Fair Supervised Learning with A Simple Random Sampler of Sensitive Attributes

Jinwon Sohn
Purdue University

Qifan Song
Purdue University

Guang Lin
Purdue University

Abstract

As the data-driven decision process becomes dominating for industrial applications, fairness-aware machine learning arouses great attention in various areas. This work proposes fairness penalties learned by neural networks with a simple random sampler of sensitive attributes for non-discriminatory supervised learning. In contrast to many existing works that critically rely on the discreteness of sensitive attributes and response variables, the proposed penalty is able to handle versatile formats of the sensitive attributes, so it is more extensively applicable in practice than many existing algorithms. This penalty enables us to build a computationally efficient group-level in-processing fairness-aware training framework. Empirical evidence shows that our framework enjoys better utility and fairness measures on popular benchmark data sets than competing methods. We also theoretically characterize estimation errors and loss of utility of the proposed neural-penalized risk minimization problem.

1 INTRODUCTION

Algorithmic fairness has been a growing research area as the prediction-based decision process becomes more and more prevalent. The legal examples include the US Equal Credit Opportunity Act, the European Union's General Data Protection Regulation, and the Fair Credit Reporting Act, to name a few. On the academic side, Mhasawade et al. (2021) discussed the importance and challenges of algorithmic fairness in public health. Kozodoi et al. (2022) studied various mitigation strategies for the credit scoring application. De-Arteaga et al. (2022) gave an overview of potential areas demanding fairness-aware business analytics with intriguing real-world examples such as dynamic pricing, distribution of vaccines, job applications, and so forth. Similar motivating examples can be easily found in other fields as well, such as education (Loukina et al., 2019), finance (Das et al., 2021), mortgage lending (Lee and Floridi, 2021), and computational medicine (Xu et al., 2022).

In pursuit of the rising demand for mitigating societal bias in decision-making processes, there have been versatile approaches, which are mainly categorized into pre-processing, in-processing, and post-processing (Barocas et al., 2017; Caton and Haas, 2020; Pagano et al., 2023; Xian et al., 2023). Pre-processing includes relabeling, reweighting, or resampling of data instances to ease possible discrimination of a learned model (Kamiran and Calders, 2011). Synthesizing data to be fair also belongs to this class as well (Xu et al., 2018; Sattigeri et al., 2019; van Breugel et al., 2021). In the field of in-processing, optimization with fairness constraints has been one of the main branches (Agarwal et al., 2018; Komiyama et al., 2018; Agarwal et al., 2019; Zafar et al., 2019; Scutari et al., 2022; Jung et al., 2023). Comparably, model-based fairness control has been also extensively studied, which regularizes the degree of discrimination through an auxiliary model (Xie et al., 2017; Beutel et al., 2017; Zhang et al., 2018; Adel et al., 2019; Mary et al., 2019; Lee et al., 2022). Following the model-agnostic spirit, post-processing directly modifies the model's outcome such that the reporting values are non-discriminatory while minimizing the loss of utility (Hardt et al., 2016; Pleiss et al., 2017; Zeng et al., 2022). Out of such mitigation categories, there have been versatile studies imposing fairness in clustering (Wang et al., 2023), feature selection (Quinzan et al., 2023), reinforcement learning (Deng et al., 2023), and so on.

There are three key notions for pursuing better algorithmic fairness: **independence**, **separation**, and **sufficiency** (Barocas et al., 2017; Caton and Haas,

2020; Pagano et al., 2023). Throughout this paper, we denote X as covariates, Y as a response, A as sensitive random variables to be protected (e.g., race or gender), and h as a scoring model that makes a data-driven decision. The independence condition requires that A is independent of $h(X)$, i.e., $A \perp h(X)$; statistical parity is an example of independence. Separation and sufficiency, on the other hand, are defined based on the conditional independence structure, i.e., $A \perp h(X)|Y$ and $A \perp Y|h(X)$ respectively. For instance, if two groups in a binary A have the same false-positive and false-negative errors in a binary classification problem, the circumstance achieves the error rate balance (Chouldechova, 2017) or equalized odds (Hardt et al., 2016), implying separation. For sufficiency, evaluating the calibration of the model across sensitive factors is a representative example (Chouldechova, 2017; Barocas et al., 2017).

Throughout this paper, let $P(X)$ denote the distribution law of a random variable X. For generic random variables X_1 and X_2, $P(X_1) = P(X_2)$ means that X_1 and X_2 have the same distribution, and $P(X_1|X_2 = x_2)$ implies the conditional distribution of X_1 given $X_2 = x_2$. If the two random variables satisfy $P(X_1, X_2) = P(X_1)P(X_2)$, we say X_1 and X_2 are independent of each other. On the other hand, $p(x)$ denotes the probability mass/density function of X, depending on the context.

In this work, we propose a model-based in-processing framework that flexibly captures and controls underlying societal discrimination realized through the output of h. Our contributions are as follows:

- First, we devise a fairness penalty by a neural network denoted as D that leverages a simple random sampler of sensitive variables for independence to measure the discrepancy between $P(h(X), A)$ and $P(h(X))P(A)$. The penalty can accommodate various scenarios even when A is a mix of continuous and discrete variables, in which many prior works fail as exhibited in Table 1.

- Second, the penalty is further extended to embrace separation while inheriting all the remarkable properties shown in the independence case.

- Third, we mathematically quantify the underlying mechanism of the proposed fairness-controlled supervised learning. Specifically, we derive the upper bounds of estimation error and loss of utility with the imposed penalty based on statistical learning theory.

2 RELATED WORKS

Density Matching Achieving fairness through density matching has been steadily implemented. Quadrianto and Sharmanska (2017) designed a privileged learning using a variant of a support vector machine on which a fairness constraint via maximum mean discrepancy is imposed. Cho et al. (2020) performed matching distributions between different sensitive groups via a differentiable kernel density estimation technique. Li et al. (2021) adopted the loss function used in the generative adversarial network (Goodfellow et al., 2014) as an extra penalty to facilitate the scoring distributions h conditional on different A values to be indistinguishable. These density-matching-based regularizations work well but are usually limited to the case when A and Y are discrete or binary. That is because their backbone optimization structures essentially depend on *sub-groups comparison*. As an example of independence for binary A, the loss function relies on two subsets of data, corresponding to $A = 1$ and $A = 0$ respectively, to measure the distributional difference between sub-groups (i.e., $P(h(X)|A = 1)$ vs $P(h(X)|A = 0)$). Obviously, this technique does not apply to continuous A. Romano et al. (2020) suggested using an extra generator that produces A conditionally on Y to ensure separation. Although this approach technically bypasses the sub-group comparison, it critically relies on the quality of the conditional generator. If A is complex, e.g., a mix of categorical and continuous variables, finding a good generator that captures the true distribution could be an extremely difficult task (see, e.g., Xu et al., 2019b; Kotelnikov et al., 2023).

HGR penalty and beyond To tackle such limitations in the literature, diverse approaches have been proposed. Mary et al. (2019) first utilized Hirschfeld-Gebelein-Rényi (HGR) maximal correlation which can capture the nonlinear correlation between A and $h(X)$ regardless of the variable type. Grari et al. (2020) and Lee et al. (2022) approximated the HGR maximal correlation via neural networks, to achieve independence and separation. These approaches conduct min-max optimization where the auxiliary neural networks approximate the HGR by maximization and the predictive model h is trained to minimize the approximated HGR. The auxiliary neural nets in the training process take variable A as the input instead of fitting A, hence the multivariate nature of A does not matter anymore. Besides, Du et al. (2021) recently proposed a way of neutralizing the hidden layer of a neural-net model, which can also handle multivariate A; Scutari et al. (2022) developed a fair (generalized) linear model that places the ridge penalty to adjust the violation of

discrimination for any kinds of A and Y.

3 METHODOLOGY

3.1 Penalty with Resampled A

Focusing on independence, this section describes the main idea which generalizes to any dimension or type of sensitive variables. The methodological novelty originates from the use of a simple random sampler of A to quantify the degree of fairness via a discriminative neural network D. Given a sufficient network capacity, the learned D can capture the violation of statistical independence, so it is used as a penalty (or adversarial) network for the risk-minimization problem of h such that the resulting h becomes fair.

To start with, let's formally define $X \in \mathcal{X} \subset \mathbb{R}^p$ and $A \in \mathcal{A} \subset \mathbb{R}^l$ as the multivariate non-sensitive and sensitive random variable respectively, and $Y \in \mathcal{Y} \subset \mathbb{R}$ as the univariate outcome variable. Given a scoring model $h : \mathcal{X} \to \mathcal{S} \subset \mathbb{R}$ and a loss function $L : \mathcal{Y} \times \mathcal{S} \to \mathbb{R}$, we denote the risk function as $R(h) = \mathbf{E}_{Y,X}[L(Y, h(X))]$ where $\mathbf{E}_{Y,X}$ means the expectation w.r.t. the joint distribution of (X,Y). Note h does not explicitly depend on A. This structure conceptually carries on fairness via blinding and does not input A in the inference phase (Quadrianto and Sharmanska, 2017; Zafar et al., 2019). In order to evaluate independence in general settings (i.e., beyond binary A and Y), we generalize the existing notion of statistical parity as follows.

Definition 1 (Generalized Statistical Parity (GSP)). *The scoring model h is called to satisfy the generalized statistical parity if $P(h(X)|A = a) = P(h(X))$ for all $a \in \mathcal{A}$.*

GSP is more general than the classical statistical parity (SP); if $\mathcal{A} = \{0, 1\}$ for the binary classification problem, GSP implies SP, i.e., $P(h(X) > \tau | A = 1) = P(h(X) > \tau | A = 0)$ for any classification threshold τ.

Now, we define the penalty that promotes the fairness of h. Inspired by the noise-contrastive loss (Gutmann and Hyvärinen, 2012; Goodfellow et al., 2014), with $D : \mathcal{S} \times \mathcal{A} \to (0, 1)$, let's define $R_F(h) = \sup_D R_F(h; D)$ where

$$R_F(h; D) = \mathbf{E}_{X,A}[\log D(h(X), A)] \\ + \mathbf{E}_{X,A'}[\log(1 - D(h(X), A'))],$$

and A' shares the same distribution with A but is independent of (X, A). We remark that A' can be easily obtained by applying the simple random sampling A' from $P(A)$. The detailed implementation for this marginal sampling is discussed in Supplementary 7.2. The following proposition theoretically validates that $R_F(h)$ captures the discrepancy between $P(h(X)|A)$ and $P(h(X))$.

Proposition 1. *Let p_A, $p_{h(X)}$, $p_{h(X)|A}$, and $p_{h(X),A}$ be the marginal densities of A and $h(X)$, the conditional density of $h(X)$ given A respectively, and the joint density of $h(X)$ and A. Denote $D^* = \arg\max_D R_F(h; D)$. Then, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,*

$$\frac{D^*(s,a)}{1 - D^*(s,a)} = \frac{p_{h(X),A}(s,a)}{p_{h(X)}(s)p_A(a)} = \frac{p_{h(X)|A}(s|a)}{p_{h(X)}(s)}.$$

The proof is shown in Supplementary 7.1. This proposition provides a theoretical justification for the use of $R_F(h; D)$ as a GSP controller. Following the argument in Theorem 1 of Goodfellow et al. (2014), we observe that $R_F(h)$ can be interpreted by the Jensen-Shannon divergence $J(\cdot, \cdot)$, i.e., $R_F(h; D^*) = 2J(P(h(X), A), P(h(X))P(A)) - 2\log 2$, and it implies $p_{h(X),A}(s,a) = p_{h(X)}(s)p_A(a)$ for all s and a if $J = 0$, which underpins that h accomplishes GSP at the minimum of $R_F(h)$. Thus, we can formulate a fairness-aware optimization problem for h by placing the extra penalty $R_F(h)$ to the (discriminatory) risk-minimization problem, i.e.,

$$\min_h R(h) + \lambda R_F(h), \tag{1}$$

where λ trades off between goodness-of-fit (utility) and the degree of GSP; as λ becomes larger, the solution model h becomes fairer (in terms of GSP) where D^* gets closer to 0.5 but meanwhile undergoes the loss of utility.

Finally, we employ an empirical min-max optimization structure since (1) is practically intractable. The population joint density of Y, X, and A is not available, and the optimal D^* is unknown in general so it requires maximization over $R_F(h; D)$. Let's denote by $\{(X_i, A_i, Y_i)\}_{i=1}^n$ the observed data set with a sample size n. We collect $\{A_i'\}_{i=1}^n$ by simply resampling $\{A_i\}_{i=1}^n$. This is an approximated but computationally efficient implementation for sampling A' from $P(A)$. Let's further denote by $\hat{R}(h) = \frac{1}{n} \sum_{i=1}^n L(Y_i, h(X_i))$ and $\hat{R}_F(h; D) = \frac{1}{n} \sum_{i=1}^n (\log D(h(X_i), A_i) + \log(1 - D(h(X_i), A_i')))$ the unbiased estimators of R and R_F respectively. Then, the empirical version of (1) is

$$\min_h \max_D \hat{R}(h) + \lambda \hat{R}_F(h; D), \tag{2}$$

where we use neural networks to model both h and D. The optimization problem (2) can be solved via an alternative min-max strategy (Supplementary 7.2). For every iteration, D is first trained to capture the degree of discrimination against GSP. The model h is then trained to minimize the risk plus the fairness penalty evaluated by D, which in turn hinders D from identifying the presence of discrimination in the next

Table 1: Summary of applicability of in-processing methods: "NN" denotes whether or not neural network models are supported. "Ind." and "Sep." imply independence and separation. Abbreviations "D.", "C.", and "M." imply discrete, continuous, and mixed (continuous and discrete) type; Abbreviations "A" and "Y" mean sensitive and outcome variables. "$\triangle$" means the framework does not explicitly control specific fairness metrics. "$\square$" means the method is technically applicable but requires a non-trivial modification. "B." implies the framework only supports a binary type variable.

Methods	NN	Ind.	Sep.	D. A	C. A	M. A	D. Y	C. Y
Edwards and Storkey (2016)	✓	✓	✗	B	✗	✗	✓	✗
Agarwal et al. (2018)	✓	✓	✓	✓	✗	✗	✓	✗
Adel et al. (2019)	✓	✓	✓	B	✗	✗	✓	✗
Cho et al. (2020)	✓	✓	✓	✓	✗	✗	✓	✗
Romano et al. (2020)	✓	✗	✓	✓	✓	$\square$	✓	✓
Li et al. (2021)	✓	✓	✓	B	✗	✗	✓	✗
Du et al. (2021)	✓	$\triangle$	$\triangle$	✓	✓	✓	✓	✗
Scutari et al. (2022)	✗	✓	✓	✓	✓	✓	✓	✓
Lee et al. (2022)	✓	✓	✓	✓	✓	✓	✓	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓

iteration. By repeating this adversarial game between the two networks D and h, the model h settles down to an equilibrium, determined by λ, between utility and fairness. We name $\hat{R}_F(h; D)$ as *simple random sampling (SRS)-based penalty (SBP)*.

Although this work focuses on the Jensen-Shannon divergence, our main idea does readily apply to other metrics. For instance, a SBP penalty utilizing the Kantorovich-Rubinstein duality of the 1-Wasserstein distance can be devised as

$$R_F(h; C) = \mathbf{E}_{X,A}[C(h(X), A)] - \mathbf{E}_{X,A'}[C(h(X), A')],$$

where C belongs to the class of 1-Lipschitz functions (Arjovsky et al., 2017). In the same way, it is straightforward to see that other popular f-divergence metrics also have the specific formulation of R_F (Nowozin et al., 2016).

This penalty is not only adequately flexible to capture nonlinear discriminatory dependency between $h(X)$ and A but also differentiable while most fairness criteria are usually non-differentiable in general (Zafar et al., 2019; Cotter et al., 2019). Noteworthily, *this penalty requires neither partitioning data sets into sub-groups nor predicting sensitive attributes* in contrast to a lot of previous works (Edwards and Storkey, 2016; Xie et al., 2017; Beutel et al., 2017; Zhang et al., 2018; Adel et al., 2019; Zhao et al., 2020; Li et al., 2021; Cho et al., 2020; Du et al., 2021). These properties greatly expand the applicability of the proposed method to numerous real-world problems having a mix of continuous and discrete sensitive attributes. We also remark that the previous sampler-based work of Romano et al. (2020) does not apply to GSP.

3.2 Extension for Separation

We also devise a penalty for separation that inherits all the advantages discussed in Section 3.1. To begin with, we define the generalized equalized odds (GEO) which is an intuitive generalization of equalized odds (Hardt et al., 2016) beyond binary outcomes.

Definition 2 (Generalized Equalized Odds (GEO)). *The scoring model h is called to satisfy the generalized equalized odds if $P(h(X)|A = a, Y = y) = P(h(X)|Y = y)$ for all $a \in \mathcal{A}$ and $y \in \mathcal{Y}$.*

To control GEO, we suggest specifying $R_F(h; D) =$

$$\mathbf{E}_{X,A,Y}[\log D(h(X), A, Y)]$$
$$+ \mathbf{E}_{A'}\mathbf{E}_{X,Y}[\beta(A', Y)\log(1 - D(h(X), A', Y))],$$

with some function $\beta : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}^+$ and $D : \mathcal{S} \times \mathcal{A} \times \mathcal{Y} \to (0, 1)$, where A' is statistically independent of (X, A, Y) but A and A' have the same distribution. The proposed penalty accompanies the next proposition.

Proposition 2. *Let $p_{h(X)|Y}$ be the conditional density function of $h(X)$ given Y and $p_{h(X)|A,Y}$ be of given Y and A. For $R_F(h; D)$, if $D^* = \underset{D}{\arg\max} \ R_F(h; D)$, then $D^*(s, a, y; \beta) =$*

$$\frac{p_{h(X)|A,Y}(s|a, y)}{p_{h(X)|A,Y}(s|a, y) + \beta(a, y)p_{h(X)|Y}(s|y)\frac{p_{A',Y}(a,y)}{p_{A,Y}(a,y)}},$$

for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $y \in \mathcal{Y}$, where $p_{A',Y}$ and $p_{A,Y}$ be the joint density functions of A' and Y and of A and Y respectively.

The proof appears in Supplementary 8.1. If β cancels out the density ratio between $p_{A',Y}$ and $p_{A,Y}$

(i.e., $\beta(a,y)p_{A',Y}(a,y)/p_{A,Y}(a,y) = 1$), then by the same argument in Proposition 1, $R_F(h; D^*)$ is minimized when h satisfies $P(h(X)|A,Y) = P(h(X)|Y)$, i.e., the penalty $R_F(h; D^*)$ promotes GEO. Thus, we fit a density-ratio estimator $\hat{\beta}$ by maximizing $R_\beta(D_\beta) =$

$$\mathbf{E}_{A,Y}[\log D_\beta(A,Y)] + \mathbf{E}_{A',Y}[\log(1 - D_\beta(A', Y))],$$

where $D_\beta : \mathcal{A} \times \mathcal{Y} \to (0,1)$ is modeled by a neural network. Then $D_\beta^* = \arg_{D_\beta} \max R_\beta(D_\beta)$ satisfies $D_\beta^*(a,y)/(1 - D_\beta^*(a,y)) = p_{A,Y}(a,y)/p_{A',Y}(a,y)$. Given an empirical minimizer $\hat{D}_\beta$ of $R_\beta(D_\beta)$, let $\hat{\beta}(a,y) = \hat{D}_\beta(a,y)/(1 - \hat{D}_\beta(a,y))$, and then by Proposition 2, $D^*(s,a,y; \hat{\beta})$ properly regularizes GEO. The numerical algorithm to expedite GEO appears in Supplementary 8.2 on the basis of the expression (2). The algorithm needs a pre-training phase to estimate $\hat{\beta}$ that is leveraged as the adaptive weights for the evaluation of the penalty $R_F(h; D)$. We observe that $\hat{\beta}$ performs a powerful density-ratio estimation on toy examples in Supplementary 8.3. It is worth mentioning that when A and Y are both discrete, $\hat{\beta}$ can be easily found without neural network training, e.g., using the empirical probability mass functions.

Proposition 2 clearly shows the fundamental differences of our simple random sampler approach compared to the conditional sampler approach (Romano et al., 2020), i.e., $A' \sim P(A|Y)$. In our case, we additionally employ β to exactly match $P(h(X)|A,Y) = P(h(X)|Y)$ in the ideal separation. In contrast, the prior work does not need β but has to obtain the conditional sampler $A' = G(\epsilon, Y)$ for some generative model G and random noise ϵ. There may be room for debate, but we believe that estimating β is generally more straightforward than training G. The former is equivalent to a DNN binary classification problem, which in modern practice is hardly impacted by the dimension or characteristics of A for tabular data sets. In contrast, the latter may necessitate the use of advanced generative models, such as generative adversarial networks (GAN) or diffusion models which are well-known for their notorious training difficulty (Xu et al., 2019b; Kotelnikov et al., 2023), when dealing with tabular-type attributes of A. Moreover, even a successfully trained generative model may only learn the support rather than the shape of a distribution (Arora et al., 2018).

Remark 1. *Although our approach necessitates β for the theoretical justification, we find that the performance of our simple random sampler-based method is fairly robust to a poor β estimation. Refer to Section 5 and Supplementary 10.4 for more details.*

Finally, it is worth mentioning that this penalty can also be used to obtain fair representation (Zhao et al., 2020; Du et al., 2021). Let's denote by $E : \mathcal{X} \to \mathcal{E}$ an encoder

which precedes $h_E : \mathcal{E} \to \mathcal{S}$ with the risk $\hat{R}(h_E \circ E)$. Then solving $\min_{h_E, E} \max_D \hat{R}(h_E \circ E) + \lambda \hat{R}_F(E; D)$ brings a fair encoder E for either independence or separation with $D : \mathcal{E} \times \mathcal{A} \to (0,1)$ or $D : \mathcal{E} \times \mathcal{A} \times \mathcal{Y} \to (0,1)$ respectively.

4 THEORY

In this section, we characterize the estimation error and the loss of utility of the proposed fairness-aware optimization scheme. For the simplicity of presentation, the analysis focuses on the solution of (2) with the GSP penalty. Our analysis borrows some proof techniques in the literature of generative adversarial modeling (Ji et al., 2021). For readers who are interested in the details of the proof, please refer to Supplementary 9.

Let's suppose $\mathcal{X} = \{x : ||x|| \leq B, x \in \mathbb{R}^p\}$, $\mathcal{Y} = [0,1]$, and $\mathcal{A} = [0,1]^l$ where $||\cdot||$ denotes the Euclidean norm. For mathematical convenience, we set $D(\cdot) = \sigma(f(\cdot))$ with $\sigma(x) = (1 + \exp(-x))^{-1}$ and consider the fully-connected neural networks: $f(x,a) = f_\mathbf{w}(x,a) = w_d^\top \kappa_{d-1}(W_{d-1}\kappa_{d-2}(\cdots W_1[x^\top, a^\top]^\top))$ and $h(x) = h_\mathbf{v}(x) = v_g^\top \psi_{g-1}(V_{g-1}\psi_{g-2}(\cdots V_1 x))$ where $\mathbf{w} = (W_1, \ldots, W_{d-1}, w_d) \in \mathbf{W}$ and $\mathbf{v} = (V_1, \ldots, V_{g-1}, v_g) \in \mathbf{V}$. We denote by $\mathcal{F}$ and $\mathcal{H}$ the function classes of f and h. Also, it is assumed that $\mathbf{W} = \bigotimes_{i=1}^{d-1}\{W_i \in \mathbb{R}^{p_{i+1} \times p_i} : ||W_i||_F \leq M_w(i)\}\bigotimes\{w_d \in \mathbb{R}^{p_d \times 1} : ||w_d|| \leq M_w(d)\}$ and $\mathbf{V} = \bigotimes_{i=1}^{g-1}\{V_i \in \mathbb{R}^{q_{i+1} \times q_i} : ||V_i||_F \leq M_v(i)\}\bigotimes\{v_g \in \mathbb{R}^{q_g \times 1} : ||v_g|| \leq M_v(g)\}$ with constants $M_w(\cdot)$ and $M_v(\cdot)$, $p_1 = p + l$, $q_1 = p$, and the Frobenius norm $||\cdot||_F$. This also induces the class of D by functional composition, defined as $\mathcal{D} = \{f_\mathbf{w}(h_\mathbf{v}(x), a) : \mathbf{w} \in \mathbf{W}, \mathbf{v} \in \mathbf{V}\}$.

We further assume that the activation functions ψ_u and κ_t are $K_\psi(u)$ and $K_\kappa(t)$-Lipschitz for all $t = 1, \ldots, d-1$ and $u = 1, \ldots, g-1$. ReLU and Sigmoid are examples of the 1-Lipschitz functions. These assumptions hint $1 > \gamma_1 \geq \sigma(h_\mathbf{v}(x)) \geq \gamma_0 > 0$ and $1 > \nu_1 \geq \sigma(f_\mathbf{w}(x,a)) \geq \nu_0 > 0$ for all x and a where the upper and the lower limits $(\gamma_1, \gamma_0, \nu_1, \nu_0)$ depend on the assumed bounds from the parameter spaces (i.e., $M_w(\cdot)$ and $M_v(\cdot)$) and the Lipschitz constants of the activation functions.

Now, let's denote $d(h; \lambda) = R(h) + \lambda R_F(h)$ and $\hat{d}(h; \lambda) = \hat{R}(h) + \lambda \hat{R}_F(h)$ as the population-level target function and its empirical version respectively. The estimation error then can be characterized by evaluating the empirical solution on the population objective based on the Rademacher complexity.

Definition 3 (Rademacher Complexity)**.** *Let $\mathcal{H}$ be the function class of h. Denote by $X_1, \ldots, X_n$ random samples that are independent and identically distributed (i.i.d.) to P_X. Then the Rademacher complexity $\mathcal{R}(\mathcal{H})$*

is defined as

$$\mathcal{R}(\mathcal{H}) = \mathbf{E}_{X,\epsilon} \left[\sup_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i h(X_i) \right| \right],$$

where $\epsilon_1, \ldots, \epsilon_n \sim \text{Unif}\{-1, 1\}$ i.i.d.

The Rademacher complexity for other function classes is defined in the same fashion. Theorem 1 holds for either the cross-entropy loss with $Y \in \{0, 1\}$ or the mean absolute error for $Y \in [0, 1]$.

Theorem 1 (Estimation Error). *Let $\hat{h}^* = \arg_{h \in \mathcal{H}} \min \hat{d}(h; \lambda)$ and define $\mathcal{L} = \{L(y, h_{\mathbf{v}}(x)) : \mathbf{v} \in V\}$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then, for any given $\lambda \geq 0$, an upper bound of the estimation error is $|d(\hat{h}^*; \lambda) - \inf_{h \in \mathcal{H}} d(h; \lambda)| \leq$*

$$\underbrace{4\mathcal{R}(\mathcal{L}) + 2F_{\mathbf{V}, \psi, B, \gamma_0, \gamma_1} \sqrt{\frac{\log(1/\delta)}{2n}}}_{\text{from } \hat{R}(h)}$$

$$+ \underbrace{2\lambda \left(F_{\nu_0, \nu_1} \mathcal{R}(\mathcal{D}) + F_{\mathbf{W}, \mathbf{V}, B, \kappa, \psi, l, \nu_0, \nu_1} \sqrt{\frac{\log(1/\delta)}{2n}} \right)}_{\text{from } \hat{R}_F(h)},$$

with the probability $1 - 3\delta$ where F_{ν_0, ν_1}, $F_{\mathbf{V}, \psi, B, \gamma_0, \gamma_1}$, and $F_{\mathbf{W}, \mathbf{V}, B, \kappa, \psi, l, \nu_0, \nu_1}$ are constants depending on the architectures of neural networks $h \in \mathcal{H}$ and $f \in \mathcal{F}$ whose exact values can be found in Supplementary 9.

Theorem 1 shows how the complexity of the neural network function classes affects the estimation accuracy through the Rademacher complexity and F. constants. As sample size n increases (while other settings are fixed), the Rademacher complexity generally decreases to zero, leading to consistency according to Theorem 1. The hyperparameter λ controls the balance between utility and fairness. Larger λ enforces better fairness at the expense of possible utility loss; the next corollary studies this utility loss.

Corollary 1 (Loss of Utility). *Define $\hat{h}^* = \arg_{h \in \mathcal{H}} \min \hat{d}(h; \lambda)$, $h_0^* = \arg_{h \in \mathcal{H}} \min d(h; \lambda = 0)$, and $h^* = \arg_{h \in \mathcal{H}} \min d(h; \lambda)$. Let's denote $\Delta(h_0^*, h^*) = R_F(h_0^*) - R_F(h^*)$. Then the loss of utility has an upper bound $|d(\hat{h}^*; \lambda = 0) - d(h_0^*; \lambda = 0)| \leq$*

$$4\mathcal{R}(\mathcal{L}) + 2F_{\mathbf{V}, \psi, B, \gamma_0, \gamma_1} \sqrt{\frac{\log(1/\delta)}{2n}} + \lambda \Delta(h_0^*, h^*),$$

with 1-δ probability for any $\lambda \geq 0$.

The loss of utility can take place from two main sources. Basically, the loss of utility could stem from sampling errors, which unfolds the first two terms in the upper bound of Corollary 1. Secondly, if the true h_0^*, which

is of the best utility, is exposed to huge discrimination against A in the population, i.e., h statistically strongly depends on A, then the penalty could lead to tremendous loss of utility in an effort to remove the dependency w.r.t. A (i.e., the term $\lambda \Delta(h_0^*, h^*)$).

5 SIMULATION

The performance of supervised learning with the proposed penalties is verified in the following three scenarios: (I) discrete outcome and sensitive attribute, (II) discrete outcome and mixed sensitive attributes, and (III) continuous outcome and sensitive attribute. The main text only delivers results about GSP in Scenario I and GEO in Scenario II due to the page limit. Results about GEO in Scenario I, GSP in Scenario II, Scenario III, and more additional studies (e.g., fair representation) are presented in Supplementary Section 10.

We compare the performance of our model (SRS-based penalty, SBP) against several newly proposed competing methods: HGR (Lee et al., 2022) that uses the neural-net approximation to calculate the soft HGR, the feature neutralization (Du et al. (2021), NEU) that interpolates feature points in a hidden layer such that the mapped space is independent of sensitive attributes, the kernel density estimation (Cho et al. (2020), KDE) that employs Q-function to approximate a distribution function, and CON (Romano et al., 2020) that uses the learned conditional sampler of A given Y. All the above in-processing models have the trade-off parameter λ to the fairness penalty L_F for fairness on top of the main loss term L_M. As a means of matching the scale of λ for the different models, all of them are tested under the formula $(1 - \lambda)L_M + \lambda L_F$ with 5 replicated experiments for various $0 < \lambda < 1$. All methods learn a neural network h with 3 hidden layers with 64 nodes. For more details about simulation setups, please refer to Supplementary Section 10.1.

Three benchmark data sets are considered for Scenario I: **Adult Data**[1] where Y is whether or not the annual income is greater than \$50K and A is whether an individual is white or non-white; **Law School Admission Data** [2] where Y is whether or not an applicant receives admission and A whether an individual is white or non-white; and **Credit Card Default Data**[1] where Y is whether or not a customer declares default and A is the gender. We refer to the work (Cho et al., 2020) to specify the sensitive attributes for analysis. Each data set is split by 80% and 20% for training and validation during the training course.

For the evaluation of fairness in Scenario I, statistical

---

[1]https://archive.ics.uci.edu/ml/datasets/
[2]http://www.seaphe.org/databases.php

parity (SP) is assessed by calculating

$$SP = \left| \mathbf{E}(\hat{Y} = 1 | A = 1) / \mathbf{E}(\hat{Y} = 1 | A = 0) - 1 \right|,$$

where $\mathcal{Y} = \{0, 1\}$, $\mathcal{A} = \{0, 1\}$, and $\hat{Y} = I(\hat{h}^*(X) > \tau)$ with the threshold τ maximizing the area under ROC curve (AUC) on the validation set. The AUC value is also used as a measure of utility. We use the Kolmogorov-Smirnov statistics (KS) to assess GSP, which is calculated as

$$\text{KS-GSP} = \sum_{a \in \mathcal{A}} \max_{h_x} |\hat{P}(\hat{h}^*(X) \leq h_x | A = a)$$
$$- \hat{P}(\hat{h}^*(X) \leq h_x)|,$$

where $\hat{P}$ denotes the empirical distribution.

Since the model intrinsically has multi-objectives (utility and fairness) in nature, the trained models at all iterations are evaluated over the validation data. Then, we plot the Pareto frontier curve of (AUC versus SP/KS) for each independent run. The Pareto frontier (Emmerich and Deutz, 2018) is a set of solutions that are not dominated by other pairs. For instance, a pair of AUC and SP, e.g., $(0.5, 0.5)$, is not dominated by $(0.4, 0.4)$ but by $(0.6, 0.4)$. This analysis helps design fair comparison studies without designing different or non-comparable early stopping strategies for the competitors (for example, NEU tends to degenerate to the trivial solution, which is perfectly fair but has no utility, for long-run iterations, so comparing the last iterates among the competing methods is not fair).

Figure 1 clearly illustrates the remarkable advantages of using SBP over the competitors. The closer the set of points is to the upper left corner of the figure, the higher performance is verified. We observe that SBP provides more consistent and better solutions in the sense that the Pareto frontiers are much more tightly gathered along the trade-off path than other competing methods. Note CON is not available for SP. Table 2 reports the fairness scores (SP/KS) of the Pareto solutions whose AUC is above the thresholds. The table succinctly demonstrates our superiority which is consistent with Figure 1. Those thresholds are chosen such that all methods yield a sufficient number of candidate SP/KS scores by referring to Figure 1. More tables with different thresholds appear in Supplementary 10.2.2.

For Scenario II, we use an additional data set **ACSEmployment**[3] from California in 2018 where Y is whether or not an individual is employed and A is a vector of age (continuous) and gender (discrete). We choose age (continuous) and race (discrete) for Adult and also age (continuous) and gender (discrete) for Credit Card

---

[3]https://github.com/socialfoundations/folktables

Default as sensitive attributes respectively while the output variables are the same as in Scenario I. For CON, we devise a conditional GAN model (Mirza and Osindero, 2014) because the trivial estimation method for $P(A|Y)$ in the original work (Romano et al., 2020) is not directly applicable when A is a mix of discrete and categorical variable. For a fair comparison, we use the same network structure for β in SBP and the discriminator for CON. To measure fairness for discrete attributes, we define

$$\text{EO} = \sum_{y \in \mathcal{Y}} \left| \frac{\mathbf{E}(\hat{Y} = 1 | A = 1, Y = y)}{\mathbf{E}(\hat{Y} = 1 | A = 0, Y = y)} - 1 \right|,$$

and KS-GEO =

$$\sum_{y \in \mathcal{Y}, a \in \mathcal{A}} \max_{h_x} |\hat{P}(\hat{h}^*(X) \leq h_x | A = a, Y = y)$$
$$- \hat{P}(\hat{h}^*(X) \leq h_x | Y = y)|.$$

We similarly define the fairness measures for continuous attributes; please refer to Supplementary 10.2 to see the formal definition.

Table 3 contrasts all models except KDE on the three data sets. Note KDE can not be implemented for continuous sensitive attributes. The thresholds are chosen based on Figure 4 in Supplementary 10.2. As is consistent with the table, the figure illustrates that SBP tends to outperform the competitors for both sensitive variables in general. For the computing time, ours is generally more efficient as shown in Table 4.

To intuitively explain that SBP outperforms others, we comment that HGR involves more burdensome approximation because it has to estimate four auxiliary neural networks in the adversarial training procedure, leading to larger estimation errors and computing time; NEU, on the other hand, does not explicitly control fairness metrics in its training process, leading to uneven performance. CON is significantly defeated by SBP mainly because estimating a good conditional generator $A|Y$ via the GAN approach is more difficult than estimating a good $\hat{\beta}$. On the other hand, we observe that SBP for separation is robust against a poor estimation of $\hat{\beta}$ (details in Supplementary 10.4).

In spite of such remarkable performance, an unavoidable issue of the proposed penalty remains regarding the adversarial game structure between h and D which is known to be unstable in general. As remedies to this obstacle, users can harness popular stabilization techniques used in the GAN literature such as penalizing the gradient of D (Zhou et al., 2019) or adding normalization layers (Miyato et al., 2018). It is also practical for the users to save the snapshots of h being learned over the Pareto frontier and deploy the final model having the desired Pareto solution for the purpose.
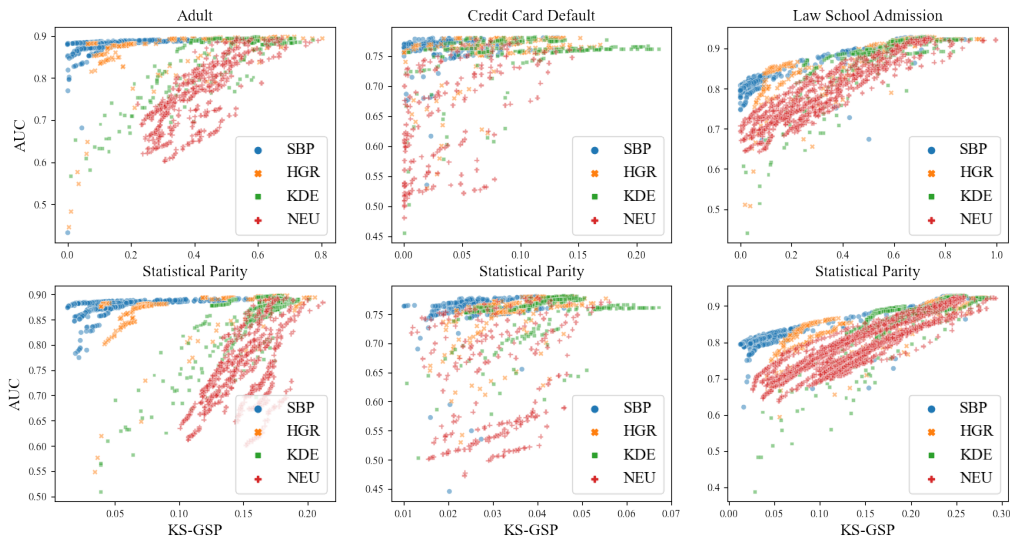
Figure 1: (Scenario I) Pareto frontiers: the first row includes pairs of SP and AUC, and the second row shows pairs of KS-GSP and AUC from 5 experiments for each λ. SBP (ours) tends to be more tightly in the upper-left corner than the competitors.

Table 2: Averages of the 5 smallest SP/KS-GSPs whose AUCs are greater than the thresholds. Those scores are selected in the Pareto solutions appearing in Figure 1. Standard deviations are in the parentheses next to the averages. All values are rounded to the third decimal place.

Model	**Adult** (AUC $\geq$ 0.85)		**Cred. Card.** (AUC $\geq$ 0.75)		**Law School.** (AUC $\geq$ 0.80)	
	SP ($\downarrow$)	KS-GSP ($\downarrow$)	SP ($\downarrow$)	KS-GSP ($\downarrow$)	SP ($\downarrow$)	KS-GSP ($\downarrow$)
SBP	0.001 ($\approx$ 0)	0.014 ($\approx$ 0)	$\approx$ 0 ($\approx$ 0)	0.011 (0.001)	0.004 (0.003)	0.018 (0.001)
HGR	0.070 (0.005)	0.040 ($\approx$ 0)	0.001 (0.001)	0.023 (0.001)	0.096 (0.003)	0.064 (0.002)
KDE	0.355 (0.010)	0.124 (0.002)	0.013 (0.009)	0.024 ($\approx$ 0)	0.239 (0.013)	0.132 (0.003)
NEU	0.446 (0.029)	0.154 (0.003)	0.040 (0.010)	0.020 (0.003)	0.214 (0.017)	0.098 (0.005)

Table 3: Averages of the 5 smallest EO/KS-GEOs whose AUCs are greater than the thresholds. Those scores are selected by referring to Figure 4 in Supplementary 10.2. Standard deviations are in the parentheses.

Metric	Model	**Adult** (AUC $\geq$ 0.80)		**Cred. Card.** (AUC $\geq$ 0.75)		**ACSEmpl.** (AUC $\geq$ 0.75)	
		Race ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)
EO	SBP	0.047 (0.004)	0.078 (0.002)	0.001 ($\approx$ 0)	0.016 (0.001)	0.066 (0.004)	0.209 (0.002)
	CON	0.358 (0.029)	0.162 (0.027)	0.006 (0.003)	0.016 (0.004)	0.155 (0.006)	0.316 (0.011)
	HGR	0.273 (0.073)	0.307 (0.002)	0.002 (0.001)	0.033 (0.001)	0.193 (0.005)	0.447 (0.002)
	NEU	0.292 (0.019)	0.201 (0.007)	0.027 (0.007)	0.055 (0.005)	0.099 (0.003)	0.276 (0.006)
KS-GEO	SBP	0.098 (0.003)	0.120 (0.001)	0.040 (0.001)	0.036 (0.001)	0.071 (0.001)	0.167 (0.002)
	CON	0.177 (0.001)	0.094 (0.007)	0.049 (0.001)	0.038 ($\approx$ 0)	0.100 (0.002)	0.207 (0.003)
	HGR	0.160 (0.003)	0.208 (0.008)	0.041 (0.003)	0.047 (0.001)	0.122 (0.001)	0.320 (0.004)
	NEU	0.156 (0.005)	0.130 (0.001)	0.063 (0.004)	0.054 (0.001)	0.074 (0.001)	0.205 (0.002)

6 CONCLUSION

The proposed fair-ML framework that employs the simple random sampler is universally applicable to clas-sification/regression problems across various fairness criteria, without worrying whether sensitive attributes are continuous or discrete, or even a mix of them. This methodological versatility is of great importance as

Table 4: Training times (mins) for the first 1000 iterations on A30 GPU for separation in **Adult**, including/excluding the pre-training time. Note that for Scenario II, CON, NEU, and SBP need a pre-training (additional NN models) course but KDE and HGR do not. Refer to Supplementary 10.1 for more details.

	Without pre-training		With pre-training
Model	Scenario I	Scenario II	Scenario II
SBP	0.22 (0.02)	0.59 (0.05)	0.80 (0.07)
CON	0.21 (0.03)	0.40 (0.03)	1.45 (0.09)
HGR	0.47 (0.08)	0.77 (0.10)	-
KDE	1.31 (0.09)	-	-
NEU	1.01 (0.09)	1.46 (0.03)	3.61 (0.09)

multifarious communities in society pay more attention to the unbiased data-driven decision process while having more and more diverse variables with distinctive characteristics to be protected together (Loukina et al., 2019; Lee and Floridi, 2021; Kozodoi et al., 2022; De-Arteaga et al., 2022). In future research, the fundamental concept of unconditionally sampling sensitive attributes will be a promising tool for promoting fairness for other related works that heavily depend on the discreteness, mixed or even high-dimensional sensitive variables, such as synthesizing a fair data set (Xu et al., 2019a) or fair uncertainty quantification (Lu et al., 2022).

Acknowledgements

References

Adel, T., Valera, I., Ghahramani, Z., and Weller, A. (2019). One-network adversarial fairness. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2412–2420. AAAI Press.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. M. (2018). A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR.

Agarwal, A., Dudík, M., and Wu, Z. S. (2019). Fair regression: Quantitative definitions and reduction-based algorithms. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 120–129. PMLR.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan.

Arora, S., Risteski, A., and Zhang, Y. (2018). Do gans learn the distribution? some theory and empirics. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Barocas, S., Hardt, M., and Narayanan, A. (2017). Fairness in machine learning. *Nips tutorial*, 1:2017.

Beutel, A., Chen, J., Zhao, Z., and Chi, E. H. (2017). Data decisions and theoretical implications when adversarially learning fair representations. *CoRR*, abs/1707.00075.

Caton, S. and Haas, C. (2020). Fairness in machine learning: A survey. *CoRR*, abs/2010.04053.

Cho, J., Hwang, G., and Suh, C. (2020). A fair classifier using kernel density estimation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Cotter, A., Jiang, H., Gupta, M. R., Wang, S. L., Narayan, T., You, S., and Sridharan, K. (2019). Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.*, 20:172:1–172:59.

Das, S., Donini, M., Gelman, J., Haas, K., Hardt, M., Katzman, J., Kenthapadi, K., Larroy, P., Yilmaz, P., and Zafar, M. B. (2021). Fairness measures for machine learning in finance. *The Journal of Financial Data Science*.

De-Arteaga, M., Feuerriegel, S., and Saar-Tsechansky, M. (2022). Algorithmic fairness in business analytics: Directions for research and practice. *CoRR*, abs/2207.10991.

Deng, Z., Sun, H., Wu, S., Zhang, L., and Parkes, D. C. (2023). Reinforcement learning with stepwise

fairness constraints. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 10594–10618. PMLR.

Du, M., Mukherjee, S., Wang, G., Tang, R., Awadallah, A. H., and Hu, X. B. (2021). Fairness via representation neutralization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12091–12103.

Edwards, H. and Storkey, A. J. (2016). Censoring representations with an adversary. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Emmerich, M. T. M. and Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat. Comput.*, 17(3):585–609.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Grari, V., Lamprier, S., and Detyniecki, M. (2020). Fairness-aware neural rényi minimization for continuous features. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2262–2268. ijcai.org.

Gutmann, M. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13:307–361.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323.

Ji, K., Zhou, Y., and Liang, Y. (2021). Understanding estimation and generalization error of generative adversarial networks. *IEEE Trans. Inf. Theory*, 67(5):3114–3129.

Jung, S., Park, T., Chun, S., and Moon, T. (2023). Re-weighting based group fairness regularization via classwise robust optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Kamiran, F. and Calders, T. (2011). Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33.

Komiyama, J., Takeda, A., Honda, J., and Shimao, H. (2018). Nonconvex optimization for regression with fairness constraints. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2742–2751. PMLR.

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. (2023). Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17564–17579. PMLR.

Kozodoi, N., Jacob, J., and Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *Eur. J. Oper. Res.*, 297(3):1083–1094.

Lee, J. K., Bu, Y., Sattigeri, P., Panda, R., Wornell, G. W., Karlinsky, L., and Feris, R. (2022). A maximal correlation approach to imposing fairness in machine learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 3523–3527. IEEE.

Lee, M. S. A. and Floridi, L. (2021). Algorithmic fairness in mortgage lending: from absolute conditions to relational trade-offs. *Minds Mach.*, 31(1):165–191.

Li, X., Cui, Z., Wu, Y., Gu, L., and Harada, T. (2021). Estimating and improving fairness with adversarial learning. *CoRR*, abs/2103.04243.

Loukina, A., Madnani, N., and Zechner, K. (2019). The many dimensions of algorithmic fairness in educational applications. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, pages 1–10. Association for Computational Linguistics.

Lu, C., Lemay, A., Chang, K., Höbel, K., and Kalpathy-Cramer, J. (2022). Fair conformal predictors for applications in medical imaging. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 12008–12016. AAAI Press.

Mary, J., Calauzènes, C., and Karoui, N. E. (2019). Fairness-aware learning for continuous attributes and treatments. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97

of *Proceedings of Machine Learning Research*, pages 4382–4391. PMLR.

Mhasawade, V., Zhao, Y., and Chunara, R. (2021). Machine learning and algorithmic fairness in public and population health. *Nat. Mach. Intell.*, 3(8):659–666.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 271–279.

Pagano, T. P., Loureiro, R. B., Lisboa, F. V., Peixoto, R. M., Guimarães, G. A., Cruz, G. O., Araujo, M. M., Santos, L. L., Cruz, M. A., Oliveira, E. L., et al. (2023). Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big data and cognitive computing*, 7(1):15.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J. M., and Weinberger, K. Q. (2017). On fairness and calibration. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5680–5689.

Quadrianto, N. and Sharmanska, V. (2017). Recycling privileged learning and distribution matching for fairness. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 677–688.

Quinzan, F., Khanna, R., Hershcovitch, M., Cohen, S., Waddington, D. G., Friedrich, T., and Mahoney, M. W. (2023). Fast feature selection with fairness constraints. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 7800–7823. PMLR.

Romano, Y., Bates, S., and Candès, E. J. (2020). Achieving equalized odds by resampling sensitive attributes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural*

Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Sattigeri, P., Hoffman, S. C., Chenthamarakshan, V., and Varshney, K. R. (2019). Fairness GAN: generating datasets with fairness properties using a generative adversarial network. *IBM J. Res. Dev.*, 63(4/5):3:1–3:9.

Scutari, M., Panero, F., and Proissl, M. (2022). Achieving fairness with a simple ridge penalty. *Stat. Comput.*, 32(5):77.

van Breugel, B., Kyono, T., Berrevoets, J., and van der Schaar, M. (2021). DECAF: generating fair synthetic data using causally-aware generative networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 22221–22233.

Wang, J., Lu, D., Davidson, I., and Bai, Z. (2023). Scalable spectral clustering with group fairness constraints. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 6613–6629. PMLR.

Xian, R., Yin, L., and Zhao, H. (2023). Fair and optimal classification via post-processing. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 37977–38012. PMLR.

Xie, Q., Dai, Z., Du, Y., Hovy, E. H., and Neubig, G. (2017). Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 585–596.

Xu, D., Wu, Y., Yuan, S., Zhang, L., and Wu, X. (2019a). Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1452–1458. ijcai.org.

Xu, D., Yuan, S., Zhang, L., and Wu, X. (2018). Fairgan: Fairness-aware generative adversarial networks. In *IEEE International Conference on Big Data (IEEE BigData 2018), Seattle, WA, USA, December 10-13, 2018*, pages 570–575. IEEE.

Xu, J., Xiao, Y., Wang, W. H., Ning, Y., Shenkman, E. A., Bian, J., and Wang, F. (2022). Algorithmic fairness in computational medicine. *EBioMedicine*, 84:104250.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019b). Modeling tabular data using

conditional GAN. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7333–7343.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification. *J. Mach. Learn. Res.*, 20:75:1–75:42.

Zeng, X., Dobriban, E., and Cheng, G. (2022). Bayes-optimal classifiers under group fairness. *CoRR*, abs/2202.09724.

Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 335–340. ACM.

Zhao, H., Coston, A., Adel, T., and Gordon, G. J. (2020). Conditional learning of fair representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhou, Z., Liang, J., Song, Y., Yu, L., Wang, H., Zhang, W., Yu, Y., and Zhang, Z. (2019). Lipschitz generative adversarial nets. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7584–7593. PMLR.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable] → Refer to Sections 3.1 and 3.2.

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]→ See Table 4.

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable] → Refer to the author's Github. The code source was anonymized during the review process.

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable] → Refer to Section 4.

   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable] → Refer to Supplementary 9.

   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable] → Refer to Section 4.

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable] → Refer to the author's Github.

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable] → Refer to Supplementary 10.1 and Section 5.

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable] → Refer to Section 5 and Supplementary 10.

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable] → Refer to Table 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable] → Refer to Section 5. Competing methods are cited.

   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

   (c) New assets either in the supplemental material or as a URL, if applicable. [**Yes**/No/Not Applicable] → Refer to the author's Github.

   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB)

Jinwon Sohn, Qifan Song, Guang Lin

approvals if applicable. [Yes/No/**Not Applicable**]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

Supplementary Materials: Fair Supervised Learning with A Simple Random Sampler of Sensitive Attributes

7 DETAILS OF SECTION 3.1

7.1 Proof of Proposition 1

Let's recall

$$R_F(h; D) = \mathbf{E}_{X,A}[\log D(h(X), A)] + \mathbf{E}_{X,A'}[\log(1 - D(h(X), A'))],$$

where A and A' are independent but identically distributed. Let $s = h(x)$ where x is a realization of X and also a is of A. The loss function can be written

$$R_F(h; D) = \int \log D(s, a) p(s, a) ds da + \int \log(1 - D(s, a')) p(s, a') ds da',$$

$$= \int \log D(s, a) p(s|a) p(a) + \log(1 - D(s, a)) p(s) p(a) ds da.$$

By the proof of Proposition 1 in Goodfellow et al. (2014), $R_F(h; D)$ is maximized at

$$D^*(s, a) = \frac{p(s|a)p(a)}{p(s|a)p(a) + p(s)p(a)} = \frac{p(s|a)}{p(s|a) + p(s)},$$

for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

7.2 Algorithm of GSP penalty

Let $\{(x_i, a_i, y_i)\}_{i=1}^n$ be a set of the realization of $\{X_i, A_i, Y_i\}_{i=1}^n$. The simple random sampler of A can be easily obtained by the minibatch construction (Algorithm 1). In practice, producing $a'_{(i)}$ directly from $\mathbf{D}$, which is an approximated implementation of Algorithm 1, works well. In every iteration, D serves as a fairness critic by quantifying the degree of discrimination against GSP. The model h is then trained to minimize the risk but at the same time to be debiased such that D would not capture the discrimination. Note $\mathbf{w}$ and $\mathbf{v}$ are parameters of D and h. The algorithm to control GSP appears in Algorithm 2. It is allowed to update $\mathbf{w}$ up to $T' \geq 1$ times for every single update of $\mathbf{v}$ for better approximation for $\hat{R}_F(h)$, e.g., $T' = 1, \ldots, 10$.

Algorithm 1: Minibatch Construction (MC) at the tth iteration

---

Data: Let $\mathbf{D}_n = \{(x_i, a_i, y_i)\}_{i=1}^n$ be the set of training data set. Set the minibatch size n_b. The subscript (i) denotes the ith drawn sample.

Result: $\mathbf{D}_{n_b}$

$\mathbf{D} = \{(x_{(i)}, a_{(i)}, y_{(i)})\}_{i=1}^{n_b}$ is randomly drawn from $\mathbf{D}_n$.

$\mathbf{D}' = \{(x'_{(i)}, a'_{(i)}, y'_{(i)})\}_{i=1}^{n_b}$ is randomly drawn from $\mathbf{D}_n \setminus \mathbf{D}$.

Construct $\mathbf{D}_{n_b} = \{(x_{(i)}, a_{(i)}, y_{(i)}, a'_{(i)})\}_{i=1}^{n_b}$ by selecting $\{a'_{(i)}\}_{i=1}^{n_b}$ from $\mathbf{D}'$ and combining it into $\mathbf{D}$.

Algorithm 2: Generalized Statistical Parity

Data: Let $\mathbf{D}_n = \{(x_i, a_i, y_i)\}_{i=1}^n$ be the set of training data set. Denote the tth iterate of model parameters by $\mathbf{w}^{(t)}$ and $\mathbf{v}^{(t)}$ and α is a learning rate. Fix the number of training iterations T and T'; set $t = 0$; and initialize $\mathbf{w}^{(0)}$ and $\mathbf{v}^{(0)}$.

Result: $h_{\mathbf{v}^{(T)}}$

while $t \leq T$ **do**

    $t = t + 1$

    $\{(x_{(i)}, a_{(i)}, y_{(i)}, a'_{(i)})\}_{i=1}^{n_b} = \text{MC}(\mathbf{D}_n)$

    $t' = 0$

    while $t' \leq T'$ **do**

        $t' = t' + 1$

        $\hat{R}_F(h_{\mathbf{v}}; D_{\mathbf{w}^{(t)}}) = \frac{1}{n_b} \sum_{i=1}^{n_b} \left(\log D_{\mathbf{w}^{(t)}}(h_{\mathbf{v}^{(t)}}(x_{(i)}), a_{(i)}) + \log(1 - D_{\mathbf{w}^{(t)}}(h_{\mathbf{v}^{(t)}}(x_{(i)}), a'_{(i)})) \right)$

        $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \alpha \frac{\partial}{\partial \mathbf{w}^{(t)}} \hat{R}_F(h_{\mathbf{v}^{(t)}}; D_{\mathbf{w}^{(t)}})$

    end

    $\hat{R}(h_{\mathbf{v}^{(t)}}) = \frac{1}{n_b} \sum_{i=1}^{n_b} L(y_{(i)}, h_{\mathbf{v}^{(t)}}(x_{(i)}))$

    $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \alpha \frac{\partial}{\partial \mathbf{v}^{(t)}} (\hat{R}(h_{\mathbf{v}^{(t)}}) + \lambda \hat{R}_F(h_{\mathbf{v}^{(t)}}; D_{\mathbf{w}^{(t)}}))$

end

---

8 DETAILS OF SECTION 3.2

8.1 Proof of Proposition 2

The penalty for separation is

$$R_F(h; D) = \mathbf{E}_{X,A,Y}[\log D(h(X), A, Y)] + \mathbf{E}_{A'}\mathbf{E}_{X,Y}[\beta(A', Y)\log(1 - D(h(X), A', Y))],$$

$$= \int \log D(s, a, y)p(s|a, y)p(a, y)dadyds + \beta(a', y)\log(1 - D(s, a', y))p(s|y)p(a')p(y)da'dyds,$$

$$= \int \log D(s, a, y)p(s|a, y)p(a, y) + \beta(a, y)\log(1 - D(s, a, y))p(s|y)p(a)p(y)dadyds.$$

By the same argument in Supplementary 7.1, $D^* = \arg_D \max R_F(h : D)$ has the form of

$$D^*(s, a, y; \beta) = \frac{p(s|a, y)p(a, y)}{p(s|a, y)p(a, y) + \beta(a, y)p(s|y)p(a)p(y)} = \frac{p(s|a, y)}{p(s|a, y) + \beta(a, y)p(s|y)\frac{p(a)p(y)}{p(a, y)}}.$$

8.2 Algorithm of GEO penalty

GEO needs a pre-training step for D_β that is also modeled by a neural network with parameter $\mathbf{u}$. We additionally denote by $D_{\mathbf{u}^{(l)}}$ the lth iterate of D_β. With the same notation in Supplementary 7.2, the numerical algorithm for GEO appears in Algorithm 3.

8.3 Estimation performance from D_β on a toy experiment

We test the estimation performance of D_β for the density ratio estimator β. Suppose $p(y|a) = P(Y = y|A = a) = (1 + \exp(-a))^{-1}$ with $P(A = a) = 0.5$ for $y, a \in \{0, 1\}$. D_β is assumed to have 2 dense hidden layers with 64 nodes and Relu activation functions, and its output layer has a single dimension with Sigmoid activation. In the total number of 10000 training iterations with 100 minibatch size, the last iterate is selected as the point estimate of $p(y|a)/p(y)$. Based on 5 replicated experiments, we calculate the averages and the standard deviation of the point estimates. As it is shown in Table 5, $\hat{\beta}$ successfully estimates the true density ratios.

Algorithm 3: Generalized Equalized Odds

Data: Let $\{(x_i, a_i, y_i)\}_{i=1}^n$ be the set of training data set and obtain $\{a_i'\}_{i=1}^n$ by permuting $\{a_i\}_{i=1}^n$. Denote the tth iterate of model parameters by $\mathbf{u}^{(t)}$, $\mathbf{w}^{(t)}$ and $\mathbf{v}^{(t)}$ and α is a learning rate. Fix L, T, and T'; set $t = 0$ and $l = 0$; and initialize $\mathbf{u}^{(0)}$, $\mathbf{w}^{(0)}$ and $\mathbf{v}^{(0)}$.

Result: $h_{\mathbf{v}^{(T)}}$

while $l \leq L$ **do**

  $l = l + 1$

  $\{(x_{(i)}, a_{(i)}, y_{(i)}, a_{(i)}')\}_{i=1}^{n_b} = \mathrm{MC}(\mathbf{D}_n)$

  $\hat{R}_\beta(D_{\mathbf{u}^{(l)}}) = \frac{1}{n_b} \sum_{i=1}^{n_b} \left(\log D_{\mathbf{u}^{(l)}}(a_{(i)}, y_{(i)}) + \log(1 - D_{\mathbf{u}^{(l)}}(a_{(i)}', y_{(i)})) \right)$

  $\mathbf{u}^{(l+1)} = \mathbf{u}^{(l)} + \alpha \frac{\partial}{\partial \mathbf{u}^{(l)}} \hat{R}_\beta(D_{\mathbf{u}^{(l)}})$

end

while $t \leq T$ **do**

  $t = t + 1$

  $\{(x_{(i)}, a_{(i)}, y_{(i)}, a_{(i)}')\}_{i=1}^{n_b} = \mathrm{MC}(\mathbf{D}_n)$

  $t' = 0$

  while $t' \leq T'$ **do**

    $t' = t' + 1$

    $\hat{R}_F(h_{\mathbf{v}}; D_{\mathbf{w}^{(t)}}) =$

    $\frac{1}{n_b} \sum_{i=1}^{n_b} \left(\log D_{\mathbf{w}^{(t)}}(h_{\mathbf{v}^{(t)}}(x_{(i)}), a_{(i)}, y_{(i)}) + \frac{D_{\mathbf{u}^{(L)}}(a_{(i)}, y_{(i)})}{1 - D_{\mathbf{u}^{(L)}}(a_{(i)}, y_{(i)})} \log(1 - D_{\mathbf{w}^{(t)}}(h_{\mathbf{v}^{(t)}}(x_{(i)}), a_{(i)}', y_{(i)})) \right)$

    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \alpha \frac{\partial}{\partial \mathbf{w}^{(t)}} \hat{R}_F(h_{\mathbf{v}^{(t)}}; D_{\mathbf{w}^{(t)}})$

  end

  $\hat{R}(h_{\mathbf{v}^{(t)}}) = \frac{1}{n_b} \sum_{i=1}^{n_b} L(y_{(i)}, h_{\mathbf{v}^{(t)}}(x_{(i)}))$

  $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \alpha \frac{\partial}{\partial \mathbf{v}^{(t)}} (\hat{R}(h_{\mathbf{v}^{(t)}}) + \lambda \hat{R}_F(h_{\mathbf{v}^{(t)}}; D_{\mathbf{w}^{(t)}}))$

end

Table 5: Estimation performance for $\hat{\beta}$ in the toy example. The point estimate is found by averaging 5 outputs. Std. implies the standard deviation calculated from the 5 outputs.

	$p(1|1)/p(1)$	$p(1|0)/p(1)$	$p(0|1)/p(0)$	$p(0|0)/p(0)$
True	1.1877	0.8123	0.6995	1.3005
Point Estimate (Std.)	1.1810 (0.0077)	0.8126 (0.0029)	0.7082 (0.0018)	1.2955 (0.0048)

9 DETAILS OF SECTION 4

9.1 Proof of Theorem 1

Our theory specifically deals with binary cross-entropy and mean absolute error for binary and continuous outcomes respectively. The overall proof strategy is to check the bounded difference condition to use McDiarmid's inequality.

The estimation error can be defined as

$$|d(\hat{h}^*; \lambda) - \inf_{h \in \mathcal{H}} d(h; \lambda)|, \tag{3}$$

where $\hat{h}^* = \arg_h \min \hat{d}(h; \lambda)$. The estimation error is further decomposed as follows.

$$d(\hat{h}^*; \lambda) - \inf_h d(h; \lambda) = d(\hat{h}^*; \lambda) - d(h^*; \lambda) = d(\hat{h}^*; \lambda) - \hat{d}(\hat{h}^*; \lambda) \tag{4}$$

$$+ \hat{d}(h^*; \lambda) - d(h^*; \lambda) \tag{5}$$

$$+ \hat{d}(\hat{h}^*; \lambda) - \hat{d}(h^*; \lambda), \tag{6}$$

where $h^* = \arg_h \min d(h; \lambda)$, and it is trivial to see (6) ≤ 0. The first line (4) is equivalent to

$$d(\hat{h}^*; \lambda) - \hat{d}(\hat{h}^*; \lambda) = R(\hat{h}^*) - \hat{R}(\hat{h}^*) + \lambda(R_F(\hat{h}^*) - \hat{R}_F(\hat{h}^*)).$$

For (5), we have

$$\hat{d}(h^*; \lambda) - d(h^*; \lambda) = \hat{R}(h^*) - R(h^*) + \lambda(\hat{R}_F(h^*) - R_F(h^*)).$$

Therefore, the estimation error is bounded by

$$|d(\hat{h}^*; \lambda) - \inf_h d(h; \lambda)| \leq 2 \underbrace{\sup_h |R(h) - \hat{R}(h)|}_{\text{I}} + 2\lambda \underbrace{\sup_h |R_F(h) - \hat{R}_F(h)|}_{\text{II}}. \tag{7}$$

Since $\hat{R}(h)$ is an empirical risk function, we denote $T((X_1, Y_1), \ldots, (X_n, Y_n)) = \sup_h |R(h) - \hat{R}(h)|$. For the binary cross-entropy loss function, i.e., $L(Y_i, h(X_i)) = Y_i \log \sigma(h(X_i)) + (1 - Y_i) \log(1 - \sigma(h(X_i)))$, the bounded difference of the ith differing variable can be bounded by

$$|T((X_1, Y_1), \ldots, (X_i, Y_i), \ldots, (X_n, Y_n)) - T((X_1, Y_1), \ldots, (X_i^\dagger, Y_i^\dagger), \ldots, (X_n, Y_n))|$$
$$\leq \sup_h |\frac{1}{n}(L(Y_i, h(X_i)) - L(Y_i^\dagger, h(X_i^\dagger)))|,$$
$$\leq \frac{1}{n} \sup_h |Y_i \log \sigma(h(X_i)) + (1 - Y_i) \log(1 - \sigma(h(X_i))) -$$
$$Y_i^\dagger \log \sigma(h(X_i^\dagger)) - (1 - Y_i^\dagger) \log(1 - \sigma(h(X_i^\dagger)))|. \tag{8}$$

Without loss of generality, let's consider (i) $Y_i = Y_i^\dagger = 1$, (ii) $Y_i = Y_i^\dagger = 0$, and (iii) $Y_i = 1$ and $Y_i^\dagger = 0$. Since the sigmoid function σ is a 1-Lipschitz function, (i) upper bounds

$$|\log \sigma(h(X_i)) - \log \sigma(h(X_i^\dagger))| \leq \frac{1}{\gamma_0} |\sigma(h(X_i)) - \sigma(h(X_i^\dagger))|,$$
$$\leq \frac{1}{\gamma_0} |h(X_i) - h(X_i^\dagger)|,$$
$$\leq \frac{1}{\gamma_0} \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B.$$

The first inequality comes from the Lipschitz property of the logarithm whose domain is bounded below by a positive constant. The Cauchy-Schwarz inequality and the Lipschitz conditions of the activation functions lead the last inequality. Similarly, for (ii), we have

$$|\log(1 - \sigma(h(X_i))) - \log(1 - \sigma(h(X_i^\dagger)))| \leq \frac{1}{1 - \gamma_1} |\sigma(h(X_i)) - \sigma(h(X_i^\dagger))|,$$
$$\leq \frac{1}{1 - \gamma_1} \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B,$$

For (iii), we obtain

$$|\log \sigma(h(X_i)) - \log(1 - \sigma(h(X_i^\dagger)))| \leq \frac{1}{\min\{\gamma_0, 1 - \gamma_1\}} |\sigma(h(X_i)) + \sigma(h(X_i^\dagger)) - 1| \leq \frac{|2\gamma_1 - 1|}{\min\{\gamma_0, 1 - \gamma_1\}}.$$

Therefore, the upper bound (8) is represented as

$$\max \left\{ \frac{|2\gamma_1 - 1|}{n\gamma_{0,1}}, \frac{1}{n\gamma_{0,1}} \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B \right\},$$

where $\gamma_{0,1} = \min\{\gamma_0, 1 - \gamma_1\}$.

On the one hand, if the underlying loss function has the mean absolute error, i.e., $L(Y_i, h(X_i)) = |Y_i - h(X_i)|$, the upper bound of the bounded difference is

$$|T((X_1, Y_1), \ldots, (X_i, Y_i), \ldots, (X_n, Y_n)) - T((X_1, Y_1), \ldots, (X_i^\dagger, Y_i^\dagger), \ldots, (X_n, Y_n))|$$
$$\leq \sup_h |\frac{1}{n}(L(Y_i, h(X_i)) - L(Y_i^\dagger, h(X_i^\dagger)))|,$$
$$\leq \frac{1}{n} \sup_h (|Y_i - Y_i^\dagger| + |h(X_i) - h(X_i^\dagger)|),$$
$$\leq \frac{1}{n} \left(1 + \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B \right).$$

Next, we take expectation to (I) in (7) with respect to the random samples. We obtain

$$\mathbf{E}_{X,Y} \sup_h |R(h) - \hat{R}(h)| = \mathbf{E}_{X,Y} \sup_h |\mathbf{E}_{\tilde{X},\tilde{Y}} \frac{1}{n} \sum_{i=1}^n L(\tilde{Y}_i, h(\tilde{X}_i)) - \frac{1}{n} \sum_{i=1}^n L(Y_i, h(X_i))|,$$
$$\leq \mathbf{E}_{X,Y,\tilde{X},\tilde{Y}} \sup_h |\frac{1}{n} \sum_{i=1}^n L(\tilde{Y}_i, h(\tilde{X}_i)) - \frac{1}{n} \sum_{i=1}^n L(Y_i, h(X_i))|,$$
$$= \mathbf{E}_{X,Y,\tilde{X},\tilde{Y},\epsilon} \sup_h |\frac{1}{n} \sum_{i=1}^n \epsilon_i (L(\tilde{Y}_i, h(\tilde{X}_i)) - L(Y_i, h(X_i)))|,$$
$$\leq 2\mathbf{E}_{X,Y,\epsilon} \sup_h |\frac{1}{n} \sum_{i=1}^n \epsilon_i L(Y_i, h(X_i))| := 2\mathcal{R}(\mathcal{L}),$$

where $\epsilon_i \sim \text{Unif}\{-1, 1\}$ are i.i.d., $(\tilde{X}_i, \tilde{Y}_i)$ are i.i.d. copies (ghost samples) of (X, Y), and $\mathcal{L} := \{L(y, h_{\mathbf{v}}(x)) : \mathbf{v} \in V\}$ is a function class of the given loss L.

Therefore, McDiarmid's inequality implies

$$\text{I} \leq 2\mathcal{R}(\mathcal{L}) + F_{\mathbf{V},\psi,B,\gamma_0,\gamma_1} \sqrt{\frac{\log(1/\delta)}{2n}}, \tag{9}$$

with the 1-δ probability, where $\mathcal{R}$ is the Rademacher complexity of $\mathcal{L}$, and the involved constant is

$$F_{\mathbf{V},\psi,B,\gamma_0,\gamma_1} = \begin{cases} \max\left\{ \frac{|2\gamma_1 - 1|}{\gamma_{0,1}}, \frac{1}{\gamma_{0,1}} \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B \right\} & \text{if } L \text{ is cross-enstropy,} \\ \left(1 + \prod_{j=1}^g M_v(j) \prod_{k=1}^{g-1} K_\psi(k) \times 2B \right) & \text{if } L \text{ is mean absolute difference.} \end{cases}$$

Similarly, we can upper bound (II). First, we check the bounded difference condition that is induced by the neural penalty, i.e.,

$$\text{II} \leq \underbrace{\sup_{h,D} [|\mathbf{E}_{X,A} \log D(h(X), A) - \hat{\mathbf{E}} \log D(h(X), A)|]}_{U_1((X_1,A_1),\ldots,(X_i,A_i),\ldots,(X_n,A_n))}$$
$$+ \underbrace{\sup_{h,D} [|\mathbf{E}_{X,A'} \log(1 - D(h(X), A')) - \hat{\mathbf{E}} \log(1 - D(h(X), A'))|]}_{U_2((X_1,A_1'),\ldots,(X_i,A_i'),\ldots,(X_n,A_n'))}.$$

Then, for any differing the ith coordinate,

$$|U_1((X_1, A_1), \ldots, (X_i, A_i), \ldots, (X_n, A_n)) - U_1((X_1, A_1), \ldots, (X_i^\dagger, A_i^\dagger), \ldots, (X_n, A_n))|$$

$$\leq \frac{1}{n} \sup_{h,D} |\log D(h(X_i), A_i) - \log D(h(X_i^\dagger), A_i^\dagger)|,$$

$$\leq \frac{1}{n\nu_0} |\sigma(f(h(X_i), A_i)) - \sigma(f(h(X_i^\dagger), A_i^\dagger))|,$$

$$\leq \frac{1}{n\nu_0} \prod_{k=1}^d M_w(k) \prod_{j=1}^{d-1} K_\kappa(j) \times ||[h(X_i), A_i] - [h(X_i^\dagger), A_i^\dagger]||,$$

$$\leq \frac{1}{n\nu_0} \prod_{k=1}^d M_w(k) \prod_{j=1}^{d-1} K_\kappa(j) \times \sqrt{l + 4B^2 \left(\prod_{k=1}^g M_v(k) \prod_{l=1}^{g-1} K_\psi(l) \right)^2},$$

$$:= \frac{F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l}}{n\nu_0}.$$

Likewise, we get the similar result for U_2 as follows,

$$|U_2((X_1, A_1'), \ldots, (X_i, A_i'), \ldots, (X_n, A_n')) - U_2((X_1, A_1'), \ldots, (X_i^\dagger, (A_i')^\dagger), \ldots, (X_n, A_n'))|$$

$$\leq \frac{F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l}}{n(1-\nu_1)}.$$

In addition,

$$\mathbf{E}_{X,A} U_1((X_1, A_1), \ldots, (X_i, A_i), \ldots, (X_n, A_n)) \leq \mathbf{E}_{X,A} \mathbf{E}_{\tilde{X},\tilde{A}} \sup_{h,D} |\frac{1}{n} \sum_{i=1}^n \log D(h(\tilde{X}_i), \tilde{A}_i) - \log D(h(X_i), A_i)|,$$

$$\leq \frac{1}{\nu_0} \mathbf{E}_{X,A} \mathbf{E}_{\tilde{X},\tilde{A}} \sup_{h,f} |\frac{1}{n} \sum_{i=1}^n f(h(\tilde{X}_i), \tilde{A}_i) - f(h(X_i), A_i)|,$$

$$= \frac{1}{\nu_0} \mathbf{E}_{X,A} \mathbf{E}_{\tilde{X},\tilde{A},\epsilon} \sup_{h,f} |\frac{1}{n} \sum_{i=1}^n \epsilon_i (f(h(\tilde{X}_i), \tilde{A}_i) - f(h(X_i), A_i))|,$$

$$\leq \frac{2}{\nu_0} \mathbf{E}_{X,A,\epsilon} \sup_{h,f} |\frac{1}{n} \sum_{i=1}^n \epsilon_i f(h(X_i), A_i)| := \frac{2}{\nu_0} \mathcal{R}(\mathcal{D}),$$

where $\epsilon_i \sim \text{Unif}\{-1,1\}$, are i.i.d., $(\tilde{X}_i, \tilde{A}_i)$ are i.i.d. copies (ghost sample) of (X, A), and $\mathcal{D} = \{f_\mathbf{w}(h_\mathbf{v}(x), a) : \mathbf{w} \in \mathbf{W}, \mathbf{v} \in \mathbf{V}\}$ is a compositional function class. Therefore, with the probability $1 - \delta$,

$$U_1((X_1, A_1), \ldots, (X_i, A_i), \ldots, (X_n, A_n)) \leq \frac{2}{\nu_0} \mathcal{R}(\mathcal{D}) + \frac{1}{\nu_0} F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l} \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Similarly for U_2, with $1 - \delta$ probability,

$$U_2((X_1, A_1'), \ldots, (X_i, A_i'), \ldots, (X_n, A_n')) \leq \frac{2}{1-\nu_1} \mathcal{R}(\mathcal{D}) + \frac{1}{(1-\nu_1)} F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l} \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Therefore, with $1 - 2\delta$ probability,

$$\text{II} \leq 2 \left(\frac{1}{\nu_0} + \frac{1}{1-\nu_1} \right) \mathcal{R}(\mathcal{D}) + \left(\frac{1 + \nu_0 - \nu_1}{\nu_0(1-\nu_1)} \right) F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l} \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Consequently, by combining I and II, the estimation error is bounded above by

$$|d(\hat{h}^*; \lambda) - \inf_h d(h; \lambda)| \leq$$

$$4\mathcal{R}(\mathcal{L}) + 2F_{\mathbf{V},\psi,B} \sqrt{\frac{\log(1/\delta)}{2n}} + 2\lambda \left(F_{\nu_0,\nu_1} \mathcal{R}(\mathcal{D}) + F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l,\nu_0,\nu_1} \sqrt{\frac{\log(1/\delta)}{2n}} \right),$$

with $1 - 3\delta$ probability, where $F_{\nu_0,\nu_1} = 2 \left(\frac{1}{\nu_0} + \frac{1}{1-\nu_1} \right)$ and $F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l,\nu_0,\nu_1} = \left(\frac{1+\nu_0-\nu_1}{\nu_0(1-\nu_1)} \right) \times F_{\mathbf{W},\mathbf{V},B,\kappa,\psi,l}$.

9.2 Proof of Corollary 1

Let $\hat{h}^* = \arg\min \hat{d}(h; \lambda)$, $h_0^* = \arg\min d(h; \lambda = 0)$, and $h^* = \arg\min d(h; \lambda)$. Let's denote by $\Delta(h_0^*, h^*) := R_F(h_0^*) - R_F(h^*)$ the amount of decision discrimination in the population with respect to the sensitive information. Since $d(h^*; \lambda) \le d(h_0^*; \lambda)$, we obtain

$$d(\hat{h}^*; \lambda = 0) - \inf_h d(h; \lambda = 0) = R(\hat{h}^*) - R(h_0^*),$$
$$= R(\hat{h}^*) - R(h^*) + R(h^*) - R(h_0^*),$$
$$\le R(\hat{h}^*) - R(h^*) + \lambda \Delta(h_0^*, h^*).$$

Similar to (4) ~ (6), the loss of utility has the following upper bound

$$|d(\hat{h}^*; \lambda = 0) - \inf_h d(h; \lambda = 0)| \le 2 \underbrace{\sup_h |R(h) - \hat{R}(h)|}_{\text{I}} + \lambda \Delta(h_0^*, h^*),$$

where I has the same upper bound (9).

10 DETAILS OF SIMULATION

This section explains simulation setups and delivers additional empirical studies to justify the performance of our method. The outline of this section is as follows.

- 10.1 explains the simulation settings concretely.

- 10.2 displays all other simulation results omitted in the main text such as simulation results for (generalized) statistical parity in Scenario II and (generalized) equalized odds in Section I, trade-off figures, and additional tables summarizing fairness scores.

- 10.3 compares our model to HGR and CON in Scenario III.

- 10.4 discusses the estimation stability of our method when the density ratio estimator (for separation) is poor.

- 10.5 illustrates our method outperforms in having fair representation.

10.1 Overall simulation setting

Target objective function   We first clarify that the notation $(1-\lambda)L_M + \lambda L_F$ in the manuscript. For the competing methods, we refer to the notation of their original papers. All methods below share $\hat{R}(h)$ to express an empirical risk function.

- For Ours,

$$L_M = \hat{R}(h), \quad L_F = \hat{R}_F(h; D).$$

- CON has the same $\hat{R}(h)$ and $\hat{R}_F(h; D)$ but it uses $A' \sim P(A|Y)$.

- For HGR (Lee et al., 2022),

$$\begin{aligned} L_M &= \hat{R}(h), \\ L_F &= \text{HGR}_{\text{soft}}(h(X), A \otimes Y) - \text{HGR}_{\text{soft}}(h(X), Y) \quad \text{for EO}, \\ L_F &= \text{HGR}_{\text{soft}}(h(X), A) \quad \text{for SP}. \end{aligned}$$

  Refer to the original paper to see the exact form of HGR_{soft}.

- For KDE (Cho et al., 2020),

$$\begin{aligned} L_M &= \hat{R}(h), \\ L_F &= \text{DDP} \quad \text{for SP}, \\ L_F &= \text{DEO} \quad \text{for EO}, \end{aligned}$$

  Refer to the original paper to see the exact expression of DDP and DEO.

- For NEU (Du et al., 2021),

$$L_M = \mathcal{L}_{\text{MSE}}, \quad L_F = \mathcal{L}_{\text{Smooth}}.$$

  Refer to the original paper to see the exact expression of $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{Smooth}}$. It is assumed that the sensitive information is available during the training phase. For NEU, the model h is pre-trained by minimizing $\hat{R}(h)$ with an early-stopping procedure. Then the discriminatory h is trained to minimize $(1-\lambda)L_M + \lambda L_F$ to make it debiased.

Network architectures for the simulation studies   All methods have the same neural network model for h. More specifically,

A. **Scenario I & II (binary classification with cross-entropy)**

- h has [Dense(64)-BN-ReLU]*3-[Dense(1)-Sigmoid].
- D has [Dense(64)-BN-ReLU]*2-[Dense(1)-Sigmoid].
- D_β has [Dense(64)-ReLU]*2-[Dense(1)-Sigmoid]   (for separation).

For $\mathrm{HGR_{soft}}$, two neural networks are needed, and each of them is set to [Dense(64)-BN-ReLU]*2-[Dense(1)]. As a result, HGR employs four auxiliary networks in total for separation and two networks for independence. For NEU, h's first output layer is neutralized. For CON in Scenario II, the discriminator has the same structure with D_β while the generator $G = [G_D, G_C]$ with $G_D = G_B$-Dense(1)-Sigmoid and $G_C = G_B$-Dense(1) where G_B =[Concat(Unif(2,[0,1]),Y)-Dense(64)-ReLU-Dense(64)-ReLU]. Unif(2,[0.1]) denotes a 2-dimensional uniform random variable each of which is in $[0, 1]$. The binary value is determined by binning the outputs of G_D with the threshold 0.5.

B. **Scenario III (regression with mean absolute error)**

- h has [Dense(16)-BN-ReLU]*3-[Dense(1)].
- D has [Dense(16)-BN-ReLU]*2-[Dense(1)-Sigmoid].
- D_β has [Dense(16)-ReLU]*2-[Dense(1)-Sigmoid]   (for separation).

$\mathrm{HGR_{soft}}$ has two [Dense(16)-BN-ReLU]*2-[Dense(1)]. For CON, the generator only has a continuous output G =[Concat(Unif(2,[0,1]),Y)-Dense(16)-ReLU-Dense(16)-Dense(1)].

Hyperparameter   All models experimented 5 times for each $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ based on 80% training and 20% validation set. Seed numbers are specified such that all models in the comparison are trained on the same data sets for each λ. For optimization, we adopt stochastic gradient descent with a learning rate of 0.005. For SBP and HGR, the number of training iterations for the maximization part of both algorithms, e.g., the notation T' in our work (Algorithm 1 and 2), is set to 1. The evaluation metrics are calculated every 100 iterations. Each metric is basically expressed as a sum, but it is equivalent to finding an average value. To see other details including a mini-batch size or the number of epochs for all data sets, please refer to the shared code scripts.

Training time comparison shown in Table 4   NEU, SBP, and CON require pre-training courses, i.e., pre-training of h for NEU, of $\hat{\beta}$ for SBP, the GAN model for CON. As a possible case of NEU, pre-training of h may need substantial training time or longer evaluation epochs for using early stopping. In the case of CON, learning the generator gets longer as the complexity of $A|Y$ increases. Tables 4 and 6 present the training times including/excluding the pre-training courses during early iterations for EO and SP to the Adult data. The same pattern is observed in other data sets, so they are omitted. All experiments, written in Tensorflow 2.4.0, run on CentOS 7 featuring Nvidia A30 GPU and 192GB of RAM.

Table 6: Training times (mins) for the first 1000 iterations on A30 GPU for independence in **Adult**. Note NEU needs the pre-training course while others do not, and KDE cannot be applied to Scenario II.

| | Without pre-training | | With pre-training |
Model	Scenario I	Scenario II	Scenario II
SBP	0.23 (0.02)	0.56 (0.05)	-
HGR	0.24 (0.02)	0.59 (0.06)	-
KDE	0.88 (0.07)	-	-
NEU	1.01 (0.09)	1.46 (0.14)	3.61 (0.73)

10.2 More simulation results in Scenario I & II

For the continuous variable, we calculate

$$
\text{SP} = |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \left| \mathbf{E}(\hat{Y} = 1 | A \le a) / \mathbf{E}(\hat{Y} = 1) - 1 \right|,
$$

$$
\text{KS-GSP} = |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \max_{h_x} |\hat{P}(\hat{h}^*(X) \le h_x | A \le a) - \hat{P}(\hat{h}^*(X) \le h_x)|,
$$

where $\mathcal{A}^* = \{\tilde{q}_{10}, \dots, \tilde{q}_{90}\}$ with the $\tilde{q}_r := \frac{r}{100}$th quantile of A. For EO and KS-GEO,

$$
\text{EO} = |\mathcal{A}^*|^{-1} \sum_{y \in \mathcal{Y}, a \in \mathcal{A}^*} \left| \frac{\mathbf{E}(\hat{Y} = 1 | A \le a, Y = y)}{\mathbf{E}(\hat{Y} = 1 | Y = y)} - 1 \right|,
$$

$$
\text{KS-GEO} = |\mathcal{A}^*|^{-1} \sum_{y \in \mathcal{Y}, a \in \mathcal{A}^*} \max_{h_x} |\hat{P}(\hat{h}^*(X) \le h_x | A \le a, Y = y) - \hat{P}(\hat{h}^*(X) \le h_x | Y = y)|.
$$

Following the same manner in the manuscript, we collect the Pareto frontiers for all 5 experiments for each λ. It is possible that each experiment produces a set of Pareto solutions. Figures 2, 3, and 4 overlay those Pareto frontiers for all competitors and fairness measures. In Figure 2, we observe that CON works as well as SBP in Scenario I. This is because $P(A|Y)$ can be easily estimated by the maximum likelihood method.

10.2.1 Trade-off curves by differing λ

Figure 5 shows that our method can capture the trade-off between utility and fairness in independence as λ differs. For clear visualization, λ is specifically chosen as 0.1, 0.5, and 0.9, respectively. The trade-off is also shown in separation (Figure 6).

10.2.2 Evaluation tables with other AUC thresholds

- (Scenario I-SP) Table 7 uses other thresholds compared to Table 2.

- (Scenario I-EO) Tables 8 and 9 show the tables with different thresholds.

- (Scenario II-SP) Table 10 show the results in SP.

- (Scenario II-EO) Table 11 uses other thresholds compared to Table 3.

These extra tables show consistent results with the tables in the main text and with their corresponding figures. In general, our method outperforms the competing methods. The thresholds are chosen such that there are a sufficient number of fairness scores to calculate their averages and standard deviations from the 5 smallest scores. For instance, if the larger threshold is chosen such as AUC ≥ 0.9 in Adult, there are no available fairness scores in all comparison methods.

Table 7: (Scenario I) Averages of the 5 smallest of SP/KS-GSPs whose AUCs are greater than the thresholds. Those scores are selected in the Pareto solutions appearing in Figure 1. Standard deviations are in the parentheses next to the averages. Note CON is not available for SP.

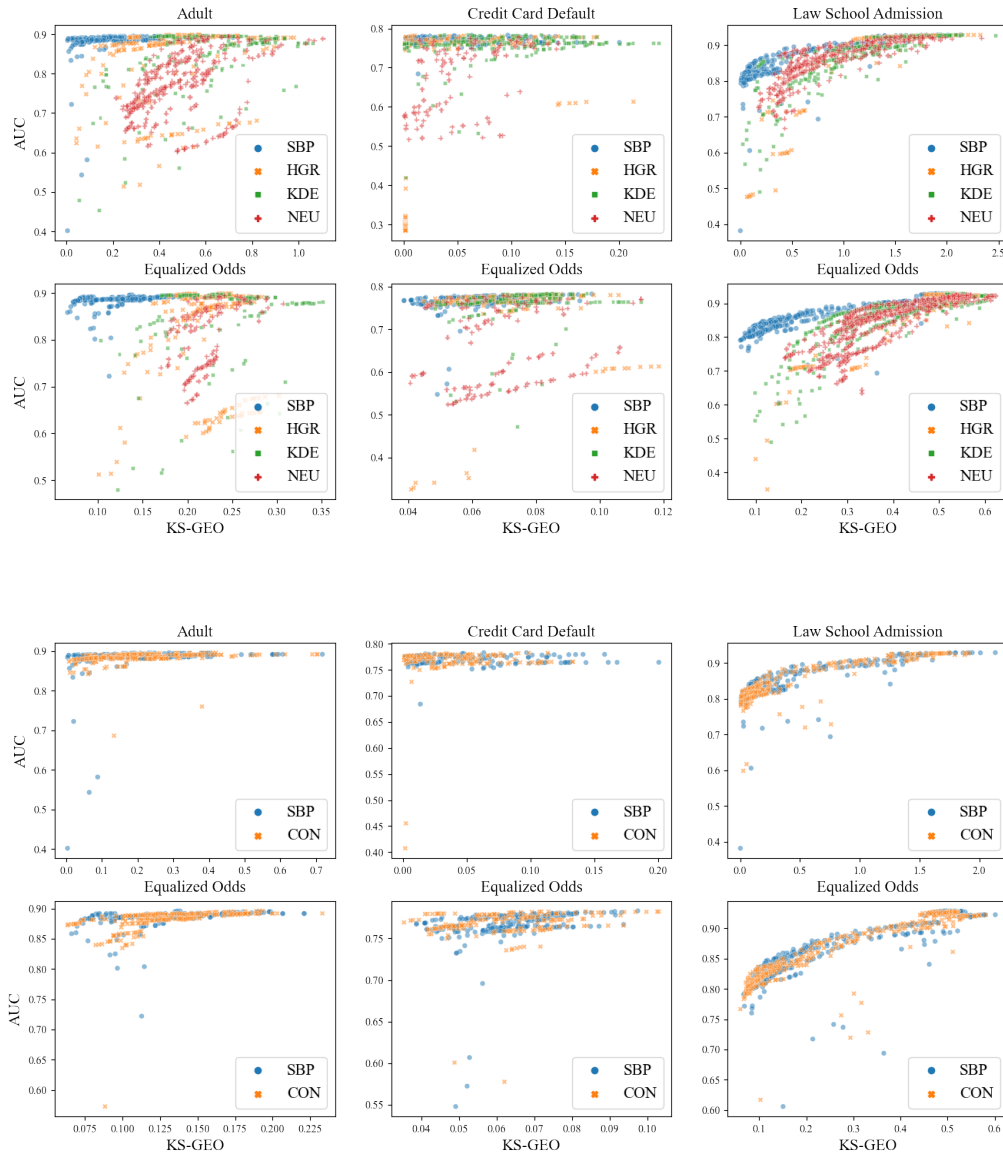|	**Adult** (AUC ≥ 0.80)		**Cred. Card.** (AUC ≥ 0.70)		**Law School.** (AUC ≥ 0.85)	
Model	SP ($\downarrow$)	KS-GSP ($\downarrow$)	SP ($\downarrow$)	KS-GSP ($\downarrow$)	SP ($\downarrow$)	KS-GSP ($\downarrow$)
SBP	0.001 (≈ 0)	0.014 (≈ 0)	≈ 0 (≈ 0)	0.011 (0.001)	0.199 (0.032)	0.112 (0.011)
HGR	0.070 (0.005)	0.040 (≈ 0)	0.001 (0.001)	0.020 (0.001)	0.123 (0.005)	0.089 (0.001)
KDE	0.280 (0.018)	0.110 (0.005)	0.005 (0.003)	0.023 (0.001)	0.252 (0.006)	0.151 (0.001)
NEU	0.372 (0.018)	0.137 (0.002)	0.002 (0.001)	0.013 (0.001)	0.385 (0.008)	0.159 (0.006)

Figure 2: (Scenario I) Pareto frontiers: the first row includes pairs of EO and AUC, and the second row shows pairs of KS-GEO and AUC from 5 experiments for each λ. Since SBP and CON are similar, they are directly compared in the below figure. SBP and CON illustrate better results than others in Adult and Law School Admission as they are tightly in the upper-left corner.

10.3 Simulation results (for both independence and separation) in Scenario III

We further compare SBP with HGR and CON in Scenario III using **Community and Crime**[1] data. Following Lee et al. (2022), the number of violent crimes per population and the ratio of black people in the population are used as a continuous outcome and a sensitive variable respectively. For the evaluation of fairness, we consider the
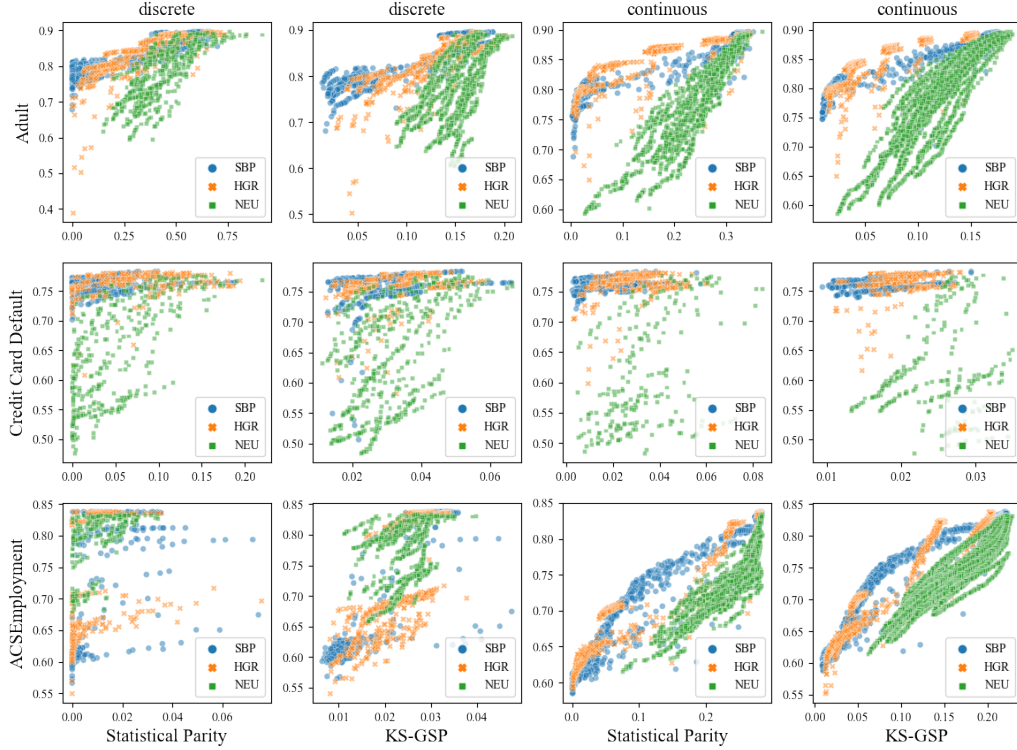
Figure 3: (Scenario II) Pareto frontiers: the first and the second row correspond to Adult, Credit Card Default, and ACSEmployment respectively from 5 experiments for each λ. SBP is superior to both NEU and HGR overall but comparable to HGR in Adult. Note CON cannot handle statistical parity.

Table 8: (Scenario I) Averages of the 5 smallest of EO/KS-GEOs whose AUCs are greater than the thresholds. Those scores are selected in the Pareto solutions appearing in Figure 2. Standard deviations are in the parentheses next to the averages.

Model	**Adult** (AUC $\geq$ 0.85)		**Cred. Card.** (AUC $\geq$ 0.75)		**Law School.** (AUC $\geq$ 0.85)	
	EO ($\downarrow$)	KS-GEO ($\downarrow$)	EO ($\downarrow$)	KS-GEO ($\downarrow$)	EO ($\downarrow$)	KS-GEO ($\downarrow$)
SBP	0.009 (0.005)	0.070 (0.002)	0.003 (0.001)	0.040 (0.001)	0.166 (0.027)	0.140 (0.005)
CON	0.019 (0.008)	0.066 (0.002)	0.001 (0.001)	0.038 (0.002)	0.217 (0.033)	0.190 (0.009)
HGR	0.164 (0.031)	0.170 (0.006)	0.002 (0.001)	0.047 (0.002)	0.891 (0.080)	0.409 (0.014)
KDE	0.383 (0.013)	0.172 (0.003)	0.002 (0.001)	0.050 (0.001)	0.311 (0.128)	0.262 (0.006)
NEU	0.450 (0.028)	0.205 (0.005)	0.030 (0.019)	0.046 (0.001)	0.421 (0.022)	0.299 (0.003)

following metrics,

$$\text{SP} = |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \left| \mathbf{E}(\hat{h}^*(X)|A \leq a)/\mathbf{E}(\hat{h}^*(X)) - 1 \right|,$$

$$\text{EO} = |\mathcal{Y}^*|^{-1} \sum_{y \in \mathcal{Y}^*} |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \left| \mathbf{E}(\hat{h}^*(X)|A \leq a, Y \leq y)/\mathbf{E}(\hat{h}^*(X)|Y \leq y) - 1 \right|,$$

$$\text{KS-GSP} = |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \max_{h_x} |\hat{P}(\hat{h}^*(X) \leq h_x|A \leq a) - \hat{P}(\hat{h}^*(X) \leq h_x)|,$$

$$\text{KS-GEO} = |\mathcal{Y}^*|^{-1} \sum_{y \in \mathcal{Y}^*} |\mathcal{A}^*|^{-1} \sum_{a \in \mathcal{A}^*} \max_{h_x} |\hat{P}(\hat{h}^*(X) \leq h_x|A \leq a, Y \leq y) - \hat{P}(\hat{h}^*(X) \leq h_x|Y \leq y)|,$$
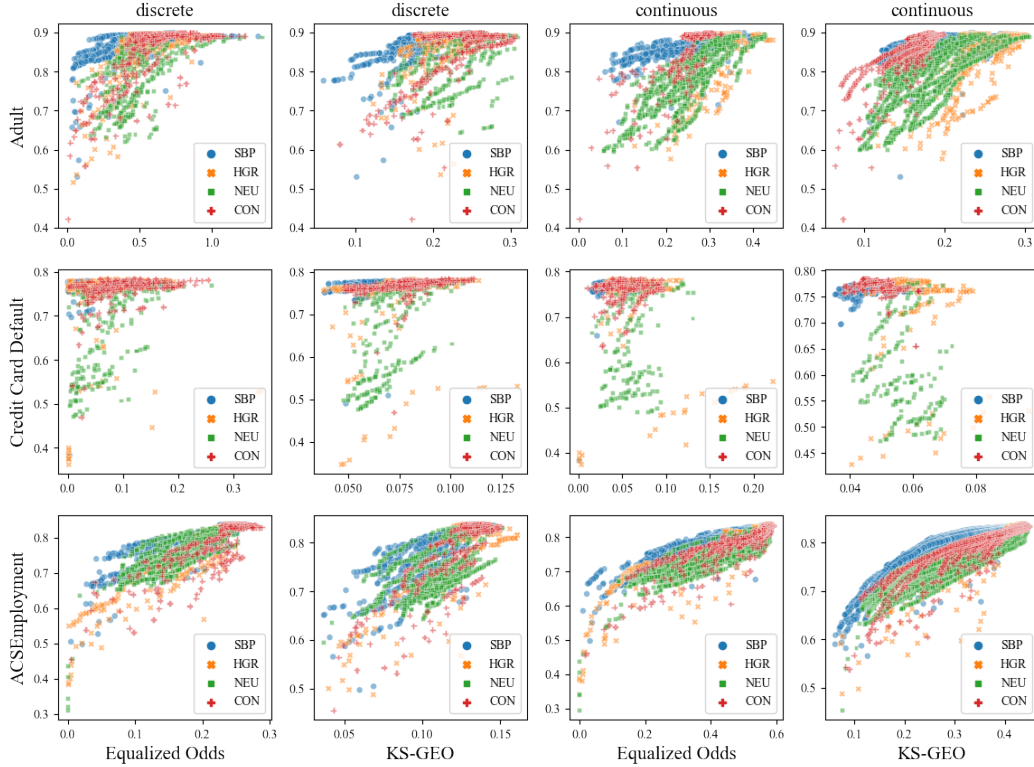
Figure 4: (Scenario II) Pareto frontiers: the first and the second row correspond to Adult, Credit Card Default, and ACSEmployment respectively from 5 experiments for each λ. Remarkably, SBP (ours) outperforms the competitors for the most part. Note KDE cannot handle continuous attributes.

Table 9: (Scenario I) Averages of the 5 smallest of EO/KS-GEOs whose AUCs are greater than the thresholds. Those scores are selected in the Pareto solutions appearing in Figure 2. Standard deviations are in the parentheses next to the averages.

	Adult (AUC ≥ 0.80)		**Cred. Card.** (AUC ≥ 0.70)		**Law School.** (AUC ≥ 0.80)	
Model	EO ($\downarrow$)	KS-GEO ($\downarrow$)	EO ($\downarrow$)	KS-GEO ($\downarrow$)	EO ($\downarrow$)	KS-GEO ($\downarrow$)
SBP	0.009 (0.005)	0.070 (0.003)	0.003 (0.001)	0.040 (0.001)	0.020 (0.010)	0.078 (0.001)
CON	0.016 (0.007)	0.066 (0.002)	0.001 (0.001)	0.038 (0.002)	0.014 (0.003)	0.074 (0.002)
HGR	0.127 (0.025)	0.167 (0.007)	0.002 (≈ 0)	0.047 (0.002)	0.587 (0.152)	0.365 (0.027)
KDE	0.282 (0.064)	0.155 (0.009)	0.001 (0.001)	0.049 (0.001)	0.194 (0.016)	0.216 (0.007)
NEU	0.364 (0.013)	0.182 (0.003)	0.016 (0.003)	0.046 (0.001)	0.380 (0.033)	0.245 (0.012)

where $\mathcal{A}^*$ and $\mathcal{Y}^*$ are the sets of quantile values of A and Y; $\mathcal{Y}^* = \{\tilde{b}_{10}, \ldots, \tilde{b}_{90}\}$ with the $\tilde{b}_r := \frac{r}{100}$th quantile of Y.

In Scenario III, NEU cannot be implemented because NEU requires partitioning a data set with respect to the outcome variable, which is practically impossible for the continuous outcome. Figure 7 and Table 12 show that SBP and HGR are comparable in Scenario III.
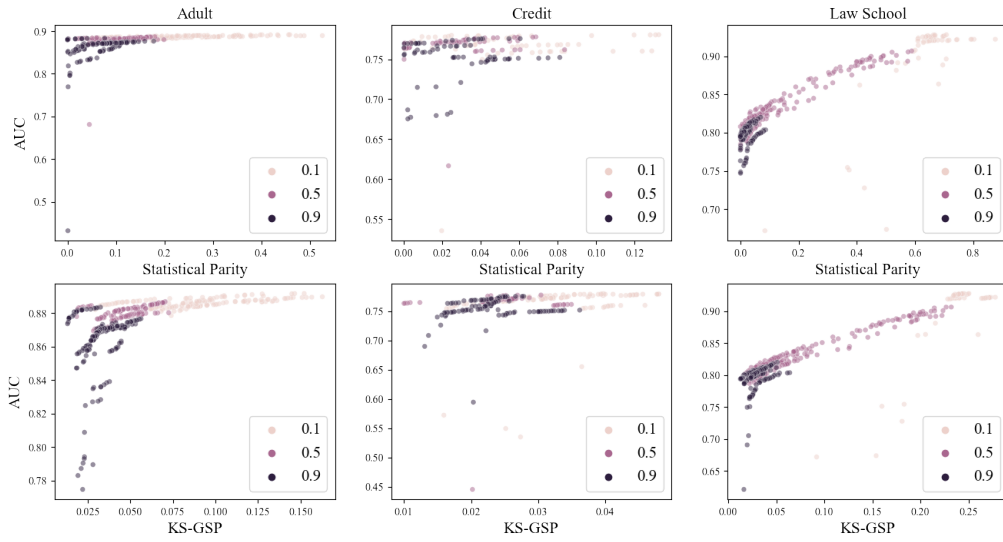
Figure 5: (Scenario I of SBP) Pareto frontiers of SP (and KS-GSP) and AUC by differing λ from 0.1 to 0.9 for all 5 experiments.
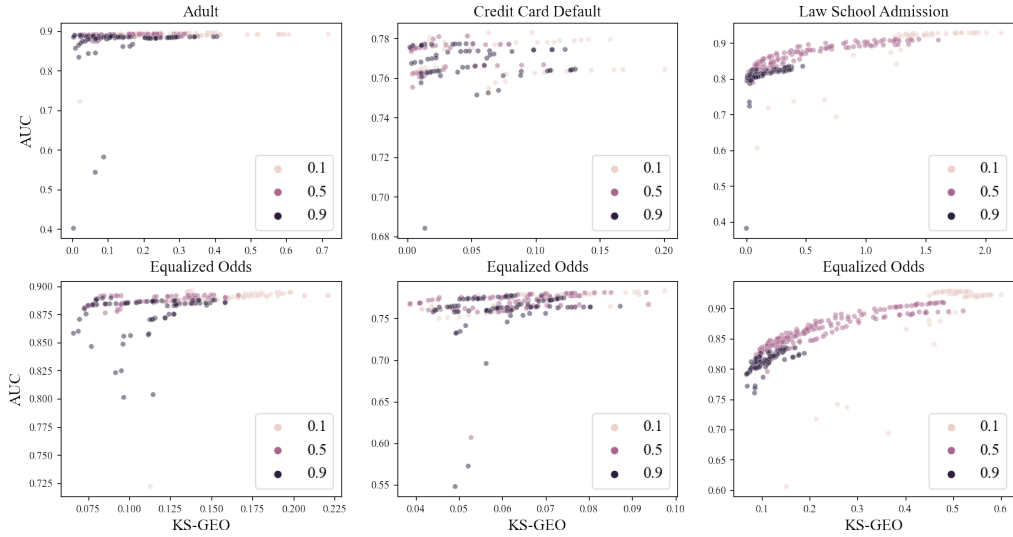

Figure 6: (Scenario I of SBP) Pareto frontiers of EO (and KS-GEO) and AUC by differing λ from 0.1 to 0.9 for all 5 experiments.

10.4 Robust estimation against the poor estimate of $\beta(a, y)$

We also investigate the impact of $\hat{\beta}$ discussed in Section 3.2 based on II. Interestingly, we observe that $\hat{\beta}$ hardly affects the performance of SBP even though the density-ratio estimator is required to guarantee GEO theoretically. Figure 9 compares the Pareto frontiers when using $\hat{\beta}(a, y) = \hat{D}_\beta(a, y)/(1 - \hat{D}_\beta(a, y))$ (SBP) or using $\hat{\beta}(a, y) = 1$ for all a, y (SBP_NW). SBP_NW is set to having the same simulation configuration as SBP. The figure describes that SBP and SBP_NW are almost the same in both scenarios. The same tendency is also found in Scenario I (Figure 8).

Table 10: (Scenario II) Averages of the 5 smallest of SP/KS-GSPs whose AUCs are greater than the thresholds. Those scores are selected by referring to Figure 3. Standard deviations are in the parentheses next to the averages.

Met.	Mod.	**Adult** (AUC $\geq$ 0.80)		**Cred. Card.** (AUC $\geq$ 0.75)		**ACSEmpl.** (AUC $\geq$ 0.75)	
		Race ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)
SP	SBP	0.002 (0.001)	0.018 (0.002)	≈ 0 (≈ 0)	0.004 (0.001)	0.001 (0.001)	0.116 (0.002)
	HGR	0.148 (0.006)	0.019 (0.001)	≈ 0 (≈ 0)	0.010 (0.001)	≈ 0 (≈ 0)	0.196 (0.004)
	NEU	0.338 (0.018)	0.216 (0.010)	0.050 (0.011)	0.040 (0.004)	≈ 0 (≈ 0)	0.216 (0.008)
KS-GSP	SBP	0.019 (0.002)	0.015 (≈ 0)	0.014 (≈ 0)	0.010 (≈ 0)	0.019 (0.003)	0.078 (0.002)
	HGR	0.062 (0.005)	0.021 (0.001)	0.015 (≈ 0)	0.012 (≈ 0)	0.017 (0.002)	0.118 (0.001)
	NEU	0.127 (0.001)	0.088 (0.003)	0.025 (0.003)	0.026 (≈ 0)	0.012 (≈ 0)	0.140 (0.001)

Table 11: (Scenario II) Averages of the 5 smallest of EO/KS-GEOs whose AUCs are greater than the thresholds. Those scores are selected by referring to Figure 4. Standard deviations are in the parentheses next to the averages.

Met.	Mod.	**Adult** (AUC $\geq$ 0.85)		**Cred. Card.** (AUC $\geq$ 0.70)		**ACSEmpl.** (AUC $\geq$ 0.80)	
		Race ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)	Gender ($\downarrow$)	Age ($\downarrow$)
EO	SBP	0.143 (0.013)	0.122 (0.002)	0.001 (≈ 0)	0.016 (0.001)	0.168 (0.004)	0.373 (0.006)
	CON	0.425 (0.017)	0.241 (0.002)	0.006 (0.003)	0.016 (0.003)	0.224 (0.004)	0.458 (0.031)
	HGR	0.355 (0.010)	0.307 (0.002)	0.002 (0.001)	0.033 (0.001)	0.195 (0.004)	0.447 (0.002)
	NEU	0.385 (0.022)	0.268 (0.012)	0.015 (0.010)	0.038 (0.004)	0.143 (0.008)	0.443 (0.017)
KS-GEO	SBP	0.147 (0.002)	0.124 (0.002)	0.040 (0.001)	0.036 (0.001)	0.094 (0.001)	0.247 (0.002)
	CON	0.177 (0.001)	0.138 (0.002)	0.049 (0.001)	0.038 (≈ 0)	0.115 (0.003)	0.312 (0.002)
	HGR	0.162 (0.004)	0.221 (0.001)	0.041 (0.003)	0.047 (0.001)	0.122 (0.001)	0.356 (0.005)
	NEU	0.196 (0.008)	0.166 (0.004)	0.053 (0.003)	0.051 (0.001)	0.101 (0.006)	0.334 (0.009)

Table 12: (Scenario III) Averages of the 5 smallest of SP/KS-GSP/EO/KS-GEOs whose MAEs are less than the thresholds. Those scores are selected in the Pareto solutions appearing in Figure 7. Standard deviations are in the parentheses next to the averages.

Model	**Community and Crime** (MAE $\leq$ 0.12)			
	SP ($\downarrow$)	KS-GSP ($\downarrow$)	EO ($\downarrow$)	KS-GEO ($\downarrow$)
SBP	0.225 (0.007)	0.141 (0.004)	0.083 (0.003)	0.060 (0.002)
CON	-	-	0.094 (0.004)	0.062 (0.002)
HGR	0.247 (0.002)	0.155 (0.001)	0.082 (0.004)	0.069 (0.002)

10.5 Fair Representation

We carry out additional simulation studies to verify that the neural penalty can be used to produce fair representation in Scenario II for EO. Following the same notation in Section 3.2, we set E as [Dense(64)-BN-ReLU] and h_E as [Dense(64)-BN-ReLU]*2-[Dense(1)-Sigmoid], so that $h = h_E \circ E$. This is the same architecture as NEU. For HGR, we calculate $\text{HGR}_{\text{soft}}(E(X), A)$ for SP and $\text{HGR}_{\text{soft}}(E(X), A \otimes Y) - \text{HGR}_{\text{soft}}(E(X), Y)$ for EO. When training SBP and HGR, the number of training iterations for the maximization part is set to 5, i.e., $T' = 5$.

Figure 11 highlights that SBP succeeds in generating fair representation as achieving comparable or even better performance than the competing methods. The collected Pareto frontiers of the models on the benchmark data sets illustrate that SBP tends to defeat others slightly. We see that the SBP can represent the fair representation for SP as well (Figure 10).
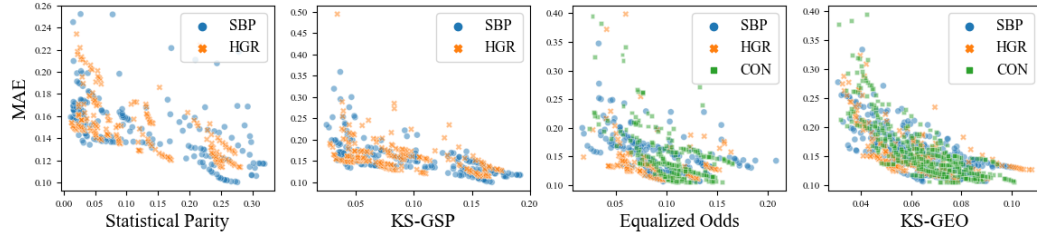
Figure 7: (Scenario III) Pareto frontiers: the points are pairs of the fairness metrics and MAE. Note the bottom-left tendency implies a better trade-off.
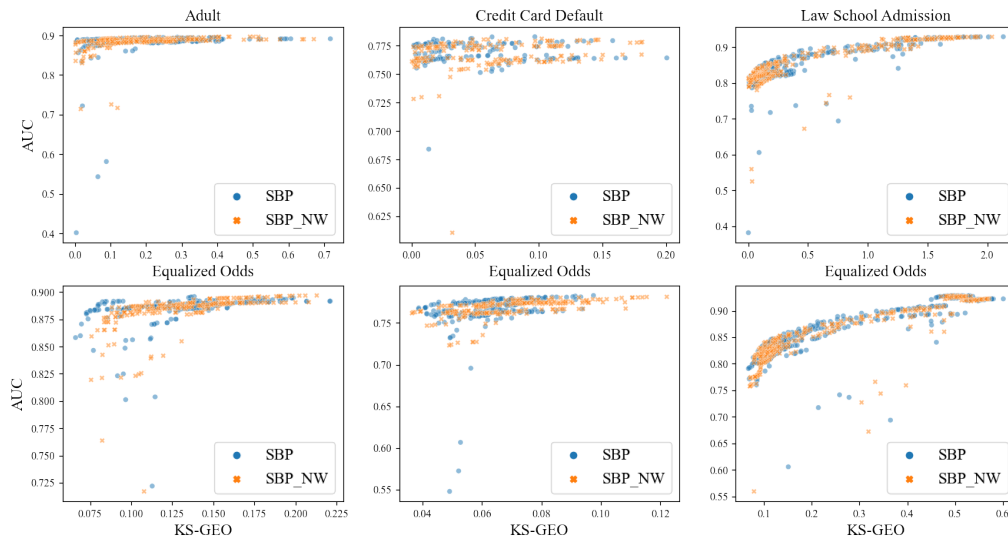

Figure 8: (Scenario I, the impact of $\hat{\beta}$) Pareto frontiers: the points are pairs of EO and AUC or KS-GEO and AUC.
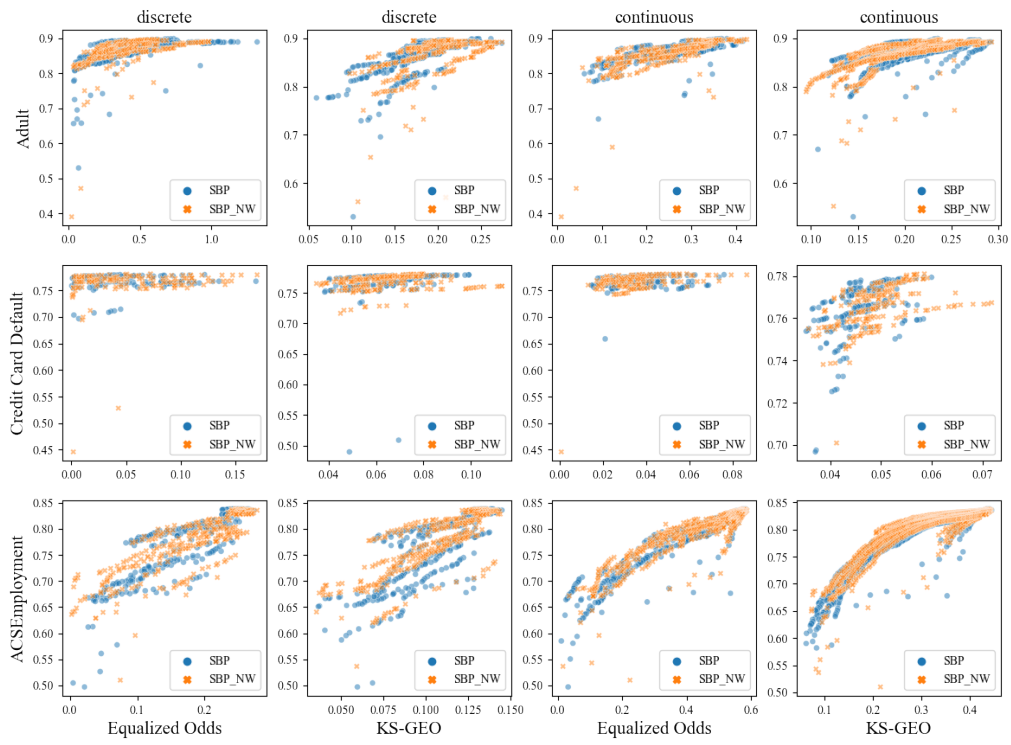
Figure 9: (Scenario II, the impact of $\hat{\beta}$) Pareto frontiers: the points are pairs of EO and AUC or KS-GEO and AUC.
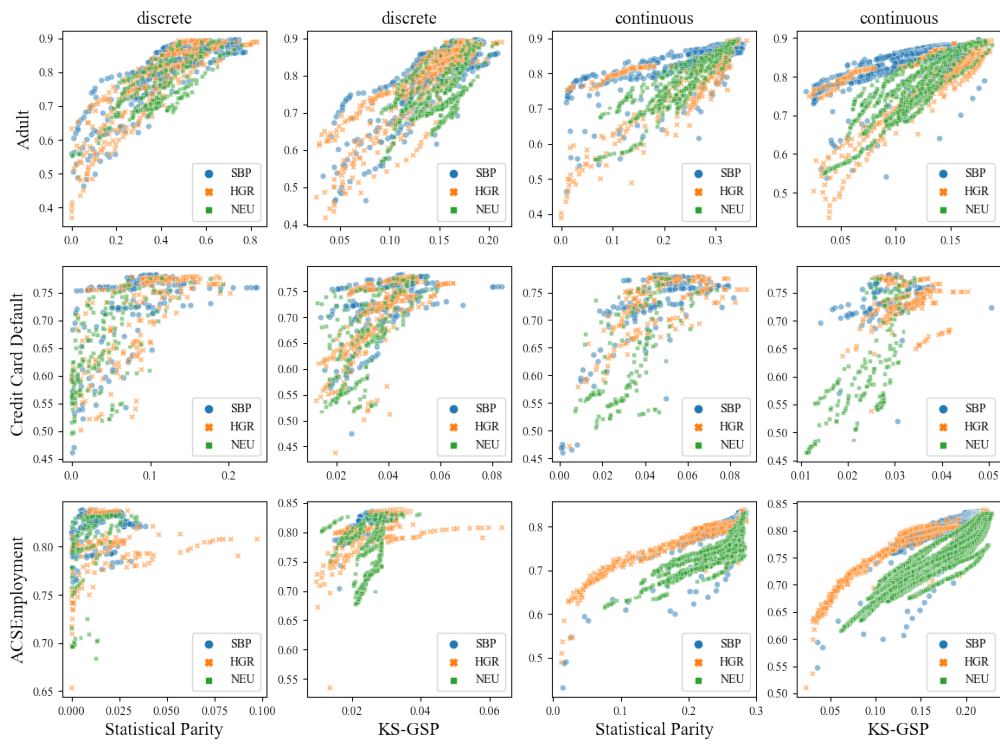
Figure 10: (Scenario II, fair representation) Pareto frontiers: the points are pairs of SP and AUC or KS-GSP and AUC.
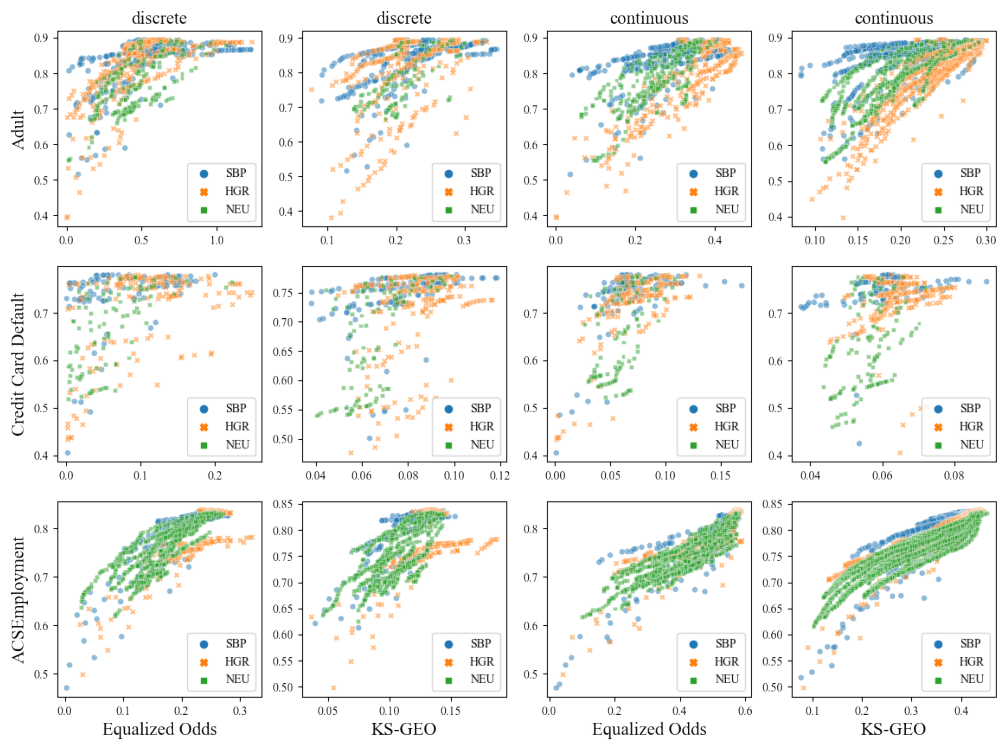
Figure 11: (Scenario II, fair representation) Pareto frontiers: the points are pairs of EO and AUC or KS-GEO and AUC.