# Beyond Adult and COMPAS: Fair Multi-Class Prediction via Information Projection

**Wael Alghamdi**[1*], **Hsiang Hsu**[1*], **Haewon Jeong**[1*],
**Hao Wang**[1], **P. Winston Michalak**[1], **Shahab Asoodeh**[2], **Flavio P. Calmon**[1]
[1]John A. Paulson School of Engineering and Applied Sciences, Harvard University
[2]Department of Computing and Software, McMaster University

## Abstract

We consider the problem of producing fair probabilistic classifiers for multi-class classification tasks. We formulate this problem in terms of "projecting" a pre-trained (and potentially unfair) classifier onto the set of models that satisfy target group-fairness requirements. The new, projected model is given by post-processing the outputs of the pre-trained classifier by a multiplicative factor. We provide a parallelizable, iterative algorithm for computing the projected classifier and derive both sample complexity and convergence guarantees. Comprehensive numerical comparisons with state-of-the-art benchmarks demonstrate that our approach maintains competitive performance in terms of accuracy-fairness trade-off curves, while achieving favorable runtime on large datasets. We also evaluate our method at scale on an open dataset with multiple classes, multiple intersectional groups, and over 1M samples.

## 1 Introduction

Machine learning (ML) algorithms are increasingly used to automate decisions that have significant social consequences. This trend has led to a surge of research on designing and evaluating fairness interventions that prevent discrimination in ML models. When dealing with *group fairness*, fairness interventions aim to ensure that a ML model does not discriminate against different groups determined by, for example, race, sex, and/or nationality. Extensive comparisons between discrimination control methods can be found in [BDH+18, FSV+19, WRC21]. As these studies demonstrate, there is still no "best" fairness intervention for ML, and the majority of existing approaches are tailored to either binary classification tasks, binary population groups, or both.[2] Moreover, discrimination control methods are often tested on overused datasets of modest sizes collected in either the US or Europe (e.g., UCI Adult [Lic13] and COMPAS [ALMK16]).

Most fairness interventions in ML focus on binary outcomes. In this case, the classification output is either positive or negative, and group-fairness metrics are tailored to binary decisions [HPS16]. While binary classification covers a range of ML tasks of societal importance (e.g., whether to approve a loan, whether to admit a student), there are many cases where the predicted variable is not binary. For example, in education, grading algorithms assign one out of several grades to students. In healthcare, predicted outcomes are frequently not binary (e.g., severity of disease).

We introduce a theoretically-grounded discrimination control method called `FairProjection`. This method ensures group fairness in multi-class classification for several, potentially overlapping population groups. We consider group fairness metrics that are natural multi-class extensions of their binary

*Equal contribution. Correspondence to: Wael Alghamdi and Flavio P. Calmon (`alghamdi@g.harvard.edu` and `flavio@seas.harvard.edu`).

[2]See Related Work and Table 1 for notable exceptions.

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

classification counterparts, such as statistical parity [FFM+15], equalized odds [HPS16], and error rate imbalance [PRW+17, Cho17]. When restricted to two predicted classes, `FairProjection` performs competitively against state-of-the-art fairness interventions tailored to binary classification tasks. `FairProjection` is model-agnostic (i.e., applicable to any model class) and scalable to datasets that are orders of magnitude larger than standard benchmarks found in the fair ML literature.

Our approach is based on an information-theoretic formulation called *information projection*. We show that this formulation is particularly well-suited for ensuring fairness in probabilistic classifiers with multi-class outputs. Given a probability distribution $P$ and a convex set of distributions $\mathcal{P}$, the goal of information projection is to find the "closest" distribution to $P$ in $\mathcal{P}$. The study of information projection can be traced back to [Csi75], which used KL-divergence to measure "distance" between distributions. Since then, information projection has been extended to other divergence measures, such as $f$-divergences [Csi95] and Rényi divergences [KS16, KS15]. Recently, [AAW+20a] studied how to project a probabilistic classifier, viewed as a conditional distribution, onto the set of classifiers that satisfy target group-fairness requirements. Remarkably, the projected classifier is obtained by multiplying (i.e., post-processing) the predictions of the original classifier by a factor that depends on the group-fairness constraints.

Prior work on information projection relies on a critical—and limiting—information-theoretic assumption: the underlying probability distributions are *known exactly*. This is infeasible in practical ML applications, where only a set of training examples sampled from the underlying data distribution is available. `FairProjection` fills this gap by using an efficient algorithm for computing the projected classifier with finite samples. We establish theoretical guarantees for this algorithm in terms of convergence and sample complexity.

Notably, our proposed fairness intervention is parallelizable (e.g., on a GPU). Hence, `FairProjection` scales to datasets with the number of samples comparable to the population of many US states ($> 10^6$ samples). We provide a TensorFlow [AAB+15] implementation of `FairProjection`[3] and apply it to post-process the outputs of probabilistic classifiers to ensure group fairness.

We benchmark our post-processing approach against several state-of-the-art fairness interventions selected based on the availability of reproducible code, and qualitatively compare it against many others. Our numerical results are among the most comprehensive comparison of fairness interventions to date. We present performance results on the HSLS (High School Longitudinal Study, used in [JWC22]), Adult [Lic13], and COMPAS [ALMK16] datasets.

We also evaluate `FairProjection` on a dataset derived from open and anonymized data from Brazil's national high school exam—the *Exame Nacional do Ensino Médio* (ENEM)—with over 1 million samples. We made use of this dataset due to the need for large-scale benchmarks for evaluating fairness interventions in multi-class classification tasks. We also answer recent calls [BZZ+21, DHMS21] for moving away from overused datasets such as Adult [Lic13] and COMPAS [ALMK16]. We hope that the ENEM dataset encourages researchers in the field of fair ML to test their methods within broader contexts.[4]

In summary, our main contributions are: **(i)** We introduce a post-processing fairness intervention for multi-class classification problems that can account for multiple protected groups and is scalable to large datasets; **(ii)** We derive finite-sample guarantees and convergence-rate results for our post-processing method. Importantly, `FairProjection` makes information projection practical without requiring exact knowledge of probability distributions; **(iii)** We demonstrate the favourable performance of our approach through comprehensive benchmarks against state-of-the-art fairness interventions; **(iv)** We put forth a new large-scale dataset (ENEM) for benchmarking discrimination control methods in multi-class classification tasks; this dataset may encourage researchers in fair ML to evaluate their methods beyond Adult and COMPAS.

**Related work.** We summarize key differentiating factors from prior work in Table 1 and provide a more in-depth discussion in Appendix B.5. The fairness interventions that are the most similar to ours

[3] Our code can be found at https://github.com/HsiangHsu/Fair-Projection.

[4] Since (to the best of our knowledge) the ENEM dataset has not been used in fair ML, we provide in Appendix C a datasheet for the ENEM dataset. The data can be found at [INE20] and code for pre-processing the data can be found at https://github.com/HsiangHsu/Fair-Projection.

| Method | Multiclass | Multigroup | Feature Scores | Curve | Parallel | Rate | Metric |
|---|---|---|---|---|---|---|---|
| Reductions [ABD+18] | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | SP, (M)EO |
| Reject-option [KKZ12] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | SP, (M)EO |
| EqOdds [HPS16] | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | EO |
| LevEqOpp [CDH+19] | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | FNR |
| CalEqOdds [PRW+17] | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | MEO |
| FACT [KCT20] | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | SP, (M)EO |
| Identifying[5] [JN20] | ✓✗ | ✓ | ✓ | ✓ | ✗ | ✗ | SP, (M)EO |
| FST [WRC20, WRC21] | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | SP, (M)EO |
| Overlapping [YCK20] | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | SP, (M)EO |
| Adversarial [ZLM18] | ✓ | ✓ | N/A[6] | ✓ | ✓ | ✗ | SP, (M)EO |
| `FairProjection` (ours) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | SP, (M)EO |

**Table 1:** Comparison between benchmark methods. **Multiclass/multigroup**: implementation takes datasets with multiclass/multigroup labels; **Scores**: processes raw outputs of probabilistic classifiers; **Curve**: outputs fairness-accuracy tradeoff curves (instead of a single point); **Parallel**: parallel implementation (e.g., on GPU) is available; **Rate**: convergence rate or sample complexity guarantee is proved; **Metric**: applicable fairness metric, with SP↔Statistical Parity, EO↔Equalized Odds, MEO↔Mean EO. Since `FairProjection` is a post-processing method, we focus our comparison on post-processing fairness intervention methods, except for Reductions [ABD+18], which is a representative in-processing method, and Adversarial [ZLM18], which we use to benchmark multi-class prediction. For comparing in-processing methods, see [LPB+21, Table 1].

are the FairScoreTransformer [WRC20, WRC21, FST] and the pre-processing method in [JN20]. The FST and [JN20] can be viewed as instantiations of `FairProjection` when restricted to the binary classification setting and to cross-entropy (for FST) or KL-divergence (for [JN20]) as the $f$-divergence of choice. Thus, our approach is a generalization of both methods to multiple $f$-divergences. Importantly, unlike our method, [JN20] requires retraining a classifier multiple times.

A reductions approach for fair classification was introduced in [ABD+18]. When restricted to binary classification, the benchmarks in Section 5 indicate that the reductions approach consistently achieves the most competitive fairness-accuracy trade-off compared to ours. `FairProjection` has two key differences from [ABD+18]: it is not restricted to binary classification tasks and does not require refitting a classifier several times over the training dataset. These are also key differentiating points from [CHKV19], which presented a meta-algorithm for fair classification that accounts for multiple constraints and groups. The reductions approach was later significantly generalized in the GroupFair method by [YCK20] to account for overlapping groups and multiple predicted classes. Unlike [YCK20], we do not require retraining classifiers.

Several other recent fairness intervention methods consider optimizing accuracy under group-fairness constraints. In [CJG+19], a "proxy-Lagrangian" formulation was proposed for incorporating non-differentiable rate constraints, including group fairness constraints. We avoid non-differentiability issues by considering the probabilities (scores) at the output of the classifier instead of thresholded decisions. In [ZVRG17], a fairness-constrained optimization was introduced that is applicable to margin-based classifiers (our approach can be used on any probabilistic classifier). In [CDPF+17] and [MW18], the fairness-accuracy trade-offs in binary classification tasks are characterized when the underlying distributions are known. A non-parity-based fairness notion was proposed in [KGZ19], called "multiaccuracy," which aims to ensure high accuracy for all subgroups even when the group information is not given in the data. We limit our analysis to parity notions of group fairness. To circumvent the non-differentiability of group-fairness constraints, approximate fairness constraints based on functionals found in information theory have been explored in [LPB+21, Rényi mutual information], [BNBR19, Rényi maximal correlation], and [PQC+19, maximum mean discrepancy]. We avoid such non-differentiability issues by casting group fairness constraints in the score domain.

[5] [JN20] mention that their method can be applied to multi-class classification, but their reported benchmarks are only for binary classification tasks.

[6] [ZLM18] is an in-processing method unlike other benchmarks in the table. It does not take a pre-trained classifier as an input.

| **Fairness Criterion** | Statistical parity | Equalized odds | Overall accuracy equality |
|---|---|---|---|
| **Expression** | $\left\lvert \frac{P_{\widehat{Y}\mid S=a}(c')}{P_{\widehat{Y}}(c')} - 1 \right\rvert \leq \alpha$ | $\left\lvert \frac{P_{\widehat{Y}\mid Y=c,S=a}(c')}{P_{\widehat{Y}\mid Y=c}(c')} - 1 \right\rvert \leq \alpha$ | $\left\lvert \frac{P(\widehat{Y}=Y \mid S=a)}{P(\widehat{Y}=Y)} - 1 \right\rvert \leq \alpha$ |

**Table 2:** Standard multi-class group fairness criteria; one fixes $\alpha > 0$ and iterates over all $(a, c, c') \in [A] \times [C]^2$.

**Notation.** Boldface Latin letters will always refer to vectors or matrices. The entries of a vector $\boldsymbol{z}$ are denoted by $z_j$, and those of a matrix $\boldsymbol{G}$ by $G_{i,j}$. The all-1 and all-0 vectors are denoted by $\mathbf{1}$ and $\mathbf{0}$. We set $[N] \triangleq \{1, \cdots, N\}$ and $\mathbb{R}_+ \triangleq [0, \infty)$. The probability simplex over $[N]$ is denoted by $\boldsymbol{\Delta}_N \triangleq \{\boldsymbol{p} \in \mathbb{R}_+^N \,;\, \mathbf{1}^T\boldsymbol{p} = 1\}$, and $\boldsymbol{\Delta}_N^+$ is its (relative) interior. If $P$ is a Borel probability measure over $\mathbb{R}^N$, $Z \sim P$ is a random variable, and $f : \mathbb{R}^N \to \mathbb{R}^K$ is Borel, then the expectation of $f(Z)$ is denoted by $\mathbb{E}[f(Z)] = \mathbb{E}_P[f] = \mathbb{E}_P[f(Z)] = \mathbb{E}_{Z\sim P}[f(Z)]$. We use the standard asymptotic notations $O$, $\Theta$, and $\Omega$.

## 2 Problem formulation and preliminaries

**Classification tasks.** The essential objects in classification are the input sample space $\mathcal{X}$, the predicted classes $\mathcal{Y}$, and the classifiers. We fix two random variables $X$ and $Y$, taking values in sets $\mathcal{X}$ and $\mathcal{Y} \triangleq [C]$. Here, $(X, Y)$ is a pair comprised of an input sample and corresponding class label randomly drawn from $\mathcal{X} \times \mathcal{Y}$ with distribution $P_{X,Y}$. A probabilistic classifier is a function $\boldsymbol{h} : \mathcal{X} \to \boldsymbol{\Delta}_C$, where $h_c(x)$ represents the probability of sample $x \in \mathcal{X}$ falling in class $c \in \mathcal{Y}$. Thus, $\boldsymbol{h}$ gives rise to a $\mathcal{Y}$-valued random variable $\widehat{Y}$ via the distribution $P_{\widehat{Y}\mid X=x}(c) \triangleq h_c(x)$.

**Group-fairness constraints.** Let $S$ be a group attribute (e.g., race and/or sex), taking values in $\mathcal{S} \triangleq [A]$. We consider multi-class generalization of three commonly used group fairness criteria in Table 2. As observed by existing works [see, e.g., ABD+18, MW18, CHKV19, WRC20, AAW+20a], each of these fairness constraints[7] can be written in the vector-inequality form $\mathbb{E}_{P_X}[\boldsymbol{G}\boldsymbol{h}] \leq \mathbf{0}$ for a closed-form matrix-valued function $\boldsymbol{G} : \mathcal{X} \to \mathbb{R}^{K\times C}$. For instance, for statistical parity, the $\boldsymbol{G}$ matrix evaluated at a fixed individual $x \in \mathcal{X}$ has $K = 2AC$ rows indexed by $(\delta, a, c') \in \{0, 1\} \times [A] \times [C]$, where the $(\delta, a, c')$-th row is equal to $\left((-1)^\delta P_S(a)^{-1} \sum_{c\in[C]} P_{S\mid X=x,Y=c}(a) h_c^{\text{base}}(x) - (\alpha + (-1)^\delta)\right) \boldsymbol{e}_{c'}$, with $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_C$ denoting the standard basis for $\mathbb{R}^C$. The expressions for the $\boldsymbol{G}$ matrix corresponding to the other fairness metrics are given in Appendix A.8, with a detailed derivation of statistical parity in Appendix A.9. Note that $\boldsymbol{G}$ depends on $P_{S\mid X,Y}$. If the group attribute $S$ is part of the input feature $X$, then $P_{S\mid X,Y}$ is simply replaced with an indicator function. Otherwise, we approximate this conditional distribution by training a probabilistic classifier.

**Goal.** Our goal is to design an efficient post-processing method that takes a pre-trained classifier $\boldsymbol{h}^{\text{base}}$ that may violate some target group-fairness criteria and finds a fair classifier that has the most similar outputs (i.e., closest utility performance) to that of $\boldsymbol{h}^{\text{base}}$.

**Fairness through information-projection.** We formulate the fairness intervention problem as follows. For a fixed search space $\mathcal{H} \subset \boldsymbol{\Delta}_C^{\mathcal{X}} \triangleq \{\boldsymbol{h} : \mathcal{X} \to \boldsymbol{\Delta}_C\}$, a loss function $\mathsf{err} : \boldsymbol{\Delta}_C^{\mathcal{X}} \times \boldsymbol{\Delta}_C^{\mathcal{X}} \to \mathbb{R}$, and a base classifier $\boldsymbol{h}^{\text{base}} \in \boldsymbol{\Delta}_C^{\mathcal{X}}$, one seeks to solve:

$$\underset{\boldsymbol{h}\in\mathcal{H}}{\text{minimize}}\ \mathsf{err}\left(\boldsymbol{h}, \boldsymbol{h}^{\text{base}}\right) \quad \text{subject to } \mathbb{E}_{P_X}\left[\boldsymbol{G}\boldsymbol{h}\right] \leq \mathbf{0}. \tag{1}$$

The function $\mathsf{err}$ quantifies the "closeness" between the scores given by $\boldsymbol{h}$ and $\boldsymbol{h}^{\text{base}}$. The constraint on $\boldsymbol{h}$ can encode any arbitrary statistical information about the joint distribution induced on the pair $(X, \widehat{Y})$. Specifically, any constraint $\mathbb{E}_{P_{X,\widehat{Y}}}[\boldsymbol{g}(X, \widehat{Y})] \leq \mathbf{0}$, where $\boldsymbol{g} : \mathcal{X} \times [C] \to \mathbb{R}^K$, may be recast in the form (1). Thus, solving the optimization (1) amounts to finding the minimal necessary perturbation to the base classifier $\boldsymbol{h}^{\text{base}}$ to make it satisfy a given on-average constraint. Since we

[7] We remark that our framework can be applied to other fairness constraints, e.g., the ones in [WRC20].

consider raw output scores, we measure "closeness" via $f$-divergences:

$$\mathsf{err}\left(\boldsymbol{h}, \boldsymbol{h}^{\text{base}}\right) = D_f(\boldsymbol{h} \| \boldsymbol{h}^{\text{base}} \mid P_X) \triangleq \mathbb{E}_{P_X}\left[\sum_{c\in[C]} h_c^{\text{base}}(X) f\left(\frac{h_c(X)}{h_c^{\text{base}}(X)}\right)\right] - f(1), \quad (2)$$

where $f$ is a convex function over $(0, \infty)$. By varying different choices of $f$, we can obtain e.g., cross-entropy (CE, $f(t) = -\log t$) and KL-divergence ($f(t) = t \log t$). For a chosen $f$-divergence, the optimization problem (1) becomes a generalization of *information projection* [Csi75].

**Preliminaries on information-projection.** In a recent work [AAW+20a], an optimal solution for the information projection formulation (1) was theoretically characterized. We briefly describe this result next. Let[8] $\mathcal{H} \triangleq \{\boldsymbol{h} \in \mathcal{C}(\mathcal{X}, \boldsymbol{\Delta}_C)\,;\ \inf_{c,x} h_c(x) > 0\}$ and we introduce the following definition and assumption.

**Definition 1.** For $\boldsymbol{p} \in \boldsymbol{\Delta}_C$, let $D_f^{\text{conj}}(\,\cdot\,, \boldsymbol{p})$ denote the convex conjugate of $D_f(\,\cdot\,\|\,\boldsymbol{p})$:

$$D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) \triangleq \sup_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} \boldsymbol{v}^T\boldsymbol{q} - D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}). \quad (3)$$

**Assumption 1.** Assume that: **(i)** $f \in \mathcal{C}^2(\mathbb{R})$, $f(1) = 0$, $f'(0^+) = -\infty$, and $f''(t) > 0$ for all $t > 0$; **(ii)** each $G_{k,c}$ is bounded, differentiable, and has bounded gradient; **(iii)** $\boldsymbol{h}^{\text{base}} \in \mathcal{H}$, and each $h_c^{\text{base}}$ has bounded partial derivatives; and **(iv)** there is an $\boldsymbol{h} \in \mathcal{H}$ such that $\mathbb{E}_{P_X}[\boldsymbol{G}\boldsymbol{h}] < \boldsymbol{0}$.

Now, the solution for (1) can be obtained by a simple "tilting" of the base classifier's output, as stated in the next theorem.

**Theorem 1** ([AAW+20a])**.** *If $f, \boldsymbol{h}^{\text{base}}$, and $\boldsymbol{G}$ satisfy Assumption 1, and $\mathcal{X} = \mathbb{R}^d$, then there is a unique solution $\boldsymbol{h}^{\text{opt}}$ for the optimization problem (1) for the $f$-divergence objective (2). Furthermore, $\boldsymbol{h}^{\text{opt}}$ is given by the tilt*

$$h_c^{\text{opt}}(x) = h_c^{\text{base}}(x) \cdot \phi\left(v_c(x; \boldsymbol{\lambda}^\star) + \gamma(x; \boldsymbol{\lambda}^\star)\right), \qquad (x, c) \in \mathcal{X} \times [C], \quad (4)$$

*where:* **(i)** *the function $\phi$ denotes the inverse of $f'$;* **(ii)** *the function $\boldsymbol{v} : \mathcal{X} \times \mathbb{R}^K \to \mathbb{R}^C$ is defined by $\boldsymbol{v}(x; \boldsymbol{\lambda}) \triangleq -\boldsymbol{G}(x)^T\boldsymbol{\lambda}$;* **(iii)** *the function $\gamma : \mathcal{X} \times \mathbb{R}^K \to \mathbb{R}$ is characterized by the equation $\mathbb{E}_{c\sim\boldsymbol{h}^{\text{base}}(x)}\left[\phi\left(v_c(x; \boldsymbol{\lambda}) + \gamma(x; \boldsymbol{\lambda})\right)\right] = 1$; and* **(iv)** *$\boldsymbol{\lambda}^\star \in \mathbb{R}^K$ is any solution to the convex problem*

$$D^\star \triangleq \min_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} \mathbb{E}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X; \boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(X)\right)\right]. \quad (5)$$

If the underlying data distribution is known, Theorem 1 yields an expression for the projected classifier as a post-processing of the base classifier. However, in practice, we do not know the underlying distribution and have to approximate it from a finite number of i.i.d. samples. In Section 3, we first describe how we approximate the solution given in Theorem 1 with finite samples. We then propose a parallelizable algorithm to solve the approximation in Section 4.

## 3 A finite-sample approximation of information projection

In practice, $P_X$ is unknown and only data points $\mathbb{X} \triangleq \{X_i\}_{i\in[N]} \subset \mathcal{X}$, drawn from $P_X$, are available. Thus, we propose the following fairness optimization problem. We search for a (multi-class) classifier $\boldsymbol{h} : \mathbb{X} \to \boldsymbol{\Delta}_C$ that solves the following:

$$\begin{aligned} \underset{\substack{\boldsymbol{h}:\mathbb{X}\to\boldsymbol{\Delta}_C \\ \boldsymbol{a}:\mathbb{X}\to\mathbb{R}^C, \boldsymbol{b}\in\mathbb{R}^K}}{\text{minimize}} \quad & D_f\left(\boldsymbol{h}\,\|\,\boldsymbol{h}^{\text{base}} \mid \widehat{P}_X\right) + \tau_1 \cdot \left(\mathbb{E}_{X\sim\widehat{P}_X}\left[\|\boldsymbol{a}(X)\|_2^2\right] + \|\boldsymbol{b}\|_2^2\right) \\ \text{subject to} \quad & \mathbb{E}_{\widehat{P}_X}\left[\boldsymbol{G}\cdot(\boldsymbol{h} + \tau_2\boldsymbol{a})\right] \le \tau_2\boldsymbol{b}, \end{aligned} \quad (6)$$

with $\widehat{P}_X$ being the empirical measure (e.g., obtained from a dataset), and $\tau_1, \tau_2 > 0$ prescribed constants. The terms $\boldsymbol{a}$ and $\boldsymbol{b}$ are added to circumvent infeasibility issues and aid convergence of our

[8] Here, $\mathcal{C}(\mathcal{X}, \boldsymbol{\Delta}_C)$ denotes the complete metric space of continuous functions from $\mathcal{X}$ to $\boldsymbol{\Delta}_C$, equipped with the sup-norm, i.e., $\|\boldsymbol{h}\| \triangleq \sup_{x\in\mathcal{X}} \|\boldsymbol{h}(x)\|_1$. In addition, we restrict attention to classifiers bounded away from the simplex boundary to simplify the proof of strong duality in Theorem 2 (see Remark 1 on our assumptions).

numerical procedure. We show in the following theorem that there is a unique solution for (6), and that it is given by a tilt (i.e., multiplicative factor) of $\boldsymbol{h}^{\text{base}}$. The tilting parameter is the solution of a finite-dimensional strongly convex optimization problem.

**Theorem 2.** *Suppose Assumption 1 holds, and set $\zeta \triangleq \tau_2^2/(2\tau_1)$. There exists a unique solution $\boldsymbol{h}^{\text{opt},N}$ to (6), and it is given by the formula*

$$h_c^{\text{opt},N}(x) = h_c^{\text{base}}(x) \cdot \phi\left(v_c(x;\boldsymbol{\lambda}^\star_{\zeta,N}) + \gamma(x;\boldsymbol{\lambda}^\star_{\zeta,N})\right), \quad (x,c) \in \mathbb{X} \times [C], \tag{7}$$

*with $\boldsymbol{v}, \phi, \gamma$ as in Theorem 1, and $\boldsymbol{\lambda}^\star_{\zeta,N} \in \mathbb{R}^K$ is the unique solution to the strongly convex problem*

$$D^\star_{\zeta,N} \triangleq \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \mathbb{E}_{\widehat{P}_X}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(X)\right)\right] + \frac{\zeta}{2}\left\|\boldsymbol{\mathcal{G}}_N^T\boldsymbol{\lambda}\right\|_2^2 \tag{8}$$

*where $\boldsymbol{\mathcal{G}}_N \triangleq \left(\boldsymbol{G}(X_1)/\sqrt{N}, \cdots, \boldsymbol{G}(X_N)/\sqrt{N}, \boldsymbol{I}_K\right) \in \mathbb{R}^{K\times(NC+K)}$.*

*Proof.* See Appendix A.1. □

Theorem 2 shows that: strong duality holds between the primal (6) and (the negative of) the dual (8); there is a unique classifier $\boldsymbol{h}^{\text{opt},N}$ minimizing our fairness formulation (6); there is a unique solution $\boldsymbol{\lambda}^\star_{\zeta,N}$ to the dual (5); and there is an explicit functional form of $\boldsymbol{h}^{\text{opt},N}$ in terms of $\boldsymbol{\lambda}^\star_{\zeta,N}$ in (7). Moreover, Theorem 2 yields a *practical* two-step procedure for solving the functional optimization in equation (6): (i) compute the dual variables $\boldsymbol{\lambda}$ by solving the strongly convex optimization in (8); (ii) tilt the base classifier by using the dual variables according to (7). This process is applied on real-world datasets using `FairProjection` (see Algorithm 1) in the next section.

The key distinctions between our formulation and Theorem 1 are that we use the empirical measure $\widehat{P}_X$ (e.g., produced using a dataset with i.i.d. samples), we have a *strongly* convex dual problem in (8) (in contrast to the convex program in (5)), and we prove strong duality in Theorem 2 (whereas an analogous strong duality is absent from the results of [AAW+20a]).

**Remark 1.** In practice, Assumption 1 is not a limiting factor for Theorem 2 and `FairProjection`. This is because: we are considering here a finite-set domain so continuity is automatic; we can perturb $\boldsymbol{h}^{\text{base}}$ by negligible noise to push it away from the simplex boundary; and the uniform classifier is strictly feasible. Nevertheless, Assumption 1 simplifies the derivation of our theoretical results.

## 4 Fair projection and theoretical guarantees

We introduce a parallelizable algorithm, `FairProjection`, that solves (6) using $N$ i.i.d. data points. We prove that its utility converges to $D^\star$ (see (5)) in the population limit and establish both sample-complexity and convergence rate guarantees. Applying `FairProjection` to the group-fairness intervention problem in (1) yields the optimal parameters in (7) for post-processing (i.e., tilting) the output of a multi-class classifier in order to satisfy target fairness constraints.

The `FairProjection` algorithm uses ADMM [BPC+11] to solve the convex program in (8). Recall that it suffices to optimize (8) for computing (6) as proved in Theorem 2. Algorithm 1 presents the steps of `FairProjection`, and its detailed derivation is given in Appendix A.2. A salient feature of `FairProjection` is its *parallelizability*. Each step that is done for $i$ varying over $[N]$ can be executed for each $i$ separately and in parallel. In particular, this applies to the most computationally intensive step, the $\boldsymbol{v}_i$-update step. We discuss next how the $\boldsymbol{v}_i$-update step is carried out.

**Inner iterations.** One approach to carry out the inner iteration in Algorithm 1 that updates $\boldsymbol{v}_i$ is to study the vanishing of the gradient of $\boldsymbol{v} \mapsto D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}_i) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}_i^T\boldsymbol{v}$ (where $\xi = (\rho+\zeta)/2$ and $\boldsymbol{a}_i \in \mathbb{R}^C$ is some vector). In the KL-divergence case, $D_{\mathsf{KL}}^{\text{conj}}$ is given by a log-sum-exp function, so its gradient is given by a softmax function, and equating the gradient to zero becomes a fixed-point equation. We give an iterative routine to solve this fixed point equation in Appendix A.3.1, whose proof of convergence is discussed in the same section. Beyond the KL-divergence case, setting the gradient to zero does not seem to be an analytically tractable problem. Nevertheless, we may reduce the vector minimization in Line 6 of Algorithm 1 to a tractable 1-dimensional root-finding problem, as the following result aids in showing.

**Lemma 1.** *For $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$, $\boldsymbol{a} \in \mathbb{R}^C$, and $\xi > 0$, if $f$ satisfies Assumption 1, we have that*

$$\min_{\boldsymbol{v}\in\mathbb{R}^C} D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v} = -\sup_{\theta\in\mathbb{R}} -\theta + \sum_{c\in[C]} \min_{q_c\geq 0} \; p_c f\left(\frac{q_c}{p_c}\right) + \frac{(a_c+q_c)^2}{4\xi} + \theta q_c. \quad (9)$$

*Proof.* See Appendix A.3.2. □

We note that the $\boldsymbol{v}_i$-update steps for both KL and CE (provided in detail in Appendix A.3.3) give, as a byproduct, the implicitly defined function $\gamma(x; \boldsymbol{\lambda})$ (see the statements of Theorems 1–2).

**Convergence guarantees.** Our proposed algorithm, `FairProjection`, enjoys the following convergence guarantees. The output after the $t$-th iteration $\boldsymbol{\lambda}_{\zeta,N}^{(t)}$ converges exponentially fast to $\boldsymbol{\lambda}_{\zeta,N}^{\star}$ (see (8)).

**Theorem 3.** *Suppose Assumption 1 holds, and that the matrix $(\boldsymbol{G}(X_i))_{i\in N} \in \mathbb{R}^{K\times NC}$ has full row-rank. Let $\boldsymbol{\lambda}_{\zeta,N}^{(t)}$ and $\boldsymbol{h}^{(t)}$ be the $t$-th iteration outputs of* `FairProjection` *for the KL-divergence case. Then, we have the exponential decay of errors $\|\boldsymbol{\lambda}_{\zeta,N}^{(t)} - \boldsymbol{\lambda}_{\zeta,N}^{\star}\|_2 = e^{-\Omega(t)}$ and $\boldsymbol{h}^{(t)}(x) = \boldsymbol{h}^{\text{opt},N}(x) \cdot \left(1 \pm e^{-\Omega(t)}\right)$ uniformly in $x \in \mathbb{X}$ as $t \to \infty$.*

*Proof.* See Appendix A.5. □

**Remark 2.** The full-rank assumption on the matrix $(\boldsymbol{G}(X_i))_{i\in N} \in \mathbb{R}^{K\times NC}$ can be ensured by adding negligible noise to it. Further, although Theorem 3 is shown for the KL-divergence, the proof directly extends to general $f$-divergences satisfying Assumption 1 (see Appendix A.6 for further discussions). Finally, we show in Theorem 6 in Appendix A.7 that carrying $t = \Omega(\log N)$ iterations of `FairProjection`, with regularizer $\zeta = \Theta(N^{-1/2})$, yields a parameter $\boldsymbol{\lambda}_{\zeta,N}^{(t)}$ that works well for the *population* problem for information projection (5); this makes `FairProjection` have a computational runtime of $O(N \log N)$.

**Benefit of parallelization.** The parallelizability of `FairProjection` provides significant speedup. In Appendix B.2, we provide an ablation study comparing the speedup due to parallelization. For the ENEM dataset (discussed next section), parallelization yields a 15-fold reduction in runtime. In addition to the parallel advantage of `FairProjection`, its inherent mathematical approach is more advantageous than gradient-based solutions. When numerically solving the dual problem (8) (or any close variant) via gradient methods, the gradient of $D_f^{\text{conj}}$ (the convex conjugate of an $f$-divergence) must be computed. However, this gradient is tractable in only a very limited number of relevant instances of $f$-divergences. `FairProjection` tackles this intractability through having its subroutines be informed by Lemma 1 and the discussion preceding it.

**Algorithm 1 :** `FairProjection` for solving (8).

**Input:** divergence $f$, predictions $\{\boldsymbol{p}_i \triangleq \boldsymbol{h}^{\text{base}}(X_i)\}_{i\in[N]}$, constraints $\{\boldsymbol{G}_i \triangleq \boldsymbol{G}(X_i)\}_{i\in[N]}$, regularizer $\zeta$, ADMM penalty $\rho$, and initializers $\boldsymbol{\lambda}$ and $(\boldsymbol{w}_i)_{i\in[N]}$.
**Output:** $h_c^{\text{opt},N}(x) \triangleq h_c^{\text{base}}(x) \cdot \phi(\gamma(x; \boldsymbol{\lambda}) + v_c(x; \boldsymbol{\lambda}))$.
$\boldsymbol{Q} \leftarrow \frac{\zeta}{2}\boldsymbol{I} + \frac{\rho}{2N}\sum_{i\in[N]} \boldsymbol{G}_i\boldsymbol{G}_i^T$
**for** $t = 1, 2, \cdots, t'$ **do**
  $\boldsymbol{a}_i \leftarrow \boldsymbol{w}_i + \rho\boldsymbol{G}_i^T\boldsymbol{\lambda}$ $\quad i \in [N]$
  $\boldsymbol{v}_i \leftarrow \operatorname{argmin}_{\boldsymbol{v}\in\mathbb{R}^C} D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}_i) + \frac{\rho+\zeta}{2}\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}_i^T\boldsymbol{v}$ $\quad i \in [N]$
  $\boldsymbol{q} \leftarrow \frac{1}{N}\sum_{i\in[N]} \boldsymbol{G}_i \cdot (\boldsymbol{w}_i + \boldsymbol{v}_i)$
  $\boldsymbol{\lambda} \leftarrow \operatorname{argmin}_{\boldsymbol{\ell}\in\mathbb{R}_+^K} \boldsymbol{\ell}^T\boldsymbol{Q}\boldsymbol{\ell} + \boldsymbol{q}^T\boldsymbol{\ell}$
  $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i + \rho \cdot \left(\boldsymbol{v}_i + \boldsymbol{G}_i^T\boldsymbol{\lambda}\right)$ $\quad i \in [N]$
**end for**

## 5 Numerical benchmarks

We present empirical results and show that `FairProjection` has competitive performance both in terms of runtime and fairness-accuracy trade-off curves compared to benchmarks—most notably the reductions approach in [ABD+18], which requires retraining. Extensive additional benchmarks and experiment details are reported in Appendix B.

**Setup.** We consider three base classifiers (`Base`): gradient boosting (GBM), logistic regression (LR), and random forest (RF), implemented by Scikit-learn [PVG+11]. For `FairProjection` (the constrained optimization in (6)), we use cross-entropy (`FairProjection-CE`) and KL-divergence (`FairProjection-KL`) as the loss function[9]. We consider two fairness constraints: mean equalized odds (MEO) and statistical parity (SP) (cf. Table 2). Particularly, to measure multi-class performance, we extend the definition of MEO as

$$\mathsf{MEO} = \max_{i \in \mathcal{Y}} \max_{s_1, s_2 \in \mathcal{S}} (|\mathsf{TPR}_i(s_1) - \mathsf{TPR}_i(s_2)| + |\mathsf{FPR}_i(s_1) - \mathsf{FPR}_i(s_2)|)/2 \tag{10}$$

where $\mathsf{TPR}_i(s) = P(\widehat{Y} = i | Y = i, S = s)$, and $\mathsf{FPR}_i(s) = P(\widehat{Y} = i | Y \neq i, S = s)$. The definition of multi-class statistical parity is provided in Appendix B.4.2. All values reported in this section are from the test set with 70/30 train-test split. When benchmarking against methods tailored to binary classification, we restrict our results to both binary $Y$ and $S$ since, unlike `FairProjection`, competing methods cannot necessarily handle multi-class predictions and multiple groups.

**Datasets.** We evaluate `FairProjection` and all benchmarks on four datasets. We use two datasets in the education domain: the high-school longitudinal study (HSLS) dataset [IPH+11, JWC22] and a novel dataset ENEM [INE20] (details in Appendix B.1). The ENEM dataset contains Brazilian college entrance exam scores along with student demographic information and socio-economic questionnaire answers (e.g., if they own a computer). After pre-processing, the dataset contains $\sim$1.4 million samples with 139 features. Race was used as the group attribute $S$, and Humanities exam score is used as the label $Y$. The score can be quantized into an arbitrary number of classes. For binary experiments, we quantize $Y$ into two classes, and for multi-class, we quantize it to 5 classes. The race feature $S$ has 5 categories, but we binarize it into White and Asian ($S = 1$) and others ($S = 0$). We call the entire ENEM dataset ENEM-1.4M. We also created smaller versions of the dataset with 50k samples: ENEM-50k-2C (binary classes) and ENEM-50k-5C (5 classes).[10] For completeness, we report results on UCI Adult [Lic13] and COMPAS [ALMK16].

**Benchmarks.** For binary classification experiments, we compare our method with five existing fair learning algorithms: `Reduction` [ABD+18], reject-option classifier [KKZ12, `Rejection`], equalized-odds [HPS16, `EqOdds`], calibrated equalized-odds [PRW+17, `CalEqOdds`], and leveraging equal opportunity [CDH+19, `LevEqOpp`].[11] The choice of benchmarks is based on the availability of reproducible codes. For the first four baselines, we use IBM AIF360 library [BDH+18]. For `Reduction` and `Rejection`, we vary the tolerance to achieve different operation points on the fairness-accuracy trade-off curves. As `EqOdds`, `CalEqOdds` and `LevEqOpp` only allow hard equality constraint on equalized odds, they each produce a single point on the plot (see Fig. 1). We include the group attribute as a feature in the training set following the same benchmark procedure described in [ABD+18, WRC21] for a consistent comparison. For multi-class classification experiments, we did not find methods that can be easily compared against `FairProjection` and use the multi-class extensions of mean equalized odds and statistical parity. For the sake of completeness, we modified the codes of adversarial debiasing [ZLM18, `Adversarial`], and compare our method against it. Note that `Reduction` [ABD+18] and `Adversarial` [ZLM18] are in-processing methods, and the rest of the benchmark algorithms are post-processing methods like `FairProjection`. Additional comparisons to [KCT20] are given in Appendix B.4.1.

There are four methods in Table 1 we did not include the experiments: FACT [KCT20], Identifying [JN20], FST [WRC21], and Overlapping [YCK20], as explained in Appendix B.1.3.

[9] We focus on `FairProjection-CE` and random forest here; results for `FairProjection-KL` and other models are in Appendix B.

[10] A datasheet (see [GMV+21]) for ENEM is given in Appendix C.

[11] https://github.com/lucaoneto/NIPS2019_Fairness.

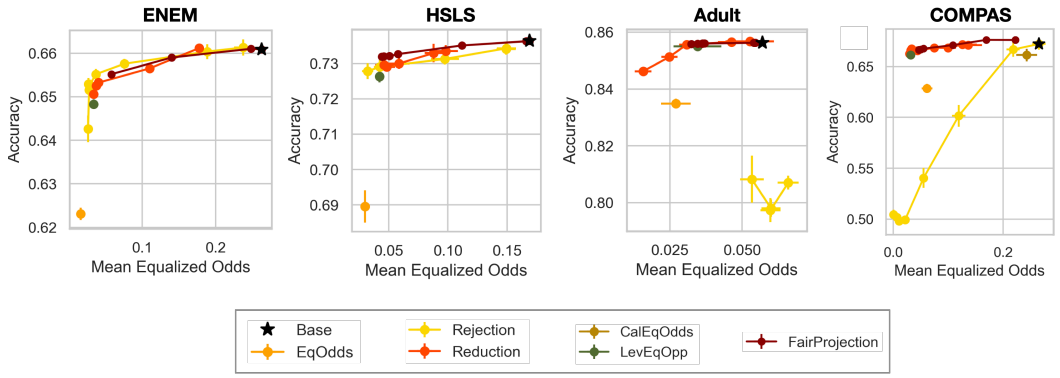



**Figure 1:** Fairness-accuracy trade-off comparisons between `FairProjection` and five baselines on ENEM-50k-2C, HSLS, Adult and COMPAS datasets. For all methods, we used random forest as a base classifier. Note that `EqOdds`, `CalEqOdds`, and `LevEqOpp` only produce a single accuracy-fairness trade-off point, whereas the rest of the methods are capable of producing the accuracy-fairness trade-off curves by varying the fairness budget $\alpha$ for the group fairness criteria listed in Table 2 — a smaller $\alpha$ corresponds to a lefter point on the accuracy-fairness trade-off curve.

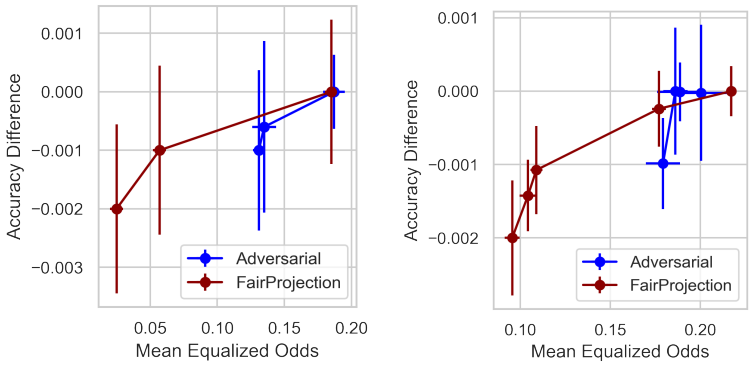
**Binary classification results.** We compare `FairProjection` with benchmarks tailored to binary classification in terms of the MEO-accuracy trade-off on the ENEM-50k-2C, HSLS, Adult, and COMPAS datasets in Fig. 1. Each point is obtained by averaging 10 runs with different train-test splits. `FairProjection-CE` curves were obtained by varying $\alpha$ values (cf. Table 2). When $\alpha = 1.0$, the outputs of `FairProjection-CE` are equivalent to the base classifier RF.

We observe that `FairProjection-CE` and `Reduction` have the overall best and most consistent performances. On ENEM-50k-2C and HSLS datasets, although `EqOdds` achieves the best fairness, that fairness comes at the cost of $4\%$ accuracy drop (additively). The other four methods, on the other hand, produce comparatively good fairness with an accuracy loss of $< 1\%$. In particular, `FairProjection-CE` has the smallest accuracy drop whilst improving MEO from 0.17 to 0.04 on HSLS. `CalEqOdds` requires strict calibration requirements and yields inconsistent performance when these requirements are not met. On ENEM-50k-2C and HSLS, `LevEqOpp` achieves comparable MEO with a slight accuracy drop, and on COMPAS, `LevEqOpp` performs equally well as `FairProjection-CE` and `Reduction`. Note that with high fairness constraints (i.e., small tolerance), the accuracy of `Rejection` deteriorates.

**Multi-Class results.** We illustrate how `FairProjection` performs on multi-class prediction using HSLS and ENEM-50k-5C. For HSLS, we divided student math performance into quartiles and generated four classes. In Figure 2, we plot fairness-accuracy trade-off of `FairProjection-CE` with logistic regression and adversarial debiasing [ZLM18, `Adversarial`]. As their base classifiers are different (`Adversarial` is a GAN-based method), we plot accuracy difference compared to the base classifier instead of plotting the absolute value of accuracy[12]. `FairProjection` reduces MEO significantly with very small loss in accuracy. While `Adversarial` is also able to reduce MEO with negligible accuracy drop, it does not reduce the MEO as much as `FairProjection`. We show more extensive results with multi-group and multi-class ($|\mathcal{Y}| = 5, = |\mathcal{S}| = 5$) in Appendix B.4.2.

**Runtime comparisons.** To demonstrate the scalability of `FairProjection`, in Table 3, we record the runtime of `FairProjection-CE` and `-KL` with the five benchmarks on ENEM-1.4M-2C, which is the biggest dataset we have. These experiments were run on a machine with AMD Ryzen 2990WX 64-thread 32-Core CPU and NVIDIA TITAN Xp 12-GB GPU. For consistency, we used the same fairness metric (MEO, $\alpha = 0.01$), base classifier (GBM), and train/test split, and each number is the average of 2 repeated experiments. `EqOdds`, `LevEqOpp`, and `CalEqOdds` are faster than `FairProjection` since they are optimized to produce one trade-off point (cf. Fig. 1). Compared to baselines that produce full fairness-accuracy trade-off curves (i.e., `Reduction` and `Rejection`), `FairProjection` has the fastest runtime. Also, the non-parallel implementation of `FairProjection-KL` takes 25.3 mins—parallelization attains $15\times$ speedup (detailed results in Appendix B.2). We further compare

[12]Base accuracy for `FairProjection` $= 0.336$, `Adversarial` $= 0.307$. Random guessing accuracy $= 0.2$.




**(a)** HSLS. **(b)** ENEM-50k-5C.

**Figure 2:** Fairness-accuracy trade-off for multi-class prediction on HSLS and ENEM-50k-5C. FairProjection is `FairProjection-CE` with LR base classifier.

| **Method** | ***Reduction*** [ABD+18] | **Rejection** [KKZ12] | EqOdds [HPS16] | LevEqOpp [CDH+19] | CalEqOdds [PRW+17] | ***FairProjection (ours)*** CE | KL |
|---|---|---|---|---|---|---|---|
| **Runtime** | 223.6 | 16.9 | 5.9 | 7.9 | 5.3 | 11.3 | 11.6 |

**Table 3:** Execution time of `FairProjection` on the ENEM-1.4M-2C compared with five baseline methods (time shown in minutes). Methods in **bold** are capable of producing a fairness-accuracy trade-off curve. Methods that are *italicized* have a uniformly superior performance. The time reported here for `FairProjection` includes the time to fit the base classifiers. If base classifiers are given, the runtime of e.g. `FairProjection-KL` is 1.63 mins. The runtimes are consistent with small standard deviations across repeated experiments.

the runtime results for the binary HSLS, which is the second biggest dataset, with the baselines that produce full fairness-accuracy trade-off curves. The runtimes for `Reduction`, `Rejection` and `FairProjection-CE` are 81.1 sec, 9.73 sec and 4.50 sec respectively—again, `FairProjection` has the fastest runtime. For a theoretical comparison between the runtime of `FairProjection` and `Reduction`, see Appendix B.3.

## 6 Final remarks and limitations

We introduce a theoretically-grounded and versatile fairness intervention method, `FairProjection`, and showcase its favorable performance in extensive experiments. We encourage the reader to peruse our theoretical result in Appendix A and extensive additional numerical benchmarks in Appendix B. `FairProjection` is able to correct bias for multigroup/multiclass datasets, and it enjoys a fast runtime thanks to its parallelizability. We also evaluate our method on the ENEM dataset (see Appendix C for a detailed description of the dataset). Our benchmarks are a step forward in moving away from the overused COMPAS and UCI Adult datasets.

We only consider group-fairness, and it would be interesting to try to incorporate other fairness notions (e.g., individual fairness [DHP+12]) into our formulation. We assume that $\boldsymbol{h}^{\text{base}}$ is a pre-trained accurate (and potentially unfair) classifier; one future research direction is understanding how the accuracy of $\boldsymbol{h}^{\text{base}}$ influences the performance of the projected classifier. Finally, the performance of `FairProjection` is inherently constrained by data availability. Performance may degrade with intersectional increases of the number of groups, the number of labels, and the number of fairness constraints.

## Acknowledgement

We thank the anonymous referees for their careful critique, which helped improve the quality of the paper considerably. This material is based upon work supported by the National Science Foundation under grants CAREER 1845852, IIS 1926925, FAI 2040880, CIF 1900750, an HDSI Bias^2 award and a gift from Oracle Research.

## References


[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AAV19] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1418–1426, 2019.

[AAW+20a] Wael Alghamdi, Shahab Asoodeh, Hao Wang, Flavio P. Calmon, Dennis Wei, and Karthikeyan Natesan Ramamurthy. Model projection: Theory and applications to fair machine learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2711–2716, 2020.

[AAW+20b] Wael Alghamdi, Shahab Asoodeh, Hao Wang, Flavio P. Calmon, Dennis Wei, and Karthikeyan Natesan Ramamurthy. Model projection: Theory and applications to fair machine learning. https://github.com/WaelAlghamdi/ModelProjection, 2020.

[ABD+18] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[ALMK16] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica*, 2016.

[BDH+18] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[BDNP19] Maria-Florina F Balcan, Travis Dick, Ritesh Noothigattu, and Ariel D Procaccia. Envy-free classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[BFG87] Jonathan M Borwein, SP Fitzpatrick, and JR Giles. The differentiability of real functions on normed linear space using generalized subgradients. *Journal of mathematical analysis and applications*, 128(2):512–534, 1987.

[BNBR19] Sina Baharlouei, Maher Nouiehed, Ahmad Beirami, and Meisam Razaviyayn. Rényi fair inference. *arXiv preprint arXiv:1906.12005*, 2019.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, jan 2011.

[BZZ+21] Michelle Bao, Angela Zhou, Samantha A Zottola, Brian Brubach, Sarah Desmarais, Aaron Seth Horowitz, Kristian Lum, and Suresh Venkatasubramanian. It's COMPASlicated: The messy relationship between RAI datasets and algorithmic fairness benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[CDH+19] Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Leveraging labeled and unlabeled data for consistent fair binary classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[CDPF+17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2017.

[CHKV19] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019.

[Cho17] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[CHS20] Jaewoong Cho, Gyeongjo Hwang, and Changho Suh. A fair classifier using kernel density estimation. *Advances in Neural Information Processing Systems*, 33:15088–15099, 2020.

[CJG+19] Andrew Cotter, Heinrich Jiang, Maya R Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.*, 20(172):1–59, 2019.

[CKV20] L Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *International Conference on Machine Learning*, pages 1349–1359. PMLR, 2020.

[Com] Creative Commons. Creative commons attribution-noderivs 3.0 unported license. https://creativecommons.org/licenses/by-nd/3.0/deed.en. 05/25/2022.

[Csi75] Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

[Csi95] Imre Csiszár. Generalized projections for non-negative functions. In *Proceedings of 1995 IEEE International Symposium on Information Theory*, page 6. IEEE, 1995.

[CST17] Liang Chen, Defeng Sun, and Kim-Chuan Toh. A note on the convergence of admm for linearly constrained convex optimization problems. *Comput. Optim. Appl.*, 66(2):327–343, mar 2017.

[DEHH21] Christophe Denis, Romuald Elie, Mohamed Hebiri, and François Hu. Fairness guarantee in multi-class classification. *arXiv preprint arXiv:2109.13642*, 2021.

[DHMS21] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6478–6490. Curran Associates, Inc., 2021.

[DHP+12] 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.

[DOBD+18] Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*, 2018.

[DV75] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975.

[DY16] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66:889–916, 2016.

[ET99a] Ivar Ekeland and Roger Témam. *Convex Analysis and Variational Problems*. Society for Industrial and Applied Mathematics, 1999.

[ET99b] Ivar Ekeland and Roger Témam. *Convex analysis and variational problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1999.

[FFM+15] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[FSV+19] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

[GMV+21] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[GP17] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[HR19] Bruce Hajek and Maxim Raginsky. Statistical learning theory. *Lecture Notes*, 387, 2019.

[INE20] INEP. Instituto nacional de estudos e pesquisas educaionais anísio teixeira, microdados do ENEM. https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem, 2020. Accessed: 2022-05-23.

[IPH+11] Steven J Ingels, Daniel J Pratt, Deborah R Herget, Laura J Burns, Jill A Dever, Randolph Ottem, James E Rogers, Ying Jin, and Steve Leinwand. High school longitudinal study of 2009 (hsls: 09): Base-year data file documentation. nces 2011-328. *National Center for Education Statistics*, 2011.

[JN20] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 702–712. PMLR, 2020.

[JSW22] Taeuk Jang, Pengyi Shi, and Xiaoqian Wang. Group-aware threshold adaptation for fair classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[JWC22] Haewon Jeong, Hao Wang, and Flavio Calmon. Fairness without imputation: A decision tree approach for fair prediction with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[KCT20] Joon Sik Kim, Jiahao Chen, and Ameet Talwalkar. FACT: A diagnostic for group fairness trade-offs. In *International Conference on Machine Learning*, pages 5264–5274. PMLR, 2020.

[KGZ19] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[KJW+21] Anilesh Krishnaswamy, Zhihao Jiang, Kangning Wang, Yu Cheng, and Kamesh Munagala. Fair for all: Best-effort fairness guarantees for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 3259–3267. PMLR, 2021.

[KKZ12] F. Kamiran, A. Karim, and X. Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929, Dec 2012.

[KS15] M Ashok Kumar and Rajesh Sundaresan. Minimization problems based on relative $\alpha$-entropy i: Forward projection. *IEEE Transactions on Information Theory*, 61(9):5063–5080, 2015.

[KS16] M Ashok Kumar and Igal Sason. Projection theorems for the Rényi divergence on $\alpha$-convex sets. *IEEE Transactions on Information Theory*, 62(9):4924–4935, 2016.

[Lic13] M. Lichman. UCI machine learning repository, 2013.

[LPB+21] Andrew Lowy, Rakesh Pavan, Sina Baharlouei, Meisam Razaviyayn, and Ahmad Beirami. Fermi: Fair empirical risk minimization via exponential Rényi mutual information. *arXiv preprint arXiv:2102.12586*, 2021.

[MW18] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118. PMLR, 2018.

[Nes04] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Springer, Boston, MA, 2004.

[PQC+19] Flavien Prost, Hai Qian, Qiuwen Chen, Ed H Chi, Jilin Chen, and Alex Beutel. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *arXiv preprint arXiv:1910.11779*, 2019.

[PRW+17] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *arXiv preprint arXiv:1709.02012*, 2017.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[Roc09] R. Tyrrell Rockafellar. *Variational analysis*. Grundlehren der mathematischen Wissenschaften ; 317. Springer, Berlin ; Heidelberg, 1st ed. 1998. edition, 2009.

[WRC20] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio P Calmon. Optimized score transformation for fair classification. In *23rd International Conference on Artificial Intelligence and Statistics*, 2020.

[WRC21] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio P Calmon. Optimized score transformation for consistent fair classification. *Journal of Machine Learning Research*, 22(258):1–78, 2021.

[YCK20] Forest Yang, Mouhamadou Cisse, and Oluwasanmi O Koyejo. Fairness with overlapping groups; a probabilistic perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[YX20] Qing Ye and Weijun Xie. Unbiased subdata selection for fair classification: A unified framework and scalable algorithms. *arXiv preprint arXiv:2012.12356*, 2020.

[ZLM18] 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.

[ZVRG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sections 3–4 for theoretical results, and Section 5 for experimental results.
   (b) Did you describe the limitations of your work? [Yes] See '**Final remarks and limitations**' in Section 6, and also the end of the '**Group-fairness constraints**' paragraph in Section 2.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See '**Final remarks and limitations**' in Section 6, e.g., lack of samples could negatively impact performance.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Assumption 1 and Remark 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See SM.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See, see SM.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See, see SM and main text.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See, all results employ cross validation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See, see numerical results section.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Yes, details (including licenses) are in the SM.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Yes, see SM.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Yes, only publicly and freely available code was used.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Yes. In particular, discussion regarding the ENEM dataset is available in the SM.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

This appendix is divided into three parts: Appendix A: Proofs of theoretical results; Appendix B: More details on the experimental setup, additional quantitative experiments, and more qualitative comparisons with related work; and Appendix C: A datasheet for ENEM (2020) dataset.

# A Proofs of theoretical results

The theoretical details of our work are included in this appendix. We prove the strong duality stated in Theorem 2 in Appendix A.1. Algorithm 1 is derived in Appendix A.2. The inner iterations of Algorithm 1 are further developed in Appendices A.3–A.4. The convergence rate result in Theorem 3 is proved in Appendix A.5, and an extension of it (to general $f$-divergences) is discussed in Appendix A.6. The performance of `FairProjection` for the population problem (5) is stated in Theorem 6 in A.7 and proved there too. Explicit formulas for the $\boldsymbol{G}$ matrix induced by the fairness metrics in Table 2 are given in Appendices A.8 and A.9.

## A.1 Proof of Theorem 2: strong duality

We use the following minimax theorem, which is a generalization of Sion's minimax theorem.

**Theorem 4** ([ET99b, Chapter VI, Prop. 2.2])**.** *Let $V$ and $Z$ be two reflexive Banach spaces, and fix two convex, closed, and non-empty subsets $\mathcal{A} \subset V$ and $\mathcal{B} \subset Z$. Let $L : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ be a function such that for each $u \in \mathcal{A}$ the function $p \mapsto L(u,p)$ is concave and upper semicontinuous, and for each $p \in \mathcal{B}$ the function $u \mapsto L(u,p)$ is convex and lower semicontinuous. Suppose that there exist points $u_0 \in \mathcal{A}$ and $p_0 \in \mathcal{B}$ such that $\lim_{p\in\mathcal{B},\|p\|\to\infty} L(u_0,p) = -\infty$ and $\lim_{u\in\mathcal{A},\|u\|\to\infty} L(u,p_0) = \infty$. Then, $L$ has at least one saddle-point $(\overline{u},\overline{p})$, and*

$$L(\overline{u},\overline{p}) = \min_{u\in\mathcal{A}} \sup_{p\in\mathcal{B}} L(u,p) = \max_{p\in\mathcal{B}} \inf_{u\in\mathcal{A}} L(u,p). \tag{11}$$

*In particular, in (11), there exists a minimizer in $\mathcal{A}$ of the outer minimization, and a maximizer in $\mathcal{B}$ of the outer maximization.*

Denote $\boldsymbol{h}_i \triangleq \boldsymbol{h}(X_i)$, $\boldsymbol{p}_i \triangleq \boldsymbol{h}^{\text{base}}(X_i)$, $\boldsymbol{a}_i \triangleq \boldsymbol{a}(X_i)$, and $\boldsymbol{G}_i \triangleq \boldsymbol{G}(X_i)$, and let the matrix $\boldsymbol{\mathcal{G}}_N \triangleq \left(\boldsymbol{G}_1/\sqrt{N}, \cdots, \boldsymbol{G}_N/\sqrt{N}, \boldsymbol{I}_K\right) \in \mathbb{R}^{K\times(NC+K)}$ be as in the theorem statement. We may rewrite the optimization (6) as

$$\begin{aligned} \underset{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]}{\text{minimize}} \quad & \frac{1}{N}\sum_{i\in[N]} D_f\left(\boldsymbol{h}_i\|\boldsymbol{p}_i\right) + \tau_1\cdot\left(\|\boldsymbol{a}_i\|_2^2 + \|\boldsymbol{b}\|_2^2\right) \\ \text{subject to} \quad & \frac{1}{N}\sum_{i\in[N]} \boldsymbol{G}_i\boldsymbol{h}_i + \tau_2\cdot\left(\boldsymbol{G}_i\boldsymbol{a}_i - \boldsymbol{b}\right) \le \boldsymbol{0}. \end{aligned} \tag{12}$$

We define $f$ at 0 by the right limit $f(0) \triangleq f(0+)$. Assume for now that $f(0+) < \infty$, and we will explain at the end of this proof how to treat the case $f(0+) = \infty$. For the optimization problem (12), the Lagrangian $L : \boldsymbol{\Delta}_C^N \times \mathbb{R}^{NC} \times \mathbb{R}^K \times \mathbb{R}_+^K \to \mathbb{R}$ is given by

$$\begin{aligned} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) \triangleq \frac{1}{N}\sum_{i\in[N]} & D_f\left(\boldsymbol{h}_i\|\boldsymbol{p}_i\right) + \tau_1\left(\|\boldsymbol{a}_i\|_2^2 + \|\boldsymbol{b}\|_2^2\right) \\ & + \boldsymbol{\lambda}^T\left(\boldsymbol{G}_i\boldsymbol{h}_i + \tau_2\left(\boldsymbol{G}_i\boldsymbol{a}_i - \boldsymbol{b}\right)\right). \end{aligned} \tag{13}$$

With $\boldsymbol{v}(x;\boldsymbol{\lambda}) \triangleq -\boldsymbol{G}(x)^T\boldsymbol{\lambda}$ as in the theorem statement, and denoting $\boldsymbol{v}_i \triangleq \boldsymbol{v}(X_i;\boldsymbol{\lambda}) = -\boldsymbol{G}_i^T\boldsymbol{\lambda}$, we may rewrite the Lagrangian as

$$\begin{aligned} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) = \frac{1}{N}\sum_{i\in[N]} & D_f\left(\boldsymbol{h}_i\|\boldsymbol{p}_i\right) - \boldsymbol{v}_i^T\boldsymbol{h}_i + \tau_1\|\boldsymbol{a}_i\|_2^2 - \tau_2\boldsymbol{v}_i^T\boldsymbol{a}_i \\ & + \tau_1\|\boldsymbol{b}\|_2^2 - \tau_2\boldsymbol{\lambda}^T\boldsymbol{b}. \end{aligned} \tag{14}$$

The optimization problem (12) can be written as

$$\inf_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} \sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right). \tag{15}$$

We check that the Lagrangian $L$ satisfies the conditions in Theorem 4. First, any Euclidean space $\mathbb{R}^M$ (for $M \in \mathbb{N}$) is a reflexive Banach space since it is finite-dimensional. In addition, the convex nonempty sets $\boldsymbol{\Delta}_C^N \times \mathbb{R}^{NC} \times \mathbb{R}^K$ and $\mathbb{R}_+^K$ are closed in their respective ambient Euclidean spaces. By continuity and convexity of $f$, and linearity of $L$ in $\boldsymbol{\lambda}$, we have that $L$ satisfies all the convexity, concavity, and semicontinuity conditions in Theorem 4. Further, fixing any $\boldsymbol{h}_i \in \boldsymbol{\Delta}_C$, $i \in [N]$, and letting $\boldsymbol{a}_i = \mathbf{0}$, $i \in [N]$, and $\boldsymbol{b} = \frac{1}{\tau_2}\left(\mathbf{1} + \frac{1}{N}\sum_{i\in[N]} \boldsymbol{G}_i \boldsymbol{h}_i\right)$, we would get that

$$L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) = -\boldsymbol{\lambda}^T\mathbf{1} + \frac{1}{N}\sum_{i\in[N]} D_f\left(\boldsymbol{h}_i \| \boldsymbol{p}_i\right) + \tau_1\|\boldsymbol{b}\|_2^2 \to -\infty \quad \text{as} \quad \|\boldsymbol{\lambda}\|_2 \to \infty. \tag{16}$$

In addition, choosing $\boldsymbol{\lambda} = \mathbf{0}$, we have the Lagrangian

$$L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) = \frac{1}{N}\sum_{i\in[N]} D_f\left(\boldsymbol{h}_i \| \boldsymbol{p}_i\right) + \tau_1\|\boldsymbol{a}_i\|_2^2 + \tau_1\|\boldsymbol{b}\|_2^2 \to \infty \tag{17}$$

as $\|\boldsymbol{b}\|_2 + \sum_{i\in[N]} \|\boldsymbol{h}_i\|_2 + \|\boldsymbol{a}_i\|_2 \to \infty$. Thus, we may apply the minimax result in Theorem 4 to obtain the existence of a saddle-point of $L$ and that

$$\begin{aligned}\min_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} \sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} & L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) \\ &= \max_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} \inf_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right).\end{aligned} \tag{18}$$

In particular, there exists a minimizer $(\boldsymbol{h}_i^{\mathrm{opt},N}, \boldsymbol{a}_i^{\mathrm{opt},N}, \boldsymbol{b}^{\mathrm{opt},N}) \in \boldsymbol{\Delta}_C \times \mathbb{R}^C \times \mathbb{R}^K, i \in [N]$, of the outer minimization in the left-hand side in (18), and a maximizer $\boldsymbol{\lambda}^\star \in \mathbb{R}_+^K$ of the outer maximization in the right-hand side of (18). By strict convexity of the objective function in (12) (and convexity of the feasibility set), we obtain that the minimizer $(\boldsymbol{h}_i^{\mathrm{opt},N}, \boldsymbol{a}_i^{\mathrm{opt},N}, \boldsymbol{b}^{\mathrm{opt},N}) \in \boldsymbol{\Delta}_C \times \mathbb{R}^C \times \mathbb{R}^K, i \in [N]$, is unique. We show next that the optimizer $\boldsymbol{\lambda}^\star$ is unique too, which we will denote by $\boldsymbol{\lambda}_{\zeta,N}^\star$ as in the theorem statement. We also show that, for each fixed $\boldsymbol{\lambda} \in \mathbb{R}_+^K$, there is a unique minimizer $(\boldsymbol{h}_i^{\boldsymbol{\lambda}}, \boldsymbol{a}_i^{\boldsymbol{\lambda}}, \boldsymbol{b}^{\boldsymbol{\lambda}}) \in \boldsymbol{\Delta}_C \times \mathbb{R}^C \times \mathbb{R}^K, i \in [N]$, of the *inner* minimization in the right-hand side of (18); by strict convexity of $f$, this would imply that $\boldsymbol{h}_i^{\mathrm{opt},N} = \boldsymbol{h}_i^{\boldsymbol{\lambda}_{\zeta,N}^\star}$.

Now, fix $\boldsymbol{\lambda} \in \mathbb{R}_+^K$, and consider the inner minimization in (18). We have that

$$\begin{aligned}&\inf_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) \\ &= \inf_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} \frac{1}{N}\sum_{i\in[N]} D_f\left(\boldsymbol{h}_i \| \boldsymbol{p}_i\right) - \boldsymbol{v}_i^T\boldsymbol{h}_i + \tau_1\|\boldsymbol{a}_i\|_2^2 - \tau_2\boldsymbol{v}_i^T\boldsymbol{a}_i + \tau_1\|\boldsymbol{b}\|_2^2 - \tau_2\boldsymbol{\lambda}^T\boldsymbol{b}\end{aligned} \tag{19}$$

$$= \frac{1}{N}\sum_{i\in[N]} \inf_{\boldsymbol{h}_i\in\boldsymbol{\Delta}_C} D_f(\boldsymbol{h}_i\|\boldsymbol{p}_i) - \boldsymbol{v}_i^T\boldsymbol{h}_i + \inf_{\boldsymbol{a}_i\in\mathbb{R}^C} \tau_1\|\boldsymbol{a}_i\|_2^2 - \tau_2\boldsymbol{v}_i^T\boldsymbol{a}_i + \inf_{\boldsymbol{b}\in\mathbb{R}^K} \tau_1\|\boldsymbol{b}\|_2^2 - \tau_2\boldsymbol{\lambda}^T\boldsymbol{b} \tag{20}$$

$$= \frac{1}{N}\sum_{i\in[N]} -D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) - \frac{1}{2}\zeta\|\boldsymbol{v}_i\|_2^2 - \frac{1}{2}\zeta\|\boldsymbol{\lambda}\|_2^2 \tag{21}$$

$$= -\frac{\zeta}{2}\left\|\boldsymbol{\mathcal{G}}_N^T\boldsymbol{\lambda}\right\|_2^2 - \frac{1}{N}\sum_{i\in[N]} D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) \tag{22}$$

where $\zeta \triangleq \tau_2^2/(2\tau_1)$. Here, the minimizers are $\boldsymbol{a}_i^{\boldsymbol{\lambda}} \triangleq \frac{\tau_2}{2\tau_1}\boldsymbol{v}_i$ and $\boldsymbol{b}_i^{\boldsymbol{\lambda}} \triangleq \frac{\tau_2}{2\tau_1}\boldsymbol{\lambda}$, and $\boldsymbol{h}_i^{\boldsymbol{\lambda}}$ is the unique probability vector in $\boldsymbol{\Delta}_C$ for which $D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) = D_f(\boldsymbol{h}_i^{\boldsymbol{\lambda}}\|\boldsymbol{p}_i) - \boldsymbol{v}_i^T\boldsymbol{h}_i^{\boldsymbol{\lambda}}$; the existence and uniqueness of $\boldsymbol{h}_i^{\boldsymbol{\lambda}}$ is guaranteed since $\boldsymbol{q} \mapsto D_f(\boldsymbol{q}\|\boldsymbol{p}_i) - \boldsymbol{v}_i^T\boldsymbol{q}$ is lower semicontinuous and strictly convex, and $\boldsymbol{\Delta}_C$ is compact. Rewriting it in the form (22), the function

$$\boldsymbol{\lambda} \mapsto \inf_{(\boldsymbol{h}_i,\boldsymbol{a}_i,\boldsymbol{b})\in\boldsymbol{\Delta}_C\times\mathbb{R}^C\times\mathbb{R}^K, i\in[N]} L\left((\boldsymbol{h}_i)_{i\in[N]}, (\boldsymbol{a}_i)_{i\in[N]}, \boldsymbol{b}, \boldsymbol{\lambda}\right) \tag{23}$$

can be seen to be strictly concave. Indeed, the function $\boldsymbol{\lambda} \mapsto \left\|\boldsymbol{\mathcal{G}}_N^T \boldsymbol{\lambda}\right\|_2^2$ is strictly convex. Also, each function $\boldsymbol{\lambda} \mapsto D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i)$ is convex as it is a pointwise supremum of linear functions: recalling that $\boldsymbol{v}_i = -\boldsymbol{G}_i^T \boldsymbol{\lambda}$, we have the formula

$$D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) = \sup_{\boldsymbol{q} \in \boldsymbol{\Delta}_C} -\boldsymbol{q}^T \boldsymbol{G}_i^T \boldsymbol{\lambda} - D_f(\boldsymbol{q} \,\|\, \boldsymbol{p}_i). \tag{24}$$

Hence, the outer maximizer $\boldsymbol{\lambda}^\star$ in (18) is indeed unique, which we denote by $\boldsymbol{\lambda}^\star_{\zeta,N}$. Note that $\boldsymbol{\lambda}^\star_{\zeta,N}$ is the unique solution to the *minimization* (8), i.e.,

$$\boldsymbol{\lambda}^\star_{\zeta,N} = \operatorname*{argmin}_{\boldsymbol{\lambda} \in \mathbb{R}_+^K} \frac{1}{N} \sum_{i \in [N]} D_f^{\mathrm{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) + \frac{\zeta}{2} \left\|\boldsymbol{\mathcal{G}}_N^T \boldsymbol{\lambda}\right\|_2^2, \tag{25}$$

as stated by the theorem.

Since $\boldsymbol{h}^{\mathrm{opt},N} = \boldsymbol{h}^{\boldsymbol{\lambda}^\star_{\zeta,N}}$, the following formula for $\boldsymbol{h}^{\boldsymbol{\lambda}}$ (for a general $\boldsymbol{\lambda} \in \mathbb{R}_+^K$) yields the desired functional form (7) for $\boldsymbol{h}^{\mathrm{opt},N}$ in terms of $\boldsymbol{\lambda}^\star_{\zeta,N}$.

**Lemma 2** ([AAW+20b, Lemma 4])**.** *Let $f : [0, \infty) \to \mathbb{R} \cup \{\infty\}$ be a strictly convex function that is continuously differentiable over $(0, \infty)$ and satisfying $f(0) = f(0+)$, $f(1) = 0$, and $f'(0+) = -\infty$. Let $\phi$ denote the inverse of $f'$. Fix $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$ and $\boldsymbol{v} \in \mathbb{R}^C$. Then, the unique minimizer of $\boldsymbol{q} \mapsto D_f(\boldsymbol{q} \| \boldsymbol{p}) - \boldsymbol{v}^T \boldsymbol{q}$ over $\boldsymbol{q} \in \boldsymbol{\Delta}_C$ is given by $q_c^\star = p_c \cdot \phi(\gamma + v_c)$, $c \in [C]$, where $\gamma \in \mathbb{R}$ is the unique number satisfying $\mathbb{E}_{c \sim \boldsymbol{p}}[\phi(\gamma + v_c)] = 1$.*

From Lemma 2, and using $\boldsymbol{v}(x; \boldsymbol{\lambda}^\star_{\zeta,N}) = -\boldsymbol{G}(x)^T \boldsymbol{\lambda}^\star_{\zeta,N}$ and $\phi = (f')^{-1}$, we get that there exists a uniquely defined function $\gamma : \mathbb{X} \times \mathbb{R}^K \to \mathbb{R}$ for which

$$\mathbb{E}_{c \sim \boldsymbol{h}^{\mathrm{base}}(x)} \left[\phi\left(\gamma(x; \boldsymbol{\lambda}^\star_{\zeta,N}) + v_c(x; \boldsymbol{\lambda}^\star_{\zeta,N})\right)\right] = 1 \tag{26}$$

for every $x \in \mathbb{X}$. For this $\gamma$, we know from Lemma 2 that

$$h_c^{\boldsymbol{\lambda}^\star_{\zeta,N}}(x) = h_c^{\mathrm{base}}(x) \cdot \phi\left(\gamma(x; \boldsymbol{\lambda}^\star_{\zeta,N}) + v_c(x; \boldsymbol{\lambda}^\star_{\zeta,N})\right) \tag{27}$$

for every $c \in [C]$ and $x \in \mathbb{X}$. Since $\boldsymbol{h}^{\mathrm{opt},N} = \boldsymbol{h}^{\boldsymbol{\lambda}^\star_{\zeta,N}}$, we obtain formula (7) for $\boldsymbol{h}^{\mathrm{opt},N}$ in terms of $\boldsymbol{\lambda}^\star_{\zeta,N}$, and the proof of Theorem 2 is complete in the case $f(0+) < \infty$.

Finally, we note how the case $f(0+) = \infty$ is treated, so assume $f(0) = f(0+) = \infty$. The only difference in this case is that the Lagrangian $L$ might attain the value $\infty$, whereas we need it to be $\mathbb{R}$-valued to apply the minimax result in Theorem 4. Nevertheless, the only way $L$ can be infinite is if some classifier $\boldsymbol{h}_i$ has an entry equal to 0, in which case the objective function in (6) (or (12)) will also be infinite, so such a classifier can be thrown out without affecting the optimization problem. More precisely, we still have strict convexity and lower semicontinuity of the objective function in (12). Thus, there is a unique minimizer $\boldsymbol{h}^{\mathrm{opt},N}$ of (12). For this optimizer, there must be an $\varepsilon_1 > 0$ such that $\boldsymbol{h}^{\mathrm{opt},N}(x) \geq \varepsilon_1 \mathbf{1}$ for *every* $x \in \mathbb{X}$. Thus, the optimization problem (12) remains unchanged if $\boldsymbol{\Delta}_C$ is restricted to classifiers bounded away from 0 by $\varepsilon_1$. Moreover, by the same reasoning, the optimization problem (24) for finding $D_f^{\mathrm{conj}}$ also remains unchanged if $\boldsymbol{\Delta}_C$ is replaced by the set of classifiers bounded away from 0 by some $\varepsilon_2 > 0$ that is *independent* of the $X_i$. Hence, choosing $\varepsilon = \min(\varepsilon_1, \varepsilon_2) > 0$, and replacing $\boldsymbol{\Delta}_C$ by $\widetilde{\boldsymbol{\Delta}}_C \triangleq \{\boldsymbol{q} \in \boldsymbol{\Delta}_C \,;\, \boldsymbol{q} \geq \varepsilon \mathbf{1}\}$ in the above proof, we attain the same results for the case $f(0+) = \infty$.

**Remark 3.** In addition to our fairness problem formulation (6) being different from that in [AAW+20b], we note that our proof techniques are distinct. Indeed, the proofs in [AAW+20b] develop several techniques since they are based only on Sion's minimax theorem, precisely because a generalized minimax result such as Theorem 4 is inapplicable in the setup of [AAW+20b]. The reason behind this inapplicability is that the ambient Banach space $\mathcal{C}(\mathcal{X}, \mathbb{R}^C)$ is *not* reflexive when $\mathcal{X}$ is infinite, e.g., when $\mathcal{X} = \mathbb{R}^d$ as is assumed in [AAW+20b], whereas it is reflexive in our case as we consider a finite set of samples $\mathbb{X} \subset \mathcal{X}$.

### A.2 Algorithm 1: derivation of the ADMM iterations

ADMM is applicable to problems taking the form

$$\begin{aligned} \underset{(\boldsymbol{V}, \boldsymbol{\lambda}) \in \mathbb{R}^V \times \mathbb{R}^K}{\text{minimize}} \quad & F(\boldsymbol{V}) + \psi(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{V} + \boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{m}, \end{aligned} \tag{28}$$

where $F : \mathbb{R}^V \to \mathbb{R} \cup \{\infty\}$ and $\psi : \mathbb{R}^K \to \mathbb{R} \cup \{\infty\}$ are closed proper convex functions, and $\boldsymbol{A} \in \mathbb{R}^{U \times V}$, $\boldsymbol{B} \in \mathbb{R}^{U \times K}$, and $\boldsymbol{m} \in \mathbb{R}^U$ are fixed.

We rewrite the convex problem (8) into the ADMM form (28) as follows. With the samples $X_1, \cdots, X_N \overset{\text{i.i.d.}}{\sim} P_X$ fixed, we denote the following fixed vectors and matrices: for each $i \in [N]$, set

$$\boldsymbol{p}_i \triangleq \boldsymbol{h}^{\text{base}}(X_i) \in \boldsymbol{\Delta}_C^+ = \{\boldsymbol{q} \in \boldsymbol{\Delta}_C \,;\, \boldsymbol{q} > \boldsymbol{0}\}, \tag{29}$$

$$\boldsymbol{G}_i \triangleq \boldsymbol{G}(X_i) \in \mathbb{R}^{K \times C}. \tag{30}$$

We introduce a variable $\boldsymbol{V} \triangleq (\boldsymbol{v}_i)_{i \in [N]} \in \mathbb{R}^{NC}$ (with components $\boldsymbol{v}_i \in \mathbb{R}^C$), and consider the objective functions

$$F(\boldsymbol{V}) \triangleq \frac{1}{N} \sum_{i \in [N]} D_f^{\text{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) + \frac{\zeta}{2} \|\boldsymbol{V}\|_2^2, \tag{31}$$

$$\psi(\boldsymbol{\lambda}) \triangleq \mathbb{I}_{\mathbb{R}_+^K}(\boldsymbol{\lambda}) + \frac{\zeta}{2} \|\boldsymbol{\lambda}\|_2^2. \tag{32}$$

Then, setting[13]

$$\boldsymbol{A} = \frac{1}{\sqrt{N}} \boldsymbol{I}_{NC}, \quad \boldsymbol{B} = \frac{1}{\sqrt{N}} (\boldsymbol{G}_i)_{i \in [N]}^T, \text{ and } \quad \boldsymbol{m} = \boldsymbol{0}_{NC}, \tag{33}$$

our finite-sample problem (8) takes the ADMM form (28).

In addition, this reparametrization allows us to parallelize the ADMM iterations, which we briefly review next. One starts with forming the augmented Lagrangian for problem (28), $L_\rho : \mathbb{R}^V \times \mathbb{R}^K \times \mathbb{R}^U \to \mathbb{R} \cup \{\infty\}$, where $\rho > 0$ is a fixed *penalty parameter* and $\boldsymbol{U} \in \mathbb{R}^U$ denotes a *dual variable*, by

$$L_\rho(\boldsymbol{V}, \boldsymbol{\lambda}, \boldsymbol{U}) \triangleq F(\boldsymbol{V}) + \psi(\boldsymbol{\lambda}) + \boldsymbol{U}^T (\boldsymbol{A}\boldsymbol{V} + \boldsymbol{B}\boldsymbol{\lambda} - \boldsymbol{m}) + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{V} + \boldsymbol{B}\boldsymbol{\lambda} - \boldsymbol{m}\|_2^2. \tag{34}$$

The ADMM iterations then repeatedly update the triplet after the $t$-th iteration $(\boldsymbol{V}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{U}^{(t)})$ into a triplet $(\boldsymbol{V}^{(t+1)}, \boldsymbol{\lambda}^{(t+1)}, \boldsymbol{U}^{(t+1)})$ that is given by

$$\boldsymbol{V}^{(t+1)} \in \operatorname*{argmin}_{\boldsymbol{V} \in \mathbb{R}^V} L_\rho(\boldsymbol{V}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{U}^{(t)}), \tag{35}$$

$$\boldsymbol{\lambda}^{(t+1)} \in \operatorname*{argmin}_{\boldsymbol{\lambda} \in \mathbb{R}^V} L_\rho(\boldsymbol{V}^{(t+1)}, \boldsymbol{\lambda}, \boldsymbol{U}^{(t)}), \tag{36}$$

$$\boldsymbol{U}^{(t+1)} = \boldsymbol{U}^{(t)} + \rho \cdot \left(\boldsymbol{A}\boldsymbol{V}^{(t+1)} + \boldsymbol{B}\boldsymbol{\lambda}^{(t+1)}\right). \tag{37}$$

We next instantiate the ADMM iterations to our problem, and we note that we will consider the scaled dual variable $\boldsymbol{W} = \sqrt{N}\boldsymbol{U}$.

In our case, the augmented Lagrangian splits into non-interacting components along the $\boldsymbol{v}_i$. This splitting allows parallelizability of the $\boldsymbol{V}$-update step, which is the most computationally intensive step. Consider a conforming decomposition $\boldsymbol{U} = (\boldsymbol{u}_i)_{i \in [N]}$ for $\boldsymbol{u}_i \in \mathbb{R}^C$, and let $\boldsymbol{W} = \sqrt{N}\boldsymbol{U}$. With some algebra, one can show that the ADMM iterations for the ADMM problem specified by (31)–(33) are expressible by[14]

$$\boldsymbol{v}_i^{(t+1)} = \operatorname*{argmin}_{\boldsymbol{v} \in \mathbb{R}^C} D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}_i) + \mathcal{R}_i^{(t)}(\boldsymbol{v}), \qquad i \in [N], \tag{38}$$

$$\boldsymbol{\lambda}^{(t+1)} = \operatorname*{argmin}_{\boldsymbol{\lambda} \in \mathbb{R}_+^K} \boldsymbol{\lambda}^T \boldsymbol{Q} \boldsymbol{\lambda} + \boldsymbol{q}^{(t)T} \boldsymbol{\lambda}, \tag{39}$$

$$\boldsymbol{w}_i^{(t+1)} = \boldsymbol{w}_i^{(t)} + \rho \cdot \left(\boldsymbol{v}_i^{(t+1)} + \boldsymbol{G}_i^T \boldsymbol{\lambda}^{(t+1)}\right), \qquad i \in [N], \tag{40}$$

[13] The prefactor $1/\sqrt{N}$ is unnecessary since $\boldsymbol{m} = \boldsymbol{0}$, but we introduce it to simplify the ensuing expressions.

[14] Note also that in these specific ADMM iterations, unlike in the general ADMM iterations, we write "$=$ argmin" as opposed to "$\in$ argmin" since strict convexity and coercivity guarantee that a unique minimizer exists (see [CST17] for a case where argmin is empty). Also, we write here $\boldsymbol{q}^{(t)T}$ instead of $\left(\boldsymbol{q}^{(t)}\right)^T$ for readability.

where $\mathcal{R}_i^{(t)} : \mathbb{R}^C \to \mathbb{R}$ is the quadratic form

$$\mathcal{R}_i^{(t)}(\boldsymbol{v}) \triangleq \frac{\rho+\zeta}{2}\|\boldsymbol{v}\|_2^2 + \left(\boldsymbol{w}_i^{(t)} + \rho \boldsymbol{G}_i^T \boldsymbol{\lambda}^{(t)}\right)^T \boldsymbol{v}, \tag{41}$$

and the fixed matrix $\boldsymbol{Q} \in \mathbb{R}^{K\times K}$ and vectors $\boldsymbol{q}^{(t)} \in \mathbb{R}^K$ are given by

$$\boldsymbol{Q} \triangleq \frac{\zeta}{2}\boldsymbol{I}_K + \frac{\rho}{2N}\sum_{i\in[N]} \boldsymbol{G}_i \boldsymbol{G}_i^T, \tag{42}$$

$$\boldsymbol{q}^{(t)} \triangleq \frac{1}{N}\sum_{i\in[N]} \boldsymbol{G}_i \cdot \left(\boldsymbol{w}_i^{(t)} + \boldsymbol{v}_i^{(t+1)}\right). \tag{43}$$

Note that both the first (38) and last (40) steps can be carried out for each sample $i \in [N]$ in parallel.

### A.3 The inner iterations: minimizing the convex conjugate of $f$-divergence

Only updating the primal-variable $\boldsymbol{v}_i$ in Algorithm 1, i.e., solving

$$\min_{\boldsymbol{v}\in\mathbb{R}^C} \; D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v} \tag{44}$$

for fixed $(\boldsymbol{p}, \xi, \boldsymbol{a}) \in \boldsymbol{\Delta}_C^+ \times (0,\infty) \times \mathbb{R}^C$, is a nonstandard task. We propose in this section two approaches to execute this step, which aim at re-expressing the required minimization as either a fixed-point or a root-finding problem. In more detail, if one has access to an explicit formula for the gradient of $D_f^{\text{conj}}$, then one can transform (44) into a fixed-point equation. This case applies for the KL-divergence, for which $\nabla D_{\mathsf{KL}}^{\text{conj}}$ is the softmax function (Appendix A.3.1). Furthermore, for the convergence of the fixed-point iterations, we derive an improved Lipschitz constant for the softmax function in Appendix A.4. On the other hand, if one does not have a tractable formula for $\nabla D_f^{\text{conj}}$, we propose the reduction provided in Lemma 1, whose proof is provided in Appendix A.3.2. We specialize the reduction provided by Lemma 1 to the cross-entropy case in Appendix A.3.3. Finally, we include in Appendix A.3.4 a general formula for $\nabla D_f^{\text{conj}}$ that can be used for the $\boldsymbol{v}_i$-update step for a general $f$-divergence, and we also utilize it in Appendices A.5–A.7 to prove the convergence rate of Algorithm 1 stated in Theorems 3–6.

#### A.3.1 Primal update for KL-divergence

Consider the case when the $f$-divergence of choice is the KL-divergence, i.e., $f(t) = t\log t$. Then, the convex conjugate $D_f^{\text{conj}}$ is given by the log-sum-exp function [DV75], namely, for $(\boldsymbol{p}, \boldsymbol{v}) \in \boldsymbol{\Delta}_C^+ \times \mathbb{R}^C$ we have

$$D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) = \log \sum_{c\in[C]} p_c e^{v_c}. \tag{45}$$

Thus, the first step in a given ADMM iteration, as in (38) (see also the beginning of the for-loop in Algorithm 1), amounts to solving

$$\min_{\boldsymbol{v}\in\mathbb{R}^C} \; \log \sum_{c\in[C]} p_c e^{v_c} + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v} \tag{46}$$

for $\xi \triangleq \frac{\rho+\zeta}{2} > 0$ and some fixed vectors $(\boldsymbol{p}, \boldsymbol{a}) \in \boldsymbol{\Delta}_C^+ \times \mathbb{R}^C$; see (29), (38) and (41) for explicit expressions. The problem (46) is strictly convex. Further, we may recast this problem, via introducing the variable $\boldsymbol{z} \in \mathbb{R}^C$ by $z_c \triangleq v_c + \log p_c$, as

$$\min_{\boldsymbol{z}\in\mathbb{R}^C} \; \log \sum_{c\in[C]} e^{z_c} + \xi\|\boldsymbol{z}\|_2^2 + \boldsymbol{b}^T\boldsymbol{z}, \tag{47}$$

where $b_c = a_c - 2\xi\log p_c$ is fixed. To solve this latter problem, it suffices to find a zero of the gradient, which is given by

$$\nabla_{\boldsymbol{z}}\left(\log \sum_{c\in[C]} e^{z_c} + \xi\|\boldsymbol{z}\|_2^2 + \boldsymbol{b}^T\boldsymbol{z}\right) = \sigma(\boldsymbol{z}) + 2\xi\boldsymbol{z} + \boldsymbol{b} \tag{48}$$

**Algorithm 2 :** $\underset{\boldsymbol{v}\in\mathbb{R}^C}{\operatorname{argmin}}\ D_{\mathsf{KL}}^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v}$

**Input:** $\xi > 0, \boldsymbol{p} \in \boldsymbol{\Delta}_C^+, \boldsymbol{a}, \boldsymbol{v} \in \mathbb{R}^C$.
$z_c \leftarrow v_c + \log p_c$ $\quad c \in [C]$
$b_c \leftarrow a_c - 2\xi \log p_c$ $\quad c \in [C]$
**repeat**
$\quad \boldsymbol{z} \leftarrow -\frac{1}{2\xi}\left(\sigma(\boldsymbol{z}) + \boldsymbol{b}\right)$
**until** convergence
**Output:** $v_c \triangleq z_c - \log p_c$ $\quad c \in [C]$

where $\sigma : \mathbb{R}^C \to \boldsymbol{\Delta}_C^+$ denotes the softmax function $\sigma(\boldsymbol{z}) \triangleq \left(\frac{e^{z_{c'}}}{\sum_{c\in[C]} e^{z_c}}\right)_{c'\in[C]}$. Thus, we arrive at the fixed-point problem $\theta(\boldsymbol{z}) = \boldsymbol{z}$ for the function

$$\theta(\boldsymbol{z}) \triangleq -\frac{1}{2\xi}\left(\sigma(\boldsymbol{z}) + \boldsymbol{b}\right). \tag{49}$$

We solve $\theta(\boldsymbol{z}) = \boldsymbol{z}$ using a fixed-point-iteration method, i.e., with some initial $\boldsymbol{z}_0$, we iteratively compute the compositions $\theta^{(m)}(\boldsymbol{z}_0)$ for $m \in \mathbb{N}$. This procedure is summarized in Algorithm 2.

The exponentially-fast convergence of Algorithm 2 is guaranteed in view of Lipschitzness of $\theta$ as defined in (49). Indeed, it is known that the softmax function is 1-Lipschitz (see, e.g., [GP17, Prop. 4]); we improve this Lipschitz constant to $1/2$ in Appendix A.4. This improvement yields a better guarantee on the convergence speed of `FairProjection`. Indeed, as a lower value of the ADMM penalty $\rho$ correlates with a faster convergence, lowering the Lipschitz constant of the softmax function allows us to speed up `FairProjection` by choosing $\rho > \frac{1}{2} - \zeta$ instead of $\rho > 1 - \zeta$.

### A.3.2 Proof of Lemma 1: primal update for general $f$-divergences

The lemma follows by the following sequence of steps:

$$\min_{\boldsymbol{v}\in\mathbb{R}^C} D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v} \overset{\text{(I)}}{=} \min_{\boldsymbol{v}\in\mathbb{R}^C}\max_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} \boldsymbol{q}^T\boldsymbol{v} - D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) + \boldsymbol{a}^T\boldsymbol{v} + \xi\|\boldsymbol{v}\|_2^2 \tag{50}$$

$$\overset{\text{(II)}}{=} \max_{\boldsymbol{q}\in\boldsymbol{\Delta}_C}\min_{\boldsymbol{v}\in\mathbb{R}^C} \boldsymbol{q}^T\boldsymbol{v} - D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) + \boldsymbol{a}^T\boldsymbol{v} + \xi\|\boldsymbol{v}\|_2^2 \tag{51}$$

$$\overset{\text{(III)}}{=} \max_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} -D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) - \frac{1}{4\xi}\|\boldsymbol{a}+\boldsymbol{q}\|_2^2 \tag{52}$$

$$= -\min_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) + \frac{1}{4\xi}\|\boldsymbol{a}+\boldsymbol{q}\|_2^2 \tag{53}$$

$$= -\min_{\boldsymbol{q}\in\mathbb{R}_+^C}\sup_{\theta\in\mathbb{R}} D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) + \frac{1}{4\xi}\|\boldsymbol{a}+\boldsymbol{q}\|_2^2 + \theta\cdot\left(\boldsymbol{1}^T\boldsymbol{q} - 1\right) \tag{54}$$

$$\overset{\text{(IV)}}{=} -\sup_{\theta\in\mathbb{R}}\min_{\boldsymbol{q}\in\mathbb{R}_+^C} D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) + \frac{1}{4\xi}\|\boldsymbol{a}+\boldsymbol{q}\|_2^2 + \theta\cdot\left(\boldsymbol{1}^T\boldsymbol{q} - 1\right) \tag{55}$$

$$\overset{\text{(V)}}{=} -\sup_{\theta\in\mathbb{R}} -\theta + \sum_{c\in[C]}\min_{q_c\geq 0} p_c f\left(\frac{q_c}{p_c}\right) + \frac{1}{4\xi}(a_c+q_c)^2 + \theta q_c, \tag{56}$$

where (I) holds by definition of $D_f^{\text{conj}}$ (see (3)), (II) by Sion's minimax theorem, (III) since the inner minimization occurs at $\boldsymbol{v} = -\frac{1}{2\xi}(\boldsymbol{q}+\boldsymbol{a})$, (IV) by generalized minimax theorems [see, e.g., Chapter VI, Proposition 2.2 in ET99a] (restated as Theorem 4 herein for convenience), and (V) by separability.

### A.3.3 Primal update for cross-entropy

In the cross-entropy (CE) case, i.e., $f(t) = -\log t$, instead of using an explicit formula for $D_f^{\text{conj}}$ (which would yield unwieldy expressions), we utilize the reduction shown in Lemma 1. Thus, we have the equality

$$\min_{\boldsymbol{v}\in\mathbb{R}^C} D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v} = -\sup_{\theta\in\mathbb{R}} -\theta + \sum_{c\in[C]} \min_{q_c\geq 0} p_c f\left(\frac{q_c}{p_c}\right) + \frac{1}{4\xi}(a_c+q_c)^2 + \theta q_c. \quad (57)$$

As per (57), we focus next on solving the inner single-variable minimization

$$\min_{q\geq 0} -p\log q + \frac{1}{4\xi}(a+q)^2 + \theta q. \quad (58)$$

It is easily seen that the solution to this minimization is the unique point making the objective's derivative vanish, i.e., it is $q^\star \in (0,\infty)$ for which

$$-\frac{p}{q^\star} + \frac{q^\star}{2\xi} + \theta + \frac{a}{2\xi} = 0. \quad (59)$$

This is easily solvable as a quadratic, yielding

$$q^\star = \sqrt{\left(\theta\xi + \frac{a}{2}\right)^2 + 2p\xi} - \left(\theta\xi + \frac{a}{2}\right). \quad (60)$$

Therefore, solving (57) amounts to finding the constant $\theta \in \mathbb{R}$ that yields a probability vector $\boldsymbol{q} \in \boldsymbol{\Delta}_C$, where

$$q_c \triangleq \sqrt{\left(\theta\xi + \frac{a_c}{2}\right)^2 + 2p_c\xi} - \left(\theta\xi + \frac{a_c}{2}\right). \quad (61)$$

Consider the function

$$g(z) \triangleq -1 + \sum_{c\in[C]} \sqrt{\left(z + \frac{a_c}{2}\right)^2 + 2p_c\xi} - \left(z + \frac{a_c}{2}\right), \quad (62)$$

so we simply are looking for a root of $g$ (then set $\theta = z/\xi$ and $\boldsymbol{v} = -\frac{1}{2\xi}(\boldsymbol{q}+\boldsymbol{a})$). This can be efficiently accomplished via Newton's method. Namely, we compute

$$g'(z) = -C + \sum_{c\in[C]} \frac{2z + a_c}{\sqrt{\left(z + \frac{a_c}{2}\right)^2 + 2p_c\xi}}, \quad (63)$$

then, starting from $z^{(0)}$, we form the sequence

$$z^{(t+1)} \triangleq z^{(t)} - \frac{g\left(z^{(t)}\right)}{g'\left(z^{(t)}\right)}. \quad (64)$$

This procedure is summarized in Algorithm 3.

### A.3.4 On the gradient of the convex conjugate of $f$-divergence

The following general result on the differentiability of $D_f^{\text{conj}}$ can be used to carry out the $\boldsymbol{v}_i$-update step for a general $f$-divergence, and it will also be useful in Appendices A.5–A.7 for proving the convergence rate of Algorithm 1 as stated in Theorems 3–6.

**Lemma 3.** *Suppose $f : (0,\infty) \to \mathbb{R}$ is strictly convex. For any fixed $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$, the function $\boldsymbol{v} \mapsto D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p})$ is differentiable, and its gradient is given by*

$$\nabla_{\boldsymbol{v}} D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) = \boldsymbol{q}_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) \in \boldsymbol{\Delta}_C, \quad (65)$$

*where*

$$\boldsymbol{q}_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) \triangleq \operatorname*{argmin}_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) - \boldsymbol{v}^T\boldsymbol{q}. \quad (66)$$

**Algorithm 3 :** $\underset{\boldsymbol{v}\in\mathbb{R}^C}{\operatorname{argmin}}\ D_{\mathsf{CE}}^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) + \xi\|\boldsymbol{v}\|_2^2 + \boldsymbol{a}^T\boldsymbol{v}$

**Input:** $\xi > 0$, $z \in \mathbb{R}$, $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$, $\boldsymbol{a} \in \mathbb{R}^C$.

**repeat**

$$g(z) \leftarrow -1 + \sum_{c\in[C]} \sqrt{\left(z + \frac{a_c}{2}\right)^2 + 2p_c\xi} - \left(z + \frac{a_c}{2}\right)$$

$$g'(z) \leftarrow -C + \sum_{c\in[C]} \frac{2z + a_c}{\sqrt{\left(z + \frac{a_c}{2}\right)^2 + 2p_c\xi}}$$

$$z \leftarrow z - \frac{g(z)}{g'(z)}$$

**until** convergence

**Output:** $v_c \triangleq \frac{1}{2\xi}\left(z - \frac{a_c}{2} - \sqrt{\left(z + \frac{a_c}{2}\right)^2 + 2p_c\xi}\right)$

*Proof.* From [Roc09, Proposition 11.3], since $\boldsymbol{q} \mapsto D_f(\boldsymbol{q}\,\|\,\boldsymbol{p})$ is a lower semicontinuous proper convex function, the subgradient of its convex conjugate $\boldsymbol{v} \mapsto D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p})$ is given by

$$\partial_{\boldsymbol{v}} D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p}) = \underset{\boldsymbol{q}\in\boldsymbol{\Delta}_C}{\operatorname{argmin}}\ D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) - \boldsymbol{v}^T\boldsymbol{q}. \tag{67}$$

Recall also that a function is differentiable at a point if and only if its subgradient there consists of a singleton [BFG87]. Thus, it only remains to show that the right-hand side in (67) is a singleton. For this, we note that $\boldsymbol{q} \mapsto D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) - \boldsymbol{v}^T\boldsymbol{q}$ is lower semicontinuous and strictly convex, and $\boldsymbol{\Delta}_C$ is compact. □

### A.4 ½-Lipschitzness of the Softmax Function

As stated in Section 4 and Appendix A.3.1, the convergence speed of the inner iteration (the $\boldsymbol{v}_i$ update step) of `FairProjection` can be guaranteed to be faster if the Lipschitz constant of the softmax function is lowered from 1 (which is proved in [GP17, Prop. 4]). By Lipschitzness here, we mean $\ell_2$-norm Lipschitzness. We prove the following proposition in this appendix.

**Proposition 1.** *For any $n \in \mathbb{N}$, the softmax function $\sigma(\boldsymbol{z}) \triangleq \left(\frac{e^{z_j}}{\sum_{i=1}^n e^{z_i}}\right)_{j\in[n]}$ is $\frac{1}{2}$-Lipschitz.*

We will need the following result.

**Lemma 4** (Theorem 2.1.6 in [Nes04])**.** *A twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is convex and has an $L$-Lipschitz continuous gradient if and only if its Hessian is positive semidefinite with maximal eigenvalue at most $L$.*

Since the softmax function is the gradient of the log-sum-exp function, and since the spectral norm is upper bounded by the Frobenius norm, it suffices to upper bound the Frobenius norm of the Jacobian of $\sigma$ by $1/2$. Suppose that $\sigma$ is operating on $n$ symbols. Consider the sum of powers functions $s_k(\boldsymbol{x}) \triangleq \sum_{i\in[n]} x_i^k$ for $\boldsymbol{x} \in \mathbb{R}^n$. For any $\boldsymbol{v} \in \mathbb{R}^n$, denoting $\boldsymbol{x} = \sigma(\boldsymbol{v})$, the square of the Frobenius norm of the Jacobian of $\sigma$ at $\boldsymbol{v}$ is given by

$$w(\boldsymbol{x}) \triangleq s_2(\boldsymbol{x})^2 + s_2(\boldsymbol{x}) - 2s_3(\boldsymbol{x}). \tag{68}$$

We show that $w(\boldsymbol{x}) \le \frac{1}{4}$ for any $n \in \mathbb{N}$ and $\boldsymbol{x} \in \boldsymbol{\Delta}_n$.

The approach we take is via reduction to the case $n \le 3$, which one can directly verify. Namely, assuming, without loss of generality, that $x_1 \le x_2 \le \cdots \le x_n$, we show that if $x_1 + x_2 \le 1/2$ then $w(\boldsymbol{y}) \ge w(\boldsymbol{x})$ where $\boldsymbol{y} \in \boldsymbol{\Delta}_{n-1}$ is given by $\boldsymbol{y} = (x_1 + x_2, x_3, \cdots, x_n)$. Note that if $n \ge 4$ then we must have $x_1 + x_2 \le 1/2$, because $x_1 + x_2 \le x_3 + x_4$ and $x_1 + x_2 + x_3 + x_4 \le 1$. Thus, we will have reduced the problem from an $n \ge 4$ to $n - 1$, which iteratively reduces the problem to $n \le 3$. Fix $n \ge 4$.

Denote $\boldsymbol{z} = (x_3, \cdots, x_n)$. A direct computation yields that

$$w(\boldsymbol{y}) - w(\boldsymbol{x}) = 2x_1x_2 \cdot (2s_2(\boldsymbol{z}) + g(x_1, x_2)) \tag{69}$$

with the quadratic

$$g(a, b) \triangleq 2a^2 + 2b^2 + 2ab - 3a - 3b + 1. \tag{70}$$

By assumption, $x_i \geq \max(x_1, x_2)$ for each $i \geq 3$, so $2s_2(\boldsymbol{z}) \geq (n-2)x_1^2 + (n-2)x_2^2 \geq x_1^2 + x_2^2$. Then,

$$w(\boldsymbol{y}) - w(\boldsymbol{x}) \geq 2x_1x_2 \cdot h(x_1, x_2) \tag{71}$$

with

$$h(a, b) \triangleq 3a^2 + 3b^2 + 2ab - 3a - 3b + 1. \tag{72}$$

Now, we show that $h$ is nonnegative for every $a, b \geq 0$ with $a + b \leq 1/2$. With $c = a + b$, we may write

$$h(a, b) = 3c^2 - (3 + 4a)c + 4a^2 + 1. \tag{73}$$

This quadratic in $c$ has its vertex at $c_{\min} = (3 + 4a)/6$. As $a \geq 0$, $c_{\min} \geq 1/2$. As $a + b \leq 1/2$, we see that the minimum of $h$ is attained for $c = 1/2$. Substituting $b = 1/2 - a$, we obtain

$$h(a, b) = \left(2a - \frac{1}{2}\right)^2, \tag{74}$$

which is nonnegative, as desired.

### A.5 Convergence rate of Algorithm 1: proof of Theorem 3

We recall a general result on the R-linear convergence rate for ADMM, which corresponds to case 1 in scenario 1 in [DY16]; see Tables 1 and 2 therein. Recall that a sequence $\{z^{(t)}\}_{t\in\mathbb{N}}$ is said to converge R-linearly to $z^\star$ if there is a constant $\eta \in (0, 1)$ and a sequence $\{\beta^{(t)}\}_{t\in\mathbb{N}}$ such that $\|z^{(t)} - z^\star\| \leq \beta^{(t)}$ and $\sup_t \left(\beta^{(t+1)}/\beta^{(t)}\right) \leq \eta$. In particular, one has exponentially small errors:

$$\|z^{(t)} - z^\star\| \leq \beta^{(0)} \cdot \eta^t. \tag{75}$$

The following theorem is used in our proof of Theorem 3.

**Theorem 5** ([DY16])**.** *Suppose that problem (28) has a saddle point, $F$ is strongly convex and differentiable with Lipschitz-continuous gradient, $\boldsymbol{A}$ has full row-rank, and $\boldsymbol{B}$ has full column-rank. Then, the ADMM iterations (35)–(37) converge R-linearly to a global optimizer.*

In Appendix A.2, we show that the dual (8) of our fairness optimization problem (6) can be written in the ADMM general form (28) with the choices

$$F(\boldsymbol{V}) = \frac{1}{N} \sum_{i\in[N]} D_f^{\text{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i) + \frac{\zeta}{2} \|\boldsymbol{V}\|_2^2 \tag{76}$$

and

$$\boldsymbol{A} = \frac{1}{\sqrt{N}} \boldsymbol{I}_{NC}, \quad \boldsymbol{B} = \frac{1}{\sqrt{N}} (\boldsymbol{G}_i)_{i\in[N]}^T. \tag{77}$$

Recall from Theorem 2 (see also the proof in Appendix A.1) that our problem (8) has a saddle point. Further, the function $F : \mathbb{R}^{NC} \to \mathbb{R}$ is $\zeta$-strongly convex and differentiable. Indeed, each $\boldsymbol{v} \mapsto D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}_i)$ is convex, and the term $\frac{\zeta}{2}\|\boldsymbol{V}\|_2^2$ is $\zeta$-strongly convex, so $F$ is $\zeta$-strongly convex too. In addition, by the formula for $\nabla D_f^{\text{conj}}$ in Lemma 3, the gradient of $F$ is

$$\nabla F(\boldsymbol{V}) = \frac{1}{N} \boldsymbol{q}_f^{\text{conj}}(\boldsymbol{V}) + \zeta \boldsymbol{V}, \tag{78}$$

where

$$\boldsymbol{q}_f^{\text{conj}}(\boldsymbol{V}) \triangleq \left(\boldsymbol{q}_f^{\text{conj}}(\boldsymbol{v}_i, \boldsymbol{p}_i)\right)_{i\in[N]}, \tag{79}$$

with $\boldsymbol{q}_f^{\text{conj}}(\boldsymbol{v}_i)$ as defined in (66).

In the KL-divergence case, i.e., $f(t) = t \log t$, the gradient of $D_f^{\text{conj}}$ is given by the softmax function (see Appendix A.3.1)

$$\boldsymbol{q}_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) = \sigma\left(\boldsymbol{v} + \log \boldsymbol{p}\right) = \left(\frac{p_c e^{v_c}}{\sum_{c' \in [C]} p_{c'} e^{v_{c'}}}\right)_{c \in [C]}. \tag{80}$$

Therefore, we have that

$$\nabla F(\boldsymbol{V}) = \frac{1}{N}\left(\sigma\left(\boldsymbol{v}_i + \log \boldsymbol{p}_i\right)\right)_{i \in [N]} + \zeta \boldsymbol{V}. \tag{81}$$

By Proposition 1, the softmax function $\sigma$ is $\frac{1}{2}$-Lipschitz. Hence, $\nabla F$ is $\left(\frac{1}{2N} + \zeta\right)$-Lipschitz.

Therefore, the general ADMM convergence rate in Theorem 5 yields that there is a constant $r > 0$ such that

$$\left\|\boldsymbol{\lambda}_{\zeta,N}^{(t)} - \boldsymbol{\lambda}_{\zeta,N}^{\star}\right\|_2 \leq \beta \cdot e^{-rt} \tag{82}$$

where $\beta \triangleq \left\|\boldsymbol{\lambda}_{\zeta,N}^{(0)} - \boldsymbol{\lambda}_{\zeta,N}^{\star}\right\|_2$. (Although Theorem 5 guarantees exponentially-fast convergence of $\boldsymbol{\lambda}_{\zeta,N}^{(t)}$ to *a* global optimizer, recall that $\boldsymbol{\lambda}_{\zeta,N}^{\star}$ is the *unique* optimizer of (8), as Theorem 2 shows.)

Finally, it remains to bound the distance between $\boldsymbol{h}^{\text{opt},N}$ and the output classifier $\boldsymbol{h}^{(t)}$ after the $t$-th iteration of Algorithm 1. Note that $\phi(u) = (f')^{-1}(u) = e^{u-1}$, so $\gamma$ may be obtained explicitly, and equation (7) becomes

$$h_{c'}^{\text{opt},N}(x) = \frac{h_{c'}^{\text{base}}(x) \cdot e^{v_{c'}(x;\boldsymbol{\lambda}_{\zeta,N}^{\star})}}{\sum_{c \in [C]} h_c^{\text{base}}(x) \cdot e^{v_c(x;\boldsymbol{\lambda}_{\zeta,N}^{\star})}}. \tag{83}$$

Thus, using $\boldsymbol{\lambda}^{(t)} \triangleq \boldsymbol{\lambda}_{\zeta,N}^{(t)}$ in place of $\boldsymbol{\lambda}_{\zeta,N}^{\star}$, we obtain that the $t$-th classifier obtained by Algorithm 1 is

$$h_{c'}^{(t)}(x) = \frac{h_{c'}^{\text{base}}(x) \cdot e^{v_{c'}(x;\boldsymbol{\lambda}^{(t)})}}{\sum_{c \in [C]} h_c^{\text{base}}(x) \cdot e^{v_c(x;\boldsymbol{\lambda}^{(t)})}}. \tag{84}$$

Therefore, we have the ratios

$$\frac{h_{c'}^{(t)}(x)}{h_{c'}^{\text{opt},N}(x)} = \frac{\sum_{c \in [C]} h_c^{\text{base}}(x) e^{v_c(x;\boldsymbol{\lambda}_{\zeta,N}^{\star})}}{\sum_{c \in [C]} h_c^{\text{base}}(x) e^{v_c(x;\boldsymbol{\lambda}^{(t)})}} \cdot \exp\left(v_{c'}(x;\boldsymbol{\lambda}^{(t)}) - v_{c'}(x;\boldsymbol{\lambda}_{\zeta,N}^{\star})\right). \tag{85}$$

By definition of $\boldsymbol{v}$, $\boldsymbol{v}(x;\boldsymbol{\lambda}) = -\boldsymbol{G}(x)^T\boldsymbol{\lambda}$. Thus, we obtain from (82) and boundedness of $\boldsymbol{G}$ that

$$\left\|\boldsymbol{v}(x;\boldsymbol{\lambda}^{(t)}) - \boldsymbol{v}(x;\boldsymbol{\lambda}_{\zeta,N}^{\star})\right\|_\infty = e^{-\Omega(t)}, \tag{86}$$

where the implicit constant is independent of $x$. Applying (86) in (85), and noting that $e^{\pm e^{-\Omega(t)}} = 1 \pm e^{-\Omega(t)}$ as $t \to \infty$, we conclude that

$$\left|\frac{h_{c'}^{(t)}(x)}{h_{c'}^{\text{opt},N}(x)} - 1\right| = e^{-\Omega(t)}, \quad c' \in [C], \tag{87}$$

uniformly in $x$. We may rewrite (87) as

$$\boldsymbol{h}^{(t)}(x) = \boldsymbol{h}^{\text{opt},N}(x) \cdot \left(1 \pm e^{-\Omega(t)}\right), \tag{88}$$

which is the desired convergence rate in the theorem statement, and the proof is complete.

### A.6 Extension of Theorem 3

Though Theorem 3 is shown for the KL-divergence, the proof directly extends to general $f$-divergences satisfying Assumption 1. In fact, Lipschitz continuity of the gradient of $D_{\mathsf{KL}}^{\text{conj}}$ is the only specific property that we apply to derive the KL-divergence case. For a general $f$-divergence, Lipschitz continuity of $\nabla D_f^{\text{conj}}$ may be derived as follows. Combining Lemmas 2–3 reveals the formula $\nabla_{\boldsymbol{v}} D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) = \left(p_c \cdot \phi\left(\gamma(\boldsymbol{v}) + v_c\right)\right)_{c \in [C]}$, where $\phi = (f')^{-1}$ and $\gamma(\boldsymbol{v})$ is uniquely defined

by $\mathbb{E}_{c\sim\boldsymbol{p}}\left[\phi(\gamma(\boldsymbol{v})+v_c)\right]=1$, with $\boldsymbol{p}\in\boldsymbol{\Delta}_C^+$ fixed. Since $\phi'=1/(f''\circ\phi)$, we have that $\phi$ is locally Lipschitz. From the proof of Theorem 5 in [AAW+20b], we know that $\boldsymbol{v}\mapsto\gamma(\boldsymbol{v})$ is locally Lipschitz. Thus, $\boldsymbol{v}\mapsto\nabla_{\boldsymbol{v}}D_f^{\text{conj}}(\boldsymbol{v},\boldsymbol{p})$ is locally Lipschitz. Further, $\boldsymbol{\lambda}\mapsto\nabla_{\boldsymbol{v}}D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}),\boldsymbol{p})$ is then also locally Lipschitz. Note that we may restrict $\boldsymbol{\lambda}$ *a priori* to be within some finite ball (see Lemma 5). Thus, if, e.g., $X$ is compactly-supported, we would obtain the desired Lipschitzness properties of the gradient of $D_f^{\text{conj}}$, and the proof of Theorem 3 carries through for $D_f$ in place of $D_{\mathsf{KL}}$.

### A.7 Convergence rate to the population problem

The following result shows, roughly, that the parameter $\boldsymbol{\lambda}_{N^{-1/2},N}^{(\log N)}$ obtainable from `FairProjection` performs well for the *population* problem for information projection (5).

**Theorem 6.** *Suppose Assumption 1 holds, let $\mathcal{X}=\mathbb{R}^d$, and consider the KL-divergence case. Then, choosing $\zeta=\Theta\left(N^{-1/2}\right)$ and $t=\Omega(\log N)$ we obtain for any $\delta\in(0,1)$ that (see (5))*

$$\Pr\left\{\mathbb{E}_X\left[D_{\mathsf{KL}}^{\text{conj}}\left(\boldsymbol{v}\left(X;\boldsymbol{\lambda}_{\zeta,N}^{(t)}\right),\boldsymbol{h}^{\text{base}}(X)\right)\right]>D^\star+O\left(\frac{1}{\sqrt{N}}\right)\right\}\leq\delta. \tag{89}$$

The proof is divided in this appendix into several lemmas. We note first that, in the course of the proof of Theorems 1 and 2 in [AAW+20a], it was shown that at least one minimizer $\boldsymbol{\lambda}^\star$ of (5) exists. Further, any such minimizer satisfies the following bound. Denote the constraint function by $\boldsymbol{\mu}(\boldsymbol{h})\triangleq\mathbb{E}_{P_X}[\boldsymbol{G}\boldsymbol{h}]$. Throughout this proof, we set $\mathcal{X}\triangleq\mathbb{R}^d$.

**Lemma 5.** *Suppose Assumption 1 holds, and fix a strictly feasible classifier $\boldsymbol{h}\in\mathcal{H}$, i.e., $\boldsymbol{\mu}(\boldsymbol{h})<\boldsymbol{0}$. Every minimizer $\boldsymbol{\lambda}^\star\in\mathbb{R}_+^K$ of (5) must satisfy the inequality*

$$\|\boldsymbol{\lambda}^\star\|_1\leq\lambda_{\max}\triangleq\frac{D_f\left(\boldsymbol{h}\,\|\,\boldsymbol{h}^{\text{base}}\mid P_X\right)}{\min\limits_{k\in[K]}-\mu_k(\boldsymbol{h})}. \tag{90}$$

We note that for the fairness metrics specified in Table 2, one valid choice of a strictly feasible $\boldsymbol{h}$ (i.e., one for which $\boldsymbol{\mu}(\boldsymbol{h})<\boldsymbol{0}$) is the uniform classifier $\boldsymbol{h}(x)\equiv\frac{1}{C}\boldsymbol{1}$. In any case, we have that $\lambda_{\max}<\infty$ since both $\boldsymbol{h}$ and $\boldsymbol{h}^{\text{base}}$ are assumed to belong to $\mathcal{H}$ and $f$ is continuous over $(0,\infty)$; e.g., one bound on $\lambda_{\max}$ is $\lambda_{\max}\leq\max_{m\leq t\leq M}f(t)/\min_{k\in[K]}-\mu_k(\boldsymbol{h})$ where $m=\inf_{c,x}h_c(x)$ and $M=1/\inf_{x,c}h_c^{\text{base}}(x)$. We will also need the following constants for the convergence analysis:

$$g_{\text{mean}}\triangleq\mathbb{E}\left[\|\boldsymbol{G}(X)\|_2^2\right], \tag{91}$$

$$g_{\max}\triangleq\sup_{x\in\mathcal{X}}\|\boldsymbol{G}(x)\|_2^2. \tag{92}$$

Clearly, $g_{\text{mean}}\leq g_{\max}$. By the boundedness of $\boldsymbol{G}$ in the second item in Assumption 1, $g_{\max}$ is finite.

**Remark 4.** Although the results in this paper are stated to hold under Assumption 1, we note that those conditions do not essentially impose any restriction on carrying our `FairProjection` algorithm. Indeed, we focus in this paper on the CE and KL cases, for which $f$ satisfies the imposed conditions. We also note that only boundedness of $\boldsymbol{G}$ is required for Theorem 2, which is true for the fairness metrics in Table 2 in non-degenerate cases (e.g., no empty groups). The condition on $\boldsymbol{h}^{\text{base}}$ being bounded away from zero can be made to hold by perturbing it if necessary with negligible noise. The condition on $\boldsymbol{h}^{\text{base}}$ being continuous is automatically satisfied if its domain is a finite set (as is the case for Theorem 2). Finally, the strict feasibility condition is verified by the uniform classifier.

Now, consider a form of $\ell_2$ regularization of (5):

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}_+^K}\mathbb{E}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}),\boldsymbol{h}^{\text{base}}(X)\right)+\frac{\zeta}{2}\left\|\widetilde{\boldsymbol{G}}(X)^T\boldsymbol{\lambda}\right\|_2^2\right] \tag{93}$$

where $\widetilde{\boldsymbol{G}}(x)\triangleq(\boldsymbol{G}(x),\boldsymbol{I}_K)\in\mathbb{R}^{K\times(K+C)}$. We show now that there is a unique minimizer $\boldsymbol{\lambda}_\zeta^\star$ of (93).

**Lemma 6.** *Under Assumption 1, there exists a unique minimizer $\boldsymbol{\lambda}_\zeta^\star$ of the regularized problem (93).*

*Proof.* Denote the function $A : \mathbb{R}_+^K \to \mathbb{R}$ by

$$A(\boldsymbol{\lambda}) \triangleq \mathbb{E}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(X)\right) + \frac{\zeta}{2}\left\|\widetilde{\boldsymbol{G}}(X)^T\boldsymbol{\lambda}\right\|_2^2\right]. \tag{94}$$

That the range of $A$ falls within $\mathbb{R}$ follows by Assumption 1, since then the function $x \mapsto D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x))$ is $P_X$-integrable. We will show that $A$ is lower semicontinuous and $\zeta$-strongly convex.

By Lemma 3, $\boldsymbol{v} \mapsto D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p})$ is differentiable for any fixed $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$, implying that it is also continuous. Thus, $\boldsymbol{\lambda} \mapsto D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x))$ is continuous for each $x \in \mathcal{X}$. Hence, by Fatou's lemma and boundedness of $\boldsymbol{G}$, $A$ is lower semicontinuous.

Next, to show strong convexity, we note that $\boldsymbol{\lambda} \mapsto D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x))$ is convex for each $x \in \mathcal{X}$. Indeed, this function is the supremum of affine functions. Further, the regularization term is $\zeta$-strongly convex, as its Hessian is given by

$$\zeta \cdot \left(\mathbb{E}\left[\widetilde{\boldsymbol{G}}(X)\widetilde{\boldsymbol{G}}(X)^T\right] + \boldsymbol{I}\right), \tag{95}$$

which is positive definite with minimal eigenvalue at least $\zeta$.

Now, for each fixed $\theta > 0$, consider the compact set $\Lambda_\theta \triangleq \{\boldsymbol{\lambda} \in \mathbb{R}_+^K \,;\, \|\boldsymbol{\lambda}\|_2^2 \le \theta\}$. By what we have shown thus far, there is a unique minimizer $\boldsymbol{\lambda}_\theta$ of $A$ over $\Lambda_\theta$. By strong convexity, if $A$ has a global minimizer then it is unique. We will show that $\boldsymbol{\lambda}_\theta$ is a global minimizer of $A$, where $\theta = 2(A(\mathbf{0}) - D^\star)/\zeta$. Suppose that $\mathbf{0}$ is not a global minimzer. Fix $\boldsymbol{\lambda} \in \mathbb{R}_+^K$ such that $A(\mathbf{0}) > A(\boldsymbol{\lambda})$. Then,

$$A(\mathbf{0}) > A(\boldsymbol{\lambda}) \ge D^\star + \frac{\zeta}{2}\left(\mathbb{E}\left[\left\|\boldsymbol{G}(X)^T\boldsymbol{\lambda}\right\|_2^2\right] + \|\boldsymbol{\lambda}\|_2^2\right) \ge D^\star + \frac{\zeta}{2}\|\boldsymbol{\lambda}\|_2^2. \tag{96}$$

Thus, $\|\boldsymbol{\lambda}\|_2^2 < \theta$. This implies that $\boldsymbol{\lambda}_\theta$ is a global minimizer of $A$, hence it is the unique global minimizer of $A$. The proof of the lemma is thus complete. □

The following bound shows that $\boldsymbol{\lambda}_\zeta^\star$ is within $O(\zeta)$ of achieving $D^\star$ (see (5)).

**Lemma 7.** *Suppose Assumption 1 holds, fix $\zeta \ge 0$, and denote the unique solution and the optimal objective value of (93) by $\boldsymbol{\lambda}_\zeta^\star$ and $D_\zeta^\star$, respectively. We have the bounds*

$$\mathbb{E}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}_\zeta^\star), \boldsymbol{h}^{\text{base}}(X)\right)\right] \le D_\zeta^\star \le D^\star + \theta_{\text{reg}} \cdot \zeta, \tag{97}$$

*where we define the constant $\theta_{\text{reg}} \triangleq \lambda_{\max}^2 \cdot (1 + g_{\text{mean}})/2$.*

*Proof.* The first bound is trivial. Using Lemma 5, we may fix a $\boldsymbol{\lambda}^\star \in \mathbb{R}_+^K$ with $\|\boldsymbol{\lambda}^\star\|_1 \le \lambda_{\max}$ such that $\boldsymbol{\lambda}^\star$ achieves $D^\star$. By definition of $D_\zeta^\star$,

$$D_\zeta^\star \le \mathbb{E}\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}^\star), \boldsymbol{h}^{\text{base}}(X)\right) + \frac{\zeta}{2}\left\|\widetilde{\boldsymbol{G}}(X)^T\boldsymbol{\lambda}^\star\right\|_2^2\right] \le D^\star + \theta_{\text{reg}} \cdot \zeta,$$

where the last inequality follows since for the 2-matrix norm, $\|\boldsymbol{M}\boldsymbol{\lambda}\|_2 \le \|\boldsymbol{M}\|_2\|\boldsymbol{\lambda}\|_2$ and $\|\boldsymbol{M}^T\|_2 = \|\boldsymbol{M}\|_2$. □

Next, we derive a sample-complexity bound for the finite-sample problem (8) via generalizing the proofs of Theorem 3 in [AAW+20a] and Theorem 13.2 in [HR19].

**Lemma 8.** *Suppose Assumption 1 holds, and let $\lambda_{\max}$ and $g_{\max}$ be as defined in Lemma 5 and equation (92). For any $\delta \in (0,1)$, with $\boldsymbol{\lambda}_{\zeta,N}^\star$ denoting the unique solution to (8), it holds with probability at least $1-\delta$ that*

$$\mathbb{E}_X\left[D_f^{\text{conj}}\left(\boldsymbol{v}(X;\boldsymbol{\lambda}_{\zeta,N}^\star), \boldsymbol{h}^{\text{base}}(X)\right)\right] \le D_\zeta^\star + \frac{2g_{\max} \cdot (1 + \zeta \cdot \lambda_{\max})^2}{\delta\zeta N}. \tag{98}$$

*Proof.* Let $\Lambda \triangleq \{\boldsymbol{\lambda} \in \mathbb{R}_+^K \; ; \; \|\boldsymbol{\lambda}\|_1 \leq \lambda_{\max}\}$, and consider the function $\ell : \Lambda \times \mathcal{X} \to \mathbb{R}$ defined by

$$\ell(\boldsymbol{\lambda}, x) \triangleq D_f^{\text{conj}}\left(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x)\right) + \frac{\zeta}{2}\left\|\widetilde{\boldsymbol{G}}(x)^T\boldsymbol{\lambda}\right\|_2^2. \tag{99}$$

Note that the regularized problem (93) can be written as

$$D_\zeta^\star \triangleq \min_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} \mathbb{E}\left[\ell(\boldsymbol{\lambda}, X)\right], \tag{100}$$

and the finite-sample version of it (8) can also be written as

$$D_{\zeta,N}^\star \triangleq \min_{\boldsymbol{\lambda}\in\mathbb{R}_+^K} \frac{1}{N}\sum_{i\in[N]} \ell(\boldsymbol{\lambda}, X_i). \tag{101}$$

We show first that, for each fixed $x \in \mathcal{X}$, the function $\boldsymbol{\lambda} \mapsto \ell(\boldsymbol{\lambda}, x)$ is $\zeta$-strongly convex over $\Lambda$. The gradient of the regularization term is $\zeta\widetilde{\boldsymbol{G}}(x)^T\boldsymbol{\lambda}$, and its Hessian is given by

$$\nabla_{\boldsymbol{\lambda}}^2 \frac{\zeta}{2}\left\|\widetilde{\boldsymbol{G}}(x)^T\boldsymbol{\lambda}\right\|_2^2 = \zeta\boldsymbol{G}(x)\boldsymbol{G}(x)^T + \zeta\boldsymbol{I}_K. \tag{102}$$

Further, the function $\boldsymbol{\lambda} \mapsto D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x))$ is convex as it is a pointwise supremum of linear functions. Indeed, for any $\boldsymbol{p} \in \boldsymbol{\Delta}_C$, recalling that $\boldsymbol{v}(x;\boldsymbol{\lambda}) = -\boldsymbol{G}(x)^T\boldsymbol{\lambda}$, we have the formula

$$D_f^{\text{conj}}(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{p}) = \sup_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} -\boldsymbol{q}^T\boldsymbol{G}(x)^T\boldsymbol{\lambda} - D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}). \tag{103}$$

Next, we show Lipschitzness of $\boldsymbol{\lambda} \mapsto \ell(\boldsymbol{\lambda}, x)$. For any fixed $\boldsymbol{v} \in \mathbb{R}^C$ and $\boldsymbol{p} \in \boldsymbol{\Delta}_C^+$, we have the gradient (see Lemma 3)

$$\nabla_{\boldsymbol{v}} D_f^{\text{conj}}(\boldsymbol{v}, \boldsymbol{p}) = \boldsymbol{q}^{\text{conj}}(\boldsymbol{v}) \in \boldsymbol{\Delta}_C, \tag{104}$$

where

$$\boldsymbol{q}^{\text{conj}}(\boldsymbol{v}) \triangleq \operatorname*{argmin}_{\boldsymbol{q}\in\boldsymbol{\Delta}_C} D_f(\boldsymbol{q}\,\|\,\boldsymbol{p}) - \boldsymbol{v}^T\boldsymbol{q}. \tag{105}$$

Thus, we have the gradient

$$\nabla_{\boldsymbol{\lambda}} D_f^{\text{conj}}\left(\boldsymbol{v}(x;\boldsymbol{\lambda}), \boldsymbol{h}^{\text{base}}(x)\right) = -\boldsymbol{G}(x)\boldsymbol{q}^{\text{conj}}\left(\boldsymbol{v}(x;\boldsymbol{\lambda})\right). \tag{106}$$

Hence, the gradient of $\boldsymbol{\lambda} \mapsto \ell(\boldsymbol{\lambda}, x)$ is

$$\nabla_{\boldsymbol{\lambda}}\ell(\boldsymbol{\lambda}, x) = -\boldsymbol{G}(x)\boldsymbol{q}^{\text{conj}}\left(\boldsymbol{v}(x;\boldsymbol{\lambda})\right) + \zeta\widetilde{\boldsymbol{G}}(x)^T\boldsymbol{\lambda}, \tag{107}$$

which therefore satisfies the bound

$$\|\nabla_{\boldsymbol{\lambda}}\ell(\boldsymbol{\lambda}, x)\|_2 \leq \|\boldsymbol{G}(x)\|_2\left(1 + \zeta\cdot\lambda_{\max}\right). \tag{108}$$

Therefore, each $\boldsymbol{\lambda} \mapsto \ell(\boldsymbol{\lambda}, x)$ is $A$-Lipschitz with

$$A = (1 + \zeta\cdot\lambda_{\max})\cdot\sup_{x\in\mathcal{X}}\|\boldsymbol{G}(x)\|_2. \tag{109}$$

Thus, by Theorem 13.1 in [HR19], with probability $1-\delta$ we have the bound

$$\mathbb{E}_X\left[\ell\left(\boldsymbol{\lambda}_{\zeta,N}^\star, X\right)\right] \leq D_\zeta^\star + \frac{2A^2}{\delta\zeta N}. \tag{110}$$

With probability one, we have the bound

$$\mathbb{E}_X\left[D_f^{\text{conj}}\left(\boldsymbol{v}\left(X;\boldsymbol{\lambda}_{\zeta,N}^\star\right), \boldsymbol{h}^{\text{base}}(X)\right)\right] \leq \mathbb{E}_X\left[\ell\left(\boldsymbol{\lambda}_{\zeta,N}^\star, X\right)\right]. \tag{111}$$

This completes the proof of the lemma. □

Now, we are ready to finish the proof of Theorem 6 by specializing the above lemmas to the KL-divergence case. So, we set $f(t) = t \log t$ for the rest of the proof. By Lemmas 7–8, we have with probability $1 - \delta$

$$\mathbb{E}_X \left[ D_f^{\text{conj}} \left( \boldsymbol{v}(X; \boldsymbol{\lambda}^\star_{\zeta,N}), \boldsymbol{h}^{\text{base}}(X) \right) \right] \leq D^\star + \theta_{\text{reg}} \cdot \zeta + \frac{2 g_{\max} \cdot (1 + \zeta \cdot \lambda_{\max})^2}{\delta \zeta N}. \tag{112}$$

Thus, by Lipschitzness (Proposition 1) and (82)

$$\mathbb{E}_X \left[ D_f^{\text{conj}} \left( \boldsymbol{v}(X; \boldsymbol{\lambda}^{(t)}_{\zeta,N}), \boldsymbol{h}^{\text{base}}(X) \right) \right] \leq D^\star + \frac{1}{2} \sqrt{g_{\text{mean}}} \beta e^{-rt} + \theta_{\text{reg}} \cdot \zeta + \frac{2 g_{\max} \cdot (1 + \zeta \cdot \lambda_{\max})^2}{\delta \zeta N}. \tag{113}$$

Here, we are choosing the constant $\beta$ independently of $N$ (as the optimal values of $\boldsymbol{\lambda}$ are bounded), and $r$ of order $\sqrt{\frac{\zeta}{\frac{1}{2N} + \zeta}}$ (as can be guaranteed from Corollary 3.1 and Theorem 3.4 in [DY16]).

Choose $\zeta = \Theta(N^{-1/2})$. Collecting the constants in (113), we obtain that

$$\mathbb{E}_X \left[ D_f^{\text{conj}} \left( \boldsymbol{v}(X; \boldsymbol{\lambda}^{(t)}_{\zeta,N}), \boldsymbol{h}^{\text{base}}(X) \right) \right] \leq D^\star + \frac{1}{2} \sqrt{g_{\text{mean}}} \beta e^{-rt} + \frac{\ell}{\delta \sqrt{N}} \tag{114}$$

for some constant $\ell$ that is completely determined by $\theta_{\text{reg}}$, $g_{\max}$, and $\lambda_{\max}$. This bound can be further upper bounded by $D^\star + O(N^{-1/2})$ by choosing $t \geq \frac{1}{2r} \log N = \Theta(\log N)$, thereby completing the proof of the theorem.

### A.8 Linearized multi-class group fairness criteria

We include, for completeness, how the group-fairness metrics in Table 2 linearize, i.e., written in the form:

$$\boldsymbol{\mu}(\boldsymbol{h}) \triangleq \mathbb{E}_{P_X} \left[ \boldsymbol{G}(X) \boldsymbol{h}(X) \right] \leq \boldsymbol{0}. \tag{115}$$

We assume that we have in hand a well-calibrated classifier that approximates $P_{Y,S|X}$, i.e., that predicts both group membership $S$ and the true label $Y$ from input variables $X$. This classifier can be directly marginalized into the following models:

- a label classifier $\boldsymbol{h}^{\text{base}} : \mathcal{X} \to \boldsymbol{\Delta}_C$ that predicts true label from input variables,
$$\boldsymbol{h}^{\text{base}}(x) \triangleq (P_{Y|X=x}(1), \cdots, P_{Y|X=x}(C)) \quad \text{for} \quad x \in \mathcal{X}, \tag{116}$$
- a group membership classifier $\boldsymbol{s} : \mathcal{X} \times \mathcal{Y} \to \boldsymbol{\Delta}_A$ that uses input and output variables to predict group membership,
$$\boldsymbol{s}(x, y) \triangleq (P_{S|X,Y}(1 \mid x, y), \cdots, P_{S|X,Y}(A \mid x, y)) \quad \text{for} \quad (x, y) \in \mathcal{X} \times \mathcal{Y}, \tag{117}$$

We let $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_C$ denote the standard basis vectors of $\mathbb{R}^C$. We suppose that the support of the group attribute $S$ is $\mathcal{S} \triangleq [A]$.

**Statistical parity.** This fairness metric measures whether the predicted outcome $\widehat{Y}$ is independent of the group attribute $S$. For statistical parity, the $\boldsymbol{G}(x)$ matrix has rows:

$$\left( (-1)^\delta \frac{\sum_{c=1}^C s_a(x, c) h_c^{\text{base}}(x)}{P_S(a)} - \left( \alpha + (-1)^\delta \right) \right) \boldsymbol{e}_{c'}.$$

There are $K = 2AC$ rows since $(\delta, a, c') \in \{0, 1\} \times [A] \times [C]$. See Appendix A.9 for a full derivation.

**Equalized odds.** This fairness metric requires the predicted outcome $\widehat{Y}$ and the group attribute $S$ to be independent conditioned on the true label $Y$. When the classification task is binary, the equalized odds becomes the equality of false positive rate and false negative rate over all groups. For equalized odds, the $\boldsymbol{G}(x)$ matrix has rows:

$$\left( (-1)^\delta \frac{s_{a'}(x, c) h_c^{\text{base}}(x)}{P_{S|Y=c}(a')} - \left( \alpha + (-1)^\delta \right) h_c^{\text{base}}(x) \right) \boldsymbol{e}_{c'}.$$

There are $K = 2AC^2$ rows.

**Overall accuracy equality.** This fairness metric requires the accuracy of the predictive model to be the same across all group groups. The $\boldsymbol{G}(x)$ matrix has rows:

$$(-1)^{\delta}\frac{s_a(x,\cdot)\odot \boldsymbol{h}^{\text{base}}(x)}{P_S(a)} - \left(\alpha + (-1)^{\delta}\right)\cdot \boldsymbol{h}^{\text{base}}(x),$$

where $\odot$ represents the element-wise product. There are $K = 2A$ rows.

### A.9 A detailed derivation of linearization of statistical parity

We derive the linearized formula for Statistical Parity given in Appendix A.8. Recall that a prediction $\widehat{Y}$ satisfies statistical parity (SP) if it is independent of the group attribute $S$, i.e., $P_{\widehat{Y}|S=a}(c') = P_{\widehat{Y}}(c')$ for every $(a, c') \in [A] \times [C]$. A relaxed notion of SP is 'approximate independence' of $\widehat{Y}$ and $S$: $|P_{\widehat{Y}|S=a}(c')/P_{\widehat{Y}}(c') - 1| \leq \alpha$ for some small $\alpha \geq 0$ and all $(a, c') \in [A] \times [C]$. Using $P_{\widehat{Y}|S} = P_{\widehat{Y},S}/P_S$ and rearranging, the above inequality is equivalent to

$$\pm P_{\widehat{Y},S}(c', a)/P_S(a) - (\alpha \pm 1)P_{\widehat{Y}}(c') \leq 0.$$

We expand via conditioning $\widehat{Y}$ on $X$, and $S$ on $(X, Y)$. Recall that $h_{c'}(x) = P_{\widehat{Y}|X=x}(c')$ and $s_a(x, c) = P_{S|X=x,Y=c}(a)$ by definition, and that we have a Markov chain $(Y, S)$–$X$–$\widehat{Y}$; hence, $P_{\widehat{Y}}(c') = \mathbb{E}[h_{c'}(X)]$ and $P_{\widehat{Y},S}(c', a) = \mathbb{E}[\sum_{c\in[C]} s_a(X, c)h_c^{\text{base}}(X)h_{c'}(X)]$. Thus, we can write approximate SP as

$$\mathbb{E}\left[\left(\pm P_S(a)^{-1}\sum_{c\in[C]} s_a(X,c)h_c^{\text{base}}(X) - (\alpha \pm 1)\right)h_{c'}(X)\right] \leq 0.$$

We denote $\boldsymbol{h}(x) = (h_1(x), h_2(x), \cdots, h_C(x))^T$, and for $(\delta, a, c') \in \{0, 1\} \times [A] \times [C]$, denote

$$\boldsymbol{g}^{(\delta,a,c')}(x) := \left((-1)^{\delta}P_S(a)^{-1}\sum_{c\in[C]} s_a(x,c)h_c^{\text{base}}(x) - (\alpha + (-1)^{\delta})\right)\boldsymbol{e}_{c'},$$

where $\{\boldsymbol{e}_{c'}\}_{c'\in[C]}$ is the standard basis of $\mathbb{R}^C$. Then, for each pair $(\delta, a, c') \in \{0, 1\} \times [A] \times [C]$, we have a linear constraint $\mathbb{E}[\boldsymbol{g}^{(\delta,a,c')}(X)^T\boldsymbol{h}(X)] \leq 0$. Since there are $K = 2AC$ possible triplets $(\delta, a, c')$, we convert the SP constraint into $K$ linear constraints.

## B Additional experiments and more details on the experimental setup

### B.1 Numerical Benchmark Details

#### B.1.1 Datasets

The HSLS dataset is collected from 23,000+ participants across 944 high schools in the USA, and it includes thousands of features such as student demographic information, school information, and students' academic performance across several years. We preprocessed the dataset (e.g., dropping rows with a significant number of missing entries, performing k-NN imputation, normalization), and the number of samples reduced to 14,509.

The ENEM dataset, collected from the 2020 Brazilian high school national exam and made available by the Brazilian Government [INE20], is comprised of student demographic information, socio-economic questionnaire answers (e.g., parents education level, if they own a computer) and exam scores. We preprocess the dataset by removing missing values, repeated exam takers, and students taking the exam before graduation ("treineiros") and obtain $\sim$1.4 million samples with 138 features.

#### B.1.2 Hyperparameters

For logistic regression and gradient boosting, we use the default parameters given by Scikit-learn. For random forest, we set the number of trees and the minimum number of samples per leaf to 10. For all classifiers, we fixed the random state to 42. When running `FairProjection` (cf. Algorithm 1), we set the hyperparameters $\zeta = 1/\sqrt{N}$ (see Theorem 6) and $\rho = 2$ (see Appendix A.3.1), where $N$ is the number of samples.

### B.1.3 Benchmark Methods

For binary classification, we compare with six different benchmark methods:

- EqOdds [HPS16]: We use AIF360 implementation of EqOddsPostprocessing and we use 50% of the test set as a validation set, i.e., 70% training set, 15% validation set, 15% test set.
- CalEqOdds [PRW+17]: We use AIF360 implementation of CalibratedEqOddsPostprocessing and we use 50% of the test set as a validation set, i.e., 70% training set, 15% validation set, 15% test set.
- Reduction [ABD+18]: We use AIF360 implementation of ExponentiatedGradientReduction, and we use 10 different epsilon values as follows: $[0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2]$. We used EqualizedOdds constraint for MEO experiments and DemographicParity for statistical parity experiments.
- Rejection [KKZ12]: We use AIF360 implementation of RejectOptionClassification. We use the default parameters except metric_ub and metric_lb, namely, low_class_thresh $= 0.01$, high_class_thresh $= 0.99$, num_class_thresh $= 100$, num_ROC_margin $= 50$. We set the values metric_ub $= \epsilon$ and metric_lb $= -\epsilon$ to obtain trade-off curves. Epsilon values we used are: $[0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2]$.
- LevEqOpp [CDH+19]: We used the code provided in the Github repo, originally programmed in R. We converted it into Python, and verified that the Python version achieved similar accuracy/fairness performance to their R version on UCI Adult dataset. We follow the same hyperparameters setup in [CDH+19].

The following four methods, despite being mentioned in Table 1, are not included in the experiments:

- FACT [KCT20]: We used the code provided on the Github repo. We did not include the results in the main text as we found that:
  (i) This method is not directly comparable because they find post-processing parameters on the entire test set and apply them on the test set. This is different from all other methods we are comparing including our method, which use training set or a separate validation set to fit the post-processing mechanism. For this reason, FACT often has a point that lies above all other curves on the accuracy-fairness plot. However, this is not a fair comparison. We include the results of FACT in the COMPAS plots for the sake of demonstration.
  (ii) We found the results produced by this method inconsistent. Partial reason is due to the problem of finding mixing rates—probability of flipping $\widehat{Y} = 1$ to $0$ (i.e., $P(\widetilde{Y} = 0|\widehat{Y} = 1)$) and vice-versa—which have to be between 0 and 1. But there are cases where these values lie outside $[0, 1]$, which leads to erroneous and inconsistent results.

  For the results we present in the COMPAS plots, we used 20 epsilon values from 1 to $10^{-4}$, equidistant in log space. We used 10 different train/test splits as we do in all other experiments. If certain splits does not produce a feasible solution, we drop those results. If none of the 10 splits produce a feasible solution, we drop the epsilon value. At the end, we had 19 epsilon values.
- Identifying [JN20]: Their optimization formulation is a special case of our formulation when $f$-divergence is KL divergence, but their algorithm requires retraining a classifier multiple times to solve the optimization problem, which results in a much slower runtime compared to ours (see Lines 1037–1046 in Appendix B.4). Nevertheless, we will add experiments for binary classification using [JN20] in the final version.
- FST [WRC20, WRC21]: Codes are not available publicly.
- Overlapping [YCK20]: We did not include this method for binary classification experiments as it reduces to the Reductions [ABD+18] approach for the binary class, binary protected group case. We could not benchmark for multi-class experiments with the code available online as it was assuming binary class (even though multiple protected groups).

For multi-class comparison, we compare with Adversarial [ZLM18]. In theory, the adversarial debiasing method is applicable to multi-class labels and groups, but its AIF360 implementation works

only for binary labels and binary groups. We adapted their implementation to work on multi-class labels by changing the last layer of the classifier model from one-neuron sigmoid activation to multi-neuron soft-max activation. We varied adversary_loss_weight to obtain a trade-off curve, values taken from $[0.001, 0.01, 0.1, 0.2, 0.35, 0.5, 0.75]$. For all other parameters, we used the default values: num_epochs $= 50$, batch_size $= 128$, classifier_num_hidden_units $= 200$.

There are some methods that are relevant to our work but we could not benchmark in our experiments due to the lack of publicly available codes, including [WRC21], [MW18], [JSW22].

### B.2 Additional experiments on runtime of FairProjection

We preform an ablation study on the runtime to illustrate that the parallelizability of `FairProjection` can significantly reduce the runtime, especially when the dataset contains hundreds of thousands of samples. We report the runtime of `FairProjection-KL` on ENEM with 2 classes, 2 groups, and with different sizes. In Table 4, we observe that when the number of samples exceeds 200k, parallelization leads to $10.1\times$ to $15.5\times$ speedup of the runtime.

| **Method** | **# of Samples (in thousands)** | | | | | |
|---|---|---|---|---|---|---|
| | 20 | 50 | 100 | 200 | 500 | ∼1400 |
| Non-Parallel | 0.37±0.00 | 0.87±0.01 | 1.72±0.01 | 3.53±0.01 | 9.09±0.01 | 25.26±0.02 |
| Parallel (GPU) | 0.18±0.00 | 0.22±0.01 | 0.25±0.01 | 0.32±0.01 | 0.64±0.01 | 1.63±0.05 |
| Speedup | 2.00× | 3.92× | 7.21× | 10.97× | 14.23× | 15.46× |

**Table 4:** Execution time of parallel (on GPU) and non-parallel (on CPU) versions of the `FairProjection-KL` ADMM algorithm on the ENEM datasets with different sizes (time shown in minutes) with gradient boosting base classifiers.

### B.3 Additional Explanation on runtime comparison

The theoretical analysis below contrasts the runtimes of both `FairProjection` and Reduction [ABD+18], which is in line with our numerically observed comparison in Table 3. Two key factors make `FairProjection` faster than Reduction:

1. `FairProjection` needs a much lower number of iterations than Reduction does (logarithmic vs. polynomial).
2. Each iteration for FairProjection is less computationally expensive than its counterpart in Reduction. In fact, it is independent of the underlying model being projected, whereas Reduction requires retraining.

In more detail, one can obtain from [ABD+18, Theorem 3] that the Reductions approach converges in $O(N^2)$ iterations (where $N$ is the number of samples and we use the suggested $\alpha = 1/2$ in [ABD+18, Theorem 3] according to the discussion at the top of page 6 therein). Taking the runtime of each iteration into consideration, one cannot hope for a runtime faster than $O(N^4)$ for Reduction. In fact, the runtime for Reduction must be higher than $O(N^4)$, since each of its iterations performs the subroutine $\text{BEST}_{\boldsymbol{h}}(\boldsymbol{\lambda})$, which is a 'cost-sensitive classification' problem (i.e., numerically solving for an optimal classifier), and the $O(N^4)$ estimate would hold only if this *retraining* procedure can be done in *constant time* (which might be overly optimistic). In contrast, `FairProjection` does not require this retraining procedure at all, runs in $O(\log N)$ iterations, has $O(N)$ runtime for each iteration, and can perform much of each iteration in a parallel way.

For the dependence of the runtime of `FairProjection` on the number of groups, we note that there is a *linear* dependence on the number of constraints $K$ when the number of samples $N$ is much larger than $K$ (which is the case for all datasets we consider), so one can say that the runtime is at most $\gamma KN \log N$ for an *absolute* constant $\gamma$. Note that there are $K = 2AC$ constraints for statistical parity, where $A$ is the number of sensitive groups, and $C$ is the number of classes; e.g., for the ENEM-1.4M-2C dataset that is used in Table 3, we get $K = 8$ for statistical parity. The $K$ factor in the $O(KN \log N)$ rate comes from the creation of the vector $\boldsymbol{q}$ in Algorithm 1. If one does not parallelize, still one gets a runtime of $O(CKN \log N)$. Interestingly, the $\boldsymbol{v}_i$-update step runtime in Algorithm 1 is $O(C)$ for a fixed $i \in [N]$ for both KL-divergence and Cross Entropy (see Appendices A.3 and A.4).

### B.4 Omitted Experimental Results on Accuracy-Fairness Trade-off

#### B.4.1 Accuracy-fairness trade-off in binary classification

We include the results of benchmark methods and Fair Projection on 4 datasets (HSLS, ENEM-50k, Adult, and COMPAS) and 3 base classifiers (Logistic regression, Random forest, and GBM) in Figures 3-10. For equalized odds experiments, we have six benchmark methods (`EqOdds`, `Rejection`, `Reduction`, `CalEqOdds`, `FACT`, `LevEqOpp`). For statistical parity experiments, we have `Rejection` and `Reduction`. We plot Fair Projection with both cross entropy and KL divergence.

When a method performs significantly worse than others, we did not plot its results. We did not include `Rejection` in the Adult plots as it did not produce consistent and reliable results on this dataset. `CalEqOdds` is included only in COMPAS as its performance was significantly worse and the point was too far away from other curves in all other datasets. `FACT` is also included only in the COMPAS plots and the reasons for this are explained in Appendix B.1.3.

We observe that Fair Projection performs consistently well in all four datasets. `FairProjection-CE` and `FairProjection-KL` have similar performance (i.e., overlapping curves) in most cases. The performance of Fair Projection is often comparable with `Reduction`. `Rejection` has competitive performance in ENEM-50k and HSLS, but its performance falters in COMPAS and Adult. `EqOdds` produces a point with very low MEO but with a substantial loss in accuracy. `LevEqOpp` also yields a point with low MEO but with a much smaller accuracy drop. Even though `LevEqOpp` only optimizes for FNR difference between two groups, it performs surprisingly well in terms of MEO in all four datasets. However, we note that `LevEqOpp` can only produce a point, not a curve, and it does not enjoy the generality of Fair Projection as it is specifically designed for binary-class, binary-group predictions and minimizing Equalized Opportunity difference.

#### B.4.2 Accuracy-fairness trade-off in multi-class/multi-group classification

In the main text, we showed the performance of `FairProjection-CE` on multi-class prediction with 5 classes and 2 groups (see Figure 2). We include results under a few different multi-class settings here. First, we show results on ENEM-50k-5-5 which has 5 classes and 5 groups in Figure 11 and 12 . We obtain 5 groups by not binarizing the race feature. Then, we show results on binary classification with 5 groups in Figure 13 and 14. Finally, we include the extended version of Figure 2 that include both `FairProjection-CE` and `FairProjection-KL` in Figure 15.

To measure multi-class performance, we extend the definition of mean equalized odds (MEO) and statistical parity as follows:

$$\mathsf{MEO} = \max_{i \in \mathcal{Y}} \max_{s_1, s_2 \in \mathcal{S}} (|\mathsf{TPR}_i(s_1) - \mathsf{TPR}_i(s_2)| + |\mathsf{FPR}_i(s_1) - \mathsf{FPR}_i(s_2)|)/2 \quad (118)$$

$$\text{Statistical Parity} = \max_{i \in \mathcal{Y}} \max_{s_1, s_2 \in \mathcal{S}} |\mathsf{Rate}_i(s_1) - \mathsf{Rate}_i(s_2)| \quad (119)$$

where we denote $\mathsf{TPR}_i(s) = P(\widehat{Y} = i \mid Y = i, S = s)$, $\mathsf{FPR}_i(s) = P(\widehat{Y} = i \mid Y \neq i, S = s)$, and $\mathsf{Rate}_i(s) = P(\widehat{Y} = i \mid S = s)$.

In all experiments, `FairProjection` reduces MEO and statistical parity significantly (e.g., 0.22 to 0.14) with a negligible sacrifice in accuracy.

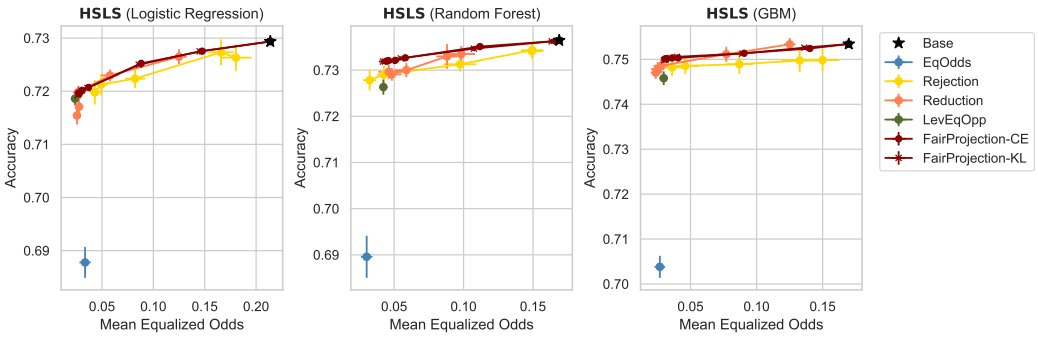



**Figure 3:** Accuracy-fairness curves of FairProjection and benchmark methods on the HSLS dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is MEO.

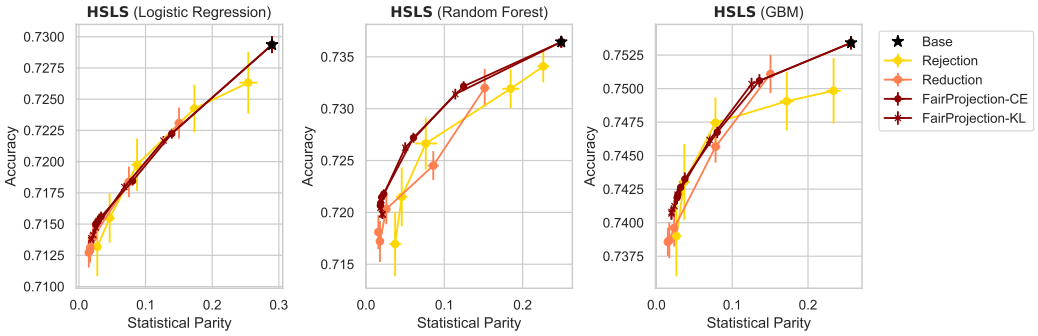



**Figure 4:** Accuracy-fairness curves of FairProjection and benchmark methods on the HSLS dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is statistical parity.

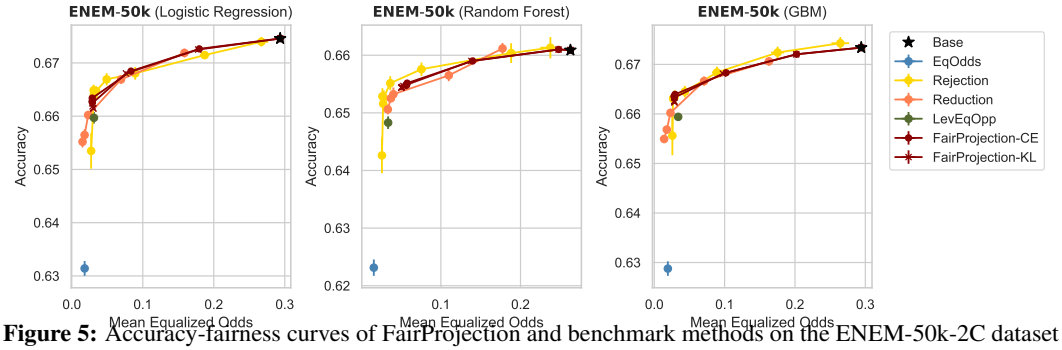



**Figure 5:** Accuracy-fairness curves of FairProjection and benchmark methods on the ENEM-50k-2C dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is MEO.

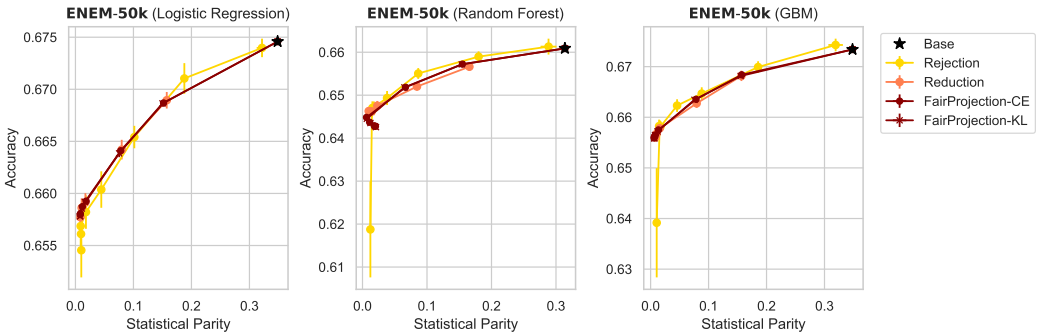



**Figure 6:** Accuracy-fairness curves of FairProjection and benchmark methods on the ENEM-50k-2C dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is statistical parity.

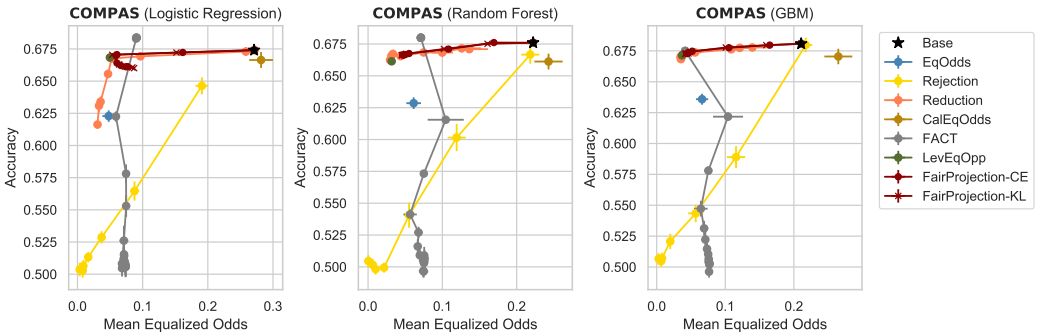



**Figure 7:** Accuracy-fairness curves of FairProjection and benchmark methods on COMPAS with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is MEO.

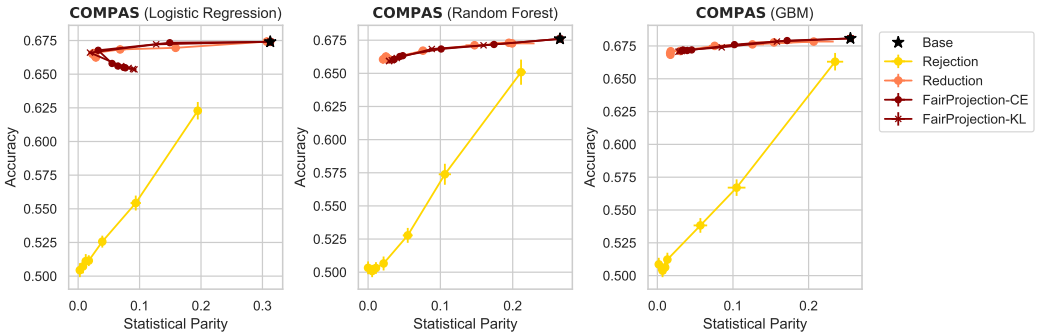



**Figure 8:** Accuracy-fairness curves of FairProjection and benchmark methods on COMPAS with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is statistical parity.

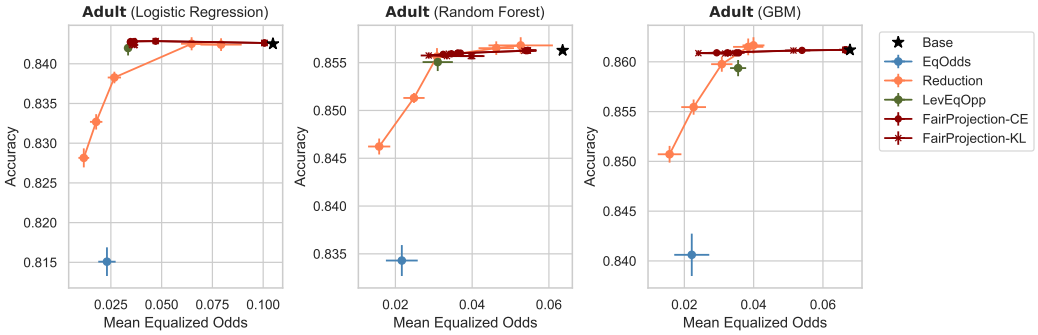



**Figure 9:** Accuracy-fairness curves of FairProjection and benchmark methods on the Adult dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is MEO.

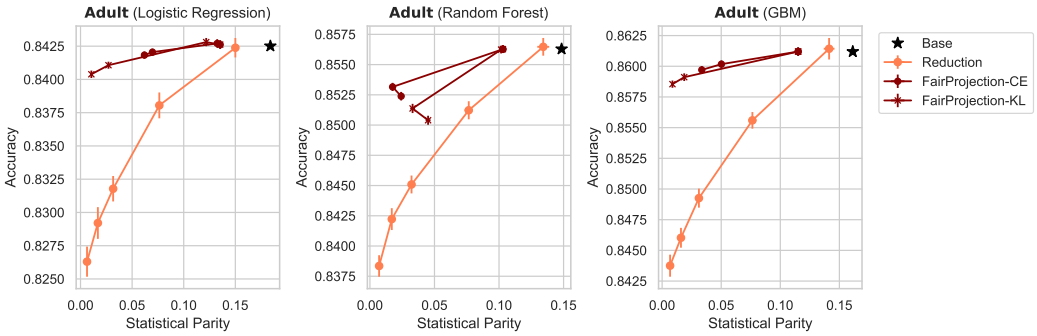



**Figure 10:** Accuracy-fairness curves of FairProjection and benchmark methods on the Adult dataset with 3 different models (Logistic regression, Random forest, GBM). The fairness constraint is statistical parity.

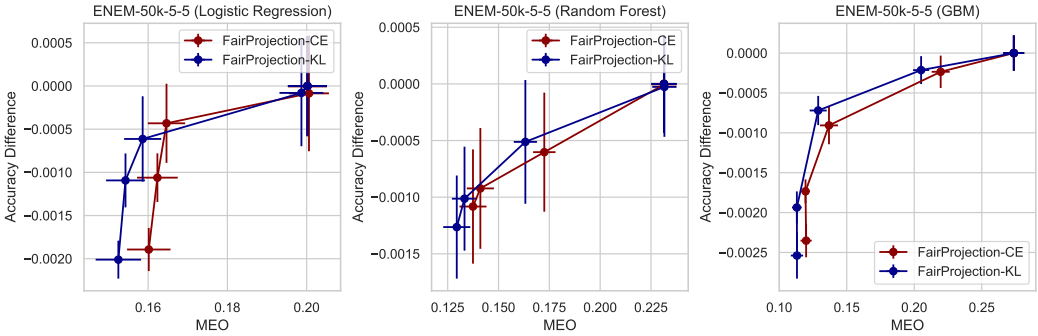



**Figure 11:** Accuracy-fairness curves of `FairProjection-CE` and `FairProjection-KL` on ENEM-50k with with 5 labels, 5 groups and different base classifiers base classifiers. The fairness constraint is MEO.

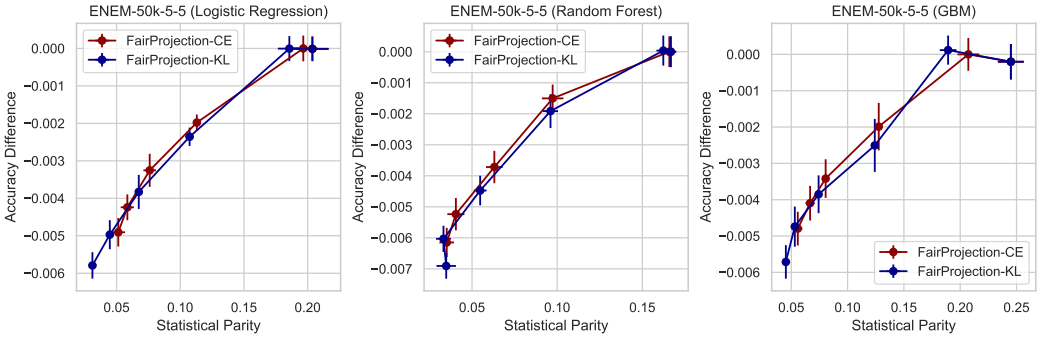



**Figure 12:** Accuracy-fairness curves of `FairProjection-CE` and `FairProjection-KL` on ENEM-50k with with 5 labels, 5 groups and different base classifiers base classifiers. The fairness constraint is SP.

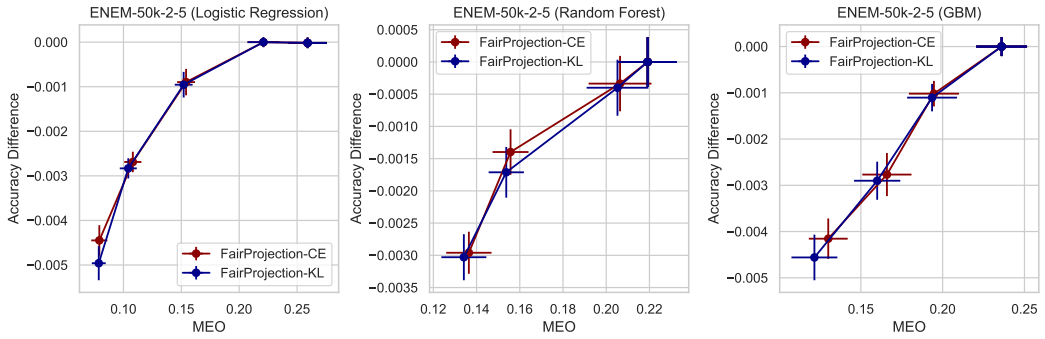



**Figure 13:** Accuracy-fairness curves of `FairProjection-CE` and `FairProjection-KL` on ENEM-50k with with 2 labels, 5 groups and different base classifiers base classifiers. The fairness constraint is MEO.

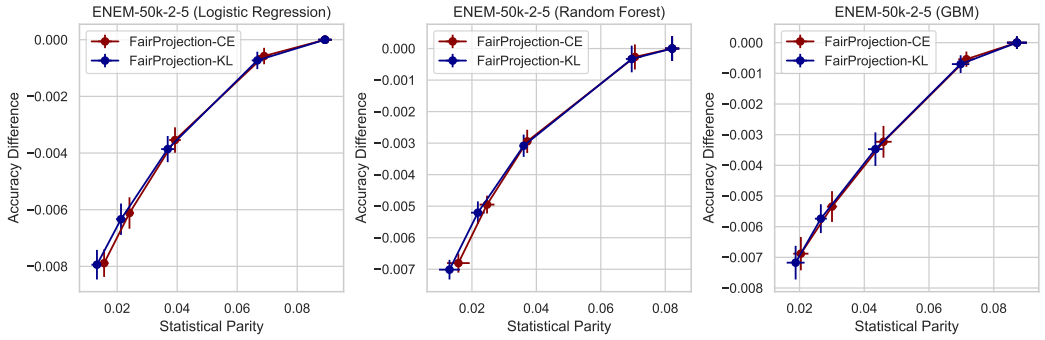



**Figure 14:** Accuracy-fairness curves of `FairProjection-CE` and `FairProjection-KL` on ENEM-50k with with 2 labels, 5 groups and different base classifiers base classifiers. The fairness constraint is SP.

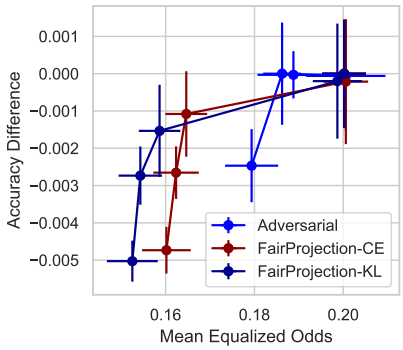



**Figure 15:** Comparison of `FairProjection-CE` and `FairProjection-KL` with `Adversarial` on ENEM-50k-5-2, meaning 5 labels, 2 groups. The reason for the difference comparing to Fig. 2 is that we resampled 50k data points from ENEM.

| Method | Multiclass | Multigroup | Feature Scores | Curve | Parallel | Rate | Metric |
|---|---|---|---|---|---|---|---|
| Reductions [ABD+18] | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | SP, (M)EO |
| Reject-option [KKZ12] | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | SP, (M)EO |
| EqOdds [HPS16] | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | EO |
| LevEqOpp [CDH+19] | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | FNR |
| CalEqOdds [PRW+17] | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | MEO |
| FACT [KCT20] | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | SP, (M)EO |
| Identifying [JN20] | ✓✗ | ✓ | ✓ | ✓ | ✗ | ✗ | SP, (M)EO |
| FST [WRC20, WRC21] | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | SP, (M)EO |
| Overlapping [YCK20] | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | SP, (M)EO |
| Adversarial [ZLM18] | ✓ | ✓ | N/A | ✓ | ✓ | ✗ | SP, (M)EO |
| `FairProjection` (ours) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | SP, (M)EO |

**Copy of Table 1.** Comparison between benchmark methods. **Multiclass/multigroup**: implementation takes datasets with multiclass/multigroup labels; **Scores**: processes raw outputs of probabilistic classifiers; **Curve**: outputs fairness-accuracy tradeoff curves (instead of a single point); **Parallel**: parallel implementation (e.g., on GPU) is available; **Rate**: convergence rate or sample complexity guarantee is proved; **Metric**: applicable fairness metric, with SP↔Statistical Parity, EO↔Equalized Odds, MEO↔Mean EO. Since `FairProjection` is a post-processing method, we focus our comparison on post-processing fairness intervention methods, except for Reductions [ABD+18], which is a representative in-processing method, and Adversarial [ZLM18], which we use to benchmark multi-class prediction. For comparing in-processing methods, see [LPB+21, Table 1].

### B.5 More on related work

Our method is a model-agnostic post-processing method, so we focus our comparison on such post-processing fairness intervention methods. In the above table, the only exception is Adversarial [ZLM18], which we use to benchmark multi-class prediction. Adversarial [ZLM18] is an in-processing method based on generative-adversarial network (GAN) where the adversary tries to guess the sensitive group attribute $S$ from $Y$ and $\widehat{Y}$. Even though this GAN-based approach is applicable to multi-class, multi-group prediction, it cannot be universally applied to any pre-trained classifier like our method.

EqOdds [HPS16], CalEqOdds [PRW+17] and LevEqOpp [CDH+19] are post-processing methods designed for binary prediction with binary groups. They find different decision thresholds for each group that equalize FNR and FPR of two groups. CalEqOdds [PRW+17] has an additional constraint that the post-processed classifier must be well-calibrated, and we observe in our experiments that this stringent constraint leads to a low-accuracy classifier especially when there is a big gap in the base rate between the two groups. FACT [KCT20] follows a similar approach but generalizes this to an optimization framework that can have both equalized odds and statistical parity constraints and flexible accuracy-fairness trade-off. The optimization formulation finds a desired confusion matrix, and their proposed post-processing method flips the predictions to match the desired confusion matrix. Reject-option [KKZ12] is similar in that it flips predictions near the decision threshold. In [KKZ12], instead of finding the optimal confusion matrix, it performs grid search to find the optimal margin around the decision threshold that can minimize either equalized odds or statistical parity. For these methods that center around modifying decision thresholds, it is not straightforward to extend to multi-class and multi-group as one will have to consider $\binom{|\mathcal{Y}|}{2} \cdot \binom{|\mathcal{S}|}{2}$ boundaries.

FST [WRC20, WRC21] tackles fairness intervention via minimizing cross-entropy for binary classes. Their method is inherently tailored to binary classification *and* only a cross-entropy objective function, and our `FairProjection`-CE reduces to FST for the case of CE and binary classification tasks. Identifying [JN20] is a method for minimizing KL-divergence for group-fairness intervention, which changes the label weights (via a convex combination) between unweighted and weighted samples, but it is not clear that this would navigate a good fairness-accuracy trade-off curve. Their method can be extended to non-binary prediction with non-binary groups by an appropriate choice of base classifier and fairness constraints, which is a non-trivial extension of the accompanying code, and we chose not to pursue this. Note that [JN20] and `FairProjection` solve the KL-divergence minimization in very different ways. In particular, the runtime of [WRC20, WRC21] on a 350k training dataset is

longer than 30 minutes using logistic regression as a base classifier (in comparison, the runtime of `FairProjection` for a 500k dataset is less than 1 minute). This is because they require reweighing the data and retraining a large number of times. Hence, it is inherently non-parallelizable.

### B.5.1 Fairness in Multi-Class Prediction

Methods that are based on optimization with a fairness regularizer often can be easily extended to multi-class prediction as it only requires a small change in the regularizer. For example, instead of using $|\text{FNR}_0(x) - \text{FNR}_1(x)|$, one can replace this with

$$\sum_{i \in \mathcal{Y}} \sum_{j \neq i \in \mathcal{Y}} |P(\widehat{Y} = j \mid Y = i, S = 0) - P(\widehat{Y} = j \mid Y = i, S = 1)|. \tag{120}$$

FERM [DOBD+18] mentions how their method can be extended to multi-class sensitive attribute. Similarly, we believe that their method can be used for multi-class labels as well. The reductions approach [ABD+18] assumes binary labels but is has natural extension to multi-class, which is explored in [YCK20]. In-processing methods proposed in [CHS20] and [ZLM18] allow for both multi-class labels and multi-class group attributes. [ZLM18] aims to achieve the independence between the sensitive attribute $S$ and $\widehat{Y}$ or $\widehat{Y}$ given $Y$ by training an adversary who tries to figure out $\widehat{S}$. [CHS20] directly estimates the fairness loss (e.g., 120) using kernel density estimation. They also demonstrate the empirical performance in a three-class classification using synthetic data. Another in-processing method is [AAV19] where the authors propose a way to incorporate multi-class fairness constraints into decision tree training. The preprocessing method suggested in [CKV20] is conceptually similar to our methods in that it aims to minimize the KL-divergence between the original distribution and preprcoessed distribution while satisfying fairness constraints. Their method, however, requires all feature vectors to be binary, and applies only to demographic parity or representation rate. There exist other notions of fairness, which is different from commonly-used group fairness metrics such as envy-freeness [BDNP19] or best-effort [KJW+21], which can be applied to multi-class prediction tasks.

Finally, there are unpublished works [DEHH21, YX20] that could handle multi-class classification. Specifically, [DEHH21] presents a post-processing method that selects different thresholds for each group to achieve demographic parity. [YX20] formulates SVM training as a mixed-integer program and integrates fairness regularizer in the objective, which can also deal with multi-class.

## C Datasheet for ENEM 2020 dataset

### Questions

The questions below are derived from [GMV+21] and aim to provide context about the ENEM-2020 dataset. We highlight that we did not create the dataset nor collect the data included in it. Instead, we simply provide a link to the ENEM-2020 data at [INE20]. At the time of writing, the ENEM-2020 dataset is open and made freely available by the Brazilian Government at [INE20] under a Creative Commons Attribution-NoDerivs 3.0 Unported License [Com]. We provide the datasheet below to clarify certain aspects of the dataset (e.g., motivation, composition, etc.) since the original information is available in Portuguese at [INE20], thus limiting its access to a broader audience. The website [INE20] contains a link to download a `.zip` file which contains the ENEM-2020 data in `.csv` format and extensive accompanying documentation.

The datasheet below is **not** a substitute for the explanatory files that are downloaded together with the dataset at [INE20], and we emphatically recommend the user to familiarize themselves with associated documentation prior to usage. We also strongly recommend the user to carefully read the "Leia-Me" (readme) file `Leia_Me_Enem_2020.pdf` available in the same `.zip` folder that contains the dataset. The answers in the datasheet below are based on an English translation of information available at [INE20] and may be incomplete or inaccurate. The datasheet below is based on our own independent analysis and in no way represents or attempts to represent the opinion or official position of the Brazilian Government and its agencies.

We also note that we do not distribute the ENEM-2020 dataset directly nor host the dataset ourselves. Instead, we provide a link to download the data from a public website hosted by the Brazilian Government. The dataset may become unavailable in case the link in [INE20] becomes inaccessible.

**Motivation**

- **For what purpose was the dataset created?** According to the "Leia-me" (Read Me) file that accompanies the data, the dataset was made available to fufill the mission of the Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) of developing and disseminating data about exams and evaluations of basic education in Brazil.
- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** The dataset was developed by INEP, which is a government agency connected to the Brazilian Ministry of Education.
- **Who funded the creation of the dataset?** The data is made freely available by the Brazilian Government.

**Composition**

- **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** The instances of the dataset are information about individual students who took the Exame Nacional do Ensino Médio (ENEM). The ENEM is the capstone exam for Brazilian students who are graduating or have graduated high school.
- **How many instances are there in total (of each type, if appropriate)?** The raw data provided in at [INE20] has approximately 5.78 million entries. The processed version we use in our experiments has approximately 1.4 million entries.
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** The data provided is the lowest level of aggregation of data collected from ENEM exam-takers made available by INEP.
- **What data does each instance consist of?** We provide a brief description of the features available in the raw public data provided at [INE20]. Upon downloading the data, a detailed description of features and their values are available (in Portuguese) in the file titled `Dicionário_Mircrodados_ENEM_2020.xsls`. The features include:
  - **Information about exam taker:** exam registration number (masked), year the exam was taken (2020), age range, sex, marriage status, race, nationality, status of high school graduation, year of high school graduation, type of high school (public, private, n/a), if they are a "treineiro" (i.e., taking the exam as practice).
  - **School data**: city and state of participant's school, school administration type (private, city, state, or federal), location (urban or rural), and school operation status.
  - **Location where exam was taken**: city and state.
  - **Data on multiple-choice questions**: The exam is divided in 4 parts (translated from Portuguese): natural sciences, human sciences, languages and codes, and mathematics. For each part there is data if the participant attended the corresponding portion of the exam, the type of exam book they received, their overall grade, answers to exam questions, and the answer sheet for the exam.
  - **Data on essay question**: if participant took the exam, grade on different evaluation criteria, and overall grade.
  - **Data on socio-economic questionnaire answers:** the data include answers to 25 socio-economic questions (e.g., number of people who live in your house, family average income, if the your house has a bathroom, etc.).
- **Is there a label or target associated with each instance?** No, there is no explicit label. In our fairness benchmarks, we use grades in various components of the exam as a predicted label.
- **Is any information missing from individual instances?** Yes, certain instances have missing values.
- **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** No explicit relationships identified.
- **Are there recommended data splits (e.g., training, development/validation, testing)?** No.

- **Are there any errors, sources of noise, or redundancies in the dataset?** The data contains missing values and, according to INEP, was collected from individual exam takers. The information is self-reported and collected at the time of the exam.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** Self-contained.
- **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)?** According to the *Leia-me* (readme) file (in Portuguese) that accompanies the dataset and our own inspection, the dataset does not contain any feature that allows direct identification of exam takers such as name, email, ID number, birth date, address, etc. The exam registration number has been substituted by a sequentially generated mask. INEP states that the released data is aligned with the Brazilian *Lei Geral de Proteção dos Dados* (LGPD, General Law for Data Protection). We emphatically recommend the user to view the Readme file prior to usage.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** The official terminology used by the Brazilian Government to denote race can be viewed as offensive. Specifically, the term used to describe the race of exam takers of Asian heritage is "Amarela," which is the Portuguese word for the color yellow. Moreover, the term "Pardo," which roughly translates to brown, is used to denote individuals of multiple or mixed ethnicity. This outdated and inappropriate terminology is still in official use by the Brazilian Government, including in its population census. The dataset itself includes integers to denote race, which are mapped to specific categories through the variable dictionary.
- **Does the dataset relate to people?** Yes.
- **Does the dataset identify any subpopulations (e.g., by age, gender)?** Yes. Information about age, sex, and race are included in the dataset.
- **Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** The *Leia-me* (readme) file notes that the individual exam-takers cannot be directly identified from the data. However, in the same file, INEP recognizes that the Brazilian data protection law (LGPD) does not clearly define what constitutes a reasonable effort of de-identification. Thus, INEP adopted a cautious approach: this dataset is a simplified/abbreviated version of the ENEM micro-data compared to prior releases and aims to remove any features that may allow identification of the exam-taker.
- **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** The data includes race information and socio-economic questionnaire answers.

**Collection Process**

Since we did not produce the data, we cannot speak directly about the collection process. Our understanding is that the data contains self-reported answers from exam-takers of the ENEM collected at the time of the exam. The exam was applied on 17 and 24 of January 2021 (delayed due to COVID). The data was aggregated and made publicly available by INEP at [INE20]. After consulting the IRB office at our institution, no specific IRB was required to use this data since it is anonymized and publicly available.

**Preprocessing/cleaning/labeling**

- **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** Some mild pre-processing was done on the data to ensure anonymity, as indicated in the "Leia-me" file. This includes aggregating participant ages, masking exam registration numbers, and removing additional information that could allow de-anonymization.

- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** The raw data is not publicly available.

**Uses**

- **Has the dataset been used for any tasks already?** We have used this dataset to benchmark fairness interventions in ML in the present paper. ENEM microdata has also been widely used in studies ranging from public policy in Brazil to item response theory in high school exams.
- **Are there tasks for which the dataset should not be used?** INEP does not clearly define tasks that should not be used on this dataset. However, no attempt should be made to de-anonymize the data.

**Distribution and Maintenance**

The ENEM-2020 dataset is open and made freely available by the Brazilian Government at [INE20] under a Creative Commons Attribution-NoDerivs 3.0 Unported License [Com] at the time of writing. The dataset may become unavailable in case the link in [INE20] becomes inaccessible.