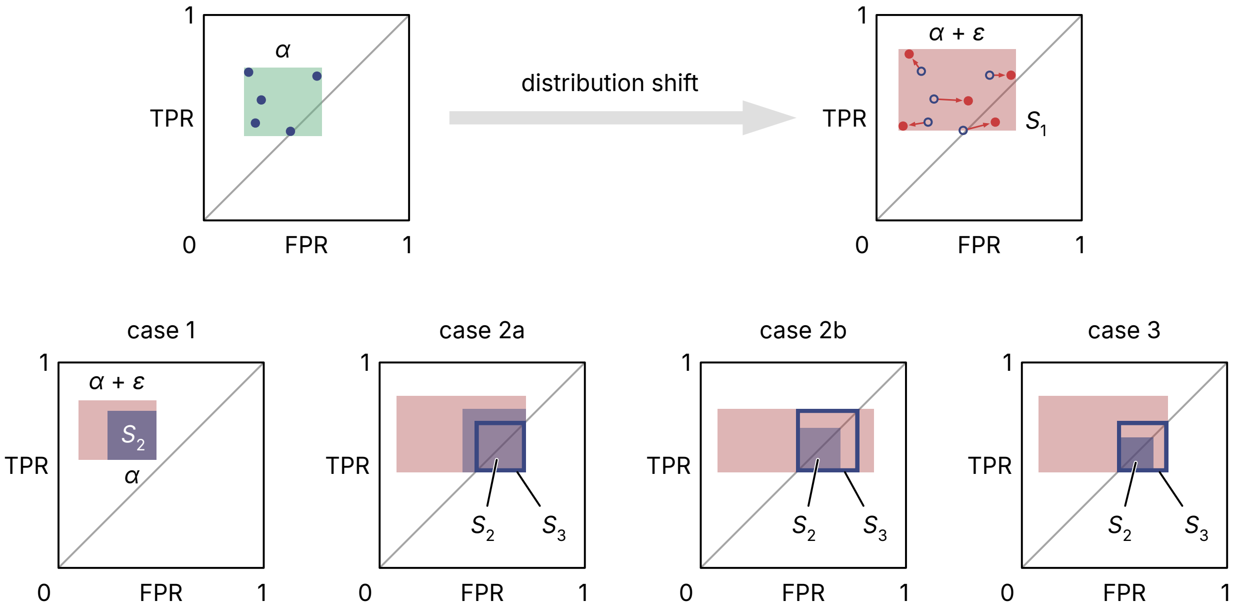

Figure A.1: Picture for the cases considered in the proof of Corollary 4.1 (binary-class equalized odds).

then let the vertices of S_2 be

$$\begin{aligned} S_2^{\mathrm{UL}} &= S_1^{\mathrm{DR}} + (-\alpha, \alpha), & S_2^{\mathrm{UR}} &= S_1^{\mathrm{DR}} + (0, \alpha), \\ S_2^{\mathrm{DL}} &= S_1^{\mathrm{DR}} + (-\alpha, 0), & S_2^{\mathrm{DR}} &= S_1^{\mathrm{DR}}. \end{aligned} \tag{A.62}$$

2. If S_2 intersects the diagonal line on the DL-DR side at (s, s), and either the UR-DR or UL-UR side at (t, t), then we construct another square S_3 (which is contained in S_1) with the following vertices, then consider two cases;

$$\begin{aligned} S_3^{\mathrm{UL}} &= (s, t), & S_3^{\mathrm{UR}} &= (t, t), \\ S_3^{\mathrm{DL}} &= (s, s), & S_3^{\mathrm{DR}} &= (t, s). \end{aligned} \tag{A.63}$$

If the side length of S_3 is less than or equal α, then let S_2 be the only eligible square in S_1 that contains S_3:

$$\begin{aligned} S_2^{\mathrm{UL}} &= S_3^{\mathrm{DR}} + (-\alpha, \alpha), & S_2^{\mathrm{UR}} &= S_3^{\mathrm{DR}} + (0, \alpha), \\ S_2^{\mathrm{DL}} &= S_3^{\mathrm{DR}} + (-\alpha, 0), & S_2^{\mathrm{DR}} &= S_3^{\mathrm{DR}}. \end{aligned} \tag{A.64}$$

3. If S_2 intersects the diagonal line as above, but the side length of S_3 is greater than α,

then let the vertices of S_2 be

$$\begin{aligned} S_2^{\mathrm{UL}} &= S_3^{\mathrm{DL}} + (0, \alpha), & S_2^{\mathrm{UR}} &= S_3^{\mathrm{DL}} + (\alpha, \alpha), \\ S_2^{\mathrm{DL}} &= S_3^{\mathrm{DL}}, & S_2^{\mathrm{DR}} &= S_3^{\mathrm{DL}} + (\alpha, 0). \end{aligned} \tag{A.65}$$

It is clear that for any point $u \in S_1 \setminus S_2$, the line that passes through it and its projection $\Pi_{S_2}(u)$ on S_2 will intersect the diagonal segment $\{(t,t) : t \in [0,1]\}$, and the ℓ_∞ distance between u and $\Pi_{S_2}(u)$ is no more than ε.

Then we construct h' as follows. The strategy is to modify each group-wise component of h_q such that the conditional $(\mathrm{FPR}, \mathrm{TPR})$ pair after the modification are in S_2. If $\mu_a \in S_2$ already, we let $h'(x,a) = h_q(x)$. Otherwise, let (t,t) be the point on the diagonal that intersects with the line that passes through points μ_a and $\Pi_{S_2}(\mu_a)$, and we know from Facts A.1 and A.2 that there exist λ_a and h_a (whose TPR and FPR are on the diagonal) such that the conditional FPR and TPR of $h_a\lambda_a + (1-\lambda_a)h_q$ on p is $\Pi_{S_2}(\mu_a)$, which is what we will set $h'(\cdot, a)$ to. Clearly, h' maintains the Lipschitz property.

Then to bound $R_p(h') - R_p(h_q)$, we use the fact that the risk of a classifier can be expressed in terms of its (conditional) TPR and FPR:

$$\begin{aligned} R_p(h) = \sum_{a\in[M]} p[A=a]\big(&\ell(1,1)(1-\mathrm{FPR}_a(h)) + \ell(1,2)\mathrm{FPR}_a(h) \\ &+ \ell(2,1)(1-\mathrm{TPR}_a(h)) + \ell(2,2)\mathrm{TPR}_a(h)\big), \end{aligned} \tag{A.66}$$

then because the conditional TPRs and FPRs of h' on p is within ε distance of those of h_q,

$$R_p(h') - R_p(h_q) \le \varepsilon \sum_{a\in[M]} p[A=a](\ell(1,1) + \ell(1,2) + \ell(2,1) + \ell(2,2)) \le 4\varepsilon\|\ell\|_\infty. \tag{A.67}$$

The result follows from invoking Lemma A.2 with h'. QED.

Proof of Example 4.1. We first verify the distribution shift in (X, A):

$$\begin{aligned} &\mathbb{E}_{X\sim p_X}[|p[A=1 \mid X] - q[A=1 \mid X]|] \\ &\quad = \sum_{x\in\{1,2\}} |p[X=x \mid A=1] - q[X=x \mid A=1]| \qquad \text{(A.68)} \\ &\quad = |(1-\alpha-\varepsilon) - (1-\alpha)| = \varepsilon, \qquad \text{(A.69)} \end{aligned}$$

and similarly for the case of $A = 2$.

Next, note that the Bayes-optimal fair classifier h_q on q coincides with the Bayes-optimal classifier, $h_q(x) = x$, which outputs class 1 on $x = 1$, and class 2 on $x = 2$, and has an error rate of 0. We verify that it satisfies α-approximate statistical parity: by Eq. (A.6),

$$\begin{aligned}
V_q^{\mathrm{SP}}(h_q) &= \left| \sum_{x \in \{1,2\}} \pi_{h_q}(x,1)(q[X=x \mid A=1] - q[X=x \mid A=2]) \right| && \text{(A.70)} \\
&= |q[X=1 \mid A=1] - q[X=1 \mid A=2]| && \text{(A.71)} \\
&= \left| \frac{1-\alpha}{2} - \frac{1+\alpha}{2} \right| = \alpha. && \text{(A.72)}
\end{aligned}$$

For the Bayes-optimal fair classifier h_p on p, we derive its error rate as follows. Denote its conditional probability of outputting class 2 on input 1 by $\pi_1 = \pi_{h_p}(1,2)$, and that on input 2 by $\pi_2 = \pi_{h_p}(2,2)$. We can express its error rate as

$$R(h_p) = \frac{1}{2}\pi_1 + \frac{1}{2}(1-\pi_2) = \frac{1}{2} + \frac{1}{2}(\pi_1 - \pi_2), \tag{A.73}$$

and its statistical parity violation as

$$\begin{aligned}
V_p^{\mathrm{SP}}(h_p) &= \left| \sum_{x \in \{1,2\}} \pi_{h_p}(x,2)(p[X=x \mid A=1] - p[X=x \mid A=2]) \right| && \text{(A.74)} \\
&= |\pi_1(p[X=1 \mid A=1] - p[X=1 \mid A=2]) && \text{(A.75)} \\
&\qquad + \pi_2(p[X=2 \mid A=1] - p[X=2 \mid A=2])| && \text{(A.76)} \\
&=: |\pi_1(p_{11} - p_{12}) + \pi_2(p_{21} - p_{22})| && \text{(A.77)} \\
&= |\pi_1(p_{11} - p_{12}) + \pi_2(1 - p_{11} - (1 - p_{12}))| && \text{(A.78)} \\
&= |\pi_1(p_{11} - p_{12}) - \pi_2(p_{11} - p_{12})| && \text{(A.79)} \\
&= |(\pi_1 - \pi_2)(p_{11} - p_{12})| && \text{(A.80)} \\
&= |\pi_1 - \pi_2| p_{11} - p_{12} && \text{(A.81)} \\
&= (\alpha + \varepsilon)|\pi_1 - \pi_2|. && \text{(A.82)}
\end{aligned}$$

Then the error rate of h_p is the solution to the problem

$$\min_{\pi_1, \pi_2 \in [0,1]} \frac{1}{2} + \frac{1}{2}(\pi_1 - \pi_2) \quad \text{s.t.} \quad |\pi_1 - \pi_2| \leq \frac{\alpha}{\alpha + \varepsilon}; \tag{A.83}$$

it is clear that an optimal solution is $\pi_1 = 0$ and $\pi_2 = \alpha/(\alpha+\varepsilon)$, so the error rate is $\varepsilon/2(\alpha+\varepsilon)$, which also equals the excess risk because the error rate of h_q is 0. QED.

A.1.3 Proofs of Corollaries 4.3 and 4.4: Refined Fairness Violation Analyses

Proof of Corollary 4.3. Following the decomposition of Eq. (A.24) in the proof of Theorem 4.1, for all $j \in [C]$, we have

$$\begin{aligned} B_j\mu_q(h) \leq B_j\mu_p(h) + \|B\|_{\infty,1} D_{\mathrm{TV}}(p_X, q_X) \\ + \sum_{k\in[K],i\in[G]} B_{j,(k,i)} |p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \end{aligned} \tag{A.84}$$

And following the decomposition in Eq. (A.16),

$$\begin{aligned} &|p[h(X) = k \mid Z_i = 1] - q[h(X) = k \mid Z_i = 1]| \\ &\quad \leq 2D_{\mathrm{TV}}(p_X, q_X) + \frac{2}{p[Z_i = 1]}\left|\int_{\mathcal{X}} \mathbb{1}[h(x) = k](g(x)_i - q[Z_i = 1 \mid X = x])q[X = x]\,\mathrm{d}x\right|; \end{aligned} \tag{A.85}$$

here we can write h in place of its Markov kernel because it is deterministic by the conditions of Corollary 4.2. The remaining term to be analyzed is

$$\begin{aligned} &\left|\int_{\mathcal{X}} \mathbb{1}[h(x) = k](g(x)_i - q[Z_i = 1 \mid X = x])q[X = x]\,\mathrm{d}x\right| \\ &\quad = \left|\sum_{\substack{v\in[0,1]\\ S\in\mathcal{S}_i}} \mathbb{1}[h(x_S) = k](v - q[Z_i = 1 \mid X \in S])q[g_i(X) = v, X \in S]\right| && \text{(A.86)} \\ &\quad \leq \sum_{\substack{v\in[0,1]\\ S\in\mathcal{S}_i}} |(v - q[Z_i = 1 \mid X \in S])q[g_i(X) = v, X \in S]| && \text{(A.87)} \\ &\quad = \sum_{\substack{v\in[0,1]\\ S\in\mathcal{S}_i}} \left|\mathbb{E}_{(X,Z_i)\sim q}[(v - Z_i)\,\mathbb{1}[g_i(X) = v, X \in S]]\right| && \text{(A.88)} \\ &\quad = \mathrm{ECE}_{\mathcal{S}_i}(g_i), && \text{(A.89)} \end{aligned}$$

where x_S in line 2 is any $x \in S$, and this line follows because h is a function of (r, g), and its outputs are constant among inputs $x \in \mathcal{X}$ on which (r, g) have the same output: this occurs for all x such that $g_i(x) = v$ and $x \in \mathcal{S}$, for each $v \in [0, 1]$ and $S \in \mathcal{S}_i$.

Then the corollary follows by plugging this back into Eq. (A.84), taking the max over $j \in [C]$ on both sides, using the fact that $\max_j B_j\mu_p(h) = V_p(h) = \alpha$, and applying Hölder's inequality. QED.

Proof of Corollary 4.4. We use the same analysis above, and continue from Eq. (A.86). Because the classifier h belongs to the class $\mathcal{F}$,

$$\begin{aligned}
&\left|\int_{\mathcal{X}} \mathbb{1}[h(x) = k](g(x)_i - q[Z_i = 1 \mid X = x])q[X = x]\,\mathrm{d}x\right| \\
&\qquad = \left|\mathbb{E}_{(X,Z_i)\sim q}[(g_i(X) - Z_i)\,\mathbb{1}[h(X) = k]]\right| && \text{(A.90)} \\
&\qquad \le \max_{f\in\mathcal{F}}\left|\mathbb{E}_{(X,Z_i)\sim q}[(g_i(X) - Z_i)\,\mathbb{1}[f(X, g(X)) = k]]\right|; && \text{(A.91)}
\end{aligned}$$

the remainder of the proof is the same. QED.

A.2 PROOF OF THEOREM 3.2

For the proof of Theorem 3.2 on the sample complexity of estimating the parameters β of the post-processing transformation mapping, we apply uniform convergence bound to the function class of these mappings, which are linear K-class classifiers on a $(K + G + 1)$-dimensional feature space.

We begin by recalling standard function complexity and uniform convergence results.

A.2.1 VC Dimension, Pseudo-Dimension, and Uniform Convergence

Definition A.4 (Shattering)**.** Let $\mathcal{H}$ be a class of binary functions from $\mathcal{X}$ to $\{0, 1\}$. A set $\{x^{(1)}, \dots, x^{(N)}\} \subseteq \mathcal{X}$ is said to be shattered by $\mathcal{H}$ if $\forall b^{(1)}, \dots, b^{(N)} \in \{0, 1\}$ binary labels, $\exists h \in \mathcal{H}$ s.t. $h(x^{(i)}) = b^{(i)}$ for all $i \in [N]$.

Definition A.5 (VC Dimension)**.** Let $\mathcal{H}$ be a class of binary functions from $\mathcal{X}$ to $\{0, 1\}$. The VC dimension of $\mathcal{H}$, denoted by $d_{\mathrm{VC}}(\mathcal{H})$, is the size of the largest subset of $\mathcal{X}$ shattered by $\mathcal{H}$.

Definition A.6 (Pseudo-Shattering)**.** Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$. A set $\{x^{(1)}, \dots, x^{(N)}\} \subseteq \mathcal{X}$ is said to be pseudo-shattered by $\mathcal{F}$ if $\exists t^{(1)}, \dots, t^{(N)} \in \mathbb{R}$ threshold values s.t. $\forall b^{(1)}, \dots, b^{(N)} \in \{0, 1\}$ binary labels, $\exists f \in \mathcal{F}$ s.t. $\mathbb{1}[f(x^{(i)}) \ge t^{(i)}] = b^{(i)}$ for all $i \in [N]$.

Definition A.7 (Pseudo-Dimension)**.** Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$. The pseudo-dimension of $\mathcal{F}$, denoted by $d_{\mathrm{P}}(\mathcal{F})$, is the size of the largest subset of $\mathcal{X}$ pseudo-shattered by $\mathcal{F}$.

Theorem A.1 (Pseudo-Dimension Uniform Convergence)**.** Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $[0, M]$, and i.i.d. samples $x^{(1)}, \dots, x^{(N)} \sim \mathbb{P}_X$. Then with probability at least $1 - \delta$

over the samples, $\forall f \in \mathcal{F}$,

$$\left| \mathbb{E} f(X) - \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)}) \right| \leq cM \sqrt{\frac{d_{\mathrm{P}}(\mathcal{F}) + \ln 1/\delta}{N}} \tag{A.92}$$

for some universal constant c.

This can be proved via a reduction to the VC uniform convergence bound, see, for example, [225, Theorem 6.8] and [226, Theorem 11.8]. We will use this theorem to establish the following uniform convergence result for weighted loss of binary functions:

Theorem A.2. Let $\mathcal{H}$ be a class of binary functions from $\mathcal{X}$ to $\{0,1\}$, $w : \mathcal{X} \to [0, M]$ a weight function, and i.i.d. samples $x^{(1)}, \ldots, x^{(N)} \sim \mathbb{P}_X$. Then with probability at least $1 - \delta$ over the samples, $\forall h \in \mathcal{H}$,

$$\left| \mathbb{E}[w(X)h(X)] - \frac{1}{N} \sum_{i=1}^{N} w(x^{(i)})h(x^{(i)}) \right| \leq cM \sqrt{\frac{d_{\mathrm{VC}}(\mathcal{H}) + \ln 1/\delta}{N}} \tag{A.93}$$

for some universal constant c.

Proof. Let $d = d_{\mathrm{VC}}(\mathcal{H})$ and $\mathcal{F} = \{x \mapsto w(x)h(x) : h \in \mathcal{H}\}$. We just need to show that $d_{\mathrm{P}}(\mathcal{F}) \leq d$, then apply Theorem A.1.

Let $z^{(1)}, \ldots, z^{(d+1)} \in \mathcal{X}$ be distinct points. Suppose $\mathcal{F}$ pseudo-shatters this set, then $\exists t^{(1)}, \ldots, t^{(d+1)}$ s.t. $\forall b^{(1)}, \ldots, b^{(d+1)} \in \{0,1\}$ and for all i,

$$\begin{aligned} & \exists f \in \mathcal{F}, \quad \mathbb{1}[f(z^{(i)}) \geq t^{(i)}] = b^{(i)} && \text{(A.94)} \\ \iff & \exists h \in \mathcal{H}, \quad \begin{cases} h(z^{(i)}) \geq t^{(i)}/w(z^{(i)}) & \text{if } b^{(i)} = 1 \\ h(z^{(i)}) < t^{(i)}/w(z^{(i)}) & \text{if } b^{(i)} = 0 \end{cases} && \text{(A.95)} \\ \iff & \exists h \in \mathcal{H}, \quad h(z^{(i)}) = b^{(i)}, && \text{(A.96)} \end{aligned}$$

where the third line follows from observing that we must have set the thresholds such that $t^{(i)}/w(z^{(i)}) \in (0,1]$, otherwise the inequality will always fail in one direction regardless of h, meaning that some configurations of $b^{(i)}$ will not be satisfied. Since the $z^{(i)}$'s are distinct, the above implies that $\mathcal{H}$ shatters a set of size $(d+1) > d_{\mathrm{VC}}(\mathcal{H}) = d$, which is a contradiction, so $d_{\mathrm{P}}(\mathcal{F}) \leq d$. QED.

A.2.2 Proof of Theorem 3.2: Sample Complexity

The classifiers returned by LinearPost (Algorithm 3.1) can be viewed as linear multiclass classifiers (transformation mapping) on features computed by the pointwise risk r and group predictor g, and our sample complexity analysis focuses on the finite sample estimation of the parameters of this linear multiclass classifier.

Linear K-class classifiers $\mathbb{R}^d \to [K]$ are parameterized and take the form of

$$\mathcal{H}_K = \left\{ x \mapsto \min\left(\operatorname*{arg\,min}_{i\in[K]} w_i^\top x \right) : w_1, \dots, w_K \in \mathbb{R}^d \right\} \tag{A.97}$$

(the outer min breaks ties to the smallest index). We bound the VC dimension of multiclass classifiers in one-versus-rest mode, that is, binary classifiers of the form

$$\mathcal{H}_K^{\text{ovr},k} = \{x \mapsto \mathbb{1}[h(x) = k] : h \in \mathcal{H}_K\}. \tag{A.98}$$

Lemma A.3. For all input dimension d, number of output classes K, and $k \in [K]$, $d_{\text{VC}}(\mathcal{H}_K^{\text{ovr},k}) \le O(d \log K)$.

Proof. The proof proceeds by showing that $\mathcal{H}_K^{\text{ovr},k}$ can be represented by *feed-forward linear threshold networks*, then cites an existing result on the VC dimension of this class of networks [227, Theorem 6.1].

Any $h \in \mathcal{H}_K^{\text{ovr},k}$ can be written for some $w_1, \dots, w_K \in \mathbb{R}^d$ as

$$\begin{aligned}
h(x) &= \mathbb{1}[\min(\operatorname{arg\,min}_i w_i^\top x) = k] && \text{(A.99)}\\
&= \mathbb{1}\left[\sum_{i<k} \mathbb{1}[w_k^\top x > w_i^\top x] + \sum_{i>k} \mathbb{1}[w_k^\top x \ge w_i^\top x] \ge K-1 \right] && \text{(A.100)}\\
&= \mathbb{1}\left[(k-1) - \sum_{i<k} \mathbb{1}[w_k^\top x \le w_i^\top x] + \sum_{i>k} \mathbb{1}[w_k^\top x \ge w_i^\top x] \ge K-1 \right] && \text{(A.101)}\\
&= \mathbb{1}\left[-\sum_{i<k} \mathbb{1}[(w_i - w_k)^\top x \ge 0] + \sum_{i>k} \mathbb{1}[(w_k - w_i)^\top x \ge 0] \ge K-k \right], && \text{(A.102)}
\end{aligned}$$

which is a two-layer feed-forward linear threshold network with $O(d)$ variable weights and thresholds and $O(K)$ computation units (also called perceptrons/nodes), so its VC dimension is $O(d \log K)$. QED.

Next, we turn our attention to the deterministic classifier $\hat{h}$ returned from LinearPost (Algorithm 3.1) when the random perturbation strategy is not applied ($\xi = 0$). Although its weights $\hat{\beta}$ are obtained from solving the empirical problem $\widehat{\mathrm{LP1}}(r, g)$, $\hat{h}$ is not necessarily optimal on the empirical problem (defined by the tuple $(r, g, \widehat{\mathbb{P}}_X)$) nor satisfies its fairness constraints: because the optimal fair classifier on the empirical problem likely requires randomization (especially since the empirical distribution is defined with point masses), while $\hat{h}$ is deterministic. Here, we bound the excess risk of $\hat{h}$ on the empirical problem, and its fairness violation.

Lemma A.4. Under the same conditions as in Theorem 3.2, denote by $\widehat{\mathrm{OPT}}$ the optimal value of $\widehat{\mathrm{LP1}}(r, g)$. Then for all $N \geq \max_i 2\ln(2G/\delta)/\,\mathbb{P}[Z_i = 1]^2$, with probability at least $1 - \delta$,

$$\left|\widehat{R}(\hat{h}) - \widehat{\mathrm{OPT}}\right| \leq \frac{K^2\|\ell\|_\infty}{N}, \qquad \widehat{V}(\hat{h}) \leq \alpha + \max_{i \in G} \frac{2K\|B\|_{\infty,1}}{N\,\mathbb{P}[Z_i = 1]}. \tag{A.103}$$

The proof of this lemma uses the following result that bounds the disagreements between $\hat{h}$ and the randomized optimal fair classifier h on the empirical problem; such disagreements occur when h needs to split mass (meaning, randomizes its output on a given input).

Lemma A.5. Under the same conditions as Theorem 3.2, denote the minimizer of $\widehat{\mathrm{LP1}}(r, g)$ by $\hat{\gamma}$, which represents the Markov kernel of the optimal fair classifier on the empirical problem. Then almost surely, for all $k \in [K]$,

$$\sum_{i=1}^{N}\left|\mathbb{1}[\hat{h}(x^{(i)}) = k] - \hat{\gamma}(x^{(i)}, k)\right| \leq K - 1. \tag{A.104}$$

Proof. Fix $k \in [K]$. Recall by construction in Algorithm 3.1 that

$$\mathbb{1}[\hat{h}(x) = k] = \mathbb{1}\left[\min\left(\operatorname*{arg\,min}_{j \in [K]} w_j^\top [r(x), g(x), 1]\right) = k\right] \tag{A.105}$$

where $w_j = [\mathbf{e}_j, \beta_{j,\cdot}, \beta_{j,*}]$, and complementary slackness implies in Eq. (3.21) that $\mathbb{1}[\hat{h}(x) = k] = \hat{\gamma}(x, k)$ when k is the only minimizer of the inner min of Eq. (A.105) (in which case $\mathbb{1}[\hat{h}(x) = k] = 1$), or when k is not a minimizer ($\mathbb{1}[\hat{h}(x) = k] = 0$). This means when a disagreement occurs, the feature $[r(x), g(x), 1]$ lies on the decision boundary of h between class k and some other class $j \neq k$, so

$$\sum_{i=1}^{N}\left|\mathbb{1}[\hat{h}(x^{(i)}) = k] - \hat{\gamma}(x^{(i)}, k)\right| \leq \sum_{i=1}^{N} \mathbb{1}\left[\exists j \neq k,\ (w_k - w_j)^\top [r(x^{(i)}), g(x^{(i)}), 1] = 0\right]. \tag{A.106}$$

Because the features $[r(x^{(i)}), g(x^{(i)}), 1]$ are samples from the pushforward distribution of $[r, g, 1]\sharp\mathbb{P}_X$, so by Assumption 3.2, no more than two sample features simultaneously occupy the same linear subspace of $(w_k - w_j)$, $j \neq k$ almost surly (we also verify that $(w_k - w_j)$ has two non-zero coordinates in the first K components). Hence, the number of disagreements is no more than $(K-1)$. QED.

Proof of Lemma A.4. Let $\hat{\gamma}$ denote the minimizer of $\widehat{\mathrm{LP1}}(r, g)$. For the excess risk, by the objective of $\widehat{\mathrm{LP1}}$,

$$\widehat{\mathrm{OPT}} = \frac{1}{N} \sum_{j=1}^{N} \sum_{k \in [K]} r(x^{(j)})_k \hat{\gamma}(x^{(j)}, k), \tag{A.107}$$

and by Lemma A.5, almost surely,

$$\begin{aligned} \left| \widehat{R}(\hat{h}) - \widehat{\mathrm{OPT}} \right| &= \frac{1}{N} \left| \sum_{j=1}^{N} \sum_{k \in [K]} r(x^{(j)})_k \Big(\mathbb{1}[\hat{h}(x^{(j)}) = k] - \hat{\gamma}(x^{(j)}, k) \Big) \right| & \text{(A.108)} \\ &\leq \frac{1}{N} \|\ell\|_\infty \sum_{k \in [K]} \sum_{j=1}^{N} \left| \mathbb{1}[\hat{h}(x^{(j)}) = k] - \hat{\gamma}(x^{(j)}, k) \right| & \text{(A.109)} \\ &\leq \frac{K(K-1)\|\ell\|_\infty}{N}. & \text{(A.110)} \end{aligned}$$

For the fairness violation, let h denote the randomized classifier whose Markov kernel is equal to $\hat{\gamma}$. First, note that by Lemma A.5, almost surely,

$$\left| \widehat{\mathbb{P}}[\hat{h}(X) = k] - \widehat{\mathbb{P}}[h(X) = k] \right| \leq \frac{1}{N} \sum_{j=1}^{N} \left| \mathbb{1}[\hat{h}(x^{(j)}) = k] - \hat{\gamma}(x^{(j)}, k) \right| \leq \frac{K-1}{N}, \tag{A.111}$$

and by Eq. (A.7), with $\widehat{\mathbb{P}}[Z_i = 1] = \frac{1}{N} \sum_{j=1}^{N} g(x^{(j)})_i$, for all $i \in [G]$,

$$\begin{aligned} &\left| \widehat{\mathbb{P}}[\hat{h}(X) = k \mid Z_i = 1] - \widehat{\mathbb{P}}[h(X) = k \mid Z_i = 1] \right| \\ &\qquad \leq \frac{1}{N} \sum_{j=1}^{N} \frac{g(x^{(j)})_i}{\widehat{\mathbb{P}}[Z_i = 1]} \left| \mathbb{1}[\hat{h}(x^{(j)}) = k] - \hat{\gamma}(x^{(j)}, k) \right| \leq \frac{K-1}{N \widehat{\mathbb{P}}[Z_i = 1]} \leq \frac{2(K-1)}{N \, \mathbb{P}[Z_i = 1]}, \end{aligned} \tag{A.112}$$

where the last inequality holds for all $N \geq \max_i 2 \ln(2G/\delta) / \mathbb{P}[Z_i = 1]^2$ with probability at least $1 - \delta$, since under this condition,

$$\left| \widehat{\mathbb{P}}[Z_i = 1] - \mathbb{P}[Z_i = 1] \right| \leq \sqrt{\frac{1}{2N} \ln \frac{2G}{\delta}} \leq \frac{\mathbb{P}[Z_i = 1]}{2} \tag{A.113}$$

by Hoeffding's inequality and a union bound; then, this implies $1/\widehat{\mathbb{P}}[Z_i = 1] \leq 2/\mathbb{P}[Z_i = 1]$.

So by Eq. (A.4), for all $j \in [C]$,

$$\begin{aligned}
B_j\hat{\mu}(\hat{h}) &\leq B_j\hat{\mu}(h) + \left| B_j\hat{\mu}(\hat{h}) - B_j\hat{\mu}(h) \right| && \text{(A.114)} \\
&\leq B_j\hat{\mu}(h) + \sum_{k\in[K]} B_{j,(k,*)} \left| \widehat{\mathbb{P}}[\hat{h}(X)=k] - \widehat{\mathbb{P}}[h(X)=k] \right| \\
&\qquad + \sum_{k\in[K],i\in[G]} B_{j,(k,i)} \left| \widehat{\mathbb{P}}[\hat{h}(X)=k \mid Z_i=1] - \widehat{\mathbb{P}}[h(X)=k \mid Z_i=1] \right| && \text{(A.115)} \\
&\leq B_j\hat{\mu}(h) + \sum_{k\in[K]} B_{j,(k,*)} \frac{K-1}{N} + \sum_{k\in[K],i\in[G]} B_{j,(k,i)} \frac{2(K-1)}{N\,\mathbb{P}[Z_i=1]} && \text{(A.116)} \\
&\leq B_j\hat{\mu}(h) + \max_{i\in G} \frac{2(K-1)\|B\|_{\infty,1}}{N\,\mathbb{P}[Z_i=1]} && \text{(A.117)}
\end{aligned}$$

Then the result follows from taking the max over $j \in [C]$ on both sides, and the fact that $\max_j B_j\hat{\mu}(h) = \alpha$. QED.

With all the technical results above, we can now prove Theorem 3.2. We first bound the fairness violation, then the excess risk.

Proof of Theorem 3.2 (Fairness Violation). Let $\hat{\gamma}$ denote the minimizer of $\widehat{\mathrm{LP1}}(r,g)$. Then by Lemma A.4, for all $N \geq \max_i 2\ln(2G/\delta)/\,\mathbb{P}[Z_i=1]^2$, with probability at least $1-\delta$,

$$\begin{aligned}
V(\hat{h}) &\leq \max_{j\in[C]} B_j\hat{\mu}(\hat{h}) + \max_{j\in[C]} \left| B_j\mu(\hat{h}) - B_j\hat{\mu}(\hat{h}) \right| && \text{(A.118)} \\
&\leq \alpha + \max_{i\in G} \frac{2K\|B\|_{\infty,1}}{N\,\mathbb{P}[Z_i=1]} + \max_{j\in[C]} \left| B_j\mu(\hat{h}) - B_j\hat{\mu}(\hat{h}) \right|. && \text{(A.119)}
\end{aligned}$$

For the last term, by Eq. (A.4), for all $j \in [C]$,

$$\begin{aligned}
\left| B_j\mu(\hat{h}) - B_j\hat{\mu}(\hat{h}) \right| &\leq \sum_{k\in[K]} B_{j,(k,*)} \left| \mathbb{P}[\hat{h}(X)=k] - \widehat{\mathbb{P}}[\hat{h}(X)=k] \right| \\
&\qquad + \sum_{k\in[K],i\in[G]} B_{j,(k,i)} \left| \mathbb{P}[\hat{h}(X)=k \mid Z_i=1] - \widehat{\mathbb{P}}[\hat{h}(X)=k \mid Z_i=1] \right|,
\end{aligned} \tag{A.120}$$

where, because h is a linear multiclass classifier with input features given by the outputs of r, g, and in this analysis we consider r, g given and fixed and only the parameters of the linear classifier are being estimated from the samples, by Theorem A.2 and the VC complexity analysis in Lemma A.3, with probability at least $1-\delta$, for all $k \in [K]$,

$$\left|\mathbb{P}[\hat{h}(X)=k]-\widehat{\mathbb{P}}[\hat{h}(X)=k]\right| = \left|\mathbb{E}[\mathbb{1}[\hat{h}(X)=k]]-\frac{1}{N}\sum_{j=1}^{N}\mathbb{1}[\hat{h}(x^{(j)})=k]\right| \tag{A.121}$$

$$\leq O\left(\sqrt{\frac{(K+G)\log K+\ln G/\delta}{N}}\right); \tag{A.122}$$

similarly, by Eq. (A.7), with $\widehat{\mathbb{P}}[Z_i=1]=\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i$,

$$\left|\mathbb{P}[\hat{h}(X)=k \mid Z_i=1]-\widehat{\mathbb{P}}[\hat{h}(X)=k \mid Z_i=1]\right|$$

$$= \left|\frac{1}{\mathbb{P}[Z_i=1]}\,\mathbb{E}[g(X)_i\hat{h}(X)]-\frac{1}{\widehat{\mathbb{P}}[Z_i=1]}\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i\hat{h}(x^{(j)})]\right| \tag{A.123}$$

$$\leq \frac{1}{\mathbb{P}[Z_i=1]}\left|\mathbb{E}[g(X)_i\hat{h}(X)]-\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i\hat{h}(x^{(j)})]\right| + \left|\frac{1}{\mathbb{P}[Z_i=1]}-\frac{1}{\widehat{\mathbb{P}}[Z_i=1]}\right|\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i\hat{h}(x^{(j)})] \tag{A.124}$$

$$\leq \frac{1}{\mathbb{P}[Z_i=1]}O\left(\sqrt{\frac{(K+G)\log K+\ln G/\delta}{N}}+\sqrt{\frac{\ln G/\delta}{N}}\right), \tag{A.125}$$

and the last inequality follows from Eq. (A.113).

Plugging Eqs. (A.122) and (A.125) back into Eq. (A.120) and Eq. (A.119), with Hölder's inequality, we get

$$V(\hat{h}) \leq \alpha + \max_{i\in G}\frac{\|B\|_{\infty,1}}{\mathbb{P}[Z_i=1]}O\left(\frac{K}{N}+\sqrt{\frac{(K+G)\log K+\ln G/\delta}{N}}+\sqrt{\frac{\ln G/\delta}{N}}\right) \tag{A.126}$$

$$\leq \alpha + \max_{i\in G}\frac{\|B\|_{\infty,1}}{\mathbb{P}[Z_i=1]}O\left(\sqrt{\frac{(K+G)\log K+\ln G/\delta}{N}}\right) \tag{A.127}$$

for all $N \geq \Omega(K)$. QED.

Proof of Theorem 3.2 (Excess Risk). We start by constructing a classifier h' from modifying h, the optimal fair classifier on the population, that satisfies fairness on the empirical problem $\widehat{\mathrm{LP1}}(r,g)$ (defined by the tuple $(r,g,\widehat{\mathbb{P}}_X)$); the strategy is similar to that used in the proof of Theorem 4.2.

First, we bound the fairness violation of h on the empirical problem. Let $\hat{\gamma}$ denote the optimal solution of $\widehat{\mathrm{LP1}}(r, g)$, then

$$\widehat{V}(h) \leq \max_{j\in[C]} B_j\mu(h) + \max_{j\in[C]}|B_j\hat{\mu}(h) - B_j\mu(h)| \leq \alpha + \max_{j\in[C]}|B_j\mu(h) - B_j\hat{\mu}(h)|, \tag{A.128}$$

and by Eq. (A.4), for all $j \in [C]$,

$$\begin{aligned}|B_j\mu(h) - B_j\hat{\mu}(h)| \leq \sum_{k\in[K]} B_{j,(k,*)}\Big|\mathbb{P}[h(X) = k] - \widehat{\mathbb{P}}[h(X) = k]\Big| \\ + \sum_{k\in[K],i\in[G]} B_{j,(k,i)}\Big|\mathbb{P}[h(X) = k \mid Z_i = 1] - \widehat{\mathbb{P}}[h(X) = k \mid Z_i = 1]\Big|.\end{aligned} \tag{A.129}$$

Since h does not depend on the samples, by Hoeffding's inequality, with probability at least $1 - \delta$, for all $k \in [K]$,

$$\Big|\mathbb{P}[h(X) = k] - \widehat{\mathbb{P}}[h(X) = k]\Big| = \left|\mathbb{E}[\mathbb{1}[h(X) = k]] - \frac{1}{N}\sum_{j=1}^{N}\mathbb{1}[h(x^{(j)}) = k]\right| \leq O\left(\sqrt{\frac{\ln G/\delta}{N}}\right); \tag{A.130}$$

similarly, by Eqs. (A.7) and (A.113),

$$\begin{aligned}&\Big|\mathbb{P}[h(X) = k \mid Z_i = 1] - \widehat{\mathbb{P}}[h(X) = k \mid Z_i = 1]\Big| \\ &= \left|\frac{1}{\mathbb{P}[Z_i = 1]}\,\mathbb{E}[g(X)_i h(X)] - \frac{1}{\widehat{\mathbb{P}}[Z_i = 1]}\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i h(x^{(j)})]\right| \qquad (A.131) \\ &\leq \frac{1}{\mathbb{P}[Z_i = 1]}\left|\mathbb{E}[g(X)_i h(X)] - \frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i h(x^{(j)})]\right| \\ &\quad + \left|\frac{1}{\mathbb{P}[Z_i = 1]} - \frac{1}{\widehat{\mathbb{P}}[Z_i = 1]}\right|\frac{1}{N}\sum_{j=1}^{N} g(x^{(j)})_i h(x^{(j)})] \qquad (A.132) \\ &\leq \frac{1}{\mathbb{P}[Z_i = 1]}O\left(\sqrt{\frac{\ln G/\delta}{N}}\right). \qquad (A.133)\end{aligned}$$

Plugging Eqs. (A.130) and (A.133) back into Eq. (A.129) and Eq. (A.128), we get

$$\widehat{V}(h) \leq \alpha + \max_{i\in G}\frac{\|B\|_{\infty,1}}{\mathbb{P}[Z_i = 1]}O\left(\sqrt{\frac{\ln G/\delta}{N}}\right) =: \alpha + \varepsilon_1. \tag{A.134}$$

Now, we construct the reference classifier h' from h as follows: let $\bar{h}$ denote the exactly fair classifier that satisfies Assumption 2.1, and $\lambda \in [0,1]$ to be determined. We set the Markov kernel of h' to

$$\pi_{h'} = \lambda \pi_{\bar{h}} + (1-\lambda)\pi_h, \tag{A.135}$$

We verify its fairness violation on the empirical problem: by Eqs. (A.4) and (A.134),

$$\widehat{V}(h') = \max_{j\in[C]} B_j \hat{\mu}(h') \leq \lambda \max_{j\in[C]} B_j \hat{\mu}(\bar{h}) + (1-\lambda) \max_{j\in[C]} B_j \hat{\mu}(h) \leq (1-\lambda)(\alpha+\varepsilon_1), \tag{A.136}$$

and we can achieve $(1-\lambda)(\alpha+\varepsilon_1) \leq \alpha$ by setting

$$\lambda = \frac{\varepsilon_1}{\alpha+\varepsilon_1} \leq \frac{\varepsilon_1}{\alpha}. \tag{A.137}$$

Using the reference classifier h', we can bound $\widehat{\mathrm{OPT}}$ in terms of the risk of h on the empirical problem, $\widehat{R}(h)$:

$$\widehat{\mathrm{OPT}} \leq \widehat{R}(h') \leq (1-\lambda)\widehat{R}(h) + \lambda \|\ell\|_\infty. \tag{A.138}$$

Next, note that by Eq. (3.8) and Theorem A.2 with the VC complexity analysis in Lemma A.3,

$$\begin{aligned}
\left| R(\hat{h}) - \widehat{R}(\hat{h}) \right| &\leq \sum_{k\in[K]} \left| \mathbb{E}_X \left[r(X)_k \, \mathbb{1}[\hat{h}(X) = k] \right] - \frac{1}{N}\sum_{j=1}^N r(x^{(j)})_k \, \mathbb{1}[\hat{h}(x^{(j)}) = k] \right| && \text{(A.139)} \\
&\leq K\|\ell\|_\infty O\left(\sqrt{\frac{(K+G)\log K + \ln G/\delta}{N}} \right) && \text{(A.140)} \\
&=: \varepsilon_2. && \text{(A.141)}
\end{aligned}$$

Then, putting everything together,

$$\begin{aligned}
R(\hat{h}) &\leq \widehat{R}(\hat{h}) + \varepsilon_2 && \text{by Eq. (A.141)} && \text{(A.142)} \\
&\leq \widehat{\mathrm{OPT}} + \frac{K^2\|\ell\|_\infty}{N} + \varepsilon_2 && \text{by Lemma A.4} && \text{(A.143)} \\
&\leq \widehat{R}(h) + \frac{\varepsilon_1\|\ell\|_\infty}{\alpha} + \frac{K^2\|\ell\|_\infty}{N} + \varepsilon_2 && \text{by Eq. (A.138)} && \text{(A.144)} \\
&\leq R(h) + \frac{\varepsilon_1\|\ell\|_\infty}{\alpha} + \frac{K^2\|\ell\|_\infty}{N} + 2\varepsilon_2 && \text{by Eq. (A.141).} && \text{(A.145)}
\end{aligned}$$

We may conclude by taking a final union bound over all events considered above. QED.

A.3 DERIVATIONS FOR SECTION 4.4.1: MULTIPLE-DISTRIBUTION LINEARPOST

We provide the derivations required to prove Theorem 4.3, and the (empirical) linear program LP2, $\widehat{\text{LP2}}$ for multiple-distribution LinearPost. The analysis mirrors that of single-distribution LinearPost in Section 3.2.1.

Recall that the multiple-distribution fair classification problem (Eq. (4.35)): we want to learn a robust fair classifier for the problem

$$\min_{\substack{h:\mathcal{X}\to[K]\\ \gamma^{(0)},\gamma^{(1)},\dots,\gamma^{(T)}\in\mathbb{R}^L}} R^{(0)}(h) \quad \text{subject to} \quad B\mu^{(t)}(h,\gamma^{(t)}) \le c \ \ \forall t\in\{0,1,\dots,T\}. \tag{A.146}$$

Assuming that the importance weights are finite, $w^{(t)} = p_X^{(t)}/p_X^{(0)} < \infty$ for all t, then following Eq. (3.9), each of the fairness constraints $j\in[C]$ can be written as

$$\begin{aligned} B_j\mu^{(t)}(h) = B_{j,(*,*)} &+ \sum_{l\in[L]} B_{j,(\mathrm{aux},l)}\gamma_l^{(t)} + \sum_{k\in[K]} B_{j,(k,*)}p^{(t)}[h(X)=k] \\ &+ \sum_{k\in[K],i\in[G]} B_{j,(k,i)}p^{(t)}[h(X)=k\mid Z_i=1], \end{aligned} \tag{A.147}$$

where

$$p^{(t)}[h(X)=k] = \int_{\mathcal{X}} \pi_h(x,k)p^{(t)}[X=x]\,\mathrm{d}x = \int_{\mathcal{X}} \pi_h(x,k)w^{(t)}(x)p^{(0)}[X=x]\,\mathrm{d}x, \tag{A.148}$$

and

$$p^{(t)}[h(X)=k\mid Z_i=1] = \int_{\mathcal{X}} \frac{g^{(t)}(X)_i}{p^{(t)}[Z_i=1]}\pi_h(x,k)p^{(t)}[X=x]\,\mathrm{d}x \tag{A.149}$$

$$= \int_{\mathcal{X}} \frac{g^{(t)}(X)_i}{p^{(t)}[Z_i=1]}\pi_h(x,k)w^{(t)}(x)p^{(0)}[X=x]\,\mathrm{d}x, \tag{A.150}$$

with $p^{(t)}[Z_i=1] = \int_{\mathcal{X}} g^{(t)}(X)_i p^{(t)}[X=x]\,\mathrm{d}x = \int_{\mathcal{X}} g^{(t)}(X)_i w^{(t)}(x)p^{(0)}[X=x]\,\mathrm{d}x$.

So the multiple-distribution fair classification problem can be formulated as a linear program over π_h, defined in terms of the reference distribution's input distribution $p_X^{(0)}$, pointwise risk function $r^{(0)}$, and all distributions' group predictors $g^{(0)}, g^{(1)}, \dots, g^{(T)}$, and importance weights $w^{(1)}, \dots, w^{(T)}$:

Primal LP2:

$$\begin{aligned} \min_{\pi_h \geq 0, \gamma} & \int_{\mathcal{X}} \sum_{k \in [K]} r^{(0)}(x)_k \pi_h(x,k) p^{(0)}[X=x] \, \mathrm{d}x \\ \text{s.t.} & \sum_{k \in [K], i \in [G]} \frac{B_{j,(k,i)}}{p^{(t)}[Z_i = 1]} \int_{\mathcal{X}} g^{(t)}(x)_i \pi_h(x,k) w^{(t)}(x) p^{(0)}[X=x] \, \mathrm{d}x + \sum_{l \in [L]} B_{j,(\mathrm{aux},l)} \gamma_l^{(t)} \\ & \quad + B_{j,(*,*)} + \sum_{k \in [K]} B_{j,(k,*)} \int_{\mathcal{X}} \pi_h(x,k) w^{(t)}(x) p^{(0)}[X=x] \, \mathrm{d}x \leq c_j \quad \forall j \in [C], \\ & \qquad \forall t \in \{0, 1, \dots, T\}, \\ & \sum_{k \in [K]} \pi_h(x,k) = 1 \qquad \forall x \in \mathcal{X}, \end{aligned} \tag{A.151}$$

with $w^{(0)} = 1$.

To derive the dual, we introduce dual variables $\phi : \mathcal{X} \to \mathbb{R}$ for the normalization constraints, and $\psi^{(0)}, \psi^{(1)}, \dots, \psi^{(T)} \in \mathbb{R}^C_{\geq 0}$ for the fairness constraints. The Lagrangian is

$$\begin{aligned} & \mathcal{L}(\pi_h, \phi, \psi) \\ & = \int_{\mathcal{X}} \sum_{k \in [K]} r^{(0)}(x)_k \pi_h(x,k) p^{(0)}[X=x] \, \mathrm{d}x + \int_{\mathcal{X}} \left(1 - \sum_{k \in [K]} \pi_h(x,k) \right) \phi(x) p^{(0)}[X=x] \, \mathrm{d}x \\ & \quad + \sum_{t=0}^{T} \sum_{j \in [C]} \psi_j^{(t)} \Bigg(\sum_{k \in [K], i \in [G]} \frac{B_{j,(k,i)}}{p^{(t)}[Z_i = 1]} \int_{\mathcal{X}} g^{(t)}(x)_i \pi_h(x,k) w^{(t)}(x) p^{(0)}[X=x] \, \mathrm{d}x \\ & \qquad + \sum_{l \in [L]} B_{j,(\mathrm{aux},l)} \gamma_l^{(t)} + \sum_{k \in [K]} B_{j,(k,*)} \int_{\mathcal{X}} \pi_h(x,k) w^{(t)}(x) p^{(0)}[X=x] \, \mathrm{d}x - \big(c_j - B_{j,(*,*)} \big) \Bigg) \end{aligned} \tag{A.152}$$

$$\begin{aligned} & = \int_{\mathcal{X}} \phi(x) p^{(0)}[X=x] \, \mathrm{d}x - \sum_{t=0}^{T} \left(\sum_{j \in [C]} \psi_j^{(t)} \big(c_j - B_{j,(*,*)} \big) + \sum_{l \in [L]} \gamma_l^{(t)} \sum_{j \in [C]} \psi_j^{(t)} B_{j,(\mathrm{aux},l)} \right) \\ & \quad + \int_{\mathcal{X}} \sum_{k \in [K]} \pi_h(x,k) p^{(0)}[X=x] \Bigg(r^{(0)}(x)_k \\ & \qquad - \Bigg(\underbrace{\phi(x) + \sum_{t=0}^{T} \sum_{j \in [C]} \psi_c^{(t)} \left(B_{j,(k,*)} + \sum_{i \in [G]} \frac{B_{j,(k,i)} g^{(t)}(x)_i w^{(t)}(x)}{p^{(t)}[Z_i = 1]} \right)}_{(\star)} \Bigg) \Bigg) \, \mathrm{d}x. \end{aligned} \tag{A.153}$$

After applying strong duality, we get

Dual LP2:

$$\begin{aligned}
&\max_{\psi\geq 0,\phi} \int_{\mathcal{X}} \phi(x) p^{(0)}[X=x]\,\mathrm{d}x - \sum_{t=0}^{T}\sum_{j\in[C]} \psi_j^{(t)}\big(c_j - B_{j,(*,*)}\big) \\
&\quad \text{s.t. } \phi(x) + \sum_{t,j} \psi_j^{(t)} \left(B_{j,(k,*)} + \sum_{i\in[G]} \frac{B_{j,(k,i)} g^{(t)}(x)_i w^{(t)}(x)}{p^{(t)}[Z_i=1]} \right) \leq r^{(0)}(x)_k \qquad \forall x\in\mathcal{X},\, k\in[K], \\
&\qquad \sum_{j\in[C]} \psi_j^{(t)} B_{j,(\mathrm{aux},l)} = 0 \qquad \forall l\in[L],\, t\in\{0,1,\dots,T\}.
\end{aligned} \tag{A.154}$$

The proof of Theorem 4.3 then follows the same analysis as that of Theorem 3.1, hence omitted.

The empirical version of LP2, used by multiple-distribution LinearPost (Algorithm 4.3), is:

Empirical $\widehat{\mathrm{LP2}}$:

$$\begin{aligned}
&\min_{\pi_h\geq 0,\gamma} \frac{1}{N} \sum_{\ell\in[N]}\sum_{k\in[K]} \Big(\hat{r}^{(0)}(x^{(\ell)})_k + \xi_k^{(j)}\Big) \pi_h(x^{(\ell)},k) \\
&\quad \text{s.t. } B_{j,(*,*)} + \sum_{l\in[L]} B_{j,(\mathrm{aux},l)} \gamma_l^{(t)} + \sum_{k\in[K]} B_{j,(k,*)} \frac{1}{N} \sum_{\ell\in[N]} \pi_h(x^{(\ell)},k) w^{(t)}(x^{(\ell)}) \\
&\qquad + \sum_{k\in[K],i\in[G]} \frac{B_{j,(k,i)}}{\hat{p}^{(t)}[Z_i=1]} \frac{1}{N} \sum_{\ell\in[N]} \hat{g}^{(t)}(x^{(\ell)})_i \pi_h(x^{(\ell)},k) w^{(t)}(x^{(\ell)}) \leq c_j \qquad \forall j\in[C], \\
&\qquad\qquad \forall t\in\{0,1,\dots,T\}, \\
&\qquad \sum_{k\in[K]} \pi_h(x^{(\ell)},k) = 1 \qquad \forall \ell\in[N],
\end{aligned} \tag{A.155}$$

where $w^{(0)}=1$, and

$$\hat{p}^{(t)}[Z_i=1] = \frac{1}{N}\sum_{j\in[N]} \hat{g}(x^{(j)})_i w(x^{(j)}). \tag{A.156}$$

This linear program has $(NK + (T+1)L)$ variables and $(N + (T+1)C)$ constraints.

A.4 PROOFS FOR CHAPTER 5

We begin with an analysis of the L^1 sensitivity of the PMF for our notion of neighboring datasets described in Section 5.2.

Remark A.1. Let $S \in \mathcal{X}^N$ be a dataset of size N, and let S' be another dataset with the same support that differs from S in at most one entry by insertion, deletion, or substitution. Let $\hat{p}$ and $\hat{p}'$ denote the empirical PMFs of S and S', respectively; that is, $\hat{p}[X = x] = \frac{1}{N} \sum_{x' \in S} \mathbb{1}[x' = x]$. Then the L^1 sensitivity of $\hat{p}$ is at most $2/N$.

Without loss of generality, assume $\mathcal{X} = \{1, 2\}$, and that the insertion/deletion is of an instance of $x = 1$. Define $N_i = \sum_{x \in S} \mathbb{1}[x = i]$.

– (Insertion). The sensitivity is

$$\begin{aligned}
\|\hat{p} - \hat{p}'\|_1 &= \sum_{x \in \mathcal{X}} |\hat{p}[X = x] - \hat{p}'[X = x]| && \text{(A.157)} \\
&= \left| \frac{N_1}{N} - \frac{N_1 + 1}{N + 1} \right| + \left| \frac{N_2}{N} - \frac{N_2}{N + 1} \right| && \text{(A.158)} \\
&= \left| \frac{N_1(N + 1) - (N_1 + 1)N}{N(N + 1)} \right| + \left| \frac{N_2(N + 1) - N_2 N}{N(N + 1)} \right| && \text{(A.159)} \\
&= \left| \frac{N_1 - N}{N(N + 1)} \right| + \left| \frac{N_2}{N(N + 1)} \right| && \text{(A.160)} \\
&= 2 \left| \frac{N_2}{N(N + 1)} \right| && \text{(A.161)} \\
&\leq \frac{2}{N}. && \text{(A.162)}
\end{aligned}$$

– (Deletion). Similarly, because $N_2 \leq n - 1$,

$$\|\hat{p} - \hat{p}'\|_1 = \left| \frac{N_1}{N} - \frac{N_1 - 1}{N - 1} \right| + \left| \frac{N_2}{N} - \frac{N_2}{N - 1} \right| = 2 \left| \frac{N_2}{N(N - 1)} \right| \leq \frac{2}{N}. \tag{A.163}$$

– (Substitution).

$$\|\hat{p} - \hat{p}'\|_1 = \left| \frac{N_1}{N} - \frac{N_1 - 1}{N} \right| + \left| \frac{N_2}{N} - \frac{N_2 + 1}{N} \right| = \frac{1}{N} + \frac{1}{N} = \frac{2}{N}. \tag{A.164}$$

A.4.1 Proof of Theorem 5.3: Optimal Fair Predictor on Regression Problems

Theorem 5.3 is a direct consequence of the lemma below. It says that, given any (randomized) predictor f with shape r (meaning its output distribution), one can derive a predictor of the form $g \circ f^*$ that has the same shape and excess risk. This predictor is obtained from the Bayes-optimal predictor f^* via post-processing, by defining g as the randomized function with Markov kernel $\pi_g(y^*, y) = \gamma(y^*, y)/\gamma(y^*, \mathbb{R})$ where γ is as in Eq. (A.165).

Lemma A.6. Let a regression problem be given. Let $f^* : \mathcal{X} \to \mathbb{R}$ be the (unconstrained) Bayes-optimal predictor, and let its output distribution be $r^* = f^* \sharp X$. Let r be an arbitrary distribution on $\mathbb{R}$. Then for any randomized predictor f with Markov kernel π_f satisfying $f \sharp X = r$, the coupling $\gamma \in \Pi(r^*, r)$ given by

$$\gamma(y^*, y) = \int_{{f^*}^{-1}(y^*)} \pi_f(x, y)\, \mathbb{P}[X = x] \,\mathrm{d}x \tag{A.165}$$

(where ${f^*}^{-1}(y^*) = \{x \in \mathcal{X} : f^*(x) = y^*\}$) satisfies

$$R(f) - R(f^*) = \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2 \,\mathrm{d}\gamma(y^*, y). \tag{A.166}$$

Conversely, for any $\gamma \in \Pi(r^*, r)$, the randomized predictor f with Markov kernel

$$\pi_f(x, y) = \frac{\gamma(f^*(x), y)}{\gamma(f^*(x), \mathbb{R})} = \frac{\gamma(f^*(x), y)}{\int_{\mathbb{R}} \gamma(f^*(x), y)\,\mathrm{d}y} \tag{A.167}$$

satisfies $f \sharp X = r$ and Eq. (A.166).

Proof. For the first direction, note that

$$\begin{aligned}
R(f) - R(f^*) &= \mathbb{E}[(f^*(X) - f(X))^2] && \text{(A.168)}\\
&= \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2\, \mathbb{P}[f^*(X) = y^*, f(X) = y] \,\mathrm{d}(y^*, y) && \text{(A.169)}\\
&= \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2 \left(\int_{\mathcal{X}} \mathbb{P}[f^*(X) = y^*, f(X) = y, X = x] \,\mathrm{d}x \right) \mathrm{d}(y^*, y) && \text{(A.170)}\\
&= \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2 \left(\int_{{f^*}^{-1}(y^*)} \mathbb{P}[f(X) = y, X = x] \,\mathrm{d}x \right) \mathrm{d}(y^*, y) && \text{(A.171)}\\
&= \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2 \left(\int_{{f^*}^{-1}(y^*)} \mathbb{P}[f(X) = y \mid X = x] \,\mathrm{d}\mu_X(x) \right) \mathrm{d}(y^*, y) && \text{(A.172)}\\
&= \int_{\mathbb{R}\times\mathbb{R}} (y^* - y)^2 \gamma(y^*, y) \,\mathrm{d}(y^*, y) && \text{(A.173)}
\end{aligned}$$

as desired, where line 4 is because $\mathbb{P}[f^*(X) = y^*, f(X) = y, X = x] = \mathbb{1}[f^*(x) = y^*]\, \mathbb{P}[f(X) =$

$y, X = x]$ as f^* is deterministic. We also verify that the constructed γ is a valid coupling:

$$\begin{aligned}\int_{\mathbb{R}} \gamma(y^*, y)\,\mathrm{d}y &= \int_{\mathbb{R}} \int_{{f^*}^{-1}(y^*)} \pi_f(x, y)\,\mathbb{P}[X = x]\,\mathrm{d}x\,\mathrm{d}y && \text{(A.174)}\\ &= \int_{{f^*}^{-1}(y^*)} \int_{\mathbb{R}} \pi_f(x, y)\,\mathrm{d}y\,\mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.175)}\\ &= \int_{{f^*}^{-1}(y^*)} \mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.176)}\\ &= \mathbb{P}[f^*(X) = y^*] && \text{(A.177)}\\ &= r^*(y^*) && \text{(A.178)}\end{aligned}$$

by Definitions A.1 and A.3, and

$$\begin{aligned}\int_{\mathbb{R}} \gamma(y^*, y)\,\mathrm{d}y^* &= \int_{\mathbb{R}} \int_{{f^*}^{-1}(y^*)} \pi_f(x, y)\,\mathbb{P}[X = x]\,\mathrm{d}x\,\mathrm{d}y^* && \text{(A.179)}\\ &= \int_{\mathcal{X}} \pi_f(x, y)\,\mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.180)}\\ &= \int_{\mathcal{X}} \mathbb{P}[f(X) = y \mid X = x]\,\mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.181)}\\ &= \mathbb{P}[f(X) = y] && \text{(A.182)}\\ &= r(y) && \text{(A.183)}\end{aligned}$$

by Definition A.2 and the assumption that $f\sharp X = r$.

For the converse direction, it suffices to show that the Markov kernel constructed for f satisfies the equality in Eq. (A.165), which would immediately imply $f\sharp X = r$, and Eq. (A.166) with the same arguments in Eq. (A.173). Let $y^*, y \in \mathbb{R}$ and $z \in {f^*}^{-1}(y^*)$ be arbitrary, then

$$\begin{aligned}\gamma(y^*, y) &= \frac{\gamma(y^*, y)}{\gamma(y^*, \mathbb{R})}\gamma(y^*, \mathbb{R}) = \frac{\gamma(f^*(z), y)}{\gamma(f^*(z), \mathbb{R})}\gamma(y^*, \mathbb{R}) && \text{(A.184)}\\ &= \pi_f(z, y)\gamma(y^*, \mathbb{R}) = \pi_f(z, y) r^*(y^*) && \text{(A.185)}\\ &= \pi_f(z, y) \int_{{f^*}^{-1}(y^*)} \mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.186)}\\ &= \int_{{f^*}^{-1}(y^*)} \pi_f(z, y)\,\mathbb{P}[X = x]\,\mathrm{d}x && \text{(A.187)}\\ &= \int_{{f^*}^{-1}(y^*)} \pi_f(x, y)\,\mathbb{P}[X = x]\,\mathrm{d}x, && \text{(A.188)}\end{aligned}$$

where line 3 is by construction of π_f, line 4 by the assumption that $\gamma \in \Pi(r^*, r)$, and the last line is because $\pi_f(x, y) = \pi_f(z, y)$ for all $x \in {f^*}^{-1}(y^*)$, also by construction. QED.

Given the Bayes-optimal predictor f^*, this lemma allows us to formulate the problem of finding the optimal predictor under a shape constraint r as that of finding the optimal coupling $\gamma \in \Pi(r^*, r)$ with the squared cost. Because statistical parity in the attribute-aware case can be interpreted as a shape constraint on the group-conditional predictors, we can leverage this lemma to prove Theorem 5.3:

Proof of Theorem 5.3. Because the predictor is attribute-aware, $f : \mathcal{X} \times [M] \to \mathbb{R}$, we can optimize the components corresponding to each group independently, $f_a = f(\cdot, a)$, $\forall a \in [M]$. Denote the overall excess risk (relative to f^*) by $ER(f) = R(f) - R(f^*)$, and the excess risk conditioned on group a by $ER_a(f) = \mathbb{E}[(f(X, A) - f^*(X, A))^2 \mid A = a]$, then

$$ER(f) = \sum_{a \in [M]} p_a ER_a(f_a). \tag{A.189}$$

Denote $r_a = f\sharp(X \mid A = a, a) = f_a\sharp(X \mid A = a)$, we have

$$\begin{aligned}\min_{f : V^{\mathrm{SP}}(f) \le \alpha} ER(f) &= \min_{\{f_a\}_{a \in [M]} : D_{\mathrm{KS}}(f_a\sharp(X|A=a), f_{a'}\sharp(X|A=a')) \le \alpha} \sum_{a \in [M]} p_a ER_a(f_a) &\text{(A.190)}\\ &= \min_{\{r_a\}_{a \in [M]} : D_{\mathrm{KS}}(r_a, r_{a'}) \le \alpha} \sum_{a \in [M]} p_a \min_{f_a : f_a\sharp(X|A=a) = r_a} ER_a(f_a), &\text{(A.191)}\end{aligned}$$

where, by Lemma A.6 and the definition of Wasserstein distance,

$$\min_{f_a : f_a\sharp(X|A=a) = r_a} ER_a(f_a) = \min_{\gamma \in \Pi(r_a^*, r_a)} \int (y^* - y)^2 \, \mathrm{d}\gamma(y^*, y) = W_2^2(r_a^*, r_a); \tag{A.192}$$

plugging this back into the previous equation proves the expression for the risk of the optimal fair predictor.

Let $\{\bar{r}_a\}_{a \in [M]}$ denote the optimal solution to the outer problem in Eq. (A.191), and subsequently, let $\bar{\gamma}_a$ denote the optimal coupling in Eq. (A.192). By definition, a randomized optimal transport T_a from r_a^* to $\bar{r}_a$ has Markov kernel $\pi_{T_a}(y^*, y) = \bar{\gamma}(y^*, y)/\bar{\gamma}(y^*, \mathbb{R})$. Lemma A.6 confirms that a minimizer of the left-hand side of Eq. (A.192) is $(x, a) \mapsto T_a \circ f^*(x, a)$, hence establishes the representation result. QED.

A.4.2 Proofs of Theorems 5.4 and 5.5: Sample Complexity

We will prove Theorem 5.5 using the bound on the error of the Laplace histogram in Theorem 5.4. The proofs require several technical lemmas. First, a concentration bound on the i.i.d. sum of Laplace random variables based on a result by Li and Tkocz [228]:

Proposition A.1. Let $X_1, \dots, X_N \sim \text{Laplace}(0,1)$, $Y_1, \dots, Y_N \sim \text{Exponential}(1)$, and $Z \sim \mathcal{N}(0,1)$ be independent, then $\sum_j X_j$ has the same distribution as $\sqrt{2\sum_j Y_j} Z$.

The identity follows from the fact that if $W_1, \dots, W_N \sim \mathcal{N}(0,1)$, then $\sqrt{2Y_j}W_j \sim \text{Laplace}(0,1)$ for each j, and then combine with the standard fact that for any $a_1, \dots, a_N \geq 0$ independent from the W_j's, $\sum_{j=1}^N a_j W_j \sim \mathcal{N}(0, \sum_{j=1}^N a_j^2)$.

Lemma A.7. Let $X_1, \dots, X_N \sim \text{Laplace}(0,1)$ be independent, then for all $t \geq 0$, with probability at least $1-\delta$, $|\sum_{j=1}^N X_j| \leq 2\sqrt{N}\ln(2N/\delta)$.

Proof. Using Proposition A.1, we bound $\sum_{j=1}^N X_j$ by analyzing $\sqrt{\sum_{j=1}^N Y_j}|Z|$. For all $t \geq 0$,

$$\mathbb{P}\left[\sum_{j=1}^N Y_j \geq t\right] \leq \mathbb{P}\left[\exists j \text{ s.t. } Y_j \geq \frac{t}{k}\right] \leq N\,\mathbb{P}\left[Y_1 \geq \frac{t}{N}\right] \leq N\exp\left(-\frac{t}{N}\right). \tag{A.193}$$

On the other hand, the Chernoff bound implies that $\mathbb{P}[|Z| \geq t] \leq 2\exp(-t^2/2)$. With a union bound, with probability at least $1-\delta$,

$$\left|\sqrt{2\sum_{j=1}^N Y_j Z}\right| = \sqrt{2\sum_{j=1}^N Y_j|Z|} \leq 2\sqrt{N}\ln\frac{2N}{\delta}. \tag{A.194}$$

QED.

Next, an L^1 (total variation) convergence result of empirical distributions with finite support, following the concentration of i.i.d. sum of Multinoulli random variables [229]:

Theorem A.3. Let $p \in \Delta^K$, the $(K-1)$-dimensional simplex, and $\hat{p}_N \sim \frac{1}{N}\text{Multinomial}(N,p)$, then with probability at least $1-\delta$, $\|p - \hat{p}_N\|_1 \leq \sqrt{(2K/N)\ln(2/\delta)}$.

Corollary A.1. Let $\mathbb{P}$ be a distribution over a finite support $\mathcal{X}$ with PMF $f \in \Delta^{\mathcal{X}}$, and $x_1, \dots, x_N \sim \mathbb{P}$ be i.i.d. samples with empirical distribution denoted by $\hat{f}_N$, then with probability at least $1-\delta$, $\|f - \hat{f}_N\|_1 \leq \sqrt{(2|\mathcal{X}|/N)\ln(2/\delta)}$.

For the proof of Theorem 5.4, we need two technical lemmas:

Lemma A.8. Let $\mathbb{P}$ be a distribution over $[K]$, $x^{(1)}, \dots, x^{(N)} \sim \mathbb{P}$ i.i.d. with their empirical PMF denoted by $\hat{f} \in \Delta^K$, where $\hat{f}_j = \frac{1}{N}\sum_{i=1}^N \mathbb{1}[x^{(i)} = j]$. Let $\mathcal{Z} \subseteq [K]$ be a subset of size L, define $p_{\mathcal{Z}} = \mathbb{P}[x \in \mathcal{Z}]$, denote the PMF conditioned on the event $x \in \mathcal{Z}$ by $f_{|\mathcal{Z}}$, and its empirical counterpart by $\hat{f}_{|\mathcal{Z}} \in \Delta^K$, where $\hat{f}_{j|\mathcal{Z}} = \frac{1}{\widehat{N}_{\mathcal{Z}}} \sum_{i=1}^N \mathbb{1}[x^{(i)} = j]\,\mathbb{1}[x^{(i)} \in \mathcal{Z}]$ and $\widehat{N}_{\mathcal{Z}} = \sum_{i=1}^N \mathbb{1}[x^{(i)} \in \mathcal{Z}]$. Then for all

$$N \geq \frac{8}{p_{\mathcal{Z}}} \ln \frac{8}{\delta}, \tag{A.195}$$

with probability at least $1 - \delta$,

$$\|f_{|\mathcal{Z}} - \hat{f}_{|\mathcal{Z}}\|_\infty \leq \sqrt{\frac{1}{Np_{\mathcal{Z}}} \ln \frac{8L}{\delta}}, \tag{A.196}$$

$$D_{\mathrm{TV}}(f_{|\mathcal{Z}}, \hat{f}_{|\mathcal{Z}}) = \frac{1}{2}\|f_{|\mathcal{Z}} - \hat{f}_{|\mathcal{Z}}\|_1 \leq \sqrt{\frac{L}{4Np_{\mathcal{Z}}} \ln \frac{8}{\delta}}, \tag{A.197}$$

$$D_{\mathrm{KS}}(f_{|\mathcal{Z}}, \hat{f}_{|\mathcal{Z}}) \leq \sqrt{\frac{1}{Np_{\mathcal{Z}}} \ln \frac{8L}{\delta}}. \tag{A.198}$$

Proof. By the Chernoff bound on the Binomial distribution, for all

$$N \geq \frac{8}{p_{\mathcal{Z}}} \ln \frac{2}{\delta}, \tag{A.199}$$

with probability at least $1 - \delta$,

$$\frac{Np_{\mathcal{Z}}}{2} \leq \widehat{N}_{\mathcal{Z}} \leq 2Np_{\mathcal{Z}}. \tag{A.200}$$

Order the samples so that the ones with $x^{(i)} \in \mathcal{Z}$ are at the front (or, equivalently, consider the first $\widehat{N}_{\mathcal{Z}}$ samples as drawn from $f_{|\mathcal{Z}}$). Conditioned on Eq. (A.200), by Hoeffding's inequality and a union bound, for all $j \in \mathcal{Z}$, with probability at least $1 - \delta$,

$$|f_{j|\mathcal{Z}} - \hat{f}_{j|\mathcal{Z}}| = \left| \frac{1}{p_{\mathcal{Z}}} \mathbb{P}[X = j] - \frac{1}{\widehat{N}_{\mathcal{Z}}} \sum_{i=1}^{\widehat{N}_{\mathcal{Z}}} \mathbb{1}[x^{(i)} = j] \right| \tag{A.201}$$

$$= \frac{1}{p_{\mathcal{Z}}} \left| \mathbb{P}[X = j] - \frac{1}{\widehat{N}_{\mathcal{Z}}} \sum_{i=1}^{\widehat{N}_{\mathcal{Z}}} p_{\mathcal{Z}}\, \mathbb{1}[x^{(i)} = j] \right| \tag{A.202}$$

$$\leq \frac{1}{p_{\mathcal{Z}}} \sqrt{\frac{p_{\mathcal{Z}}^2}{2\widehat{N}_{\mathcal{Z}}} \ln \frac{2L}{\delta}} \leq \sqrt{\frac{1}{Np_{\mathcal{Z}}} \ln \frac{2L}{\delta}}. \tag{A.203}$$

Next, by Corollary A.1, with probability at least $1 - \delta$,

$$D_{\mathrm{TV}}(f_{|\mathcal{Z}}, \hat{f}_{|\mathcal{Z}}) = \frac{1}{2}\sum_{j\in\mathcal{Z}}\left|f_{j|\mathcal{Z}} - \hat{f}_{j|\mathcal{Z}}\right| = \frac{1}{2}\sum_{j\in\mathcal{Z}}\left|\frac{1}{p_{\mathcal{Z}}}\,\mathbb{P}[X = j] - \frac{1}{\widehat{N}_{\mathcal{Z}}}\sum_{i=1}^{\widehat{N}_{\mathcal{Z}}}\mathbb{1}[x^{(i)} = j]\right| \tag{A.204}$$

$$\leq \frac{1}{2}\sqrt{\frac{L}{2\widehat{N}_{\mathcal{Z}}}\ln\frac{2}{\delta}} \leq \sqrt{\frac{L}{4Np_{\mathcal{Z}}}\ln\frac{2}{\delta}}. \tag{A.205}$$

Lastly, D_{KS} computes the L^{∞}-distance between two CDFs, so similar to the L_{∞} bound, by Hoeffding's inequality and a union bound, with probability at least $1 - \delta$,

$$D_{\mathrm{KS}}(f_{|\mathcal{Z}}, \hat{f}_{|\mathcal{Z}}) = \max_{j\in\mathcal{Z}}\left|\frac{1}{p_{\mathcal{Z}}}\,\mathbb{P}[X \leq j] - \frac{1}{\widehat{N}_{\mathcal{Z}}}\sum_{i=1}^{\widehat{N}_{\mathcal{Z}}}\mathbb{1}[x^{(i)} \leq j]\right| \tag{A.206}$$

$$= \frac{1}{p_{\mathcal{Z}}}\left|\mathbb{P}[X \leq j] - \frac{1}{\widehat{N}_{\mathcal{Z}}}\sum_{i=1}^{\widehat{N}_{\mathcal{Z}}} p_{\mathcal{Z}}\,\mathbb{1}[x^{(i)} \leq j]\right| \tag{A.207}$$

$$\leq \frac{1}{p_{\mathcal{Z}}}\sqrt{\frac{p_{\mathcal{Z}}^2}{2\widehat{N}_{\mathcal{Z}}}\ln\frac{2L}{\delta}} \tag{A.208}$$

$$\leq \sqrt{\frac{1}{Np_{\mathcal{Z}}}\ln\frac{2L}{\delta}}. \tag{A.209}$$

The result follows by taking a final union bound over the four events during the analysis and rescaling $\delta \leftarrow \delta/4$. QED.

The following is an analysis of the error of Laplace histograms with NormalizeCumSum normalization.

Lemma A.9. Let $f \in \mathbb{R}_{\geq 0}^K$ be a constant vector (corresponds to the PMF to be privatized), and $\xi_1, \ldots, \xi_K \sim \text{Laplace}(0, b)$ be independent. Define $F \in \mathbb{R}_{\geq 0}^K$, where $F_j = \sum_{\ell=1}^{j} f_\ell$ (the CDF of f). Denote $s = F_K$, and let $t \geq 0$ be an arbitrary clipping threshold. For all $j \in [K]$, define

$$\text{(add noise)} \qquad p_j \coloneqq f_j + \xi_j, \qquad P_j = \sum_{\ell=1}^{j} p_\ell, \tag{A.210}$$

$$\text{(isotonic regression)} \qquad q_j = Q_j - Q_{j-1}, \qquad Q_j \coloneqq \frac{1}{2}\big(P_{l_j} + P_{r_j}\big), \tag{A.211}$$

$$\text{(clipping)} \qquad g_j = G_j - G_{j-1}, \qquad G_j \coloneqq \begin{cases} \mathrm{proj}_{[0,t]} Q_j & \text{if } j < K, \\ t & \text{else,} \end{cases} \tag{A.212}$$

where

$$(l_j, r_j) = \arg\max_{l \le j \le r}(P_l - P_r). \tag{A.213}$$

Then with probability at least $1-\delta$,

$$\|f-g\|_1 \le 3|s-t| + 74bK \ln\frac{4K}{\delta}, \tag{A.214}$$

$$\|f-g\|_\infty \le 2|s-t| + 32b\sqrt{K}\ln\frac{4K}{\delta}, \tag{A.215}$$

$$\|F-G\|_\infty \le |s-t| + 12b\sqrt{K}\ln\frac{4K}{\delta}. \tag{A.216}$$

Proof. Our analysis proceeds by using the triangle inequality to decompose and bound each of the following terms (shown here for $\|\cdot\|_1$, analogously for $\|\cdot\|_\infty$ and the partial sums):

$$\|f-g\|_1 \le \|f-p\|_1 + \|p-q\|_1 + \|q-g\|_1. \tag{A.217}$$

We will use the following concentration result of Laplace random variables: by Lemma A.7, with probability at least $1-\delta$, for all $0 \le \ell \le m \le K$,

$$\left|\sum_{j=\ell+1}^{m} \xi_j\right| \le 4b\sqrt{m-\ell}\ln\frac{2(m-\ell)K^2}{\delta} \le 12b\sqrt{m-\ell}\ln\frac{2K}{\delta}. \tag{A.218}$$

First Term in Eq. (A.217). By the tail of the Laplace (equivalently, two-sided exponential) distribution and a union bound, with probability at least $1-\delta$,

$$|f_j - p_j| = |\xi_j| \le 2b\ln\frac{K}{\delta}, \quad \forall j \in [K], \tag{A.219}$$

and it follows that

$$\|f-p\|_1 = \sum_{j=1}^{K}|f_j - p_j| \le 2bK\ln\frac{K}{\delta}. \tag{A.220}$$

For the partial sums, by Eq. (A.218),

$$\|F-P\|_\infty = \max_j|F_j - P_j| = \max_j\left|\sum_{\ell=1}^{j}\xi_\ell\right| \le \max_j 12b\sqrt{j}\ln\frac{2K}{\delta} = 12b\sqrt{K}\ln\frac{2K}{\delta}. \tag{A.221}$$

Second Term in Eq. (A.217). Note that for any $\ell \leq m$ such that $P_\ell \geq P_m$ (i.e., a violating pair for isotonic regression),

$$P_\ell - P_m = -\sum_{j=\ell+1}^{m} (f_j + \xi_j) \leq -\sum_{j=\ell+1}^{m} \xi_j \leq 12b\sqrt{m-\ell}\ln\frac{2K}{\delta}, \tag{A.222}$$

because $f_j \geq 0$. So for all $j \in [K]$,

$$\begin{aligned}
0 &\leq |p_j - q_j| && \text{(A.223)}\\
&= |Q_j - Q_{j-1} - (P_j - P_{j-1})| && \text{(A.224)}\\
&\leq |Q_j - P_j| + |Q_{j-1} - P_{j-1}| && \text{(A.225)}\\
&= \begin{Bmatrix} Q_j - P_j & \text{if } Q_j > P_j \\ P_j - Q_j & \text{else} \end{Bmatrix} + \begin{Bmatrix} Q_{j-1} - P_{j-1} & \text{if } Q_{j-1} > P_{j-1} \\ P_{j-1} - Q_{j-1} & \text{else} \end{Bmatrix} && \text{(A.226)}\\
&= \begin{Bmatrix} Q_j - \dfrac{Q_{l_j} + Q_{r_j}}{2} & \text{if } Q_j > P_j \\ \dfrac{Q_{l_j} + Q_{r_j}}{2} - Q_j & \text{else} \end{Bmatrix} + \begin{Bmatrix} Q_{j-1} - \dfrac{Q_{l_{j-1}} + Q_{r_{j-1}}}{2} & \text{if } Q_{j-1} > P_{j-1} \\ \dfrac{Q_{l_{j-1}} + Q_{r_{j-1}}}{2} - Q_{j-1} & \text{else} \end{Bmatrix} && \text{(A.227)}\\
&\leq \begin{Bmatrix} Q_j - \dfrac{Q_j + Q_{r_j}}{2} & \text{if } Q_j > P_j \\ \dfrac{Q_{l_j} + Q_j}{2} - Q_j & \text{else} \end{Bmatrix} + \begin{Bmatrix} Q_{j-1} - \dfrac{Q_{j-1} + Q_{r_{j-1}}}{2} & \text{if } Q_{j-1} > P_{j-1} \\ \dfrac{Q_{l_{j-1}} + Q_{j-1}}{2} - Q_{j-1} & \text{else} \end{Bmatrix} && \text{(A.228)}\\
&= \frac{1}{2}\begin{Bmatrix} Q_j - Q_{r_j} & \text{if } Q_j > P_j \\ Q_{l_j} - Q_j & \text{else} \end{Bmatrix} + \frac{1}{2}\begin{Bmatrix} Q_{j-1} - Q_{r_{j-1}} & \text{if } Q_{j-1} > P_{j-1} \\ Q_{l_{j-1}} - Q_{j-1} & \text{else} \end{Bmatrix} && \text{(A.229)}\\
&\leq 12b\sqrt{\max(r_j - j, j - l_j)}\ln\frac{2K}{\delta} \leq 12b\sqrt{\max(K-j, j)}\ln\frac{2K}{\delta}, && \text{(A.230)}
\end{aligned}$$

where the above uses $Q_{r_j} \leq Q_j \leq Q_{l_j}$ for all j, and then Eq. (A.222). It then follows that

$$\|p - q\|_1 = \sum_{j=1}^{K} |p_j - q_j| \leq 24bK\ln\frac{2K}{\delta}. \tag{A.231}$$

Lastly, using the fact that the L^∞ error of L^∞ isotonic regression is $\frac{1}{2}\max_{\ell\leq m}(Q_\ell - Q_m)$ [230], and by Eq. (A.222) again,

$$0 \leq \|P - Q\|_\infty = \frac{1}{2}\max_{\ell \leq m}(Q_\ell - Q_m) \leq 6b\sqrt{K}\ln\frac{2K}{\delta}. \tag{A.232}$$

Third Term in Eq. (A.217). Note that because $f_j \in [0, s]$, by Eq. (A.218),

$$\min_j Q_j = \min_j \frac{1}{2}(P_{l_j} + P_{r_j}) \geq \min_j P_j = \min_j \sum_{m=1}^{j} (f_m + \xi_m) \tag{A.233}$$

$$\geq \min_j \sum_{m=1}^{j} \xi_m \geq -\max_j \left| \sum_{m=1}^{K} \xi_m \right| \geq -12b\sqrt{K} \ln \frac{2K}{\delta}, \tag{A.234}$$

and similarly,

$$\max_j Q_j \leq \max_j \sum_{m=1}^{j} (f_m + \xi_m) \leq s + 12b\sqrt{K} \ln \frac{2K}{\delta}. \tag{A.235}$$

Since Q is nondecreasing after isotonic regression, clipping only affects its prefix and/or suffix. For the prefix, let $l = \max\{j \in [K] : G_j = 0\}$. If l does not exist, then no clipping to zero has occurred. Otherwise, for all $j \leq l$, by Eq. (A.234),

$$|Q_j - G_j| = \max(-Q_j, 0) \leq 12b\sqrt{K} \ln \frac{2K}{\delta}, \tag{A.236}$$

and

$$|q_j - g_j| = |Q_j - Q_{j-1} - (G_j - G_{j-1})| \leq -Q_j - Q_{j-1} \tag{A.237}$$

$$\leq 2 \max\left(-\min_j Q_j, 0 \right) \leq 24b\sqrt{K} \ln \frac{2K}{\delta}. \tag{A.238}$$

For the suffix, let $r = \min\{j \in [K] : G_j = t\}$, then for all $r \leq j < K$, by Eq. (A.235),

$$|Q_j - G_j| = Q_j - t \leq |s - t| + 12b\sqrt{K} \ln \frac{2K}{\delta}, \tag{A.239}$$

and

$$|q_j - g_j| \leq (Q_j - t) + \max(Q_{j-1} - t, 0) \tag{A.240}$$

$$\leq 2 \max\left(\max_j Q_j - t, 0 \right) \leq 2|s - t| + 24b\sqrt{K} \ln \frac{2K}{\delta}; \tag{A.241}$$

for $j = K$,

$$|Q_K - G_K| \leq \begin{cases} Q_K - t & \text{if } Q_K \geq t \\ t - Q_K & \text{else} \end{cases} = \begin{cases} Q_K - t & \text{if } Q_K \geq t \\ t - \left(s + \sum_{m=1}^{K} \xi_m\right) & \text{else} \end{cases} \tag{A.242}$$

$$\leq |s-t| + 12b\sqrt{K} \ln \frac{2K}{\delta}, \tag{A.243}$$

and

$$|q_K - g_K| \leq |Q_K - t| + \max(Q_{K-1} - t, 0) \leq 2|s-t| + 24b\sqrt{K} \ln \frac{2K}{\delta}. \tag{A.244}$$

Finally, for $\| \cdot \|_1$,

$$\|q-g\|_1 = \sum_{j=1}^{l} |Q_j - Q_{j-1} - (G_j - G_{j-1})| + \sum_{j=r}^{K} |Q_j - Q_{j-1} - (G_j - G_{j-1})| \tag{A.245}$$

$$= \sum_{j=1}^{l} (Q_j - Q_{j-1}) + |q_r - g_r| + \sum_{j=r+1}^{K-1} (Q_j - Q_{j-1}) + |q_K - g_K| \tag{A.246}$$

$$= Q_l - Q_1 + |q_r - g_r| + Q_{K-1} - Q_r + |q_K - g_K| \tag{A.247}$$

$$\leq -Q_1 + |q_r - g_r| + Q_{K-1} - t + |q_K - g_K| \tag{A.248}$$

$$\leq 12b\sqrt{K} \ln \frac{2K}{\delta} + 2\left(|s-t| + 12b\sqrt{K} \ln \frac{2K}{\delta}\right) + \left(s + 12b\sqrt{K} \ln \frac{2K}{\delta}\right) - t \tag{A.249}$$

$$\leq 3|s-t| + 48b\sqrt{K} \ln \frac{2K}{\delta}, \tag{A.250}$$

keeping in mind that $1 \leq l$ and $r \leq K-1, K$ on line 3 and onward.

The result follows by taking a final union bound over the two events above and rescaling $\delta \leftarrow \delta/2$. QED.

Proof of Theorem 5.4. Because $p_a \geq 0$, by Lemma A.7, with probability at least $1 - \delta$,

$$|p_a - \tilde{p}_a| = \left| p_a - \max\left(p_a + \sum_{k \in \mathcal{Y}} \text{Laplace}\left(0, \frac{2}{\varepsilon N}\right), 0\right) \right| \tag{A.251}$$

$$= \left| \max\left(\sum_{k \in \mathcal{Y}} \text{Laplace}\left(0, \frac{2}{\varepsilon N}\right), -p_a\right) \right| \tag{A.252}$$

$$\leq \left| \sum_{k \in \mathcal{Y}} \text{Laplace}\left(0, \frac{2}{\varepsilon N}\right) \right| \leq O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon N} \ln \frac{M|\mathcal{Y}|}{\delta} \right). \tag{A.253}$$

Next, define

$$\hat{r}_a(k) = \frac{1}{N_a}\sum_{i=1}^{N} \mathbb{1}\big[h \circ f(x^{(i)}, a^{(i)}) = k,\ a^{(i)} = a\big], \qquad \check{r}_a(k) = \frac{1}{\tilde{p}_a}\tilde{r}(a, k), \tag{A.254}$$

where $N_a = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[a^{(i)} = a]$. Note that $\check{F}_a(k) = \sum_{\ell \le k} \check{r}_a(k)$. By the triangle inequality,

$$\begin{aligned}
&\|r_a - \tilde{r}_a\|_\infty && \text{(A.255)}\\
&\le \|\tilde{r}_a - \check{r}_a\|_\infty + \|\check{r}_a - \hat{r}_a\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.256)}\\
&= \left\|\tilde{r}_a - \frac{1}{\tilde{p}_a}\check{r}(a,\cdot)\right\|_\infty + \left\|\frac{1}{\tilde{p}_a}\check{r}(a,\cdot) - \frac{1}{\hat{p}_a}\hat{r}(a,\cdot)\right\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.257)}\\
&\le \left\|\tilde{r}_a - \frac{1}{\hat{p}_a}\check{r}(a,\cdot)\right\|_\infty + 2\left\|\frac{1}{\tilde{p}_a}\check{r}(a,\cdot) - \frac{1}{\hat{p}_a}\check{r}(a,\cdot)\right\|_\infty + \left\|\frac{1}{\hat{p}_a}\check{r}(a,\cdot) - \frac{1}{\hat{p}_a}\hat{r}(a,\cdot)\right\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.258)}\\
&= \frac{1}{\hat{p}_a}\|\hat{p}_a\tilde{r}_a - \check{r}(a,\cdot)\|_\infty + \frac{2}{\hat{p}_a}|\hat{p}_a - \tilde{p}_a| + \frac{1}{\hat{p}_a}\|\check{r}(a,\cdot) - \hat{r}(a,\cdot)\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.259)}\\
&\le \frac{1}{\hat{p}_a}\|\tilde{p}_a\tilde{r}_a - \hat{p}_a\tilde{r}_a\|_\infty + \frac{1}{\hat{p}_a}\|\tilde{p}_a\tilde{r}_a - \check{r}(a,\cdot)\|_\infty + \frac{2}{\hat{p}_a}|\hat{p}_a - \tilde{p}_a| \\
&\qquad + \frac{1}{\hat{p}_a}\|\check{r}(a,\cdot) - \hat{r}(a,\cdot)\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.260)}\\
&\le \frac{1}{\hat{p}_a}\|\tilde{p}_a\tilde{r}_a - \check{r}(a,\cdot)\|_\infty + \frac{3}{\hat{p}_a}|\hat{p}_a - \tilde{p}_a| + \frac{1}{\hat{p}_a}\|\check{r}(a,\cdot) - \hat{r}(a,\cdot)\|_\infty + \|\hat{r}_a - r_a\|_\infty && \text{(A.261)}\\
&\le O\left(\frac{1}{\hat{p}_a}\left(|\hat{p}_a - \tilde{p}_a| + \frac{\sqrt{|\mathcal{Y}|}}{\varepsilon N}\ln\frac{M|\mathcal{Y}|}{\delta} + \frac{1}{\varepsilon N}\ln\frac{M|\mathcal{Y}|}{\delta}\right) + \sqrt{\frac{1}{p_a N}\ln\frac{M|\mathcal{Y}|}{\delta}}\right) && \text{(A.262)}\\
&\le O\left(\frac{1}{\hat{p}_a}\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon N}\ln\frac{M|\mathcal{Y}|}{\delta} + \frac{\sqrt{|\mathcal{Y}|}}{\varepsilon N}\ln\frac{M|\mathcal{Y}|}{\delta} + \frac{1}{\varepsilon N}\ln\frac{M|\mathcal{Y}|}{\delta}\right) + \sqrt{\frac{1}{p_a N}\ln\frac{M|\mathcal{Y}|}{\delta}}\right) && \text{(A.263)}\\
&\le O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon\hat{p}_a N}\ln\frac{M|\mathcal{Y}|}{\delta} + \sqrt{\frac{1}{p_a N}\ln\frac{M|\mathcal{Y}|}{\delta}}\right) && \text{(A.264)}\\
&\le O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon p_a N}\ln\frac{M|\mathcal{Y}|}{\delta} + \sqrt{\frac{1}{p_a N}\ln\frac{M|\mathcal{Y}|}{\delta}}\right) && \text{(A.265)}
\end{aligned}$$

by Eq. (A.253) and Lemmas A.8 and A.9; the bound on the $\|\check{r}(a,\cdot) - \hat{r}(a,\cdot)\|_\infty$ follows the same analysis in the proof of Lemma A.9 for the first term.

Similarly,

$$
\begin{aligned}
\|r_a - \tilde{r}_a\|_1 &\le \frac{1}{\hat{p}_a}\|\tilde{p}_a\tilde{r}_a - \check{r}(a,\cdot)\|_1 + \frac{3}{\hat{p}_a}|\hat{p}_a - \tilde{p}_a| + \frac{1}{\hat{p}_a}\|\check{r}(a,\cdot) - \hat{r}(a,\cdot)\|_1 + \|\hat{r}_a - r_a\|_1 \qquad \text{(A.266)} \\
&\le O\left(\frac{|\mathcal{Y}|}{\varepsilon p_a N}\ln\frac{M|\mathcal{Y}|}{\delta} + \sqrt{\frac{|\mathcal{Y}|}{p_a N}\ln\frac{M}{\delta}}\right), \qquad \text{(A.267)}
\end{aligned}
$$

and

$$
\begin{aligned}
D_{\mathrm{KS}}(r_a, \tilde{r}_a) &\le \frac{1}{\hat{p}_a}\max_k\left|\sum_{\ell\le k}(\tilde{p}_a\tilde{r}_a(\ell) - \check{r}(a,\ell))\right| + \frac{3}{\hat{p}_a}|\hat{p}_a - \tilde{p}_a| \\
&\quad + \frac{1}{\hat{p}_a}\max_k\left|\sum_{\ell\le k}(\check{r}(a,\cdot) - \hat{r}(a,\cdot))\right| + D_{\mathrm{KS}}(\hat{r}_a, r_a) \qquad \text{(A.268)} \\
&\le O\left(\frac{\sqrt{|\mathcal{Y}|}}{\varepsilon p_a N}\ln\frac{M|\mathcal{Y}|}{\delta} + \sqrt{\frac{|\mathcal{Y}|}{p_a N}\ln\frac{M|\mathcal{Y}|}{\delta}}\right). \qquad \text{(A.269)}
\end{aligned}
$$

QED.

Finally, for the proof of Theorem 5.5, we need the following technical lemma on the difference of W_2^2 distances, adapted from [231]:

Lemma A.10. Let p, p', q, q' be distributions whose supports are contained in a ball of radius R in $\mathbb{R}^d$ centered at 0, then

$$
\begin{aligned}
&\left|W_2^2(p,q) - W_2^2(p',q')\right| \\
&\quad\le \left|\int \|x\|_2^2\,\mathrm{d}(p-p')(x)\right| + \left|\int \|x\|_2^2\,\mathrm{d}(q-q')(x)\right| \\
&\qquad + 2R\sup_{\text{convex } f\in\mathrm{Lip}(1)}\left|\int f(x)\,\mathrm{d}(p-p')(x)\right| + 2R\sup_{\text{convex } g\in\mathrm{Lip}(1)}\left|\int g(x)\,\mathrm{d}(q-q')(x)\right| \qquad \text{(A.270)} \\
&\quad\le 4RW_1(p,p') + 4RW_1(q,q'). \qquad \text{(A.271)}
\end{aligned}
$$

The last line follows from the dual representation of W_1 distance for distributions with bounded support:

$$
W_1(p,q) = \sup_{f\in\mathrm{Lip}(1)}\left|\int f(x)\,\mathrm{d}(p-q)(x)\right|, \tag{A.272}
$$

and the fact that $x \mapsto \|x\|_2^2$ is $2R$-Lipschitz on the centered ball of radius R.

Also, recall the fact that the W_1 distance of distributions supported on a ball of radius R can be upper bounded by total variation distance as follows:

$$
\begin{aligned}
W_1(p,q) &= \inf_{\gamma\in\Pi(p,q)} \int d(x,y)\,\mathrm{d}\gamma(x,y) && \text{(A.273)}\\
&\le 2R \inf_{\gamma\in\Pi(p,q)} \int \mathbb{1}[x\neq y]\,\mathrm{d}\gamma(x,y) && \text{(A.274)}\\
&= 2R\left(1-\sup_{\gamma\in\Pi(p,q)} \int \mathbb{1}[x=y]\,\mathrm{d}\gamma(x,y)\right) && \text{(A.275)}\\
&= 2R\left(1-\int \min(p(x),q(x))\,\mathrm{d}x\right) && \text{(A.276)}\\
&= 2R\int \max(0,q(x)-p(x))\,\mathrm{d}x && \text{(A.277)}\\
&= R\int |q(x)-p(x)|\,\mathrm{d}x && \text{(A.278)}\\
&= R\|p-q\|_1 && \text{(A.279)}\\
&= 2RD_{\mathrm{TV}}(p,q), && \text{(A.280)}
\end{aligned}
$$

where line 6 is because $\int (p(x)-q(x))\,\mathrm{d}x = 0$.

And, note the following simple fact regarding one-dimensional optimal transports T^* under the squared cost. In the special case where T^* is a Monge transportation plan (meaning, a deterministic mapping), the lemma is equivalent to saying that T^* is a nondecreasing function (see the last panel of Fig. 5.1 for a picture):

Lemma A.11. Let p,q be two distributions over $[K]$, and $\gamma \in \arg\min_{\gamma'\in\Pi(p,q)} \sum_{m,\ell}(m-\ell)^2\gamma(m,\ell)$, then for all $k\in[K]$, there exists $m_k\in[K]$ such that

$$
g_k(m) \coloneqq \sum_{\ell=1}^{k} \frac{\gamma(m,\ell)}{p(m)} \begin{cases} = 1 & \text{if } m < m_k \\ \in [0,1] & \text{if } m = m_k \\ = 0 & \text{if } m > m_k, \end{cases} \quad \forall m\in[K] \text{ where } p(m)>0. \tag{A.281}
$$

Proof. Let $k\in[K]$ be arbitrary. Suppose to the contrary that $\nexists m_k$ such that Eq. (A.281) holds, then $\exists m$ where $g_k(m) < g_k(m+1)$ or $1 > g_k(m) \geq g_k(m+1) > 0$. We show that either of these cases contradicts the optimality of γ.

In both cases, there must exist $l \le k < r$ such that $\gamma(m,r), \gamma(m+1,l) \ge q$ for some $q>0$, because $\sum_{\ell=1}^{k}\gamma(m+1,\ell) = p(m+1)g_k(m+1) > 0$ and $\sum_{\ell=k+1}^{K}\gamma(m,\ell) = p(m)\big(1-g_k(m)\big) > 0$. Then a coupling γ' with a lower cost can be constructed by (partially) exchanging the two entries:

$$\gamma'(i,\ell) = \begin{cases} \gamma(m,r) - q & \text{if } i = m, \ell = r \\ \gamma(m,l) + q & \text{if } i = m, \ell = l \\ \gamma(m+1,r) + q & \text{if } i = m+1, \ell = r \\ \gamma(m+1,l) - q & \text{if } i = m+1, \ell = l \\ \gamma(i,\ell) & \text{else.} \end{cases}$$

We verify that it has a lower cost than γ:

$$\sum_{i,\ell} (i-\ell)^2(\gamma'(i,\ell) - \gamma(i,\ell)) \tag{A.282}$$

$$= -(m-r)^2 q + (m-l)^2 q + (m+1-r)^2 q - (m+1-l)^2 q \tag{A.283}$$

$$= 2(l-r)q \tag{A.284}$$

$$< 0. \tag{A.285}$$

QED.

Proof of Theorem 5.5. The proof proceeds by decomposing the excess risk and fairness violation using triangle inequality, then bounding the difference between each pair of decomposed terms:

- Denote the (unconstrained) Bayes-optimal predictor by f^*, and its output distribution conditioned on group a by $r^*_a = f^*\sharp(X \mid A = a, a)$. We will bound its difference from $r_a = f\sharp(X \mid A = a, a)$, the output distribution of the predictor f being post-processed.

- Given a discretizer $h : \mathbb{R} \to \mathcal{Y}$, denote the conditional discretized output distribution of f^* by $\check{r}^*_a = h\sharp r^*_a$, and that of f by $\check{r}_a = h\sharp r_a$. We will compare r^*_a to its discretized version $\check{r}^*_a$, and $\check{r}_a$ to $\tilde{r}_a$, the private empirical conditional PMF of the discretized output of f.

- We will bound the difference between the group marginal distribution estimated privately from the samples, $\tilde{p}_a$ (Line 8 of Algorithm 5.1), from the true $p_a = \mathbb{P}[A = a]$.

- Let $\{\tilde{q}'_a\}_{a\in[M]}$ be the optimal solution to Eq. (5.12) on inputs $(\{\tilde{r}_a\}_{a\in[M]}, \{\tilde{p}_a\}_{a\in[M]}, \alpha)$, and $\{\tilde{q}_a\}_{a\in[M]}$ the optimal shape of LP3$(\{\tilde{r}_a\}_{a\in[M]}, \{\tilde{p}_a\}_{a\in[M]}, \alpha)$. The difference between $\tilde{q}'_a$ and $\tilde{q}_a$ is that the support of the latter is restricted to $\mathcal{Y}$ in LP3, otherwise they both optimize the same objective. We will compare and bound these two quantities.

- Lastly, recall that the fair predictor returned from Algorithm 5.1 has the form $\bar{f}(x,a) = T_a \circ h \circ f(x,a)$, where T_a is the optimal transport from $\tilde{r}_a$ to $\tilde{q}_a$. The $\tilde{q}_a$'s will be compared to the shape of the Bayes optimal fair predictor $\bar{f}^*$, denoted by $\bar{r}_a^*$, via $\tilde{q}_a'$. Note that $\{\bar{r}_a^*\}_{a\in[M]}$ is the optimal solution to Eq. (5.12) on inputs $(\{r_a^*\}_{a\in[M]}, \{p_a\}_{a\in[M]}, \alpha)$.

Error Bound. Note that $R(\bar{f}) - R(\bar{f}^*) = ER(\bar{f}) - ER(\bar{f}^*)$. By the orthogonality principle,

$$\begin{aligned}
ER(\bar{f}) &= \mathbb{E}\Big[\big(\bar{f}(X,A) - f^*(X,A)\big)^2\Big] && \text{(A.286)}\\
&= \sum_{a\in[M]} p_a\, \mathbb{E}_{X|A=a}\Big[\big(\bar{f}(X,a) - f^*(X,a)\big)^2\Big] && \text{(A.287)}\\
&= \sum_{a\in[M]} p_a\, \mathbb{E}_{X|A=a}\Big[\big(\bar{f}(X,a) - f(X,a) + (f(X,a) - f^*(X,a))\big)^2\Big] && \text{(A.288)}\\
&= \sum_{a\in[M]} p_a \Big(\mathbb{E}_{X|A=a}\, \Big[\big(\bar{f}(X,a) - f(X,a)\big)^2 + (f(X,a) - f^*(X,a))^2 && \text{(A.289)}\\
&\qquad\qquad + 2\big(\bar{f}(X,a) - f(X,a)\big)(f(X,a) - f^*(X,a))\Big]\Big)\\
&\le \sum_{a\in[M]} p_a\, \mathbb{E}_{X|A=a}\Big[\big(\bar{f}(X,a) - f(X,a)\big)^2\Big] + \underbrace{3\,\mathbb{E}[|f(X,A) - f^*(X,A)|]}_{\mathcal{E}_1}, && \text{(A.290)}
\end{aligned}$$

where line 5 is because of the assumption that the images of $\bar{f}, f$ are contained in $[0,1]$. The second term on the last line is the L^1 excess risk of f; for the first term,

$$\begin{aligned}
&\mathbb{E}_{X|A=a}\Big[\big(\bar{f}(X,a) - f(X,a)\big)^2\Big]\\
&\quad= \mathbb{E}_{X|A=a}\big[(T_a \circ h \circ f(X,a) - f(X,a))^2\big] && \text{(A.291)}\\
&\quad= \mathbb{E}_{X|A=a}\big[(T_a \circ h \circ f(X,a) - h \circ f(X,a) + (h \circ f(X,a) - f(X,a)))^2\big] && \text{(A.292)}\\
&\quad\le \mathbb{E}_{X|A=a}\big[(T_a \circ h \circ f(X,a) - h \circ f(X,a))^2 + 3|h \circ f(X,a) - f(X,a)|\big] && \text{(A.293)}\\
&\quad\le \mathbb{E}_{X|A=a}\big[(T_a \circ h \circ f(X,a) - h \circ f(X,a))^2\big] + \frac{3}{2K} && \text{(A.294)}\\
&\quad= \sum_{j\le K} \check{r}_a(j)\, \mathbb{E}\big[(T_a(j) - j)^2\big] + \frac{3}{2K} && \text{(A.295)}\\
&\quad= \sum_{j\le K} \tilde{r}_a(j)\, \mathbb{E}\big[(T_a(j) - j)^2\big] + \sum_{j\le K} (\check{r}_a(j) - \tilde{r}_a(j))\, \mathbb{E}\big[(T_a(j) - j)^2\big] + \frac{3}{2K} && \text{(A.296)}\\
&\quad\le W_2^2(\tilde{r}_a, \tilde{q}_a) + \|\check{r}_a - \tilde{r}_a\|_1 + \frac{3}{2K} && \text{(A.297)}\\
&\quad\le W_2^2(\tilde{r}_a, \tilde{q}_a) + O\left(\sqrt{\frac{K}{p_a N}\ln\frac{MK}{\delta}} + \frac{K}{\varepsilon p_a N}\ln\frac{MK}{\delta}\right) + \frac{3}{2K}, && \text{(A.298)}
\end{aligned}$$

where line 4 is because h discretizes the input to the midpoint of the bin that it falls in, which displaces it by up to $(t-s)/2K \le 1/2K$ by assumption; line 5 is because $\check{r}_a(j) = \mathbb{P}[h \circ f(X,a) = j]$; the first term on line 7 is because T_a is the optimal transport from $\tilde{r}_a$ to $\tilde{q}_a$. Then, combining the above, by Theorem 5.4, with probability at least $1-\delta$,

$$ER(\bar{f}) \le \sum_{a\in[M]} \left(p_a W_2^2(\tilde{r}_a, \tilde{q}_a) + O\left(\sqrt{\frac{p_a K}{N} \ln \frac{MK}{\delta}} + \frac{K}{N\varepsilon} \ln \frac{MK}{\delta} \right) \right) + \mathcal{E}_1 + \frac{3}{2K} \tag{A.299}$$

$$\le \sum_{a\in[M]} p_a W_2^2(\tilde{r}_a, \tilde{q}_a) + \mathcal{E}_1 + \underbrace{O\left(\sqrt{\frac{MK}{N} \ln \frac{MK}{\delta}} + \frac{MK}{N\varepsilon} \ln \frac{MK}{\delta} \right)}_{\mathcal{E}_2} + \frac{3}{2K} \tag{A.300}$$

$$\le \sum_{a\in[M]} \left(\tilde{p}_a W_2^2(\tilde{r}_a, \tilde{q}_a) + |\tilde{p}_a - p_a| W_2^2(\tilde{r}_a, \tilde{q}_a) \right) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.301}$$

$$\le \sum_{a\in[M]} \left(\tilde{p}_a W_2^2(\tilde{r}_a, \tilde{q}_a) + |\tilde{p}_a - p_a| \right) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.302}$$

$$\le \sum_{a\in[M]} \tilde{p}_a W_2^2(\tilde{r}_a, \tilde{q}_a) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.303}$$

$$\le \sum_{a\in[M]} \tilde{p}_a W_2^2(\tilde{r}_a, h\sharp\tilde{q}'_a) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.304}$$

$$\le \sum_{a\in[M]} \tilde{p}_a (W_2(\tilde{r}_a, \tilde{q}'_a) + W_2(\tilde{q}'_a, h\sharp\tilde{q}'_a))^2 + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.305}$$

$$= \sum_{a\in[M]} \tilde{p}_a \left(W_2^2(\tilde{r}_a, \tilde{q}'_a) + 2W_2(\tilde{r}_a, \tilde{q}'_a) W_2(\tilde{q}'_a, h\sharp\tilde{q}'_a) + W_2^2(\tilde{q}'_a, h\sharp\tilde{q}'_a) \right) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{2K} \tag{A.306}$$

$$\le \sum_{a\in[M]} \tilde{p}_a W_2^2(\tilde{r}_a, \tilde{q}'_a) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K} \tag{A.307}$$

$$\le \sum_{a\in[M]} \tilde{p}_a W_2^2(\tilde{r}_a, q^*_a) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K}, \tag{A.308}$$

where line 6 follows by noting that $\{h\sharp\tilde{q}'_a\}_{a\in[M]}$ is a feasible shape for LP3, as it can be verified that $D_{\mathrm{KS}}(h\sharp\tilde{q}'_a, h\sharp\tilde{q}'_{a'}) \le \alpha$ given that $D_{\mathrm{KS}}(\tilde{q}'_a, \tilde{q}'_{a'}) \le \alpha$, $\forall a, a' \in [M]$ (hence restricting the support of the barycenter to $\mathcal{Y}$ introduces an additional error of $3/2K$), and the last line is because $\{\tilde{q}'_a\}_{a\in[M]}$ is a minimizer of Eq. (5.12) on inputs $(\{\tilde{r}_a\}_{a\in[M]}, \{\tilde{p}_a\}_{a\in[M]}, \alpha)$.

So for the suboptimality of $\bar{h}$, by Theorem 5.3,

$$
\begin{aligned}
&ER(\bar{f}) - ER(\bar{f}^*) \\
&\quad \leq \sum_{a\in[M]} \tilde{p}_a \big(W_2^2(\tilde{r}_a, q_a^*) - W_2^2(r_a^*, q_a^*) \big) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K} && \text{(A.309)} \\
&\quad \leq \sum_{a\in[M]} \tilde{p}_a \big(W_2^2(\tilde{r}_a, q_a^*) - (W_2(q_a^*, \check{r}_a^*) - W_2(r_a^*, \check{r}_a^*)) \big)^2 + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K} && \text{(A.310)} \\
&\quad \leq \sum_{a\in[M]} \tilde{p}_a \big(W_2^2(\tilde{r}_a, q_a^*) - W_2^2(\check{r}_a^*, q_a^*) + 2W_2(\check{r}_a^*, q_a^*) W_2(\check{r}_a^*, r_a^*) \big) + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K} && \text{(A.311)} \\
&\quad \leq \sum_{a\in[M]} \tilde{p}_a \big(W_2^2(\tilde{r}_a, q_a^*) - W_2^2(\check{r}_a^*, q_a^*) \big) + \frac{1}{K} + \mathcal{E}_1 + \mathcal{E}_2 + \frac{3}{K}, && \text{(A.312)}
\end{aligned}
$$

where the last line is because h is a transport from r_a^* to $\check{r}_a^*$ with displacements of at most $1/2K$; for the first term, by Lemma A.10 and Eq. (A.280),

$$
\begin{aligned}
&\sum_{a\in[M]} \tilde{p}_a \big(W_2^2(\tilde{r}_a, q_a^*) - W_2^2(\check{r}_a^*, q_a^*) \big) \\
&\quad \leq 4 \sum_{a\in[M]} \tilde{p}_a W_1(\tilde{r}_a, \check{r}_a^*) && \text{(A.313)} \\
&\quad \leq 4 \sum_{a\in[M]} \tilde{p}_a (W_1(\tilde{r}_a, \check{r}_a) + W_1(\check{r}_a, r_a) + W_1(r_a, r_a^*) + W_1(r_a^*, \check{r}_a^*)) && \text{(A.314)} \\
&\quad \leq 4 \sum_{a\in[M]} \tilde{p}_a \left(\|\tilde{r}_a - \check{r}_a\|_1 + \frac{1}{2K} + \mathbb{E}_{X|A=a}[|f^*(X,a) - f(X,a)|] + \frac{1}{2K} \right) && \text{(A.315)} \\
&\quad \leq 4 \sum_{a\in[M]} \tilde{p}_a \, \mathbb{E}_{X|A=a}[|f^*(X,a) - f(X,a)|] + \mathcal{E}_2 + \frac{4}{K} && \text{(A.316)} \\
&\quad \leq 4 \sum_{a\in[M]} (p_a - p_a + \tilde{p}_a) \, \mathbb{E}_{X|A=a}[|f^*(X,a) - f(X,a)|] + \mathcal{E}_2 + \frac{4}{K} && \text{(A.317)} \\
&\quad \leq 4 \sum_{a\in[M]} p_a \, \mathbb{E}_{X|A=a}[|f^*(X,a) - f(X,a)|] + 4 \sum_{a\in[M]} |\tilde{p}_a - p_a| + 4\mathcal{E}_2 + \frac{4}{K} && \text{(A.318)} \\
&\quad \leq 4\mathcal{E}_1 + \mathcal{E}_2 + \frac{4}{K}, && \text{(A.319)}
\end{aligned}
$$

the third inequality is because the joint distribution of $(f(X,a), f^*(X,a))$ is a valid coupling belonging to $\Pi(r_a, r_a^*)$ that incurs a transportation cost of

$$
\mathbb{E}[|f^*(X,a) - f(X,a)|] = \int |y - y^*| \, \mathrm{d}\,\mathbb{P}[f(X,a) = y, f^*(X,a) = y^*] \geq W_1(r_a, r_a^*). \qquad \text{(A.320)}
$$

Putting everything together, and with a union bound over the two events above, gives the result in the theorem statement.

Fairness Guarantee. Let $\bar{r}_a$ denote the output distribution of the post-processed predictor $\bar{f}$ conditioned on group a. Using triangle inequality, for any $a, a' \in [M]$,

$$D_{\mathrm{KS}}(\bar{r}_a, \bar{r}_{a'}) \leq D_{\mathrm{KS}}(\tilde{q}_a, \tilde{q}_{a'}) + D_{\mathrm{KS}}(\bar{r}_a, \tilde{q}_a) + D_{\mathrm{KS}}(\bar{r}_{a'}, \tilde{q}_{a'}) \leq \alpha + D_{\mathrm{KS}}(\bar{r}_a, \tilde{q}_a) + D_{\mathrm{KS}}(\bar{r}_{a'}, \tilde{q}_{a'}), \tag{A.321}$$

where

$$\begin{aligned}
&D_{\mathrm{KS}}(\bar{r}_a, \tilde{q}_a) \\
&\quad = \max_k \left| \sum_{j \leq k} (\bar{r}_a(j) - \tilde{q}_a(j)) \right| && \text{(A.322)} \\
&\quad = \max_k \left| \sum_{j \leq k} \big(\mathbb{P}[\bar{f}(X, a) = j \mid A = a] - \tilde{q}_a(j) \big) \right| && \text{(A.323)} \\
&\quad = \max_k \left| \sum_{j \leq k} \left(\sum_{\ell \leq K} \check{r}_a(\ell) \, \mathbb{P}[T_a(\ell) = j \mid A = a] - \sum_{\ell \leq K} \tilde{r}_a(\ell) \, \mathbb{P}[T_a(\ell) = j \mid A = a] \right) \right| && \text{(A.324)} \\
&\quad = \max_k \left| \sum_{\ell = 1}^{K} (\check{r}_a(\ell) - \tilde{r}_a(\ell)) \sum_{j \leq k} \mathbb{P}[T_a(\ell) = j \mid A = a] \right| && \text{(A.325)} \\
&\quad \leq \max_k \left(\left| \sum_{\ell \leq m_k - 1} (\check{r}_a(\ell) - \tilde{r}_a(\ell)) \right| + |(\check{r}_a(m_k) - \tilde{r}_a(m_k)) \, \mathbb{P}[T_a(m_k) \leq k]| \right) && \text{(A.326)} \\
&\quad \leq D_{\mathrm{KS}}(\check{r}_a, \tilde{r}_a) + \max_k |\check{r}_a(m_k) - \tilde{r}_a(m_k)| && \text{(A.327)} \\
&\quad \leq O\left(\frac{\sqrt{K}}{\varepsilon p_a N} \ln \frac{MK}{\delta} + \sqrt{\frac{K}{p_a N} \ln \frac{MK}{\delta}} \right); && \text{(A.328)}
\end{aligned}$$

line 3 is because T_a is a transport from $\tilde{r}_a$ to $\tilde{q}_a$, line 5 uses Lemma A.11 and the fact that $\sum_{\ell \leq j} \gamma_a(m, \ell) / \tilde{r}_a(m) = \mathbb{P}[T_a(m) \leq j]$ (γ_a is defined on Line 12 in Algorithm 5.1), and line 7 is by Theorem 5.4. QED.

APPENDIX B: EXPERIMENT DETAILS

The setup described here applies to the experiments in Chapters 3, 4 and 6, but not to the regression experiments in Section 5.4, whose setup was described in Section 5.4.5.

B.1 EVALUATION PROTOCOL

Each algorithm is run with varying fairness tolerance settings to generate different accuracy-fairness tradeoffs, and repeated 5 times under different random dataset splits and random seeds. Model selection is performed on the validation set. Unless otherwise specified, the primary performance metric is classification accuracy, and fairness violations are measured using the approximate fairness definitions in Section 2.2.1, based on pairwise differences. An exception is the violation measurement on CivilComments in Chapter 6 (Eq. (6.8)), where we extend the definition to overlapping groups and adopt the weighted mean-difference measure.

For a fixed random seed, we construct the accuracy-fairness tradeoff curve as follows. From the collection of classifiers trained under different tolerance settings, we first remove classifiers that are not Pareto-optimal on the validation set, that is, those dominated by another classifier with both higher accuracy and lower violation. The remaining classifiers are sorted by their violation. A continuous tradeoff curve is then obtained by interpolating between adjacent models, spanning validation violations from $V_{\min}^{(\text{seed})}$ to $V_{\max}^{(\text{seed})}$.

To aggregate tradeoff curves across random seeds and estimate uncertainty, we evaluate each curve on the *test set* at six equally spaced validation-violation percentage levels $t \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. At each level t, we identify (possibly by interpolation) the classifier whose validation-set fairness violation is $t \cdot (V_{\max}^{(\text{seed})} - V_{\min}^{(\text{seed})})$, then record its test-set accuracy and violation. These results are then averaged across seeds at each percentage level to obtain the mean and standard deviation, which are used to plot the aggregated tradeoff curves throughout the thesis.

B.2 HYPERPARAMETERS

In all experiments, we train logistic regression models using the `LogisticRegression` method from scikit-learn [232] with `max_iter` set to 10000 and all other settings left at their defaults; for the Reductions experiments in Chapter 6, we instead use the GPU-accelerated cuML implementation.[1] Gradient-boosted decision tree (GBDT) models are trained with LightGBM's `LGBMClassifier` using default hyperparameters [233].

The multilayer perceptron (MLP) models in Sections 3.4 and 4.5.4 experiments have three hidden layers of sizes $(512, 256, 128)$ with ReLU activation, and are optimized with Adam [234] for 20 epochs (batch size 128, initial learning rate 1e-3) with multiplicative learning rate decay of 0.8 per epoch.

For BERT [108] in Sections 3.4 and 4.5.4 experiments, we use the Transformers API.[2] Starting from the `bert-base-uncased` checkpoint, we attach a randomly initialized linear classification head and fine-tune with AdamW [235] for 3 epochs (batch size 32, learning rate 2e-5), applying gradient norm clipping at 1.0, and a linear schedule with warmup for the first 10% of training steps via Transformers library's `get_linear_schedule_with_warmup` method.

B.2.1 Fair Algorithms

LinearPost. We use the Gurobi optimizer to solve the linear programs involved,[3] and sweep the fairness tolerance parameter α over 16 evenly spaced values between $\alpha_{\min} = 0.001$ and $\alpha_{\max}$, where $\alpha_{\max}$ is set to the fairness violation on the training set without any mitigation (namely, the training-set violation of the classifier returned under $\alpha = \infty$). This produces 16 classifiers spanning different accuracy-fairness tradeoffs.

For experiments under EO fairness on the multiclass BiasBios dataset, we report results from LinearPost configured for multiclass TPR parity. This is because the fairness violation is mostly dominated by disparities in recall, rather than by other types of misclassification errors (reflected in the off-diagonal entries of the confusion matrix). Additionally, the run time is relatively long due to the large number of classes ($K = 28$) in BiasBios.

Robust LinearPost. We sweep the fairness tolerance parameter similar to above, except that the sweep between $\alpha_{\min} = 0.001$ and $\alpha_{\max}$ is performed evenly in the log space.

The number of robustification iterations in Algorithm 4.2 is limited to $T = 20$, and the tolerance parameter for the pessimization step is set to $\tau = 0.001$. We use one-hidden-layer neural networks with width 128 and LeakyReLU activation to generate the uncertainty set modeling bounded shifts (Section 4.4.2). During pessimization, these models are optimized to maximize fairness violation via full-batch gradient ascent using Adam (default hyperpa-

[1] `https://github.com/rapidsai/cuml`

[2] `https://huggingface.co/docs/transformers`

[3] `https://www.gurobi.com`

rameters, learning rate 0.01) for 1000 epochs. To reduce variance, we warm-start training by minimizing only the regularization terms for 1000 epochs, so that the perturbed distribution equals the reference distribution initially. We run 5 trials with different random initializations and select the model that induces the highest fairness violation.

MBS [99]. This post-processing algorithm is based on the same underlying representation result as LinearPost, and therefore returns classifiers of the same form. However, it is limited to the binary class and binary group setting, and is only applicable to SP and EO fairness. In the authors' implementation, the parameters of the post-processing transformation are optimized via a (heuristic) grid search on labeled training examples, with the number of candidate parameter combinations depending on the sample size; this procedure has time complexity exponential in the number of fairness constraints.

We use the code released by the authors.[4] The number of training examples is limited to $M = 5000$ due to long runtime, and the `threshold` parameter for controlling the accuracy-fairness tradeoff is swept in the same manner as the fairness tolerance α in LinearPost.

FairProjection [103]. This post-processing algorithm differs from LinearPost and MBS in that it outputs probabilities over labels (soft scores) rather than hard class assignments. The optimization objective is to minimize the divergence between the transformed and original scores (we use KL divergence, also denoted FairProjection-KL), subject to fairness constraints being satisfied on the distribution induced by the transformed scores.

We use the code released by the authors.[5] We sweep the `tol` parameter for controlling the accuracy-fairness tradeoff over the set $\{0.9, 0.8, 0.7, 0.6, 0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.001\}$.

As with LinearPost, when evaluating EO fairness on BiasBios, we report results from FairProjection configured for multiclass TPR parity.

Reductions [73]. This is an in-processing algorithm based on a two-player game formulation of the fair classification problem. The algorithm relies on a cost-sensitive classification oracle and techniques from no-regret learning, and it returns a randomized ensemble of classifiers that satisfy the specified fairness constraints.

We use the implementation provided in the AIF360 package with default hyperparameters [236], and sweep the `eps` parameter for "allowed fairness constraint violation" over the set $\{100, 50, 20, 10, 5, 2, 1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001\}$.

[4] `https://github.com/chenw20/BiasScore`

[5] `https://github.com/HsiangHsu/Fair-Projection`

Adversarial [84, 85]. Adversarial debiasing is a fair representation learning algorithm (in-processing), where a neural-network-based classifier is trained while enforcing alignment of the penultimate layer's feature distributions across groups. Distributional alignment is achieved through a setup analogous to generative adversarial networks [86, 87], by minimizing Jensen-Shannon divergence (corresponds to training the adversary with a cross-entropy loss).

We use our own implementation. When the base model is an MLP (described in Appendix B.2), the adversary is a two-hidden-layer ReLU MLP with widths $(1024, 512)$, trained jointly with the base model via a gradient reversal layer, but with a slightly larger learning rate of 2e-3. When the base model is BERT, the adversary MLP uses hidden sizes $(2048, 1024)$ and a learning rate of 2e-4. For (multiclass) TPR/FPR parity and equalized odds, the adversary needs to match conditional (on label Y) feature distributions across sensitive groups; this is implemented by injecting label information into the adversary's input following CDAN [237].

The accuracy-fairness tradeoff here is controlled by the multiplier applied to the reversed gradient from the adversary to the base model (which provides signal for distribution matching), which we sweep over $\{0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$.

For experiments under multiclass TPR parity, we report results from Adversarial configured for equalized odds, which imposes a stronger requirement.

MinDiff [206]. This is another in-processing algorithm based on fair representation learning, but unlike Adversarial, which matches distributions of intermediate feature representations, MinDiff aligns distributions of the model's output probabilities (for example, on MLP, the outputs after the final softmax activation). More specifically, it adds a regularization term to the training objective that penalizes the distance between the relevant distributions, measured using *maximum mean discrepancy* (MMD) [89]. MMD admits closed-form, differentiable expressions; following Prost et al. [206], we use a Gaussian kernel $k(x, x') = \exp(-\|x - x'\|^2/\sigma^2)$ with bandwidth $\sigma = 0.1$.

We use our own implementation. In binary group settings, we directly minimize the MMD between the two groups' distributions. In multigroup (including overlapping groups) settings, we match each group's distribution to the overall distribution. For instance, letting $S \in \Delta^K$ denote the class probabilities output by the model, the MinDiff regularization term for multigroup EO is

$$\frac{1}{MK} \sum_{a \in [M], k \in [K]} D_{\mathrm{MMD}} \Big(\overbrace{(S|A = a, Y = k)}^{\text{group } a\text{'s distribution conditioned on class } k}, \underbrace{(S|Y = k)}_{\text{overall distribution conditioned on class } k} \Big). \tag{B.1}$$

For experiments in Chapter 6 that train a fair classification head using MinDiff, we employ a single-hidden-layer ReLU network with width 512 and a final softmax output, optimized with Adam for 3 epochs (batch size 512, learning rate 10^{-4}). The regularization strength is swept over $\{0, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1, 1.5, 2, 2.5, 3, 3.5, 4, 6, 8, 10\}$.

Similar to Adversarial, for experiments under multiclass TPR parity, we report results from MinDiff configured for equalized odds.

B.3 DATASETS

Adult [238]. This dataset contains tabular features describing individuals' socioeconomic information. The task is to predict whether an individual's annual income exceeds \$50,000 in US dollars (binary classification). We take (biological) sex as the sensitive attribute (binary).

ACSIncome [109]. The task is the same as in Adult, but the features are drawn from the U.S. Census Bureau's American Community Survey Public Use Microdata Sample (ACS PUMS), which is more recent and substantially larger. We download the dataset through the folktables package.[6]

We consider either a binary classification task of predicting whether annual income exceeds \$50,000 (ACSIncome2), or a multiclass variant that partitions income into five approximately equal-frequency buckets (ACSIncome5); the buckets are (0, 15,000], (15,000, 30,000], (30,000, 48,600], (48,600, 78,030], and (78,030, ∞).

For the sensitive attribute, we use either sex (ACSIncomeX-Sex) or race (ACSIncomeX-Race). For race, we group individuals into five categories, mirroring that of Adult: "White", "Asian, Native Hawaiian or Other Pacific Islander", "Black or African American", "American Indian or Alaska Native", and "Other".

For the geographic shift experiments in Chapter 4, we further partition the dataset by the individual's state of residence (including Puerto Rico), which is denoted with a suffix. For example, ACSIncome2-Sex-CA refers to the binary task with sex as the sensitive attribute restricted to individuals residing in California (CA).

COMPAS [11]. This dataset contains tabular features such as age, charge degree (and, when used, the natural language charge description), number of prior convictions, and length of stay in jail. The task is to predict whether an individual will recidivate within two years (binary classification). We use race as the sensitive attribute and restrict the dataset to individuals labeled as either "African-American" or "Caucasian".

[6] `https://github.com/socialfoundations/folktables`

BiasBios [65]. Given a short biography, the task is to classify the person's occupation (28-way classification). The sensitive attribute is sex (binary), which is not explicitly given but often inferable from the biography. We use the version scrapped and hosted by Ravfogel et al. [194], containing 393,423 examples in total.

CivilComments [13]. The task is to detect whether a public comment is toxic (binary), and the sensitive attribute is the religion(s) mentioned in the comment (Christianity, Judaism, Islam, Hinduism, Buddhism). Since comments may mention multiple religions (or none), the groups are overlapping abd the sensitive attribute is multilabel, $A \in \{0,1\}^5$.

Table B.1: Examples from each dataset used in Chapter 6 experiments. For tabular datasets, the examples are shown after list serialization.

Dataset	Example
Adult (raw encoding)	age: 39 workclass: State-gov education: Bachelors education-num: 13 marital-status: Never-married occupation: Adm-clerical relationship: Not-in-family capital-gain: 2174 capital-loss: 0 hours-per-week: 40 native-country: United-States
Adult	Age: 39 Class of worker: State government employee Educational attainment: Bachelor's degree Education level (numeric): 13 Marital status: Never married or under 15 years old Occupation: Administrative support and clerical workers Relationship: Other nonrelative Capital gain in the previous year: 2174 Capital loss in the previous year: 0 Hours worked per week: 40 Country of origin: United States

Table B.1: Examples from each dataset used in Chapter 6 experiments (continued).

Dataset	Example
ACSIncome	Age: 38 Class of worker: Employee of a private not-for-profit, tax-exempt, or charitable organization Educational attainment: Bachelor's degree Marital status: Never married or under 15 years old Occupation recode for 2018 and later based on 2018 OCC codes: MGR-Social And Community Service Managers Place of birth (Recode): South Carolina/SC Relationship: Reference person Usual hours worked per week past 12 months: 50
COMPAS	Age at the time of survey: 42 Charge degree: Felony Charge description: Tampering With Physical Evidence Number of prior convictions: 1 Length of stay in jail: 1
BiasBios	She earned her B.A. from the State University of New York at Geneseo and M.A. and certification from the University at Buffalo. She is currently serving as GALA's Chair. Her research interests include identity development and gender within the English Language Arts classroom. She hopes to continue to develop GALA into a community of educators dedicated to how gender impacts learning.
CivilComments	The Philippian hymn can be interpreted various ways. It is thought to have preceded Paul and may contain the concept of the First and Second Adam theology that Paul would reference in 1 Cor and Romans. "Emptying" may refer to Jesus emptying himself of the desire to be godlike that caused Adam's downfall (Eve appears to have dropped out of the picture).

B.3.1 Dataset Split Sizes

Table B.2: Dataset split sizes used in Section 4.5 geographic shift, group label noise, and worst-case covariate shift experiments (under random seed 33; sizes vary slightly across seeds). For post-processing fair algorithms, the train examples are further divided into pre-training and post-processing splits.

	Train			
Dataset	Pre-Train	Post-Process	Validation	Test
ACSIncome2-Sex-CA	97,798	19,586	19,535	58,746
ACSIncome2-Sex-FL	49,589	9,789	9,800	29,747
ACSIncome2-Sex-AL	-	-	2,258	6,678
ACSIncome2-Sex-AZ	-	-	3,407	9,942
ACSIncome2-Sex-CO	-	-	3,131	9,504
ACSIncome2-Sex-GA	-	-	5,114	15,103
ACSIncome2-Sex-IL	-	-	6,520	20,283
ACSIncome2-Sex-IN	-	-	3,495	10,467
ACSIncome2-Sex-KY	-	-	2,208	6,718
ACSIncome2-Sex-LA	-	-	2,125	6,115
ACSIncome2-Sex-MA	-	-	4,017	11,981
ACSIncome2-Sex-MD	-	-	3,412	9,852
ACSIncome2-Sex-MI	-	-	5,054	15,133
ACSIncome2-Sex-MN	-	-	3,108	9,193
ACSIncome2-Sex-MO	-	-	3,137	9,536
ACSIncome2-Sex-NC	-	-	5,109	15,719
ACSIncome2-Sex-NJ	-	-	4,634	14,333
ACSIncome2-Sex-NY	-	-	10,231	30,819
ACSIncome2-Sex-OH	-	-	6,246	18,509
ACSIncome2-Sex-OR	-	-	2,191	6,541
ACSIncome2-Sex-PA	-	-	6,774	20,531
ACSIncome2-Sex-SC	-	-	2,581	7,404
ACSIncome2-Sex-TN	-	-	3,455	10,124
ACSIncome2-Sex-TX	-	-	13,548	40,824
ACSIncome2-Sex-VA	-	-	4,646	13,919
ACSIncome2-Sex-WA	-	-	4,096	12,066
ACSIncome2-Sex-WI	-	-	3,282	9,752

Table B.3: Dataset split sizes used in Section 3.4, and Section 4.5 calibration experiments. For post-processing fair algorithms, the train examples are further divided into pre-training and post-processing splits.

	Train			
Dataset	Pre-Train	Post-Process	Validation	Test
Adult	14,652	9,769	9,768	14,653
ACSIncome2-Sex	832,250	166,450	166,450	499,350
ACSIncome5-Race	832,250	166,450	166,450	499,350
COMPAS	1,583	1,056	1,055	1,584
BiasBios	196,711	39,342	39,343	118,027

Table B.4: Dataset split sizes used in Chapter 6 experiments.

Dataset	Train	Validation	Test
Adult (Sections 6.4.1 and 6.4.3)	2,000	2,000	20,000
Adult (Section 6.4.2)	100–20,000	5,000	20,000
ACSIncome2-Race	10,000	10,000	20,000
COMPAS	2,000	1,000	2,000
BiasBios	20,000	20,000	50,000
CivilComments	20,000	20,000	50,000

B.3.2 Prompt Templates for Section 6 Experiments

Listing B.1: Prompt templates used on the Adult dataset. For the group label prompt, the placeholder `{class_condition}` is replaced with either “`less than or equal to 50K`” or “`greater than 50K`” to condition the query on a (hypothetical) class label.

Template for class label (income)

```
Answer with a single letter.

Question: What is the annual income (USD) of the following individual?
{example}

A. Less than or equal to 50K
B. Greater than 50K

Answer:
```

Template for conditional group label (sex)

```
Answer with a single letter.

Question: What is the sex of the following individual? Their annual
income (USD) is {class_condition}.
{example}

A. Female
B. Male

Answer:
```

Template for embeddings

```
Question: What is the annual income (USD) and sex of the following
individual?
{example}

Answer:
```

Listing B.2: Prompt templates used on the ACSIncome2-Race dataset. For the group label prompt, the placeholder `{class_condition}` is replaced with either "`less than or equal to 50K`" or "`greater than 50K`" to condition the query on a (hypothetical) class label.

Template for class label (income)

```
Answer with a single letter.

Question: What is the annual income (USD) of the following individual?
{example}

A. Less than or equal to 50K
B. Greater than 50K

Answer:
```

Template for conditional group label (race)

```
Answer with a single letter.

Question: What is the race of the following individual? Their annual
income (USD) is {class_condition}.
{example}

A. White
B. Black or African American
C. American Indian or Alaska Native
D. Asian, Native Hawaiian or Pacific Islander
E. None of the above

Answer:
```

Template for embeddings

```
Question: What is the annual income (USD) and race of the following
individual?
{example}

Answer:
```

Listing B.3: Prompt templates used on the COMPAS dataset.

Template for class label (recidivism)

```
Answer with a single letter.

Question: Will the following individual recidivate within two years?
{example}

A. Unlikely to recidivate
B. Likely to recidivate

Answer:
```

Template for conditional group label (race)

```
Answer with a single letter.

Question: What is the race of the following individual? They are
{class_condition} within two years.
{example}

A. African American
B. Caucasian

Answer:
```

Template for embeddings

```
Question: Will the following individual recidivate within two years, and
what is their race?
{example}

Answer:
```

Listing B.4: Prompt templates used on the BiasBios dataset. For the group label prompt, the placeholder {class_condition} is replaced with either "unlikely to recidivate" or "likely to recidivate" to condition the query on a (hypothetical) class label.

Template for class label (occupation)

```
Answer with one of the following: accountant, architect, attorney,
chiropractor, comedian, composer, dentist, dietitian, dj, filmmaker,
interior designer, journalist, model, nurse, painter, paralegal, pastor,
personal trainer, photographer, physician, poet, professor, psychologist,
 rapper, software engineer, surgeon, teacher, yoga teacher.

Question: The following is a biography of an individual, what is their
occupation?
{example}

Answer:
```

Template for group label (sex)

```
Answer with a single letter.

Question: The following is a biography of an individual, what is their
sex?
{example}

A. Female
B. Male

Answer:
```

Template for embeddings

```
Question: The following is a biography of an individual, what is their
occupation and sex?
{example}

Answer:
```

Listing B.5: Prompt templates used on the CivilComments dataset. For the group label prompt, the placeholders `{religion}` and `{religionists}` are replaced with one of the following pairs: (`Christianity`, `Christians`), (`Judaism`, `Jewish people`), (`Islam`, `Muslims`), (`Hinduism`, `Hindus`), (`Buddhism`, `Buddhists`).

Template for class label (toxicity)

```
Answer with a single letter.

Question: Is the following comment toxic (harmful, malicious, derogatory,
 threat, insult, identity attack, etc.)?
{example}

A. Non-toxic
B. Toxic

Answer:
```

Template for group label (religion)

```
Answer with a single letter.

Question: Does the following comment mention {religion} or
{religionists} in any way?
{example}

A. {religion} is not mentioned
B. {religion} is mentioned

Answer:
```

Template for embeddings

```
Question: Is the following comment toxic (harmful, malicious, derogatory,
 threat, insult, identity attack, etc.), and what religion(s) does it
mention in any way?
{example}

Answer:
```

REFERENCES

[1] S. Barocas and A. D. Selbst, "Big Data's Disparate Impact," *California Law Review*, vol. 104, no. 3, 2016.

[2] M. Ali, P. Sapiezynski, M. Bogen, A. Korolova, A. Mislove, and A. Rieke, "Discrimination through Optimization: How Facebook's Ad Delivery Can Lead to Biased Outcomes," in *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, 2019.

[3] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith, "The Risk of Racial Bias in Hate Speech Detection," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[4] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in Criminal Justice Risk Assessments: The State of the Art," *Sociological Methods & Research*, vol. 50, no. 1, 2021.

[5] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science*, vol. 366, no. 6464, 2019.

[6] A. Fuster, P. Goldsmith-Pinkham, T. Ramadorai, and A. Walther, "Predictably Unequal? The Effects of Machine Learning on Credit Markets," *The Journal of Finance*, vol. 77, no. 1, 2022.

[7] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, 2017.

[8] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, vol. 54, no. 6, 2021.

[9] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[10] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," in *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, 2018.

[11] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine Bias," *ProPublica*, 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[12] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and Mitigating Unintended Bias in Text Classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[13] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

[14] G. Vardi, "On the Implicit Bias in Deep-Learning Algorithms," *Communications of the ACM*, vol. 66, no. 6, 2023.

[15] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness Without Demographics in Repeated Loss Minimization," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[16] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization," in *The Eighth International Conference on Learning Representations*, 2020.

[17] Y. Hu, R. Xian, Q. Wu, Q. Fan, L. Yin, and H. Zhao, "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[18] M. Hardt, E. Price, and N. Srebro, "Equality of Opportunity in Supervised Learning," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[19] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities.* The MIT Press, 2023.

[20] T. Calders and S. Verwer, "Three naive Bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, 2010.

[21] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness Through Awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012.

[22] J. Chen, N. Kallus, X. Mao, G. Svacha, and M. Udell, "Fairness Under Unawareness: Assessing Disparity When Protected Class Is Unobserved," in *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, 2019.

[23] Consumer Financial Protection Bureau, "Equal Credit Opportunity Act (Regulation B)," 2023, 12 CFR Part 1002, Section 5B.

[24] I. Globus-Harris, V. Gupta, C. Jung, M. Kearns, J. Morgenstern, and A. Roth, "Multicalibrated Regression for Downstream Fairness," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023.

[25] S. Wang, W. Guo, H. Narasimhan, A. Cotter, M. Gupta, and M. I. Jordan, "Robust Optimization for Fairness with Noisy Protected Groups," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[26] D. Mandal, S. Deng, S. Jana, J. M. Wing, and D. Hsu, "Ensuring Fairness Beyond the Training Data," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[27] A. Barrainkua, P. Gordaliza, J. A. Lozano, and N. Quadrianto, "Preserving the Fairness Guarantees of Classifiers in Changing Environments: A Survey," *ACM Computing Surveys*, vol. 57, no. 6, 2025.

[28] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, 2014.

[29] R. Cummings, V. Gupta, D. Kimpara, and J. Morgenstern, "On the Compatibility of Privacy and Fairness," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 2019.

[30] S. Agarwal, "Trade-Offs between Fairness and Privacy in Machine Learning," in *IJCAI 2020 Workshop on AI for Social Good*, 2020.

[31] M. Kearns and A. Roth, *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2020.

[32] K. R. Varshney, *Trustworthy Machine Learning*, 2022.

[33] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[34] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Gray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[35] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models are Zero-Shot Learners," in *The Tenth International Conference on Learning Representations*, 2022.

[36] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu, "Black-Box Tuning for Language-Model-as-a-Service," in *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[37] W. Gan, S. Wan, and P. S. Yu, "Model-as-a-Service (MaaS): A Survey," in *2023 IEEE International Conference on Big Data*, 2023.

[38] A. Cruz and M. Hardt, "Unprocessing Seven Years of Algorithmic Fairness," in *The Twelfth International Conference on Learning Representations*, 2024.

[39] H. Song, T. Diethe, M. Kull, and P. Flach, "Distribution Calibration for Regression," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[40] S. Zhao, M. P. Kim, R. Sahoo, T. Ma, and S. Ermon, "Calibrating Predictions to Decisions: A Novel Approach to Multi-Class Calibration," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[41] A. Mutapcic and S. Boyd, "Cutting-set methods for robust convex optimization with pessimizing oracles," *Optimization Methods and Software*, vol. 24, no. 3, 2009.

[42] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu, "Differentially Private Histogram Publication," in *IEEE 28th International Conference on Data Engineering*, 2012.

[43] I. Diakonikolas, M. Hardt, and L. Schmidt, "Differentially Private Learning of Structured Discrete Distributions," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[44] S. Vadhan, "The Complexity of Differential Privacy," in *Tutorials on the Foundations of Cryptography*. Springer, 2017.

[45] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag, "TabLLM: Few-shot Classification of Tabular Data with Large Language Models," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 2023.

[46] R. Xian and H. Zhao, "A Unified Post-Processing Framework for Group Fairness in Classification," 2024. [Online]. Available: https://arxiv.org/pdf/2405.04025

[47] R. Xian, L. Yin, and H. Zhao, "Fair and Optimal Classification via Post-Processing," in *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[48] R. Xian and H. Zhao, "Efficient Post-Processing for Equal Opportunity in Fair Multi-Class Classification," 2023. [Online]. Available: https://openreview.net/forum?id=zKjSmbYFZe

[49] R. Xian and H. Zhao, "Group Fairness Under Distribution Shifts: Analysis and Robust Post-Processing," 2025. [Online]. Available: https://openreview.net/forum?id=FL98GeTuwf

[50] R. Xian, Q. Li, G. Kamath, and H. Zhao, "Differentially Private Post-Processing for Fair Regression," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[51] R. Xian, Y. Wan, and H. Zhao, "Group Fairness Meets the Black Box: Enabling Fair Algorithms on Closed LLMs via Post-Processing," 2025. [Online]. Available: https://arxiv.org/pdf/2508.11258

[52] R. Xian, H. Ji, and H. Zhao, "Cross-Lingual Transfer with Class-Weighted Language-Invariant Representations," in *The Tenth International Conference on Learning Representations*, 2022.

[53] R. Xian, H. Zhuang, Z. Qin, H. Zamani, J. Lu, J. Ma, K. Hui, H. Zhao, X. Wang, and M. Bendersky, "Learning List-Level Domain-Invariant Representations for Ranking," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[54] Y. Hu, R. Xian, Q. Wu, Q. Fan, L. Yin, and H. Zhao, "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[55] S. Agarwal and A. Deshpande, "On the Power of Randomization in Fair Classification and Representation," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[56] N. Konstantinov and C. H. Lampert, "Fairness-Aware PAC Learning from Corrupted Data," *Journal of Machine Learning Research*, vol. 23, no. 160, 2022.

[57] A. Blum, P. Okoroafor, A. Saha, and K. M. Stangl, "On the Vulnerability of Fairness Constrained Learning to Malicious Noise," in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 2024.

[58] S. Agarwal, A. Deshpande, R. Rajaraman, and R. Sundaram, "Optimal Fair Learning Robust to Adversarial Distribution Shift," in *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[59] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic Decision Making and the Cost of Fairness," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[60] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," OJ L 119, 4.5.2016, pp. 1–88.

[61] D. Pedreshi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[62] Consumer Financial Protection Bureau, "Using publicly available information to proxy for unidentified race and ethnicity," 2014. [Online]. Available: https://www.consumerfinance.gov/data-research/research-reports/using-publicly-available-information-to-proxy-for-unidentified-race-and-ethnicity

[63] S. L. Garfinkel, "De-identification of personal information." National Institute of Standards and Technology, 2015. [Online]. Available: https://dx.doi.org/10.6028/NIST.IR.8053

[64] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building Classifiers with Independency Constraints," in *2009 IEEE International Conference on Data Mining Workshops*, 2009.

[65] M. De-Arteaga, A. Romanov, H. Wallach, J. Chayes, C. Borgs, A. Chouldechova, S. Geyik, K. Kenthapadi, and A. T. Kalai, "Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting," in *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, 2019.

[66] P. Putzel and S. Lee, "Blackbox Post-Processing for Multiclass Fairness," in *Proceedings of the Workshop on Artificial Intelligence Safety 2022*, 2022.

[67] J. Rouzot, J. Ferry, and M.-J. Huguet, "Learning Optimal Fair Scoring Systems for Multi-Class Classification," in *2022 IEEE 34th International Conference on Tools with Artificial Intelligence*, 2022.

[68] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[69] A. K. Menon and R. C. Williamson, "The Cost of Fairness in Binary Classification," in *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, 2018.

[70] H. Zhao and G. J. Gordon, "Inherent Tradeoffs in Learning Fair Representations," *Journal of Machine Learning Research*, vol. 23, no. 57, 2022.

[71] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and Removing Disparate Impact," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[72] Equal Employment Opportunity Commission, "Uniform Guidelines on Employee Selection Procedure," 1978, 29 CFR Part 1607, Section 4D.

[73] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A Reductions Approach to Fair Classification," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[74] A. Chouldechova, "Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments," *Big Data*, vol. 5, no. 2, 2017.

[75] Ú. Hébert-Johnson, M. P. Kim, O. Reingold, and G. N. Rothblum, "Multicalibration: Calibration for the (Computationally-Identifiable) Masses," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[76] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[77] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, 2012.

[78] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized Pre-Processing for Discrimination Prevention," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[79] K. Lu, P. Mardziel, F. Wu, P. Amancharla, and A. Datta, "Gender Bias in Neural Natural Language Processing," in *Logic, Language, and Security.* Springer, 2020.

[80] M. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual Fairness," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[81] J. Byrd and Z. C. Lipton, "What is the Effect of Importance Weighting in Deep Learning?" in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[82] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness Constraints: Mechanisms for Fair Classification," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[83] R. Zemel, Y. L. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning Fair Representations," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[84] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning Adversarially Fair and Transferable Representations," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[85] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating Unwanted Biases with Adversarial Learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[86] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[87] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research*, vol. 17, no. 59, 2016.

[88] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[89] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A Kernel Two-Sample Test," *Journal of Machine Learning Research*, vol. 13, no. 25, 2012.

[90] E. Z. Liu, B. Haghgoo, A. S. Chen, A. Raghunathan, P. W. Koh, S. Sagawa, P. Liang, and C. Finn, "Just Train Twice: Improving Group Robustness without Training Group Information," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[91] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[92] X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong, "Pareto Multi-Task Learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[93] O. Sener and V. Koltun, "Multi-Task Learning as Multi-Objective Optimization," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[94] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient Surgery for Multi-Task Learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[95] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil, "Leveraging Labeled and Unlabeled Data for Consistent Fair Binary Classification," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[96] S. Gaucher, N. Schreuder, and E. Chzhen, "Fair learning with Wasserstein barycenters for non-decomposable performance measures," in *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, 2023.

[97] C. Denis, R. Elie, M. Hebiri, and F. Hu, "Fairness Guarantees in Multi-Class Classification with Demographic Parity," *Journal of Machine Learning Research*, vol. 25, no. 130, 2024.

[98] P. Li, J. Zou, and L. Zhang, "FaiREE: Fair classification with finite-sample and distribution-free guarantee," in *The Eleventh International Conference on Learning Representations*, 2023.

[99] W. Chen, Y. Klochkov, and Y. Liu, "Post-Hoc Bias Scoring is Optimal for Fair Classification," in *The Twelfth International Conference on Learning Representations*, 2024.

[100] X. Zeng, G. Cheng, and E. Dobriban, "Bayes-Optimal Fair Classification with Linear Disparity Constraints via Pre-, In-, and Post-processing," 2024. [Online]. Available: https://arxiv.org/pdf/2402.02817

[101] E. Diana, W. Gill, M. Kearns, K. Kenthapadi, A. Roth, and S. Sharifi-Malvajerdi, "Multiaccurate Proxies for Downstream Fairness," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[102] D. Wei, K. N. Ramamurthy, and F. Calmon, "Optimized Score Transformation for Fair Classification," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.

[103] W. Alghamdi, H. Hsu, H. Jeong, H. Wang, P. W. Michalak, S. Asoodeh, and F. P. Calmon, "Beyond Adult and COMPAS: Fair Multi-Class Prediction via Information Projection," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[104] A. Țifrea, P. Lahoti, B. Packer, Y. Halpern, A. Beirami, and F. Prost, "FRAPPÉ: A Group Fairness Framework for Post-Processing Everything," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[105] Z. Ji, J. Li, and M. Telgarsky, "Early-stopped neural networks are consistent," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[106] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, "Learning Non-Discriminatory Predictors," in *Proceedings of the 30th Conference on Learning Theory*, 2017.

[107] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, 1998.

[108] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019.

[109] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring Adult: New Datasets for Fair Machine Learning," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[110] Y. Chen, R. Raab, J. Wang, and Y. Liu, "Fairness Transferability Subject to Bounded Distribution Shift," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[111] S. Giguere, B. Metevier, B. C. da Silva, Y. Brun, P. S. Thomas, and S. Niekum, "Fairness Guarantees under Demographic Shift," in *The Tenth International Conference on Learning Representations*, 2022.

[112] M. Kang, L. Li, M. Weber, Y. Liu, C. Zhang, and B. Li, "Certifying Some Distributional Fairness with Subpopulation Decomposition," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[113] X. Hou and L. Zhang, "Finite-Sample and Distribution-Free Fair Classification: Optimal Trade-off Between Excess Risk and Fairness, and the Cost of Group-Blindness," 2024. [Online]. Available: https://arxiv.org/pdf/2410.16477

[114] C. Schumann, X. Wang, A. Beutel, J. Chen, H. Qian, and E. H. Chi, "Transfer of Machine Learning Fairness across Domains," 2019. [Online]. Available: https://arxiv.org/pdf/1906.09688

[115] A. Coston, K. N. Ramamurthy, D. Wei, K. R. Varshney, S. Speakman, Z. Mustahsan, and S. Chakraborty, "Fair Transfer Learning with Missing Protected Attributes," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

[116] Y. Roh, K. Lee, S. E. Whang, and C. Suh, "FR-Train: A Mutual Information-Based Approach to Fair and Robust Training," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[117] A. Rezaei, A. Liu, O. Memarrast, and B. D. Ziebart, "Robust Fairness Under Covariate Shift," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021.

[118] H. Singh, R. Singh, V. Mhasawade, and R. Chunara, "Fairness Violations and Mitigation under Covariate Shift," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[119] B. An, Z. Che, M. Ding, and F. Huang, "Transferring Fairness under Distribution Shifts via Fair Consistency Regularization," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[120] S. Wu, M. Gong, B. Han, Y. Liu, and T. Liu, "Fair Classification with Instance-dependent Label Noise," in *Proceedings of the First Conference on Causal Learning and Reasoning*, 2022.

[121] Z. Jiang, X. Han, H. Jin, G. Wang, R. Chen, N. Zou, and X. Hu, "Chasing Fairness Under Distribution Shift: A Model Weight Perturbation Approach," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[122] S. Baharlouei, S. Patel, and M. Razaviyayn, "F-FERM: A Scalable Framework for Robust Fair Empirical Risk Minimization," in *The Twelfth International Conference on Learning Representations*, 2024.

[123] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," 2019. [Online]. Available: https://arxiv.org/pdf/1812.11806

[124] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A Brief Review of Domain Adaptation," in *Advances in Data Science and Information Engineering*, ser. Transactions on Computational Science and Computational Intelligence, 2021.

[125] S. Zhao, A. Sinha, Y. He, A. Perreault, J. Song, and S. Ermon, "Comparing Distributions by Measuring Differences that Affect Decision Making," in *The Tenth International Conference on Learning Representations*, 2022.

[126] C. Villani, *Topics in Optimal Transportation*, ser. Graduate Studies in Mathematics. American Mathematical Society, 2003, vol. 58.

[127] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware Minimization for Efficiently Improving Generalization," in *The Ninth International Conference on Learning Representations*, 2021.

[128] P. Gopalan, A. T. Kalai, O. Reingold, V. Sharan, and U. Wieder, “Omnipredictors,” in *13th Innovations in Theoretical Computer Science Conference*, vol. 215, 2022.

[129] M. P. Naeini, G. Cooper, and M. Hauskrecht, “Obtaining Well Calibrated Probabilities Using Bayesian Binning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[130] J. C. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” in *Advances in Large Margin Classifiers*, 1999.

[131] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[132] B. Zadrozny and C. Elkan, “Transforming Classifier Scores into Accurate Multiclass Probability Estimates,” in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[133] A. Kumar, P. S. Liang, and T. Ma, “Verified Uncertainty Calibration,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[134] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, “Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[135] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, “Classifier calibration: A survey on how to assess and improve predicted class probabilities,” *Machine Learning*, vol. 112, no. 9, 2023.

[136] S. Ben-David, P. M. Long, and Y. Mansour, “Agnostic Boosting,” in *Computational Learning Theory*, 2001.

[137] L. Trevisan, M. Tulsiani, and S. Vadhan, “Regularity, Boosting, and Efficiently Simulating Every High-Entropy Distribution,” in *2009 24th Annual IEEE Conference on Computational Complexity*, 2009.

[138] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of Statistical Planning and Inference*, vol. 90, no. 2, 2000.

[139] G. Cybenko, “Approximation by Superpositions of a Sigmoidal Function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, 1989.

[140] K.-I. Funahashi, “On the Approximate Realization of Continuous Mappings by Neural Networks,” *Neural Networks*, vol. 2, no. 3, 1989.

[141] K. Hornik, M. Stinchcombe, and H. White, “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, no. 5, 1989.

[142] Z. Ji, M. Telgarsky, and R. Xian, “Neural tangent kernels, transportation mappings, and universal approximation,” in *The Eighth International Conference on Learning Representations*, 2020.

[143] M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Sharifi-Malvajerdi, and J. Ullman, “Differentially Private Fair Learning,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[144] D. Xu, S. Yuan, and X. Wu, “Achieving Differential Privacy and Fairness in Logistic Regression,” in *Companion Proceedings of the 2019 World Wide Web Conference*, 2019.

[145] H. Mozannar, M. I. Ohannessian, and N. Srebro, “Fair Learning with Private Demographic Data,” in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[146] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, “What Can We Learn Privately?” in *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

[147] C. Tran, F. Fioretto, and P. V. Hentenryck, “Differentially Private and Fair Deep Learning: A Lagrangian Dual Approach,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021.

[148] S. Song, K. Chaudhuri, and A. D. Sarwate, “Stochastic gradient descent with differentially private updates,” in *IEEE Global Conference on Signal and Information Processing*, 2013.

[149] R. Bassily, A. Smith, and A. Thakurta, “Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds,” in *IEEE 55th Annual Symposium on Foundations of Computer Science*, 2014.

[150] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep Learning with Differential Privacy,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[151] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” in *Theory of Cryptography*, 2006.

[152] I. Mironov, “Rényi Differential Privacy,” in *IEEE 30th Computer Security Foundations Symposium*, 2017.

[153] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data,” in *The Fifth International Conference on Learning Representations*, 2017.

[154] S. Fletcher and M. Z. Islam, “Decision Tree Classification with Differential Privacy: A Survey,” *ACM Computing Surveys*, vol. 52, no. 4, 2019.

[155] Y.-X. Wang, "Revisiting differentially private linear regression: Optimal and adaptive prediction & estimation in unbounded domain," in *Proceedings of the 34th Uncertainty in Artificial Intelligence Conference*, 2018.

[156] C. Covington, X. He, J. Honaker, and G. Kamath, "Unbiased Statistical Estimation and Valid Confidence Intervals Under Differential Privacy," 2021. [Online]. Available: https://arxiv.org/pdf/2110.14465

[157] D. Alabi, A. McMillan, J. Sarathy, A. Smith, and S. Vadhan, "Differentially Private Simple Linear Regression," *Proceedings on Privacy Enhancing Technologies*, 2022.

[158] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially Private Empirical Risk Minimization," *Journal of Machine Learning Research*, vol. 12, no. 29, 2011.

[159] D. Kifer, A. Smith, and A. Thakurta, "Private Convex Empirical Risk Minimization and High-dimensional Regression," in *Proceedings of the 25th Annual Conference on Learning Theory*, 2012.

[160] M. Bun, G. Kamath, T. Steinke, and S. Z. Wu, "Private Hypothesis Selection," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[161] I. Aden-Ali, H. Ashtiani, and G. Kamath, "On the Sample Complexity of Privately Learning Unbounded High-Dimensional Gaussians," in *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 2021.

[162] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil, "Fair Regression with Wasserstein Barycenters," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[163] T. Le Gouic, J.-M. Loubes, and P. Rigollet, "Projection to Fairness in Statistical Learning," 2020. [Online]. Available: https://arxiv.org/pdf/2005.11720

[164] E. Anderes, S. Borgwardt, and J. Miller, "Discrete Wasserstein Barycenters: Optimal Transport for Discrete Data," *Mathematical Methods of Operations Research*, vol. 84, no. 2, 2016.

[165] J. M. Altschuler and E. Boix-Adserà, "Wasserstein Barycenters can be Computed in Polynomial Time in Fixed Dimension," *Journal of Machine Learning Research*, vol. 22, no. 44, 2021.

[166] M. Cuturi and A. Doucet, "Fast Computation of Wasserstein Barycenters," in *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[167] M. Staib, S. Claici, J. Solomon, and S. Jegelka, "Parallel Streaming Wasserstein Barycenters," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[168] M. Hardt and K. Talwar, "On the Geometry of Differential Privacy," in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, 2010.

[169] F. Hu, P. Ratz, and A. Charpentier, "Parametric Fairness with Statistical Guarantees," 2023. [Online]. Available: https://arxiv.org/pdf/2310.20508

[170] G. Taturyan, E. Chzhen, and M. Hebiri, "Regression under demographic parity constraints via unlabeled post-processing," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[171] M. Rudemo, "Empirical Choice of Histograms and Kernel Density Estimators," *Scandinavian Journal of Statistics*, vol. 9, no. 2, 1982.

[172] J. Liu and K. Talwar, "Private Selection from Private Candidates," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

[173] N. Papernot and T. Steinke, "Hyperparameter Tuning with Renyi Differential Privacy," in *The Tenth International Conference on Learning Representations*, 2022.

[174] S. Mohapatra, S. Sasy, X. He, G. Kamath, and O. Thakkar, "The Role of Adaptive Optimizers for Honest Private Hyperparameter Selection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022.

[175] M. Redmond and A. Baveja, "A data-driven software tool for enabling cooperative information sharing among police departments," *European Journal of Operational Research*, vol. 141, no. 3, 2002.

[176] L. F. Wightman, *LSAC National Longitudinal Bar Passage Study.* Law School Admission Council, 1998.

[177] L. Team, "The Llama 3 Herd of Models," 2024. [Online]. Available: https://arxiv.org/pdf/2407.21783

[178] G. Team, "Gemma 3 Technical Report," 2025. [Online]. Available: https://arxiv.org/pdf/2503.19786

[179] OpenAI, "GPT-4 Technical Report," 2024. [Online]. Available: https://arxiv.org/pdf/2303.08774

[180] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, A. Babiker, N. Schärli, A. Chowdhery, P. Mansfield, D. Demner-Fushman, B. Agüera y Arcas, D. Webster, G. S. Corrado, Y. Matias, K. Chou, J. Gottweis, N. Tomasev, Y. Liu, A. Rajkomar, J. Barral, C. Semturs, A. Karthikesalingam, and V. Natarajan, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, 2023.

[181] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners." OpenAI, 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[182] X. Han, T. Baldwin, and T. Cohn, "Balancing out Bias: Achieving Fairness Through Balanced Training," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[183] S. Garg, V. Perot, N. Limtiaco, A. Taly, E. H. Chi, and A. Beutel, "Counterfactual Fairness in Text Classification through Robustness," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

[184] Z. Fatemi, C. Xing, W. Liu, and C. Xiong, "Improving Gender Fairness of Pre-Trained Language Models without Catastrophic Forgetting," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, vol. 2, 2023.

[185] X. Han, T. Baldwin, and T. Cohn, "Diverse Adversaries for Mitigating Bias in Training," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

[186] V. Cherepanova, C.-J. Lee, N.-J. Akpinar, R. Fogliato, M. A. Bertran, M. Kearns, and J. Zou, "Improving LLM Group Fairness on Tabular Data via In-Context Learning," in *NeurIPS 2024 Safe Generative AI Workshop*, 2024.

[187] J. Atwood, N. Scherrer, P. Lahoti, A. Balashankar, F. Prost, and A. Beirami, "Inducing Group Fairness in Prompt-Based Language Model Decisions," in *ICLR 2025 Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*, 2025.

[188] I. Baldini, D. Wei, K. Natesan Ramamurthy, M. Singh, and M. Yurochkin, "Your fairness may vary: Pretrained language model fairness in toxic text classification," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

[189] Y. Liu, S. Gautam, J. Ma, and H. Lakkaraju, "Confronting LLMs with Traditional ML: Rethinking the Fairness of Large Language Models in Tabular Classifications," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

[190] J. Hu, W. Liu, and M. Du, "Strategic Demonstration Selection for Improved Fairness in LLM In-Context Learning," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[191] Y. Li, L. Zhang, and Y. Zhang, "Fairness of ChatGPT," 2024. [Online]. Available: https://arxiv.org/pdf/2305.18569

[192] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate Before Use: Improving Few-shot Performance of Language Models," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[193] S. Bordia and S. R. Bowman, "Identifying and Reducing Gender Bias in Word-Level Language Models," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2019.

[194] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, and Y. Goldberg, “Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[195] S. Singh, S. Ravfogel, J. Herzig, R. Aharoni, R. Cotterell, and P. Kumaraguru, “Representation Surgery: Theory and Practice of Affine Steering,” in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[196] C. May, A. Wang, S. Bordia, S. R. Bowman, and R. Rudinger, “On Measuring Social Biases in Sentence Encoders,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019.

[197] N. Nangia, C. Vania, R. Bhalerao, and S. R. Bowman, “CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

[198] M. Nadeem, A. Bethke, and S. Reddy, “StereoSet: Measuring stereotypical bias in pretrained language models,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, vol. 1, 2021.

[199] O. Shaikh, H. Zhang, W. Held, M. Bernstein, and D. Yang, “On Second Thought, Let’s Not Think Step by Step! Bias and Toxicity in Zero-Shot Reasoning,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2023.

[200] X. Bai, A. Wang, I. Sucholutsky, and T. L. Griffiths, “Measuring Implicit Bias in Explicitly Unbiased Large Language Models,” 2024. [Online]. Available: https://arxiv.org/pdf/2402.04105

[201] A. Parrish, A. Chen, N. Nangia, V. Padmakumar, J. Phang, J. Thompson, P. M. Htut, and S. R. Bowman, “BBQ: A Hand-Built Bias Benchmark for Question Answering,” in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

[202] OpenAI, “OpenAI o1 System Card,” 2024. [Online]. Available: https://arxiv.org/pdf/2412.16720

[203] N. Cecere, A. Bacciu, I. Fernández-Tobías, and A. Mantrach, “Monte Carlo Temperature: A robust sampling strategy for LLM’s uncertainty quantification methods,” in *Proceedings of the 5th Workshop on Trustworthy NLP*, 2025.

[204] M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi, “Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs,” in *The Twelfth International Conference on Learning Representations*, 2024.

[205] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy, E. Wallace, D. Rolnick, and F. Tramèr, "Stealing Part of a Production Language Model," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[206] F. Prost, H. Qian, Q. Chen, E. H. Chi, J. Chen, and A. Beutel, "Toward a better trade-off between performance and fairness with kernel-based distribution matching," 2019. [Online]. Available: https://arxiv.org/pdf/1910.11779

[207] Z. C. Lipton, Y.-X. Wang, and A. J. Smola, "Detecting and Correcting for Label Shift with Black Box Predictors," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[208] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization.* Princeton University Press, 2009.

[209] J. R. Anthis, K. Lum, M. Ekstrand, A. Feller, and C. Tan, "The Impossibility of Fair LLMs," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2025.

[210] A. Tamkin, A. Askell, L. Lovitt, E. Durmus, N. Joseph, S. Kravec, K. Nguyen, J. Kaplan, and D. Ganguli, "Evaluating and Mitigating Discrimination in Language Model Decisions," 2023. [Online]. Available: https://arxiv.org/pdf/2312.03689

[211] K. Morehouse, W. Pan, J. M. Contreras, and M. R. Banaji, "Bias Transmission in Large Language Models: Evidence from Gender-Occupation Bias in GPT-4," in *ICML 2024 Next Generation of AI Safety Workshop*, 2024.

[212] A. Salinas, A. Haim, and J. Nyarko, "What's in a Name? Auditing Large Language Models for Race and Gender Bias," 2025. [Online]. Available: https://arxiv.org/pdf/2402.14875

[213] T. Eloundou, A. Beutel, D. G. Robinson, K. Gu, A.-L. Brakman, P. Mishkin, M. Shah, J. Heidecke, L. Weng, and A. T. Kalai, "First-Person Fairness in Chatbots," in *The Thirteenth International Conference on Learning Representations*, 2025.

[214] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: Long Form Question Answering," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[215] E. Sheng, K.-W. Chang, P. Natarajan, and N. Peng, "The Woman Worked as a Babysitter: On Biases in Language Generation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019.

[216] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, "Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[217] M. Sap, M. C. Prasettio, A. Holtzman, H. Rashkin, and Y. Choi, “Connotation Frames of Power and Agency in Modern Films,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[218] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, “RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[219] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. El Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan, “Constitutional AI: Harmlessness from AI Feedback,” 2022. [Online]. Available: https://arxiv.org/pdf/2212.08073

[220] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, “Plug and Play Language Models: A Simple Approach to Controlled Text Generation,” in *The Eighth International Conference on Learning Representations*, 2020.

[221] B. Krause, A. D. Gotmare, B. McCann, N. S. Keskar, S. Joty, R. Socher, and N. F. Rajani, “GeDi: Generative Discriminator Guided Sequence Generation,” 2020. [Online]. Available: https://arxiv.org/pdf/2009.06367

[222] K. Li, T. Liu, N. Bashkansky, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, “Measuring and Controlling Instruction (In)Stability in Language Model Dialogs,” in *The First Conference on Language Modeling*, 2024.

[223] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, “Fine-Tuning Language Models from Human Preferences,” 2020. [Online]. Available: https://arxiv.org/pdf/1909.08593

[224] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan, “Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback,” 2022. [Online]. Available: https://arxiv.org/pdf/2204.05862

[225] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[226] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed., ser. Adaptive Computation and Machine Learning Series. The MIT Press, 2018.

[227] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, 1999.

[228] J. Li and T. Tkocz, "Tail Bounds for Sums of Independent Two-Sided Exponential Random Variables," in *High Dimensional Probability IX*, 2023.

[229] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Inequalities for the L_1 Deviation of the Empirical Distribution," Hewlett-Packard Laboratories, Technical Report HPL-2003-97R1, 2003.

[230] Q. F. Stout, "L_∞ Isotonic Regression for Linear, Multidimensional, and Tree Orders," 2017.

[231] L. Chizat, P. Roussillon, F. Léger, F.-X. Vialard, and G. Peyré, "Faster Wasserstein Distance Estimation with the Sinkhorn Divergence," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[232] 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 É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, 2011.

[233] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[234] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *The Third International Conference on Learning Representations*, 2015.

[235] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *The Seventh International Conference on Learning Representations*, 2019.

[236] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias," 2018. [Online]. Available: https://arxiv.org/pdf/1810.01943

[237] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional Adversarial Domain Adaptation," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[238] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.