

---

# Efficient Post-Processing for Equal Opportunity in Fair Multi-Class Classification

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Fairness in machine learning is of growing concern as more instances of biased  
2 model behavior are documented while their adoption continues to rise. The majority  
3 of studies have focused on binary classification settings, despite the fact that many  
4 real-world problems are inherently multi-class. This paper considers fairness  
5 in multi-class classification under the notion of parity of true positive rates—an  
6 extension of binary class equalized odds [25]—which ensures equal opportunity  
7 to qualified individuals regardless of their demographics. We focus on algorithm  
8 design and provide a post-processing method that derives fair classifiers from pre-  
9 trained score functions. The method is developed by analyzing the representation  
10 of the optimal fair classifier, and is efficient in both sample and time complexity,  
11 as it is implemented by linear programs on finite samples. We demonstrate its  
12 effectiveness at reducing disparity on benchmark datasets, particularly under large  
13 numbers of classes, where existing methods fall short.

## 14 1 Introduction

15 Algorithmic fairness has emerged as a topic of significant concern in the field of machine learning,  
16 due to the potential for models to exhibit discriminatory behavior towards historically disadvantaged  
17 demographics [11, 5, 7], all while their adoption continues to rise in domains including high-stakes  
18 areas such as criminal justice, healthcare, and finance [4, 8]. To address the concern, a variety of  
19 fairness criteria have been proposed (e.g., demographic parity, equalized odds) along with mitigation  
20 methods [12, 21, 25, 28]. On classification problems, the majority of work focuses on the binary class  
21 setting [2, Table 1], where one class is typically considered to be more favorable (e.g., the approval  
22 vs. rejection of a credit card application).

23 Yet, many real-world problems are multi-class in nature. In the case of credit card applications, issuers  
24 may prefer assigning higher-tier interest rates to high-risk applicants compared to outright rejection,  
25 which creates opportunities to applicants who would otherwise be denied credit and also generates  
26 returns for the banks. Similarly, in online advertising, recruiting platforms can employ machine  
27 learning models to match users to relevant job postings across multiple occupation categories. There  
28 are evidences, however, for such systems to exhibit gender bias [9, 15, 49]; for instance, models  
29 that are trained to identify occupation from biography tend to show higher accuracy (recall) on male  
30 biographies than on their female counterparts in occupations that are historically male-dominated [16].

31 In the example above, unfairness is manifested in a disparity of *true positive rates* (TPRs) across  
32 demographic groups  $A$  (generalizing the true positive and negative rates in binary classification),

$$\text{TPR}_a(\hat{Y})_y := \mathbb{P}(\hat{Y} = y \mid Y = y, A = a), \quad \forall y \in [k], a \in [m].$$

33 A classifier satisfying parity of TPRs, i.e.,  $\text{TPR}_a = \text{TPR}_{a'}$  for all  $a, a'$ , ensures that individuals with  
34 the same qualification ( $Y$ ) will have *equal opportunity* of receiving their favorable outcome ( $\hat{Y} = Y$ )

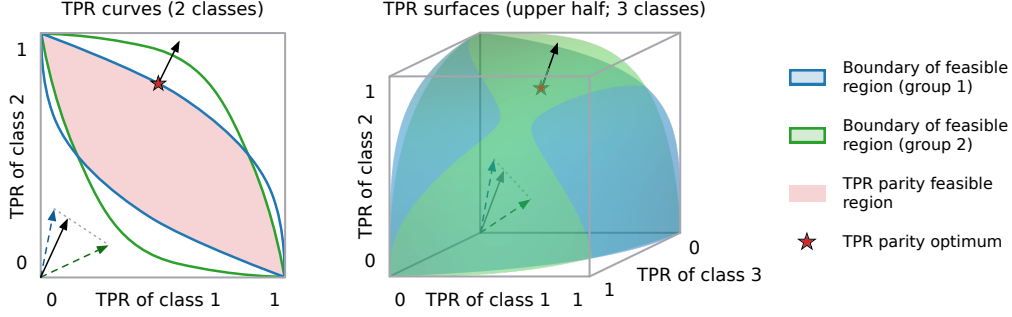


Figure 1: Feasible region of TPRs on a binary class (left) and a three-class problem (right). The black (resp. colored) arrow indicates the utility-maximizing direction (of each group).

regardless of demographics [22], e.g., being shown job postings on recruiting platforms for which the user is qualified. When the classes are binary, this fairness notion recovers *equalized odds* [25].

In this paper, we focus on the design of algorithm for mitigating TPR disparity and provide an efficient *post-processing* method that derives *attribute-aware* fair classifiers from (pre-trained) scoring models. Our method works on multi-class and multi-group classification problems, guarantees fairness by a sample complexity bound, can be implemented by linear programs, and achieves higher reductions in disparity compared to existing algorithms that are applicable to multi-class—a recently proposed post-processing method based on model projection [2], and adversarial debiasing [46], an in-processing method—especially when the number of classes is large.

**Organization.** We introduce the problem setup and objectives in Section 2, then describe our post-processing method for TPR parity in Section 3, along with suboptimality analyses; in particular, our method yields the optimal fair classifier when applied to the *Bayes optimal* score function. Our method is instantiated for finite sample estimation in Section 4, and we also provide sample complexity bounds to complete the analysis. Finally, in Section 5, we compare our algorithm with existing methods for disparity reduction on benchmark datasets.<sup>1</sup> A high-level summary of our results is provided in Section 1.1.

## 1.1 Summary of Results

One way to interpret and understand TPR parity is through visualizing the feasible regions of TPRs. In Fig. 1, we plot the feasible regions (achievable by probabilistic classifiers) of two groups on a (hypothetical) binary classification problem on the left, and those on a three-class problem on the right, where each axis represents the TPR of a class. Achieving optimal TPR parity amounts to first finding the TPR that maximizes the overall utility (e.g., accuracy) in the intersection of feasible regions, and subsequently an (attribute-aware) classifier attaining that target TPR on all groups. Note that the left figure is equivalent to the ROC curve (with a flip of the horizontal axis, because the TPR of class 1 equals one minus the false negative rate by treating class-1 as the negative class), which was used by Hardt et al. [25] for studying equalized odds. And thus, the TPR (hyper)surface plots in higher dimensions are a natural generalization of the ROC curve to multi-class settings.

Step one of finding the optimal fair TPR can be formulated as a linear program when estimating from finite samples. For the second step, our method derives a classifier attaining the target TPR from the score function; in particular, it yields the optimal fair classifier when the score is Bayes optimal:

**Theorem 1.1.** Let  $f_1^*, \dots, f_m^* : \mathcal{X} \rightarrow \Delta_k$  denote the Bayes score function on each group,  $f_a^*(x) := \mathbb{E}[Y | X = x, A = a]$ , and  $q_1, \dots, q_m \in \Delta_k$  be arbitrary. Then under a continuity assumption (2.4),  $\exists \beta_1, \dots, \beta_m \in [0, 1]$  and  $\lambda_1, \dots, \lambda_m \in \mathbb{R}^k$  s.t. the probabilistic attribute-aware classifier

$$(x, a) \mapsto \begin{cases} \arg \max_{y'} (\lambda_a)_{y'} \cdot f_a^*(x)_{y'} & \text{w.p. } 1 - \beta_a \\ y & \text{w.p. } \beta_a \cdot (q_a)_y, \forall y \in [k] \end{cases} \quad (1a)$$

achieves the maximum utility subject to TPR parity.

<sup>1</sup>Our code is provided in the supplemental material.

The post-processed classifier returned by our method is a mixture of two models (weighted by  $\beta$ ). Eq. (1a) returns the class with the highest likelihood after a class-wise rescaling, called a *tilting* [2], which generalizes the concept of *thresholding* in binary classifiers. Eq. (1b) makes random assignments sampled from a Multinoulli( $q$ ) distribution, which handles situations where the fair TPR lies in the interior of the feasible region (see Fig. 1, where the optimum is located within the interior of group 2 feasible region). To alleviate potential ethical concerns regarding this randomization, we point out that the parameter  $q_a$ 's used in class sampling can be specified per-group by the practitioner responsibly, e.g., uniform  $1/k$ , or  $e_{y'}$  with  $y'$  being an advantaged outcome.

Among the possibly infinitely many fair classifiers derived from the score function  $f$ , we specifically seek the simplistic representation in Eq. (1) because it immediately extrapolates to unseen examples, can be estimated via linear programs from finite samples, and provides good generalization performance at the rate of  $\tilde{O}(\sqrt{k/n})$  thanks to its low function complexity (Theorem 4.2).

When the score function being post-processed is not Bayes optimal, our method is still applicable, but the result may not be optimal nor exactly achieve TPR parity (without leveraging labeled data or additional knowledge of the model, as our method only needs unlabeled data). But these suboptimality are minimized if the model is *calibrated* (Theorem 3.5); this answers the question raised in [2] about the effects of base model inaccuracies on downstream post-processing.

## 1.2 Related Work

**Fairness Criteria.** The notion of TPR parity has appeared in the literature as *conditional procedure accuracy equality* [8], *avoiding disparate mistreatment* [44], and (multi-class) *equal opportunity* [16, 33, 35] (to be distinguished from the fairness criterion with the same name in [25]). Other group fairness notions that extend to multi-class include (but not limited to) *equalized odds* [25] (of which TPR parity is a necessary condition), and *demographic parity* (DP) [12] (where Xian et al. [40] recently proposed an optimal post-processing method). However, DP may be less desirable than TPR parity in some use cases because the perfect classifier is not permitted under DP when the base rates differ [47]. It is worth noting that TPR parity implies *accuracy parity* [11]. In addition to group fairness, there are notions defined on the individual level [21].

**Mitigation Methods.** Our method is based on post-processing [27, 25]. There are also in-processing methods via fair representation learning [45, 46, 48, 34], solving zero-sum games [1, 41], and pre-processing methods that debias the data prior to training [13, 49]; see [5, 14] for a survey.

For multi-class TPR parity, the only applicable post-processing method to date, to our knowledge, is due to Alghamdi et al. [2] (which is the primary baseline for our method in our experiments). It is a general-purpose method that transforms the scores to satisfy fairness while minimizing the distributional divergence (e.g., KL) between the transformed scores and the original. However, the tradeoff between model performance and fairness is unclear as they did not relate the divergence to utility. Furthermore, while the authors provided a sample complexity bound for their optimization objective, it is not explicitly related to the violation of the fairness criteria.

## 2 Preliminaries

A  $k$ -class classification problem is defined by a joint distribution  $\mu$  of input  $X \in \mathcal{X}$ , demographic group membership  $A \in [m] := \{1, \dots, m\}$  (a.k.a. the sensitive attribute), and class label  $Y \in [k]$ . We assume that all classes have nonzero probability on each group, otherwise TPR would not be well-defined. Denote the joint distribution of  $(X, A)$  by  $\mu^{X,A}$ , and, the  $(k-1)$ -dimensional probability simplex by  $\Delta_k := \{z \in \mathbb{R}_{\geq 0}^k : \|z\|_1 = 1\}$ .

**Assumption 2.1.**  $\mathbb{P}_\mu(Y = y \mid A = a) > 0$ , for all  $y \in [k], a \in [m]$ .

Let  $f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  be an attribute-aware (pre-trained) score function, whose outputs are probability vectors that estimate the class probabilities as in  $f(x, a)_y \approx \mathbb{P}_\mu(Y = y \mid X = x, A = a)$ . We will write  $f_a : \mathcal{X} \rightarrow \Delta_k$  to denote the component of  $f$  associated with group  $a$ , i.e.,  $f_a(x) \equiv f(x, a)$ . Our goal is to find fair (probabilistic) post-processing maps  $g_1, \dots, g_m : \Delta_k \rightarrow \mathcal{Y}$  to derive a classifier  $(x, a) \mapsto g_a \circ f_a(x)$  that satisfies TPR parity while maximizing utility (e.g., classification accuracy). We will sometimes write  $R := f(X, A)$ .

116 We allow for controllable tradeoffs between utility and fairness through the following relaxation of  
 117 TPR parity, and call a classifier  $\alpha$ -fair if it satisfies  $\alpha$ -TPR parity:

118 **Definition 2.2** (Approximate TPR Parity). Let  $\alpha \in [0, 1]$ . A predictor  $\hat{Y}$  is said to satisfy  $\alpha$ -TPR  
 119 parity if  $\Delta_{\text{TPR}}(\hat{Y}) \leq \alpha$ , where

$$\Delta_{\text{TPR}}(\hat{Y}) := \max_{a, a' \in \mathcal{A}} \left\| \text{TPR}_a(\hat{Y}) - \text{TPR}_{a'}(\hat{Y}) \right\|_{\infty}, \quad (2)$$

120 and  $\text{TPR}_a(\hat{Y}) := \mathbb{P}(\hat{Y} \mid Y = y, A = a) \in [0, 1]^k$ ;  $\mathbb{P}$  includes the randomness of the predictor.

121 Beyond classification accuracy, we also allow for any utility functions that depend only on the TPRs:<sup>2</sup>

122 **Definition 2.3** (Utility). The utility function  $u : [k] \times [k] \rightarrow \mathbb{R}$  is defined for some  $v \in \mathbb{R}^k$  by

$$u(\hat{y}, y) := \sum_{y' \in [k]} v_{y'} \mathbb{1}[y = y', \hat{y} = y'].$$

123 E.g., accuracy,  $\mathbb{1}[y = \hat{y}]$ , is obtained by setting  $v = \mathbf{1}_k$ . The term  $v$  will appear in our analyses,  
 124 and the significance of considering utilities of this form is that we could evaluate a classifier by a  
 125 weighted sum of its TPRs. Define  $p_{ay} := \mathbb{P}_{\mu}(A = a, Y = y)$ , then

$$\mathcal{U}(\hat{Y}) = \mathbb{E} u(\hat{Y}, Y) = \sum_{a \in [m], y \in [k]} v_y p_{ay} \text{TPR}_a(\hat{Y})_y \equiv \mathcal{U}(\text{TPR}_1(\hat{Y}), \dots, \text{TPR}_m(\hat{Y})). \quad (3)$$

126 Finally, we make the following continuity assumption on the distributions of score to avoid technical  
 127 complexities related to tie-breaking (on the atoms). This assumption has also appeared in prior work  
 128 on fair post-processing [17, 23, 40]; it holds when the input distributions are continuous and the score  
 129 function is injective, or can be satisfied by adding small random perturbations to the scores.

130 **Assumption 2.4.** The conditional distribution of score,  $\mathbb{P}(f_a(X) \mid A = a)$ , is (Lebesgue absolutely)  
 131 continuous, for all  $a \in [m]$ .

### 132 3 TPR Parity via Post-Processing

133 Given a score function  $f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$ , and access to the (unlabeled) joint distribution  $\mu^{X, A}$  (i.e.,  
 134 no estimation error), we describe a method for deriving an attribute-aware  $\alpha$ -fair classifier while  
 135 maximizing utility, in the form of  $(x, a) \mapsto g_a \circ f_a(x)$ , where the  $g_a$ 's are (probabilistic) fair  
 136 post-processing maps for each group. That is, we want to solve

$$\max_{g_1, \dots, g_m} \mathcal{U}(\hat{Y}) \quad \text{s.t.} \quad \Delta_{\text{TPR}}(\hat{Y}) \leq \alpha \quad \text{where} \quad \hat{Y} = g_A \circ f_A(X).$$

137 Although the method only returns classifiers derived from  $f$  as opposed to searching over the space of  
 138 all classifiers  $h : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$ , it would yield the optimal fair classifier provided that the information  
 139 of  $(A, Y)$  is preserved in the output of  $f$ ; this is the case when the score function is Bayes optimal.

#### 140 3.1 Deriving Optimal Fair Classifier From Bayes Score Function

141 In this section, we explain how to obtain an optimal fair classifier by deriving from the Bayes score  
 142 function  $f^*$ , thereby providing a *proof of Theorem 1.1* (omitted proofs are in Appendices A and B).

143 **Step 1** (Finding Utility-Maximizing Fair TPRs). Let  $D_a \subseteq [0, 1]^k$  denote the set of feasible TPRs  
 144 on group  $a$  achieved by probabilistic classifiers. The first step is to find utility-maximizing fair TPRs  
 145 contained in an  $\ell_{\infty}$ -ball of diameter  $\alpha$  per Definition 2.2 of  $\alpha$ -TPR parity (left figure of Fig. 2):

$$\max_{t_1 \in D_1, \dots, t_m \in D_m} \mathcal{U}(t_1, \dots, t_m) \quad \text{s.t.} \quad \|t_a - t_{a'}\|_{\infty} \leq \alpha, \quad \forall a, a' \in [m]. \quad (4)$$

146 When  $\alpha = 0$ , this reduces to finding a single  $t \in \bigcap_a D_a$ , and because each  $D_a$  is convex (since  
 147 probabilistic classifiers are allowed), it can be found with ternary search as suggested in [25]. If

<sup>2</sup>This includes all possible utility/loss functions in binary classification, since  $\text{TPR}(\hat{Y})_1$  (true negative rate) and  $\text{TPR}(\hat{Y})_2$  (true positive rate) fully determine the  $2 \times 2$  confusion matrix.

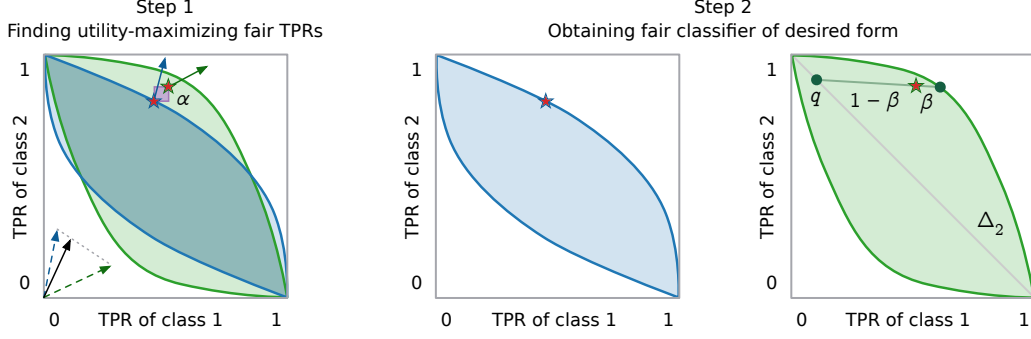


Figure 2: Achieving  $\alpha$ -TPR parity on a binary class problem. First, the utility-maximizing TPRs residing in an  $\ell_\infty$ -ball of diameter  $\alpha$  are found (left). Then, classifiers achieving the fair TPRs are obtained: a tilting of the scores when the TPR lies on the boundary (middle), otherwise, a mixture of tilting and randomization (right). The simplex  $\Delta_k$  is always inscribed in the feasible region.

148 instead the  $t_a$ 's are to be estimated from finite samples, then the empirical  $\hat{D}_a$ 's are described by  
 149 polytopes and the problem can be formulated as a linear program (Section 4).

150 The feasible regions of TPR generally differ across groups, due to uncertainties inherent to the task  
 151 of interest, or to inadequate and biased sourcing of data. The more the  $D_a$ 's differ, the greater the  
 152 tradeoff between fairness and utility. Hence TPR parity incentivizes the learner to improve data  
 153 collection and aspects of modeling that induces a balanced predictive capability on all groups [25].

154 Because  $f^*(X, A)$  is *sufficient statistic* for  $Y$ , the fair TPRs we found above are always achievable  
 155 by classifiers derived from  $f^*$ . Or more concretely,

156 **Proposition 3.1.** Let  $f^* : \mathcal{X} \rightarrow \Delta_k$  denote the Bayes score function, then  $D := \{\text{TPR}(h) \in [0, 1]^k \mid$   
 157  $h : \mathcal{X} \rightarrow \mathcal{Y} \text{ (probabilistic)}\} = \{\text{TPR}(g \circ f^*) \in [0, 1]^k \mid g : \Delta_k \rightarrow \mathcal{Y} \text{ (probabilistic)}\}.$

158 **Step 2 (Obtaining Fair Classifier of Desired Form).** Having found the utility-maximizing fair TPR  
 159  $t_a$ 's, the next step is to derive a classifier that attains  $t_a$  on each group:

160 **Theorem 3.2.** Let  $f^* : \mathcal{X} \rightarrow \Delta_k$  denote the Bayes score function, and  $q \in \Delta_k$  be arbitrary. Then  
 161 under Assumptions 2.1 and 2.4,  $\forall t \in D$ , there exists  $\beta \in [0, 1]$  and  $\lambda \in \mathbb{R}^k$  s.t.  $\text{TPR}(h) = t$ , where

$$h(x) = \begin{cases} \arg \max_{y'} \lambda_{y'} f^*(x)_{y'} & \text{w.p. } 1 - \beta \\ y & \text{w.p. } \beta q_y, \forall y \in [k]. \end{cases}$$

162 The construction uses the observation that the boundary of  $D$ , denoted by  $\partial D$ , is given by the set of  
 163 TPRs attained by tiltings of the Bayes score:

164 **Proposition 3.3.** Let  $f^* : \mathcal{X} \rightarrow \Delta_k$  denote the Bayes score function. Then under Assumption 2.1,  
 165  $h : \mathcal{X} \rightarrow \mathcal{Y}$  (probabilistic) satisfies  $\text{TPR}(h) \in \partial D$  if and only if  $\exists \lambda \in \mathbb{R}^k, \lambda \neq 0$  s.t.  $h(x) \in$   
 166  $\arg \max_y \lambda_y f^*(x)_y$  almost surely.

167 *Proof of Theorem 3.2.* If the target TPR lies on the boundary of  $D$ , then by Proposition 3.3, it is  
 168 achieved by a tilting of the Bayes score without any randomization (i.e.,  $\beta = 0$ ; center figure of  
 169 Fig. 2). This holds due to Assumption 2.4, because we may break ties arbitrarily without affecting  
 170 TPR, since the set of tied scores (finite union of  $(k - 2)$ -d subspaces) has (Lebesgue) measure zero.

171 Otherwise, and generally, there must exist  $t' \in \partial D$  and  $\beta \in [0, 1]$  s.t.  $t$  can be written as a linear  
 172 combination of  $t = \beta q + (1 - \beta)t'$ . This is simply because  $q \in \Delta_k \subseteq D$ , and the line connecting  $q$   
 173 and  $t$  must intersect  $\partial D$  at some point  $t'$  (right figure of Fig. 2). Since the TPR of the input-agnostic  
 174 randomization according to  $\text{Multinoulli}(q)$  equals  $q$ , and  $t'$  is achieved by a tilting of the score per  
 175 Proposition 3.3, their  $\beta$ -mixture achieves the target TPR  $t$  by linearity.  $\square$

## 176 3.2 Deriving From Any Score Function

177 The post-processing method described in the previous section, which only requires unlabeled data  
 178  $(X, A)$ , yields the optimal  $\alpha$ -fair classifier when applied to Bayes scores  $f^*$ . Yet, in practice, there

---

**Algorithm 1** Post-Process Score Function for  $\alpha$ -TPR parity
 

---

1: **Input:**  $\alpha \in [0, 1]$ ,  $q_1, \dots, q_m \in \Delta_k$ , score function  $f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$ , distribution  $\mu^{X,A}$   
 2:  $\tilde{D}_a := \{\widetilde{\text{TPR}}_a(h) \mid h : \mathcal{X} \rightarrow \mathcal{Y} \text{ (probabilistic)}\} \quad \triangleright \text{Eq. (5), induced TPR feasible region}$   
 3:  $\tilde{t}_1, \dots, \tilde{t}_m \leftarrow \arg \max_{\tilde{t}_1 \in \tilde{D}_1, \dots, \tilde{t}_m \in \tilde{D}_m} \mathcal{U}(\tilde{t}_1, \dots, \tilde{t}_m)$  s.t.  $\|\tilde{t}_a - \tilde{t}_{a'}\|_\infty \leq \alpha, \forall a, a' \in [m]$   
 $\triangleright$  utility-maximizing fair TPRs  
 4: **for**  $a = 1$  **to**  $m$  **do**  
 5:   Find  $h_a, \beta_a \in [0, 1]$  s.t.  $\widetilde{\text{TPR}}_a(h_a) \in \partial \tilde{D}_a$  and  $\tilde{t}_a = (1 - \beta_a) \widetilde{\text{TPR}}_a(h_a) + \beta_a q_a$   
 6:   Find  $\lambda_a \in \mathbb{R}^k$  s.t.  $h_a(x) \in \arg \max_{y'} (\lambda_a)_{y'} \cdot f_a(x)_{y'}, \forall x \in \text{supp}(\mu_a^X)$   
 7: **end for**  
 8: **Return:**  $(x, a) \mapsto \arg \max_{y'} (\lambda_a)_{y'} \cdot f_a(x)_{y'}$  w.p.  $1 - \beta_a$ , and  $y$  w.p.  $\beta_a \cdot (q_a)_y$  for each  $y \in [k]$

---

179 is the concern that Bayes score functions could be arbitrarily complex and are often not exactly  
 180 learnable due to limited data or computational constraints [39].

181 Nonetheless, our method is still applicable to arbitrary (approximations to the Bayes) score functions  
 182  $f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  for deriving classifiers that are approximately fair and optimal, by treating them  
 183 as if they were *Bayes optimal* (Algorithm 1). Where, the only tweak we made is replacing the  
 184 ground-truth TPRs and feasible regions (which are unknown without access to the Bayes score) by  
 185 approximations *induced* by  $f$ , i.e.,

$$\tilde{D}_a := \left\{ \widetilde{\text{TPR}}_a(h) \in [0, 1]^k \mid h : \mathcal{X} \rightarrow \mathcal{Y} \text{ (probabilistic)} \right\}, \quad (5)$$

186 where

$$\widetilde{\text{TPR}}_a(h)_y := \frac{1}{\tilde{p}_{ay}} \int_{x \in \mathcal{X}} f_a(x)_y \mathbb{P}(h(x) = y) d\mu^{X,A}(x, a), \quad \tilde{p}_{ay} := \int_{x \in \mathcal{X}} f_a(x)_y d\mu^{X,A}(x, a). \quad (6)$$

187 It is not hard to show that they are equal to their ground-truth counterparts when  $f = f^*$ .

188 The suboptimality of the classifier returned from Algorithm 1 are upper bounded by the  $L^1$  difference  
 189 between the score function  $f$  and a *group-wise distribution calibrated* reference score with *finer*  
 190 *granularity*. In other words, better results can be obtained by recalibrating  $f$  prior to post-processing.

191 **Definition 3.4.** A score  $\bar{R}$  is said to be (group-wise) distribution calibrated if  $\mathbb{P}(Y = y \mid \bar{R} = s) = s_y$ ,  
 192  $\forall s \in \Delta_k, y \in [k]$  (resp.  $\mathbb{P}(Y = y \mid \bar{R} = s, A = a) = s_y, \forall a \in [m]$ ). Furthermore, it is said to have  
 193 finer granularity than score  $R$  if  $\mathbb{P}(Y = y \mid \bar{R} = s, R = s') = s_y, \forall s, s' \in \Delta_k, y \in [k]$ .

194 Distribution calibration is a multi-class generalization of the original definition of calibration for  
 195 binary predictors [30, 37], requiring the predicted score to match the underlying class distribution  
 196 conditioned on the score across all classes, not just the most confident one [24]. Although this  
 197 definition is convenient to work with mathematically, it could be difficult to achieve in practice. We  
 198 will relax it to the notion of *decision calibration* [50] when we prove our results in Appendix B (w.r.t.  
 199 the set of all tiltings; derived from *multicalibration* [26]), which can be achieved in polynomial time.

200 **Theorem 3.5.** Let  $h : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$  be the (probabilistic) classifier derived from a score function  
 201  $f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  using Algorithm 1. Then under Assumptions 2.1 and 2.4, for any group-wise  
 202 distribution calibrated reference score function  $\bar{f} : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  with finer granularity than  $f$ ,

$$|\bar{\mathcal{U}} - \mathcal{U}(h)| \leq 6\|v\|_1 \max_{a,y} \frac{\epsilon_{ay}}{p_{ay}}, \quad \Delta_{\text{TPR}}(h) \leq \alpha + 4 \max_{a,y} \frac{4\epsilon_{ay}}{p_{ay}},$$

203 where  $p_{ay} := \mathbb{P}_\mu(A = a, Y = y)$ ,  $v$  is from Definition 2.3 of the utility,  $\bar{\mathcal{U}}$  denotes the utility of the  
 204 optimal  $\alpha$ -fair classifier derived from the reference  $\bar{f}$ , and

$$\epsilon_{ay} := \mathbb{E}[|\bar{f}_a(X)_y - f_a(X)_y| \mathbb{1}[A = a]] \quad (7)$$

205 measures the miscalibration of  $f$  w.r.t.  $\bar{f}$  on group  $a$  and class  $y$ .

206 We draw two conclusions from this result. First, by using the Bayes score function  $f^*$  as the reference,  
 207 it states that the suboptimality of the derived classifier when  $f \neq f^*$  is upper bounded by the  $L^1$   
 208 difference between the approximate scores and the ground-truth; this answers the question raised

in [2] regarding the impact of base model inaccuracies. Second, if  $f$  satisfies calibration, then by using itself as the reference, the result guarantees that the classifier derived using Algorithm 1 exactly achieves the desired level of fairness, and is optimal among all fair classifiers derived from  $f$  (which can only be further improved using labeled data).

## 4 Finite-Sample Algorithm and Guarantees

We instantiate the post-processing method above for TPR parity to the case where we do not have access to the distribution  $\mu^{X,A}$  but only samples drawn from it (i.e., to perform estimation), and analyze the sample complexity.

**Assumption 4.1.** We have  $n$  i.i.d. samples of  $(X, A)$  that are independent of the score function  $f$  being post-processed. Denote the samples from group  $a$  by  $(x_{a,i})_{i \in [n_a]}$ , and their number by  $n_a$ .

### 4.1 Algorithm

We adapt Algorithm 1 to handle finite samples by replacing  $\tilde{D}_a$  and  $\mathcal{U}$  with their empirical counterparts (essentially calling it with the empirical distribution  $\hat{\mu}^{X,A}$  formed by the samples as the argument), and implement the optimization problems on Lines 3, 5 and 6 using linear programs.

**Step 1** (Finding Utility-Maximizing Fair TPRs). The empirical induced feasible region of TPRs,  $\hat{D}_a$ , can be computed via evaluating the TPRs of all (probabilistic) classifiers acting on the samples—by representing them using  $n_a \times k$  lookup tables (each row gives the probabilities of the random class assignment on the corresponding sample):

$$\hat{D}_a := \left\{ \widehat{\text{TPR}}_a(\gamma_a) \mid \gamma_a \in \mathbb{R}_{\geq 0}^{n_a \times k}, \sum_{y \in [k]} (\gamma_a)_{i,y} = 1, \forall i \in [n_a] \right\},$$

where

$$\widehat{\text{TPR}}_a(\gamma)_y := \frac{1}{n \hat{p}_{ay}} \sum_{i \in [n_a]} f_a(x_{a,i})_y \cdot (\gamma_a)_{i,y}, \quad \hat{p}_{ay} := \frac{1}{n} \sum_{i \in [n_a]} f_a(x_{a,i})_y \quad (8)$$

(cf. Line 2 and Eqs. (5) and (6)). Note that  $\hat{D}_a$  is a polygon, since it is specified by linear constraints.

To obtain the utility-maximizing fair TPR  $\hat{t}_a$ 's, we take the empirical maximizer subject to the  $\alpha$ -TPR constraint via solving a linear program (cf. Line 3 and Eqs. (3) and (4)):

$$\text{LP1}(\alpha) : \max_{\hat{t}_1, \dots, \hat{t}_m \in \hat{D}_m} \widehat{\mathcal{U}}(\hat{t}_1, \dots, \hat{t}_m) \quad \text{s.t.} \quad \|\hat{t}_a - \hat{t}_{a'}\|_\infty \leq \alpha, \forall a, a' \in [m],$$

where  $\widehat{\mathcal{U}}(\hat{t}_1, \dots, \hat{t}_m) := \sum_{a,y} v_y \hat{p}_{ay}(\hat{t}_a)_y$  is the empirical utility.

**Step 2** (Obtaining Fair Classifier of Desired Form). The next step is finding a classifier that achieves  $\hat{t}_a$ 's on the empirical distribution, i.e., Lines 5 and 6. To implement Line 5, note that another way of approaching this problem is to realize that among all eligible  $(\beta_a, h_a)$ -pairs, the  $h_a$  associated with the maximum  $\beta_a$  value must satisfy  $\widehat{\text{TPR}}_a(h_a) \in \partial \hat{D}_a$  (otherwise, a contradiction can be reached using the fact that  $\hat{D}_a \subseteq [0, 1]^k$  is compact; also see the right figure of Fig. 2). Combined with the strategy above of representing classifiers using lookup tables, we get the following linear program:

$$\text{LP2}(t, q) : \max_{\gamma, \beta} \beta \quad \text{s.t.} \quad t = (1 - \beta) \widehat{\text{TPR}}(\gamma) + \beta q \quad \text{and} \quad \gamma \in \mathbb{R}_{\geq 0}^{n \times k}, \sum_{y \in [k]} \gamma_{i,y} = 1, \forall i \in [n].$$

Finally, on Line 6, we find a tilting  $\lambda_a$  s.t. after coordinate-wise multiplied by the scores, the argmax class assignment has nonzero probability according to the classifier  $\gamma_a$  found in the preceding step:

$$\text{LP3}(\gamma) : \min_{\lambda} 0 \quad \text{s.t.} \quad \lambda_y f(x_i)_y \geq \lambda_{y'} f(x_i)_{y'} \quad \forall i \in [n], y, y' \in [k], \gamma_{i,y} > 0.$$

The feasible set of this problem is nonempty by Proposition 3.1, because we are *treating*  $f$  as if it were the Bayes score function, and the empirical distribution  $\hat{\mu}^{X,A}$  as the population.

All combined, our algorithm involves solving  $(2m + 1)$  linear programs, where LP1 is the dominating one with  $O(nk)$  variables and constraints; solving which (to near-optimality) takes, e.g.,  $\tilde{O}(\text{poly}(nk))$  time using interior point methods [38].

## 245 4.2 Sample Complexity

246 Thanks to the low function complexity of the post-processing maps (Proposition B.12) used in our  
247 algorithm to derive classifiers of the form in Eq. (1), it has the following sample complexity:

248 **Theorem 4.2.** *Let  $h : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$  be the (probabilistic) classifier derived from a score function*  
249  *$f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  using Algorithm 1 with the empirical distribution of samples in Assumption 4.1*  
250 *as the argument. Then under Assumptions 2.1 and 2.4, for any group-wise distribution calibrated*  
251 *reference score function  $\bar{f} : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  with finer granularity than  $f$  (Definition 3.4), and*  
252  *$n \geq \Omega(\max_{a,y} \ln(mk/\delta)/p_{ay})$ ,*

$$\begin{aligned} |\bar{U} - \mathcal{U}(h)| &\leq \|v\|_1 O\left(\max_{a,y} \frac{1}{p_{ay}} \left(\epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n}\right)\right), \\ \Delta_{\text{TPR}}(h) &\leq \alpha + O\left(\max_{a,y} \frac{1}{p_{ay}} \left(\epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n}\right)\right), \end{aligned}$$

253 where  $p_{ay} := \mathbb{P}_\mu(A = a, Y = y)$ ,  $v$  is from Definition 2.3 of the utility,  $\bar{U}$  denotes the utility of the  
254 optimal  $\alpha$ -fair classifier derived from the reference  $\bar{f}$ , and  $\epsilon_{ay} := \mathbb{E}[|\bar{f}_a(X)_y - f_a(X)_y| \mathbf{1}[A = a]]$   
255 measures the miscalibration of  $f$  w.r.t.  $\bar{f}$  on group  $a$  and class  $y$ .

256 The bound consists of a calibration error  $\epsilon_{ay}$  as discussed in the remarks of Theorem 3.5, an estimation  
257 error from applying uniform convergence (the Natarajan dimension of the set of tiltings is  $O(k)$ ), and  
258 a  $k/n$  term that comes from the disagreement over class assignments on the samples between the  
259 (deterministic) tilting found on Line 6 and the (probabilistic) classifier on Line 5 due to tie-breaking.

## 260 5 Experiments

261 We evaluate Algorithm 1 for reducing TPR disparity on benchmark datasets, and demonstrate its  
262 effectiveness compared to existing post-processing as well as in-processing bias mitigation methods.

263 **Datasets.** The first task is income prediction, for which, we use the ACSIncome dataset [20]—an  
264 extension of the UCI Adult dataset [29] with much more examples (1.6 million vs. 30,162), allowing  
265 us to compare methods confidently. We consider a binary setting where the sensitive attribute is  
266 gender and the target is whether the income is over \$50k, as well as a multi-group multi-class setting  
267 with five race categories and five income buckets. The second is text classification, of identifying  
268 occupations (28 in total) from biographies in the BiasBios dataset [16]; sensitive attribute is gender.

269 **Baselines and Setup.** The main baseline is FairProjection [2]—the only post-processing algo-  
270 rithm applicable for multi-class TPR parity to our knowledge.<sup>3</sup> In the binary setting, we also compare  
271 to RejectOption [27]. To demonstrate the deficiencies of existing methods at reducing TPR dispar-  
272 ity, we additionally include in-processing results using Reductions [1] and Adversarial [46].<sup>45</sup>

273 On each task, we first create a pre-training split from the dataset and train a linear logistic regression  
274 scoring model (with isotonic calibration and five-fold cross-validation as implemented in `scikit-`  
275 `learn` [42, 43, 32]), then randomly split the remaining data for post-processing and testing with 10  
276 different seeds and aggregate the results (the pre-trained model remains the same). For in-processing,  
277 we use the same splits but merge the pre-training and post-processing data for training. On BiasBios,  
278 linear logistic regression is performed on embeddings of biographies computed by a pre-trained BERT  
279 model from the `bert-base-uncased` checkpoint [18] (in other words, head-tuning). Additional  
280 details, including hyperparameters, are included in Appendix C.1.

281 **Results.** In Fig. 3, we plot the tradeoff curves from varying the fairness tolerance ( $\alpha$  for our  
282 method). Our method is consistently the most effective at minimizing TPR disparity, particularly  
283 under multi-class settings, where existing algorithms only manage to partially reduce  $\Delta_{\text{TPR}}$  (and

<sup>3</sup>We use the authors' code, where TPR parity is equivalent to the `meo` constraint. The results from using the KL divergence variant is included, which are better than the cross-entropy variant in our experiments.

<sup>4</sup>Although Reductions is extended to multi-class by Yang et al. [41], an implementation was not provided.

<sup>5</sup>The implementation (with minor modifications) in the AIF360 library is used for the latter methods [6].

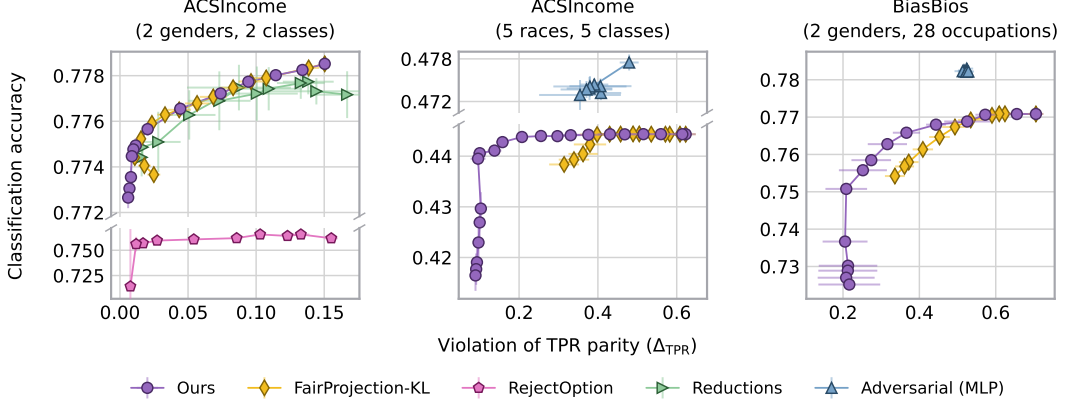


Figure 3: Tradeoff curves between accuracy and  $\Delta_{\text{TPR}}$  (Eq. (2)). Base model is logistic regression, except for Adversarial, which uses a feedforward network. Error bars indicate the standard deviation over 10 runs with different random splits. Running time is reported in appendix Table 1.

at a greater cost to accuracy when using FairProject and RejectOption). It also outperforms the in-processing Reductions on binary ACSIncome, and Adversarial in terms of  $\Delta_{\text{TPR}}$ , which, although enjoys higher accuracies because of the use of the more expressive feedforward networks as the prediction model, fails to reduce TPR parity. Sharper drops in accuracies are observed when applying our method with small  $\alpha$  settings, e.g.,  $\leq 0.02$ . We saw this happen when the randomized component in Eq. (1b) is activated (i.e.,  $\beta > 0$ ), meaning that Line 3 has found fair TPRs that lie in the interior of the feasible region of the better-performing group in order to match the feasible TPR on the worse-performing one(s). Hence the drop is expected because utility is being sacrificed to achieve TPR parity.

Although our method greatly reduces TPR disparity, there remains a gap to reaching  $\Delta_{\text{TPR}} = 0$ , especially on tasks with more classes (i.e., BiasBios, where a higher variance is also observed). While this could be due to miscalibration, or potentially a violation of Assumption 2.4, the main reason is suspected to be insufficient sample size. Recall from Theorem 4.2 that the sample complexity for  $\Delta_{\text{TPR}}$  scales as  $\tilde{O}(\sqrt{k/np_{ay}})$  in the worse-case  $(a, y)$ , which is itself at least  $\tilde{O}(\sqrt{mk^2/n})$ . Thus, learning generalizable classifiers that satisfy TPR parity under more groups and classes is much harder in terms of data requirement (and by extension, computing resource).

Lastly, we emphasize the necessity of group-wise calibration for achieving low  $\Delta_{\text{TPR}}$ , as the definition of the criterion involves conditioning on the true label (it is also reflected by the calibration error term  $\epsilon_{ay}$  in Theorem 4.2). In an ablation study (appendix Fig. 4), a larger (minimum achievable)  $\Delta_{\text{TPR}}$  is observed when no efforts are made to calibrate the scoring model. It is therefore necessary for model vendors to provide accurate uncertainty quantifications, and for practitioners building fair classifiers to verify and improve calibration.


## 6 Conclusions and Limitations

We described a post-processing method for reducing TPR disparity for equal opportunity in multi-class classification, and demonstrated its performance in comparison to existing algorithms on benchmark datasets, especially when the number of classes is large. We analyzed the sample complexity of our method, and established its optimality under model calibration.

The effectiveness of our method at reducing TPR disparity is largely contributed to the tailored analysis, although it limits our method to this fairness notion only. Some use cases may demand equalized odds ( $\hat{Y} \perp A \mid Y$ ) beyond TPR parity ( $\mathbb{1}[\hat{Y} = Y] \perp A \mid Y$ ), which is a more stringent criterion: TPR parity only needs to match the main diagonal of the (conditional) confusion matrix across groups, whereas equalized odds requires matching all  $k^2$  entries. The design of efficient algorithms for achieving equalized odds remains an open problem.<sup>6</sup>

<sup>6</sup>Most algorithms, e.g., [2], are only evaluated for TPR parity but not (multi-class) equalized odds.

## References

- [1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning*, pages 60–69, 2018.
- [2] Wael Alghamdi, Hsiang Hsu, Haewon Jeong, Hao Wang, P. Winston Michalak, Shahab Asoodeh, and Flavio P. Calmon. Beyond Adult and COMPAS: Fair Multi-Class Prediction via Information Projection. In *Advances in Neural Information Processing Systems*, 2022.
- [3] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.
- [4] Solon Barocas and Andrew D. Selbst. Big Data’s Disparate Impact. *California Law Review*, 104(3):671–732, 2016.
- [5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.
- [6] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias, 2018. *arxiv:1810.01943 [cs.AI]*.
- [7] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?  . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.
- [8] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research*, 50(1):3–44, 2021.
- [9] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- [10] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [11] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, pages 77–91, 2018.
- [12] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building Classifiers with Independency Constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18, 2009.
- [13] Flavio P. Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized Pre-Processing for Discrimination Prevention. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [14] Simon Caton and Christian Haas. Fairness in Machine Learning: A Survey, 2020. *arxiv:2010.04053 [cs.LG]*.
- [15] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*, October 2018. URL <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G>.

- [16] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In *Proceedings of the 2019 ACM Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.
- [17] Christophe Denis, Romuald Elie, Mohamed Hebiri, and François Hu. Fairness guarantee in multi-class classification, 2023. *arxiv:2109.13642 [math.ST]*.
- [18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [19] Steven Diamond and Stephen Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- [20] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring Adult: New Datasets for Fair Machine Learning. In *Advances in Neural Information Processing Systems*, 2021.
- [21] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness Through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.
- [22] Executive Office of the President. Big Data: A Report on Algorithmic Systems, Opportunity, and Civil Rights. The White House, 2016. URL <https://www.fdlp.gov/GPO/gpo90618>.
- [23] Solenne Gaucher, Nicolas Schreuder, and Evgenii Chzhenn. Fair learning with Wasserstein barycenters for non-decomposable performance measures. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 2436–2459, 2023.
- [24] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330, 2017.
- [25] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- [26] Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (Computationally-Identifiable) Masses. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1939–1948, 2018.
- [27] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision Theory for Discrimination-Aware Classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929, 2012.
- [28] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2564–2572, 2018.
- [29] Ron 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*, pages 202–207, 1996.
- [30] Meelis Kull and Peter Flach. Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration. In *Machine Learning and Knowledge Discovery in Databases*, pages 68–85, 2015.
- [31] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, second edition, 2018.

- [32] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
- [33] Preston Putzel and Scott Lee. Blackbox Post-Processing for Multiclass Fairness. In *Proceedings of the Workshop on Artificial Intelligence Safety 2022*, volume 3087, 2022.
- [34] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, 2020.
- [35] Julien Rouzot, Julien Ferry, and Marie-José Huguet. Learning Optimal Fair Scoring Systems for Multi-Class Classification. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence*, 2022.
- [36] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- [37] Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution Calibration for Regression. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5897–5906, 2019.
- [38] Pravin M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989.
- [39] Blake Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning Non-Discriminatory Predictors. In *Proceedings of the 2017 Conference on Learning Theory*, pages 1920–1953, 2017.
- [40] Ruicheng Xian, Lang Yin, and Han Zhao. Fair and Optimal Classification via Post-Processing Predictors. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.
- [41] Forest Yang, Mouhamadou Cisse, and Sanmi Koyejo. Fairness with Overlapping Groups. In *Advances in Neural Information Processing Systems*, volume 33, pages 4067–4078, 2020.
- [42] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the 18th International Conference on Machine Learning*, pages 609–616, 2001.
- [43] Bianca Zadrozny and Charles Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002.
- [44] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180, 2017.
- [45] Richard Zemel, Yu Ledell Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning Fair Representations. In *Proceedings of the 30th International Conference on Machine Learning*, pages 325–333, 2013.
- [46] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating Unwanted Biases with Adversarial Learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.
- [47] Han Zhao and Geoffrey J. Gordon. Inherent Tradeoffs in Learning Fair Representations. *Journal of Machine Learning Research*, 23(57):1–26, 2022.
- [48] Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J. Gordon. Conditional Learning of Fair Representations. In *International Conference on Learning Representations*, 2020.

- 454 [49] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender Bias  
455 in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018*  
456 *Conference of the North American Chapter of the Association for Computational Linguistics:*  
457 *Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, 2018.
- 458 [50] Shengjia Zhao, Michael P. Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating  
459 Predictions to Decisions: A Novel Approach to Multi-Class Calibration. In *Advances in Neural*  
460 *Information Processing Systems*, 2021.

461 **A Omitted Proofs for Section 3.1**

462 *Proof of Proposition 3.1.* For the *only if* direction, let  $h$  be a probabilistic classifier, then set  $g$  s.t.

$$\mathbb{P}(g(s) = y) = \mathbb{P}(h(X) = y \mid f^*(X) = s), \quad \forall s \in \Delta_k.$$

463 We verify that

$$\begin{aligned} \text{TPR}(g \circ f^*)_y &= \mathbb{P}(g \circ f^*(X) = y \mid Y = y) \\ &= \frac{1}{p_y} \int_{s \in \Delta_k} \mathbb{P}(g(s) = y, f^*(X) = s, Y = y) \\ &= \frac{1}{p_y} \int_{s \in \Delta_k} \mathbb{P}(g(s) = y, Y = y \mid f^*(X) = s) \mathbb{P}(f^*(X) = s) \\ &= \frac{1}{p_y} \int_{s \in \Delta_k} \mathbb{P}(g(s) = y) \mathbb{P}(Y = y \mid f^*(X) = s) \mathbb{P}(f^*(X) = s) \\ &= \frac{1}{p_y} \int_{s \in \Delta_k} \mathbb{P}(h(X) = y \mid f^*(X) = s) \mathbb{P}(Y = y \mid f^*(X) = s) \mathbb{P}(f^*(X) = s) \\ &= \frac{1}{p_y} \int_{s \in \Delta_k} \mathbb{P}(h(X) = y, f^*(X) = s, Y = y) \\ &= \mathbb{P}(h(X) = y \mid Y = y) \\ &= \text{TPR}(h)_y. \end{aligned}$$

464 For the *only if* direction, let  $g$  be a probabilistic post-processing map, then set  $h$  s.t.

$$\mathbb{P}(h(x) = y) = \mathbb{P}(g \circ f^*(x) = y), \quad \forall x \in \mathcal{X}.$$

465 Since the two classifiers agree (probabilistically),  $\text{TPR}(h)_y = \text{TPR}(g \circ f^*)_y$ . □

466 The proof of Proposition 3.3 makes use of the supporting hyperplane of the convex set  $D$ , which  
467 could be proved from the separating hyperplane theorem [10, Section 2.5.2]:

468 **Theorem A.1** (Supporting Hyperplane). *Let  $C \subset \mathbb{R}^d$  be a nonempty convex set, and  $x \in \partial C$  a point  
469 on its boundary, then  $\exists v \in \mathbb{R}^d$ ,  $v \neq 0$ , s.t.  $v^\top x \geq v^\top x'$  for all  $x' \in C$ .*

470 *Proof of Proposition 3.3.* First of all, recall that  $D$  is a convex set because randomized classifiers are  
471 allowed. Let  $f^*$  denote the Bayes score function on  $\mu$ , and note that the ground-truth TPR can be  
472 computed via

$$\begin{aligned} \text{TPR}(h)_y &= \mathbb{P}(h(X) = y \mid Y = y) \\ &= \frac{1}{p_y} \mathbb{P}(h(X) = y, Y = y) \\ &= \frac{1}{p_y} \int_{x \in \mathcal{X}} f^*(x)_y \mathbb{P}(h(x) = y) d\mu^X(x) \end{aligned} \tag{9}$$

473 where  $p_y = \mathbb{E}[f^*(x)_y]$ . Also, for all  $v \in \mathbb{R}^k$ ,  $v \neq 0$ , the classifier  $h_v^*$  maximizes the utility  
474  $\sum_y v_y p_y \text{TPR}(h_v)_y$  if and only if

$$h_v^*(x) \in \arg \max_y v_y \mathbb{P}(Y = y \mid X = x), \quad \forall x \in \mathcal{X}. \tag{10}$$

475 For the *only if* direction, let  $h$  be s.t.  $\text{TPR}(h) \in \partial D$ , and suppose to the contrary that  $\forall \lambda \in \mathbb{R}^k$ ,  
476  $\lambda \neq 0$ ,  $\exists A_\lambda \subseteq \mathcal{X}$  with measure nonzero s.t.  $h(x) \notin \arg \max_y \lambda_y f^*(x)_y$ ,  $\forall x \in A_\lambda$ . This implies

477 that for all  $v \in \Delta_k$ , by Eqs. (9) and (10),

$$\begin{aligned}
& \sum_{y \in [k]} v_y p_y \text{TPR}(h_v^*)_y - \sum_{y \in [k]} v_y p_y \text{TPR}(h)_y \\
&= \sum_{y \in [k]} v_y p_y (\mathbb{P}(h_v^*(X) = y \mid Y = y) - \mathbb{P}(h(X) = y \mid Y = y)) \\
&= \int_{x \in A_{v/p}} \sum_{y \in [k]} v_y f^*(x)_y (\mathbb{P}(h_v^*(X) = y) - \mathbb{P}(h(X) = y)) d\mu^X(x) \\
&> 0;
\end{aligned}$$

478 given that  $p \in \mathbb{R}_{>0}^k$ , this contradicts the fact that since  $\text{TPR}(h) \in \partial D$ , by Theorem A.1,  $\exists v \in \Delta^k$ ,  
479  $v \neq 0$ , s.t.  $\sum_y v_y \text{TPR}(h)_y \geq \sum_y v_y \text{TPR}(h')_y$  for all  $h'$ .

480 For the *if* direction, let  $h$  be s.t.  $\exists \lambda \in \mathbb{R}^k$ ,  $\lambda \neq 0$  s.t.  $h(x) \in \arg \max_y \lambda_y f^*(x)_y$ . Then we know  
481 from Eq. (10) that

$$\sum_{y \in [k]} \lambda_y p_y \text{TPR}(h)_y \geq \sum_{y \in [k]} \lambda_y p_y \text{TPR}(h')_y, \quad \forall h' : \mathcal{X} \rightarrow \mathcal{Y},$$

482 which implies that  $\text{TPR}(h) \in \partial D$ . □

## 483 B Omitted Proofs for Sections 3.2 and 4

484 In this section, we provide the proofs to Theorems 3.5 and 4.2. As mentioned in the remarks following  
485 Definition 3.4, we relax the requirement for *distribution calibration* by replacing the miscalibration  
486 measure in Eq. (7) with

$$\epsilon'_{ay} := \max_{g \in \Lambda_k} |\mathbb{E}[(\bar{f}_a(X)_y - f_a(X)_y) \mathbb{1}[g \circ f_a(X) = y] \mathbb{1}[A = a]]|, \quad (11)$$

487 where

$$\Lambda_k := \left\{ s \mapsto \arg \max_{y'} \lambda_{y'} s_{y'} : \lambda \in \mathbb{R}^k \right\}$$

488 is the set of tilings. The relaxed measure  $\epsilon'_{ay}$  is clearly upper bounded by Eq. (7).

489 When  $\epsilon'_{ay} = 0$ ,  $f$  is said to satisfy the notion of (group-wise)  $\mathcal{L}^k$ -*decision calibration* proposed by  
490 Zhao et al. [50, Definition 2 and Proposition 2], who also provided a polynomial time post-processing  
491 algorithm for recalibration.

492 We restate and prove Theorems 3.5 and 4.2 with the relaxed miscalibration measure defined in  
493 Eq. (11):

494 **Theorem B.1.** *Let  $h : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$  be the (probabilistic) classifier derived from a score function*  
495  *$f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  using Algorithm 1. Then under Assumptions 2.1 and 2.4, for any group-wise*  
496 *distribution calibrated reference score function  $\bar{f} : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  with finer granularity than*  
497  *$f$  (Definition 3.4),*

$$\begin{aligned}
|\bar{\mathcal{U}} - \mathcal{U}(h)| &\leq 6 \|v\|_1 \max_{a,y} \frac{\epsilon'_{ay}}{p_{ay}}, \\
\Delta_{\text{TPR}}(h) &\leq \alpha + 4 \max_{a,y} \frac{4\epsilon'_{ay}}{p_{ay}},
\end{aligned}$$

498 where  $p_{ay} := \mathbb{P}_\mu(A = a, Y = y)$ ,  $v$  is from Definition 2.3 of the utility,  $\bar{\mathcal{U}}$  denotes the utility of the  
499 optimal  $\alpha$ -fair classifier derived from the reference  $\bar{f}$ , and  $\epsilon'_{ay}$  is defined in Eq. (11).

500 **Theorem B.2.** *Let  $h : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Y}$  be the (probabilistic) classifier derived from a score function*  
501  *$f : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  using Algorithm 1 with the empirical distribution of samples in Assumption 4.1*  
502 *as the argument. Then under Assumptions 2.1 and 2.4, for any group-wise distribution calibrated*

reference score function  $\bar{f} : \mathcal{X} \times \mathcal{A} \rightarrow \Delta_k$  with finer granularity than  $f$  (Definition 3.4), and  
 $n \geq \Omega(\max_{a,y} \ln(mk/\delta)/p_{ay})$ ,

$$|\bar{\mathcal{U}} - \mathcal{U}(h)| \leq \|v\|_1 O\left(\max_{a,y} \frac{1}{p_{ay}} \left(\epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n}\right)\right),$$

$$\Delta_{\text{TPR}}(h) \leq \alpha + O\left(\max_{a,y} \frac{1}{p_{ay}} \left(\epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n}\right)\right),$$

where  $p_{ay} := \mathbb{P}_\mu(A = a, Y = y)$ ,  $v$  is from Definition 2.3 of the utility,  $\bar{\mathcal{U}}$  denotes the utility of the  
optimal  $\alpha$ -fair classifier derived from the reference  $\bar{f}$ , and  $\epsilon'_{ay}$  is defined in Eq. (11).

We first state and prove Lemmas B.3, B.4 and B.13 below that we will use when proving the theorems.

**Lemma B.3.** Let  $f : \mathcal{X} \rightarrow \Delta_k$  be a score function,  $\bar{f}$  a distribution calibrated reference score  
function with finer granularity, and

$$h(x) = \begin{cases} \arg \max_{y'} \lambda_{y'} f(x)_{y'} & \text{w.p. } 1 - \beta \\ y & \text{w.p. } \beta q_y, \forall y \in [k] \end{cases}$$

for some  $\beta \in [0, 1]$  and  $\lambda \in \mathbb{R}^k$  (as in Theorem 3.2). Then under Assumptions 2.1 and 2.4,  $\forall y \in [k]$ ,

$$\left| \tilde{p}_y \widetilde{\text{TPR}}(h)_y - p_y \text{TPR}(h)_y \right| \leq \epsilon'_y,$$

$$\left| \widetilde{\text{TPR}}(h)_y - \text{TPR}(h)_y \right| \leq \frac{2\epsilon'_y}{p_y},$$

where  $p_y := \mathbb{P}(Y = y)$ ,  $\widetilde{\text{TPR}}$  and  $\tilde{p}_y$  are quantities induced by  $f$  as defined in Eq. (6), and  
 $\epsilon'_y := \max_{g \in \Lambda_k} |\mathbb{E}[(\bar{f}(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]|$ .

*Proof.* Define  $\bar{\beta} := 1 - \|\beta\|_1$ ,  $g(s) := \arg \max_{y'} \lambda_{y'} s_{y'}$ , and let  $f^*(x) := \mathbb{E}[Y \mid X = x]$  denote the  
Bayes score function. Note from Eq. (9) that

$$\text{TPR}(h)_y = \frac{1}{p_y} \int_{x \in \mathcal{X}} f^*(x)_y \mathbb{P}(h(x) = y) d\mu^X(x) = \beta_y + \frac{\bar{\beta}}{p_y} \mathbb{E}[f^*(X)_y \mathbb{1}[g \circ f(X) = y]]$$

and  $p_y = \mathbb{E}[f^*(x)_y] = \mathbb{E}[\bar{f}(x)_y]$  by calibration; on the other hand, by definition of  $\widetilde{\text{TPR}}$  in Eq. (6),

$$\widetilde{\text{TPR}}(h)_y := \frac{1}{\tilde{p}_y} \int_{x \in \mathcal{X}} f(x)_y \mathbb{P}(h(x) = y) d\mu^X(x) = \beta_y + \frac{\bar{\beta}}{\tilde{p}_y} \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]]$$

and  $\tilde{p}_y := \mathbb{E}[f(x)_y]$ .

Therefore,

$$\left| \tilde{p}_y \widetilde{\text{TPR}}(h)_y - p_y \text{TPR}(h)_y \right| = \beta_y |p_y - \tilde{p}_y| + \bar{\beta} |\mathbb{E}[(f^*(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]|; \quad (12)$$

where, for the first term, using the fact that the constant function  $s \mapsto y \in \Lambda$  (via setting, e.g.,  
 $\lambda = e_y$ ),

$$\begin{aligned} |p_y - \tilde{p}_y| &= |\mathbb{E}[\bar{f}(X)_y - f(X)_y]| \\ &\leq \max_{g' \in \Lambda} |\mathbb{E}[(\bar{f}(X)_y - f(X)_y) \mathbb{1}[g' \circ f(X) = y]]| \\ &\leq \epsilon'_y, \end{aligned} \quad (13)$$

and for the second term,

$$\begin{aligned} &|\mathbb{E}[(f^*(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]| \\ &\leq |\mathbb{E}[(f^*(X)_y - \bar{f}(X)_y) \mathbb{1}[g \circ f(X) = y]]| + |\mathbb{E}[(\bar{f}(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]| \\ &\leq \left| \mathbb{E} \left[ (\mathbb{1}[Y = y] - \bar{f}(X)_y) \mathbb{1} \left[ \bigvee_{s \in \Delta_k} f(X) = s \text{ and } y \in \arg \max_{y'} \lambda_{y'} s_{y'} \right] \right] \right| + \epsilon'_y \\ &= \epsilon'_y \end{aligned} \quad (14)$$

by Definition 3.4, because  $\bar{f}$  is calibrated and has finer granularity than  $f$ . In the arguments above, ties can be arbitrarily broken because by Assumption 2.4, the contribution from the set of tied scores—which has measure zero—can be ignored. Then the first claim follows by plugging Eqs. (13) and (14) back into Eq. (12) and using the fact that  $\bar{\beta} + \sum_{y'} \beta_{y'} = 1$ .

For the second claim, we have

$$\begin{aligned}
& \left| \widehat{\text{TPR}}(h)_y - \text{TPR}(h)_y \right| \\
&= \bar{\beta} \left| \frac{1}{p_y} \mathbb{E}[f^*(X)_y \mathbb{1}[g \circ f(X) = y]] - \frac{1}{\tilde{p}_y} \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] \right| \\
&= \bar{\beta} \left| \mathbb{E} \left[ \left( \frac{f^*(X)_y}{p_y} - \frac{f(X)_y}{\tilde{p}_y} \right) \mathbb{1}[g \circ f(X) = y] \right] \right| \\
&\leq \frac{\bar{\beta}}{p_y} |\mathbb{E}[(f^*(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]| + \frac{\bar{\beta}}{p_y} \left| 1 - \frac{p_y}{\tilde{p}_y} \right| \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] \\
&\leq \frac{\bar{\beta}}{p_y} \epsilon'_y + \frac{\bar{\beta}}{p_y} \left| 1 - \frac{p_y}{\tilde{p}_y} \right| \mathbb{E}[f(X)_y] \\
&= \frac{\bar{\beta}}{p_y} \epsilon'_y + \frac{\bar{\beta}}{p_y} |p_y - \tilde{p}_y| \\
&\leq \frac{2\bar{\beta}}{p_y} \epsilon'_y
\end{aligned} \tag{15}$$

by Eqs. (13) and (14). The claim then follows by noting that  $\bar{\beta} \leq 1$ .  $\square$

**Lemma B.4.** Let  $f : \mathcal{X} \rightarrow \Delta_k$  be a score function,  $x_1, \dots, x_n \sim \mu^X$  i.i.d. samples, and

$$h(x) = \begin{cases} \arg \max_{y'} \lambda_{y'} f(x)_{y'} & \text{w.p. } 1 - \beta \\ y & \text{w.p. } \beta q_y, \forall y \in [k] \end{cases}$$

for some  $\beta \in [0, 1]$  and  $\lambda \in \mathbb{R}^k$  (as in Theorem 3.2). Then under Assumptions 2.1 and 2.4, w.p. at least  $1 - \delta$ ,  $\forall y \in [k]$ ,

$$\begin{aligned}
\left| \tilde{p}_y \widehat{\text{TPR}}(h)_y - \hat{p}_y \widehat{\text{TPR}}(h)_y \right| &\leq O \left( \sqrt{\frac{k \ln k / \delta}{n}} \right), \\
\left| \widehat{\text{TPR}}(h)_y - \widehat{\text{TPR}}(h)_y \right| &\leq O \left( \frac{1}{\tilde{p}_y} \sqrt{\frac{k \ln k / \delta}{n}} \right),
\end{aligned}$$

where  $\tilde{p}_y := \mathbb{E}[f(x)_y]$ ,  $\widehat{\text{TPR}}$  is induced by  $f$  as defined in Eq. (6), and  $\widehat{\text{TPR}}$  and  $\hat{p}_y$  their finite sample estimates as defined in Eq. (8).

The proof to this lemma requires concentration inequality and uniform convergence results.

**Theorem B.5** (Hoeffding's Inequality). Let  $x_1, \dots, x_n \in \mathbb{R}$  be i.i.d. random variables s.t.  $a_i \leq x_i \leq b_i$  almost surely. Then w.p. at least  $1 - \delta$ ,  $|\frac{1}{n} \sum_{i=1}^n (x_i - \mathbb{E} x_i)| \leq \sqrt{\sum_{i=1}^n (b_i - a_i)^2 / 2n^2 \cdot \ln 2 / \delta}$ .

**Definition B.6** (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 pseudo-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 B.7** (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  $d_{\text{VC}}(\mathcal{H})$ , is the size of the largest subset of  $\mathcal{X}$  shattered by  $\mathcal{H}$ .

**Definition B.8** (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.  $\mathbb{1}[f(x_i) \geq t_i] = b_i$  for all  $i \in [n]$ .

**Definition B.9** (Pseudo-Dimension). Let  $\mathcal{F}$  be a class of functions from  $\mathcal{X}$  to  $\mathbb{R}$ . The pseudo-dimension of  $\mathcal{F}$ , denoted by  $d_{\text{P}}(\mathcal{F})$ , is the size of the largest subset of  $\mathcal{X}$  pseudo-shattered by  $\mathcal{F}$ .

**Theorem B.10** (Pseudo-Dimension Uniform Convergence). *Let  $\mathcal{H}$  be a class of functions from  $\mathcal{X}$  to  $\mathbb{R}$ ,  $\ell : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$  a nonnegative loss function upper bounded by  $M$ ,  $p$  a distribution over  $\mathcal{X} \times \mathcal{Y}$ , of which  $(x_1, y_1), \dots, (x_n, y_n) \sim p$  are i.i.d. samples. Then w.p. at least  $1 - \delta$  over the random draw of the samples,  $\forall h \in \mathcal{H}$ ,*

$$\left| \mathbb{E}_{(X,Y) \sim p} \ell(h(X), Y) - \frac{1}{n} \sum_{i=1}^n \ell(h(x_i), y_i) \right| \leq cM \sqrt{\frac{d + \ln 1/\delta}{n}}$$

for some universal constant  $c$ , where  $d := d_P(\{(x, y) \mapsto \ell(h(x), y) : h \in \mathcal{H}\})$ .

This can be proved via a reduction to the VC uniform convergence bound; see [36, Theorem 6.8] and [31, Theorem 11.8]. We will use this theorem to establish the following VC bound for weighted 0-1 error:

**Theorem B.11.** *Let  $\mathcal{H}$  be a class of binary functions from  $\mathcal{X}$  to  $\{0, 1\}$ ,  $p$  a distribution over  $\mathcal{X} \times \mathcal{Y}$ , of which  $(x_1, y_1), \dots, (x_n, y_n) \sim p$  are i.i.d. samples. Define nonnegative weighting  $w(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ , and let  $M := \sup_{x,y} w(x, y)$ . Then w.p. at least  $1 - \delta$  over the random draw of the samples,  $\forall h \in \mathcal{H}$ ,*

$$\left| \mathbb{E}_{(X,Y) \sim p} [w(X, Y) \mathbb{1}[h(X) \neq Y]] - \frac{1}{n} \sum_{i=1}^n w(x_i, y_i) \mathbb{1}[h(x_i) \neq y_i] \right| \leq cM \sqrt{\frac{d_{VC}(\mathcal{H}) + \ln 1/\delta}{n}}$$

for some universal constant  $c$ .

*Proof.* We are essentially considering a weighted variant of the 0-1 loss, for which Theorem B.10 can be applied. We only need to show that the weighting does not increase the complexity of  $\mathcal{H}$ .

Let  $d := d_{VC}(\mathcal{H})$ , and  $\{(x_1, y_1), \dots, (x_{d+1}, y_{d+1})\} \subseteq \mathcal{X} \times \{0, 1\}$  s.t. the  $x_i$ 's are distinct w.l.o.g. Suppose  $\mathcal{F} := \{(x, y) \mapsto w(x, y) \mathbb{1}[h(x) \neq y]\}$  pseudo-shatters this set, then  $\exists t_1, \dots, t_{d+1}$  s.t.  $\forall b_1, \dots, b_{d+1} \in \{0, 1\}$  and for all  $i$ ,

$$\begin{aligned} & \exists f \in \mathcal{F}, \mathbb{1}[f(x_i, y_i) \geq t_i] = b_i \\ \iff & \exists h \in \mathcal{H}, \begin{cases} \mathbb{1}[h(x_i) \neq y_i] \geq t_i/w(x_i, y_i) & \text{if } b_i = 1 \\ \mathbb{1}[h(x_i) \neq y_i] < t_i/w(x_i, y_i) & \text{if } b_i = 0 \end{cases} \\ \iff & \exists h \in \mathcal{H}, \mathbb{1}[h(x_i) \neq y_i] = b_i \\ \iff & \exists h \in \mathcal{H}, h(x_i) = b_i \text{ XOR } y_i, \end{aligned}$$

where the third line follows from realizing that  $t_i/w(x_i, y_i) \in (0, 1]$ , otherwise the inequality will always fail in one direction regardless of  $h$  (note that  $b_i$  can be arbitrary). Since the  $x_i$ 's are distinct, the above implies that  $\mathcal{H}$  shatters a set of size  $(d+1) > d_{VC}(\mathcal{H}) = d$ , which is a contradiction, so  $d_P(\mathcal{F}) < d+1$ .  $\square$

Last but not least, we bound the VC dimension of tiltings in one-vs.-all mode by  $k$ .

**Proposition B.12.** *Let  $k \geq 2$ , and fix  $y \in [k]$ . Let  $d$  denote the VC dimension of the class of binary functions from  $\Delta_k$  to  $\{0, 1\}$  given by  $\{s \mapsto \mathbb{1}[\lambda_y s_y \in \max_{y'} \lambda_{y'} s_{y'}] : \lambda \in \mathbb{R}^k\}$ , then  $d \leq O(k \ln k)$ .*

*Proof.* Note that any  $g \in \{s \mapsto \mathbb{1}[\lambda_y s_y \in \max_{y'} \lambda_{y'} s_{y'}] : \lambda \in \mathbb{R}^k\}$  can be written as

$$s \mapsto \mathbb{1} \left[ \sum_{y'} \mathbb{1}[\lambda_y s_y - \lambda_{y'} s_{y'} \geq 0] \geq k \right],$$

which is implemented by a two-layer feed-forward linear threshold network with  $(3k+1)$  weights and thresholds in total and  $(2k+1)$  computation units, so  $d \leq 2(3k+1) \log_2(2(2k+1)/\ln 2)$ ; see [3, Chapter 6 and Theorem 6.1].  $\square$

575 *Proof of Lemma B.4.* Define  $\bar{\beta} := 1 - \|\beta\|_1$ ,  $g(s) := \arg \max_{y'} \lambda_{y'} s_{y'}$ . Recall from Eqs. (6) and (8)  
 576 that

$$\begin{aligned}\widehat{\text{TPR}}(h)_y &:= \frac{1}{\tilde{p}_y} \int_{x \in \mathcal{X}} f(x)_y \mathbb{P}(h(x) = y) d\mu^X(x) = \beta_y + \frac{\bar{\beta}}{\tilde{p}_y} \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]], \\ \widehat{\text{TPR}}(h)_y &:= \frac{1}{n\hat{p}_y} \sum_{i \in [n]} f(x_i)_y \mathbb{P}(h(x_i) = y) = \beta_y + \frac{\bar{\beta}}{n\hat{p}_y} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y],\end{aligned}$$

577 where

$$\tilde{p}_y := \mathbb{E}[f(x)_y], \quad \hat{p}_y := \frac{1}{n} \sum_{i \in [n]} f(x_i)_y.$$

578 Therefore,

$$\begin{aligned}& \left| \tilde{p}_y \widehat{\text{TPR}}(h)_y - \hat{p}_y \widehat{\text{TPR}}(h)_y \right| \\ &= \beta_y |\tilde{p}_y - \hat{p}_y| + \bar{\beta} \left| \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] - \frac{1}{n} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y] \right|; \quad (16)\end{aligned}$$

579 where, for the first term, by Theorem B.5, w.p. at least  $1 - \delta$ ,  $\forall y$ ,

$$|\tilde{p}_y - \hat{p}_y| \leq \sqrt{\frac{\ln 2k/\delta}{2n}}, \quad (17)$$

580 and for the second term, by Theorem B.11 and Proposition B.12, w.p. at least  $1 - \delta$ ,  $\forall g, y$ ,

$$\left| \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] - \frac{1}{n} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y] \right| \leq c \sqrt{\frac{O(k \ln k) + \ln k/\delta}{n}} \quad (18)$$

581 since  $f(x_i)_y \in [0, 1]$ . Then the first claim follows by plugging Eqs. (17) and (18) back into Eq. (16).  
 582 Again, ties that arise when applying the tilting  $g$  can be arbitrarily broken because by Assumption 2.4,  
 583 the contribution from the set of tied scores—which has measure zero—can be ignored.

584 For the second claim, we have w.p. at least  $1 - \delta$ ,  $\forall y$ ,

$$\begin{aligned}& \left| \widehat{\text{TPR}}(h)_y - \widehat{\text{TPR}}(h)_y \right| \\ &= \bar{\beta} \left| \frac{1}{\tilde{p}_y} \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] - \frac{1}{n\hat{p}_y} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y] \right| \\ &\leq \frac{\bar{\beta}}{\tilde{p}_y} \left| \mathbb{E}[f(X)_y \mathbb{1}[g \circ f(X) = y]] - \frac{1}{n} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y] \right| \\ &\quad + \frac{\bar{\beta}}{\tilde{p}_y} \left| 1 - \frac{\tilde{p}_y}{\hat{p}_y} \right| \frac{1}{n} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y] \\ &\leq \frac{\bar{\beta}}{\tilde{p}_y} c \sqrt{\frac{O(k \ln k) + \ln k/\delta}{n}} + \frac{\bar{\beta}}{\tilde{p}_y} \left| 1 - \frac{\tilde{p}_y}{\hat{p}_y} \right| \frac{1}{n} \sum_{i \in [n]} f(x_i)_y \\ &= \frac{\bar{\beta}}{\tilde{p}_y} c \sqrt{\frac{O(k \ln k) + \ln k/\delta}{n}} + \frac{\bar{\beta}}{\tilde{p}_y} |\hat{p}_y - \tilde{p}_y| \\ &\leq \frac{\bar{\beta}}{\tilde{p}_y} \left( c \sqrt{\frac{O(k \ln k) + \ln k/\delta}{n}} + \sqrt{\frac{\ln 2k/\delta}{2n}} \right)\end{aligned}$$

585 by Eqs. (17) and (18). The claim then follows by noting that  $\bar{\beta} \leq 1$ . □

586 We combine Lemma B.4 and the analysis of Lemma B.3 to obtain the following lemma that will be  
 587 applied directly in the proof of Theorem B.2:

588 **Lemma B.13.** *Let  $f : \mathcal{X} \rightarrow \Delta_k$  be a score function,  $\bar{f}$  a distribution calibrated reference score*  
 589 *function with finer granularity,  $x_1, \dots, x_n \sim \mu^X$  i.i.d. samples, and*

$$h(x) = \begin{cases} \arg \max_{y'} \lambda_{y'} f(x)_{y'} & \text{w.p. } 1 - \beta \\ y & \text{w.p. } \beta q_y, \forall y \in [k] \end{cases}$$

590 *for some  $\beta \in [0, 1]$  and  $\lambda \in \mathbb{R}^k$  (as in Theorem 3.2). Then under Assumptions 2.1 and 2.4, w.p. at*  
 591 *least  $1 - \delta$ ,  $\forall y \in [k]$ ,*

$$\left| \hat{p}_y \widehat{\text{TPR}}(h)_y - p_y \text{TPR}(h)_y \right| \leq \epsilon'_y + O\left(\sqrt{\frac{k \ln k / \delta}{n}}\right),$$

592 *and for all  $n \geq \Omega(\max_y \ln(k/\delta)/p_y)$ ,*

$$\left| \widehat{\text{TPR}}(h)_y - \text{TPR}(h)_y \right| \leq O\left(\frac{\epsilon'_y}{p_y} + \frac{1}{p_y} \sqrt{\frac{k \ln k / \delta}{n}}\right),$$

593 *where  $p_y := \mathbb{P}(Y = y)$ ,  $\widehat{\text{TPR}}$  and  $\hat{p}_y$  are finite sample estimates of quantities induced by  $f$  as defined*  
 594 *in Eq. (8), and  $\epsilon'_y := \max_{g \in \Lambda_k} |\mathbb{E}[(f(X)_y - \bar{f}(X)_y) \mathbb{1}[g \circ f(X) = y]]|$ .*

595 *Proof.* Define  $\bar{\beta} := 1 - \|\beta\|_1$ ,  $g(s) := \arg \max_{y'} \lambda_{y'} s_{y'}$ , and let  $f^*(x) := \mathbb{E}[Y \mid X = x]$  denote the  
 596 Bayes score function. Note that the ground-truth TPR can be computed as follows, where we also  
 597 define its finite sample estimate:

$$\begin{aligned} \text{TPR}(h)_y &= \beta_y + \frac{\bar{\beta}}{p_y} \mathbb{E}[f^*(X)_y \mathbb{1}[g \circ f(X) = y]], \\ \widehat{\text{TPR}}(h)_y &:= \beta_y + \frac{\bar{\beta}}{n \check{p}_y} \sum_{i \in [n]} f^*(x_i)_y \mathbb{1}[g \circ f(x_i) = y], \end{aligned}$$

598 *where  $p_y := \mathbb{P}(Y = y) = \mathbb{E}[f^*(x)_y] = \mathbb{E}[\bar{f}(x)_y]$  by calibration, and  $\check{p}_y := \frac{1}{n} \sum_{i \in [n]} f^*(x_i)_y$ ; on*  
 599 *the other hand, recall from Eq. (8) that*

$$\widehat{\text{TPR}}(h)_y := \frac{1}{n \hat{p}_y} \sum_{i \in [n]} f(x_i)_y \mathbb{P}(h(x_i) = y) = \beta_y + \frac{\bar{\beta}}{n \hat{p}_y} \sum_{i \in [n]} f(x_i)_y \mathbb{1}[g \circ f(x_i) = y],$$

600 *where  $\hat{p}_y := \frac{1}{n} \sum_{i \in [n]} f(x_i)_y$ .*

601 *Applying Lemma B.4 to  $f^*$ , we get w.p. at least  $1 - \delta$ ,  $\forall y \in [k]$ ,*

$$\left| p_y \text{TPR}(h)_y - \check{p}_y \widehat{\text{TPR}}(h)_y \right| \leq O\left(\sqrt{\frac{k \ln k / \delta}{n}}\right), \quad (19)$$

$$\left| \text{TPR}(h)_y - \widehat{\text{TPR}}(h)_y \right| \leq O\left(\frac{1}{p_y} \sqrt{\frac{k \ln k / \delta}{n}}\right). \quad (20)$$

602 *Now, for the first claim,*<sup>7</sup>

$$\left| \check{p}_y \widehat{\text{TPR}}(h)_y - \hat{p}_y \widehat{\text{TPR}}(h)_y \right| = \beta_y |\check{p}_y - \hat{p}_y| + \bar{\beta} \left| \frac{1}{n} \sum_{i \in [n]} (f^*(x_i)_y - f(x_i)_y) \mathbb{1}[g \circ f(x_i) = y] \right|; \quad (21)$$

<sup>7</sup>The indicator in the summation should have been  $\mathbb{1}[g \circ f^*(x_i) = y]$ , but the score function in that term is decoupled in the analysis of Theorem B.2, hence interchangeable.

where, for the first term, by applying Theorem B.5 twice and following the same analysis in Eq. (13),  
w.p. at least  $1 - \delta$ ,  $\forall y$ ,

$$\begin{aligned} |\check{p}_y - \hat{p}_y| &= \left| \frac{1}{n} \sum_{i \in [n]} (\bar{f}(x_i)_y - f(x_i)_y) \right| \\ &\leq |\mathbb{E}[\bar{f}(X)_y - f(X)_y]| + O\left(\sqrt{\frac{\ln k/\delta}{n}}\right) \\ &\leq \epsilon'_y + O\left(\sqrt{\frac{\ln k/\delta}{n}}\right), \end{aligned} \quad (22)$$

and for the second term, by Theorem B.11 and Proposition B.12,<sup>8</sup> followed by the same analysis in Eq. (14), w.p. at least  $1 - \delta$ ,  $\forall g, y$ ,

$$\begin{aligned} &\left| \frac{1}{n} \sum_{i \in [n]} (f^*(x_i)_y - f(x_i)_y) \mathbb{1}[g \circ f(x_i) = y] \right| \\ &\leq |\mathbb{E}[(f^*(X)_y - f(X)_y) \mathbb{1}[g \circ f(X) = y]]| + O\left(\sqrt{\frac{k \ln k/\delta}{n}}\right) \\ &\leq \epsilon'_y + O\left(\sqrt{\frac{k \ln k/\delta}{n}}\right). \end{aligned} \quad (23)$$

Again, we have relied on Assumption 2.4 to avoid tie-breaking issues. Then the first claim follows by plugging Eqs. (22) and (23) back into Eq. (21), combining with Eq. (19), and using the fact that  $\bar{\beta} + \sum_{y'} \beta_{y'} = 1$ .

For the second claim, by Eqs. (22) and (23) and the same analysis in Eq. (15), w.p. at least  $1 - \delta$ ,  $\forall g, y$ ,

$$\left| \widetilde{\text{TPR}}(h)_y - \widehat{\text{TPR}}(h)_y \right| \leq O\left(\frac{\bar{\beta}}{\check{p}_y} \epsilon'_y + \frac{\bar{\beta}}{\check{p}_y} \sqrt{\frac{k \ln k/\delta}{n}}\right)$$

The claim then follows by noting that  $\bar{\beta} \leq 1$ , combining with Eq. (20), and the fact that  $\check{p}_y = \Theta(p_y)$  when  $n \geq \Omega(\max_y \ln(k/\delta)/p_y)$  by Theorem B.5.  $\square$

Finally, we get back to the proofs of Theorems B.1 and B.2.

*Proof of Theorem B.1.* We begin with the second claim, where by Lemma B.3,  $\forall a, a', y$ ,

$$\begin{aligned} &|\text{TPR}_a(h)_y - \text{TPR}_{a'}(h)_y| \\ &\leq \left| \widetilde{\text{TPR}}_a(h)_y - \widetilde{\text{TPR}}_{a'}(h)_y \right| + \left| \widetilde{\text{TPR}}_a(h)_y - \text{TPR}_a(h)_y \right| + \left| \widetilde{\text{TPR}}_{a'}(h)_y - \text{TPR}_{a'}(h)_y \right| \\ &\leq \alpha + \frac{2\epsilon'_{ay}}{p_{ay}} + \frac{2\epsilon'_{a'y}}{p_{a'y}}, \end{aligned}$$

and the claim follows by taking the max over  $a, a', y$ .

For the first claim, let  $\bar{h} := \arg \max_{\Delta_{\text{TPR}}(h') \leq \alpha} \sum_{a,y} v_y p_{ay} \text{TPR}_a(h')_y$  denote the classifier that achieves  $\bar{U}$ , and recall that  $h := \arg \max_{\Delta_{\widetilde{\text{TPR}}}(h') \leq \alpha} \sum_{a,y} v_y p_{ay} \widetilde{\text{TPR}}_a(h')_y$ , where analogous to Eq. (2) we defined

$$\Delta_{\widetilde{\text{TPR}}}(\hat{Y}) := \max_{a, a' \in \mathcal{A}} \left\| \widetilde{\text{TPR}}_a(\hat{Y}) - \widetilde{\text{TPR}}_{a'}(\hat{Y}) \right\|_{\infty}.$$

<sup>8</sup>Note that the weight  $(f^*(x_i)_y - f(x_i)_y) \in [-1, 1]$  can be made nonnegative by adding 1.

620 Then by Lemma B.3,

$$\begin{aligned}
\bar{\mathcal{U}} - \mathcal{U}(h) &= \sum_{a \in [m], y \in [k]} v_y p_{ay} (\text{TPR}_a(\bar{h})_y - \text{TPR}_a(h)_y) \\
&\leq \sum_{a \in [m], y \in [k]} v_y \tilde{p}_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h)_y \right) + \sum_{a \in [m], y \in [k]} 2v_y \epsilon'_{ay} \\
&\leq \sum_{a \in [m], y \in [k]} v_y \tilde{p}_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h)_y \right) + \sum_{a \in [m], y \in [k]} 2v_y p_{ay} \max_{a', y'} \frac{\epsilon'_{a'y'}}{p_{a'y'}} \\
&\leq \sum_{a \in [m], y \in [k]} v_y \tilde{p}_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h)_y \right) + 2\|v\|_1 \max_{a, y} \frac{\epsilon'_{ay}}{p_{ay}} \tag{24}
\end{aligned}$$

621 by Hölder's inequality, since  $\sum_a p_{ay} \leq 1$  for all  $y$ .

622 Our next step is to use the fact that  $h$  is the maximizer of  $\tilde{\mathcal{U}}$  subject to  $\Delta_{\widetilde{\text{TPR}}}(h) \leq \alpha$  to eliminate  
623 the first summation, except that the constraint may not be satisfied by  $\bar{h}$ . So we introduce a third  
624 classifier  $h'$  s.t.

$$\widetilde{\text{TPR}}_a(h')_y = \text{proj}_{[t_y - \alpha, t_y + \alpha]} \left( \widetilde{\text{TPR}}_a(h)_y \right), \quad \forall y \in [k],$$

625 where  $t_y := \widetilde{\text{TPR}}_b(h)_y$  with  $b$  satisfying  $\widetilde{\text{TPR}}_b(h)_y \in \{t_y : t \in \bigcap_{a \in [m]} \tilde{D}_a\}$ , the intersection of  
626 the feasible regions restricted to coordinate  $y$ ; or in other words,  $t_y$  is the  $\widetilde{\text{TPR}}$  of class  $y$  on the  
627 worst-performing group  $b$ . Because  $|\widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_b(\bar{h})_y| \leq \alpha + 2\epsilon'_{ay}/p_{ay} + 2\epsilon'_{by}/p_{by}$  (by a  
628 similar argument to the one above) for all  $a, y$ , it follows by a case analysis that

$$\left| \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h')_y \right| \leq \frac{2\epsilon'_{ay}}{p_{ay}} + \frac{2\epsilon'_{by}}{p_{by}} \leq 4 \max_{a', y'} \frac{\epsilon'_{a'y'}}{p_{a'y'}}, \quad \forall a \in [m], y \in [k].$$

629 Then,

$$\begin{aligned}
&\sum_{a \in [m], y \in [k]} v_y p_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h)_y \right) \\
&= \sum_{a \in [m], y \in [k]} v_y p_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h')_y + \widetilde{\text{TPR}}_a(h')_y - \widetilde{\text{TPR}}_a(h)_y \right) \\
&\leq \sum_{a \in [m], y \in [k]} v_y p_{ay} \left( \widetilde{\text{TPR}}_a(\bar{h})_y - \widetilde{\text{TPR}}_a(h')_y \right) \\
&\leq \sum_{a \in [m], y \in [k]} 4v_y p_{ay} \max_{a', y'} \frac{\epsilon'_{a'y'}}{p_{a'y'}} \\
&\leq 4\|v\|_1 \max_{a, y} \frac{\epsilon'_{ay}}{p_{ay}}, \tag{25}
\end{aligned}$$

630 then one side of the claim follows from plugging this back into Eq. (24); the other side follows  
631 symmetrically by using the fact that  $\bar{h}$  is the maximizer of  $\mathcal{U}$  subject to  $\Delta_{\text{TPR}}(\bar{h}) \leq \alpha$ .  $\square$

632 *Proof of Theorem B.2.* Let  $\gamma$  denote the probabilistic classifier found on Line 5 of Algorithm 1 that  
633 operates on the samples. We begin with the second claim, where by Lemma B.13, w.p. at least  $1 - \delta$ ,  
634  $\forall a, a', y$  and  $n \geq \Omega(\max_{a, y} \ln(mk/\delta)/p_{ay})$ ,

$$\begin{aligned}
&|\text{TPR}_a(h)_y - \text{TPR}_{a'}(h)_y| \\
&\leq \left| \widehat{\text{TPR}}_a(\gamma)_y - \widehat{\text{TPR}}_{a'}(\gamma)_y \right| + \left| \widehat{\text{TPR}}_a(h)_y - \text{TPR}_a(h)_y \right| + \left| \widehat{\text{TPR}}_{a'}(h)_y - \text{TPR}_{a'}(h)_y \right| \\
&\quad + \left| \widehat{\text{TPR}}_a(h)_y - \widehat{\text{TPR}}_a(\gamma)_y \right| + \left| \widehat{\text{TPR}}_{a'}(h)_y - \widehat{\text{TPR}}_{a'}(\gamma)_y \right| \\
&\leq \alpha + O\left( \frac{\epsilon'_{ay}}{p_{ay}} + \frac{\epsilon'_{a'y}}{p_{a'y}} + \left( \frac{1}{p_{ay}} + \frac{1}{p_{a'y}} \right) \left( \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n} \right) \right),
\end{aligned}$$

where the last two terms are upper bounded via

$$\begin{aligned} \left| \widehat{\text{TPR}}_a(h)_y - \widehat{\text{TPR}}_a(\gamma)_y \right| &= \left| \frac{\bar{\beta}}{n_a \hat{p}_{y|a}} \sum_{i \in [n_a]} f_a(x_i)_y (\mathbb{1}[h_a(x_i) = y] - \mathbb{P}(\gamma_a(x_i) = y)) \right| \\ &\leq \frac{k \bar{\beta}}{n_a \hat{p}_{y|a}}, \end{aligned}$$

since by construction, the interior of the set  $\{s \in \Delta_k \mid \exists x \in \mathcal{X} : f_a(x) = s, h_a(x) = y\}$  is equivalent to that of  $\{s \in \Delta_k \mid \exists x \in \mathcal{X} : f_a(x) = s, \mathbb{P}(\gamma_a(x) = y) = 1\}$ , so  $h_a$  disagrees with  $\gamma_a$  only on the boundary, when  $\lambda_y f_a(x)_y = \lambda_{y'} f_a(x)_{y'}$  for some  $y'$  (i.e., there is a tie). Because locations where ties can occur is specified by  $k$  hyperplanes, and the probability of having two random samples  $f_a(x_i), f_a(x_j)$ ,  $i \neq j$  lying on the same hyperplane is zero by Assumption 2.4, the number of disagreements on the samples is no more than  $k$ . Finally, note that  $n_a \hat{p}_{y|a} = \Theta(n \hat{p}_{ya})$  when  $n \geq \Omega(\max_{a,y} \ln(mk/\delta)/p_{ay})$ .

The first claim then follows by taking the max over  $a, a', y$ .

For the first claim, applying Lemma B.13 followed by the same analysis above and in Eq. (25),

$$\begin{aligned} \bar{\mathcal{U}} - \mathcal{U}(h) &= \sum_{a \in [m], y \in [k]} v_y p_{ay} (\text{TPR}_a(\bar{h})_y - \text{TPR}_a(h)_y) \\ &\leq \sum_{a \in [m], y \in [k]} v_y \hat{p}_{ay} \left( \widehat{\text{TPR}}_a(\bar{h})_y - \widehat{\text{TPR}}_a(\gamma)_y + O\left(\frac{1}{p_{ay}} \left( \epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n} \right) \right) \right) \\ &\leq \sum_{a \in [m], y \in [k]} v_y \hat{p}_{ay} O\left( \max_{a', y'} \frac{\epsilon'_{a'y'}}{\hat{p}_{a'y'}} + \frac{1}{p_{ay}} \left( \epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n} \right) \right) \\ &\leq \|v\|_1 O\left( \max_{a,y} \frac{1}{p_{ay}} \left( \epsilon'_{ay} + \sqrt{\frac{k \ln mk/\delta}{n}} + \frac{k}{n} \right) \right); \end{aligned}$$

again, we have used the fact that  $\check{p}_{ay} = \Theta(p_{ay})$  when  $n \geq \Omega(\max_{a,y} \ln(mk/\delta)/p_{ay})$  by Theorem B.5. The other side follows symmetrically.  $\square$

## C Additional Experiment Details and Results

### C.1 Experiment Setup

**Dataset.** On ACSIncome, the dataset is randomly split 0.63/0.07/0.3 for training and calibrating the score function, post-processing, and testing, respectively. On BiasBios, it is split 0.35/0.35/0.3. In-processing methods are run on the two training splits merged.

We use the version of BiasBios scrapped and prepared by Ravfogel et al. [34], which contains 393,423 examples in total (vs. the 397,340 gathered by De-Arteaga et al. [16]).

**Scoring Model.** For training the logistic regression scoring model on which the post-processing methods are based, and to achieve (some level of) group-wise calibration, we use `CalibratedClassifierCV` (with one-vs.-all isotonic calibration method) from the `scikit-learn` package using logistic regression as the base model (with 10,000 iterations), which trains the logistic regression model while performing isotonic calibration with five-fold cross validation [42, 43]. In addition, to achieve *group-wise* calibration, we train one `CalibratedClassifierCV` on each group separately.

In Fig. 4, we compare the post-processing results of our algorithm to those applied on a scoring method without any attempts at calibration. The uncalibrated scoring model is trained directly with `LogisticRegression` on all groups in aggregate.

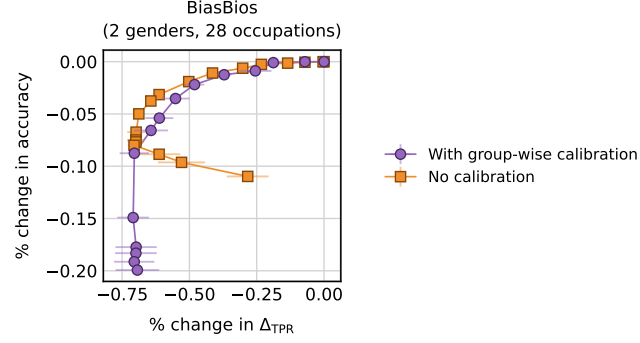


Figure 4: Tradeoff curves between accuracy and  $\Delta_{\text{TPR}}$  (Eq. (2)) by Algorithm 1 on a group-wise calibrated logistic regression scoring model and an uncalibrated one. Error bars indicate the standard deviation over 10 runs with different random splits.

Table 1: Running time of post-processing methods, averaged over three random splits.

	ACSIIncome		BiasBios
Groups	2	5	2
Classes	2	5	28
Examples (post-processing split)	116,515		137,698
Ours	137	1829	4764
RejectOption	75	-	-
FairProject-KL (GPU)	22	30	84

**Hyperparameters.** For all baseline methods, we use default settings that came with the code/package. In particular, for FairProject, increasing the number of iterations to over 1,000 did not improve performance. The tradeoff curves in Fig. 3 are generated with the following fairness tolerance/strictness settings.

For our method,  $\alpha$  is set to:

- ACSIIncome (binary). 0.14, 0.12, 0.1, 0.08, 0.05, 0.02, 0.01, 0.008, 0.005, 0.002, 0.001, 0.0001.
- ACSIIncome (5-group, 5-class). 0.6, 0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.08, 0.05, 0.02, 0.01, 0.008, 0.005, 0.002, 0.001, 0.0001.
- BiasBios. 0.5, 0.45, 0.3, 0.25, 0.2, 0.15, 0.1, 0.08, 0.05, 0.02, 0.01, 0.008, 0.005, 0.002.

For FairProject-KL, tolerance is set to:

- ACSIIncome (binary). 0.2, 0.12, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.0.
- ACSIIncome (5-group, 5-class). 0.6, 0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.08, 0.05, 0.02, 0.01, 0.008, 0.005, 0.002, 0.001, 0.0001.
- BiasBios. 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.0.

For RejectOption on binary ACSIIncome,  $\text{metric\_ub}$  and  $\text{metric\_lb}$  are set to plus/minus 200, 100, 50, 40, 35, 30, 20, 15, 10, 5, 1.

For Reductions on binary ACSIIncome, tolerance is set to 1.0, 0.14, 0.12, 0.1, 0.08, 0.05, 0.02, 0.01, 0.001, 0.0001. We use LogisticRegression as the base model.

For Adversarial, the strength of the adversarial loss is set to 0.0001, 0.001, 0.01, 0.1, 0.2, 0.5, 1. The base model is a one-hidden-layer feedforward ReLU network.

686 **Algorithm 1 Implementation.** In our code, the linear programs of Algorithm 1 introduced when  
687 it is instantiated to finite sample estimation in Section 4 are implemented through the `cvxpy` [19]  
688 interface, and for solving which we use the COIN-OR Cbc solver based on *branch and cut*.<sup>9</sup>

## 689 C.2 Additional Results

690 **Running Time.** We report the running time of the post-processing algorithms considered in  
691 Section 5 in Table 1. The experiments are run on an Intel Xeon Silver 4314 for CPU implementations,  
692 and an NVIDIA RTX A6000 for GPU implementations (namely, FairProject). Note that our  
693 method runs on a single core, so multiple experiments with different levels of fairness constraint  $\alpha$  can  
694 be run in parallel. The times are recorded under the strictest tolerance setting (see **Hyperparameters**  
695 in the previous section).

696 **Calibration.** In Fig. 4, we compare the results of our post-processing Algorithm 1 on a score  
697 function trained with group-wise calibration in mind to those of an uncalibrated one. It is observed  
698 that under smaller settings of  $\alpha$ , TPR disparity increases instead of seeing further reductions. This is  
699 because when the scores are uncalibrated, they do not reflect the true class probabilities, and on the  
700 other hand, smaller  $\alpha$  settings means the algorithm will have less tolerance to errors of the scores;  
701 these two reasons combined cause the rebound of  $\Delta_{\text{TPR}}$  observed with the uncalibrated model.

---

<sup>9</sup><https://github.com/coin-or/Cbc>.