### MULTIFAIR: Multi-Group Fairness in Machine Learning

Jian Kang<sup>1</sup>, Tiankai Xie<sup>2</sup>, Xintao Wu<sup>3</sup> Ross Maciejewski<sup>2</sup>, and Hanghang Tong<sup>1</sup> University of Illinois at Urbana-Champaign, {jiank2, htong}@illinois.edu;

<sup>2</sup>Arizona State University, {txie21, rmacieje}@asu.edu;

<sup>3</sup>University of Arkansas, {xintaowu}@uark.edu

#### **ABSTRACT**

Algorithmic fairness is becoming increasingly important in data mining and machine learning, and one of the most fundamental notions is *group fairness*. The vast majority of the existing works on group fairness, with a few exceptions, primarily focus on debiasing with respect to a single sensitive attribute, despite the fact that the co-existence of multiple sensitive attributes (e.g., gender, race, marital status, etc.) in the real-world is commonplace. As such, methods that can ensure a fair learning outcome with respect to all sensitive attributes of concern simultaneously need to be developed. In this paper, we study multi-group fairness in machine learning (MULTIFAIR), where statistical parity, a representative group fairness measure, is guaranteed among demographic groups formed by multiple sensitive attributes of interest. We formulate it as a mutual information minimization problem and propose a generic end-to-end algorithmic framework to solve it. The key idea is to leverage a variational representation of mutual information, which considers the variational distribution between learning outcomes and sensitive attributes, as well as the density ratio between the variational and the original distributions. Our proposed framework is generalizable to many different settings, including other statistical notions of fairness, and could handle any type of learning task equipped with a gradient-based optimizer. Empirical evaluations in the fair classification task on three real-world datasets demonstrate that our proposed framework can effectively debias the classification results with minimal impact to the classification accuracy.

#### **ACM Reference Format:**

Jian Kang<sup>1</sup>, Tiankai Xie<sup>2</sup>, Xintao Wu<sup>3</sup> Ross Maciejewski<sup>2</sup>, and Hanghang Tong<sup>1</sup>. 2018. Multifair: Multi-Group Fairness in Machine Learning. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/1122445.1122456

#### 1 INTRODUCTION

The increasing amount of data and computational power have empowered machine learning algorithms to play crucial roles in automated decision-making for a variety of real-world applications, including college admission [33], credit scoring [21], criminal justice [5] and healthcare analysis [2]. As the application landscape

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Woodstock '18, June 03–05, 2018, Woodstock, NY © 2018 Association for Computing Machinery. ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00 https://doi.org/10.1145/1122445.1122456

of machine learning continues to broaden and deepen, so does the concern regarding the potential, often unintentional, bias it could introduce or amplify. For example, recent media coverage have revealed that a well-trained image generator could turn a low-resolution picture of a black man into a high-resolution image of white man due to the skewed data distribution that causes the model to disfavor the minority group<sup>1</sup>, and another article high-lighted an automated credit card application system assigning a dramatically higher credit limit to a man than to his female partner, even though his partner has a better credit history<sup>2</sup>.

As such, algorithmic fairness, which aims to mitigate unintentional bias caused by automated learning algorithms, has become increasingly important in recent years. To date, researchers have proposed a variety of fairness notions [10, 11]. Among them, one of the most fundamental notions is group fairness<sup>3</sup>. Generally speaking, to ensure group fairness, the first step is to partition the entire population into a few demographic groups based on a pre-defined sensitive attribute (e.g., gender). Then the fair learning algorithm will enforce parity of a certain statistical measure among those demographic groups. Group fairness can be instantiated with many statistical notions of fairness. Statistical parity [35] enforces the learned classifier to accept equal proportion of population from the pre-defined majority group and minority group. Likewise, disparate impact [11] ensures the acceptance rate for the minority group should be no less than four-fifth of that for the group with the highest acceptance rate, which is analogous to the famous 'fourfifth' rule in the legal support area [23]. In addition, equalized odds and equal opportunity [14] are used to enforce the classification accuracies to be equal across all demographic groups conditioned on ground-truth outcomes or positively labeled populations, respectively. The vast majority of the existing works in group fairness primarily focus on debiasing with respect to a single sensitive attribute. However, it is quite common for multiple sensitive attributes (e.g., gender, race, marital status, etc.) to co-exist in a real-world application. We ask: would a debiasing algorithm designed to ensure the group fairness for a particular sensitive attribute (e.g., marital status) unintentionally amplify the group bias with respect to another sensitive attribute (e.g., gender)? If so, how can we ensure a fair learning outcome with respect to all sensitive attributes of concern simultaneously?

The sparse literature for answering these questions [7, 11, 17, 35] has two major limitations. The first limitation is that some existing

<sup>&</sup>lt;sup>1</sup>https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-nulse-stylegan-ohama-hiss

pulse-stylegan-obama-bias <sup>2</sup>https://www.nytimes.com/2019/11/10/business/Apple-credit-card-investigation.html

<sup>&</sup>lt;sup>3</sup>An orthogonal work in algorithmic fairness is individual fairness. Although it promises fairness by 'treating similar individuals similarly' in principle, it is often hard to be operationalized in practice due to its strong assumption on distance metrics and data distributions.

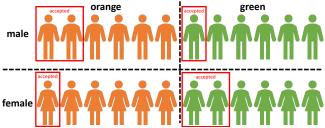


Figure 1: An illustrative example of bias in job application classification when considering multiple sensitive attributes. Rows indicate gender (e.g., male vs. female) and columns indicate race (e.g., orange vs. green)<sup>4</sup>. Boxed individuals receive job offers. If we consider gender or race alone, statistical parity is enforced due to the equal acceptance rate. However, when considering gender and race (i.e., forming finer-grained gender-race groups), the classification result is biased in the fine-grained gender-race groups. This is because, the acceptance rates in two fine-grained groups (i.e., male-green group and female-orange) are lower than that of the two other fine-grained groups (i.e., male-orange and female-green).

works could only debias multiple *distinct* sensitive attributes [7], which fails to mitigate bias on the fine-grained groups formed by multiple sensitive attributes. Figure 1 provides an illustrative example of the difference between fairness with respect to multiple distinct sensitive attributes and fairness among fine-grained groups of multiple sensitive attributes. The second limitation is that the optimization problems behind other existing works are often subject to surrogate constraints of statistical parity [11, 17, 35] instead of directly optimizing statistical parity itself, resulting in unstable performance on bias mitigation unless the learned models are Bayes optimal.

In this paper, we tackle these two limitations by studying the problem of multi-group fairness (MULTIFAIR), which aims to directly enforce statistical parity on multiple sensitive attributes simultaneously. Though our focused fairness notion is statistical parity, the proposed method can be generalized to other statistical fairness notions (e.g., equalized odds and equal opportunity) with minor modifications. The key idea in solving the MultiFair problem is to consider all sensitive attributes of interest as a vectorized sensitive attribute in order to partition the demographic groups and then minimize the dependence between learning outcomes and this vectorized attribute. More specifically, we measure the dependence using mutual information originated in information theory [30]. Building upon it, we formulate the MULTIFAIR problem as an optimization problem regularized on mutual information minimization. To the best of our knowledge, we are the first to debias multiple sensitive attributes without relying on surrogate proxy constraints.

The main contributions of this paper are summarized as follows.

• **Problem Definition.** We formally define the problem of multi-group fairness in machine learning (MULTIFAIR) and formulate it as an optimization problem, where the key idea

is to minimize both the task-specific loss function (e.g., crossentropy loss in classification) and mutual information between learning outcomes and the vectorized sensitive attribute.

- End-to-End Algorithmic Framework. We propose a novel end-to-end bias mitigation framework, named MULTIFAIR, by optimizing a variational representation of mutual information. The proposed framework is extensible and capable of solving any learning task equipped with a gradient-based optimizer.
- Empirical Evaluations. We perform empirical evaluations in the fair classification task on three real-world datasets. The evaluation results demonstrate that our proposed frame-work can effectively mitigate bias with little sacrifice in the classification accuracy.

#### 2 PROBLEM DEFINITION

In this section, we first present a table of the main symbols used in this paper. Then, we briefly review the concepts of statistical parity and mutual information, as well as their relationships. Finally, we formally define the problem of multi-group fairness.

**Symbols Definitions** D a set W a matrix h a vector h[i]the *i*-th element in **h**  $Pr(\cdot)$ the probability of an event happening joint distribution of two random variables p.,. marginal distribution of a random variable p.  $H(\cdot)$ entropy  $H(\cdot|\cdot)$ conditional entropy mutual information  $I(\cdot,\cdot)$ 

Table 1: Table of symbols.

In this paper, matrices are denoted by bold uppercase letters (e.g., X), vectors are denoted by bold lowercase letters (e.g., y), scalars are denoted by italic lowercase letters (e.g., c) and sets are denoted by calligraphic letters (e.g., D). We use superscript  $^T$  to denote transpose (e.g.,  $\mathbf{h}^T$  is the transpose of  $\mathbf{h}$ ) and superscript C to denote the complement of a set (e.g., set  $D^C$  is the complement of set D). We use a convention similar to NumPy for vector indexing (e.g.,  $\mathbf{h}[i]$  is the i-th element in vector  $\mathbf{h}$ ).

#### 2.1 Preliminaries

**Statistical Parity** is one of the most intuitive and widely-used group fairness notions. Given a set of data points X, their corresponding labels y and a sensitive attribute s, classification with statistical parity aims to learn a classifier to predict outcomes that (1) are as accurate as possible with respect to y and (2) do not favor one group over another with respect to y. Mathematically, statistical parity is defined as follows.

DEFINITION 1. (Statistical Parity [35]). Suppose we have (1) a population X, (2) a hypothesis  $h: X \to \{0,1\}$  which assigns a binary label to individual x drawn from X and (3) a sensitive attribute which splits the population X into majority group M and minority group  $M^C$  (i.e.,  $X = M \cup M^C$ ). An individual x is accepted if h(x) = 1

<sup>&</sup>lt;sup>4</sup>We use imaginary race groups to avoid potential offenses.

and rejected if h(x) = 0. The hypothesis  $h: X \to \{0, 1\}$  is said to have statistical parity on the population X as long as

$$Pr[h(x) = 1 | x \in \mathcal{M}] = Pr[h(x) = 1 | x \in \mathcal{M}^C]$$

where  $Pr[\cdot]$  denotes the probability of an event happening.

Several methods have been proposed to achieve statistical parity. For example, Zemel et al. [36] learn fair representation by regularizing the difference in expected positive rate for majority and minority groups. Zhang et al. [37] propose an adversarial learning-based framework for fair classification, in which the output of the predictor is used to predict the sensitive attribute by the adversary. Kearns et al. [17] propose a learner-auditor framework to enforce subgroup fairness through fictitious play strategy.

**Mutual Information** was first introduced in 1940s [30]. Given two random variables, mutual information measures the dependence between them by quantifying the amount of information in bits obtained on one random variable through observing the other one.

DEFINITION 2. (Mutual Information [30]). Let (x, y) be a pair of random variables x and y. Suppose their joint distribution is  $p_{x,y}$  and the marginal distributions are  $p_x$  and  $p_y$ . The mutual information between x and y is defined as

$$I(x;y) = H(x) - H(x|y) = \int_{x} \int_{y} p_{x,y} \log \frac{p_{x,y}}{p_{x}p_{y}} dxdy$$

where  $H(x) = -\int_{x} p_{x} \log p_{x} dx$  is the entropy of x and  $H(x|y) = -\int_{x} \int_{u} p_{x,y} \log p_{x|y} dx dy$  is the conditional entropy of x given y.

Unlike correlation coefficients (e.g., Pearson's correlation coefficient) which could only capture the linear dependence between two random variables, mutual information is more general in capturing both the linear and nonlinear dependence between two random variables. We have I(x;y)=0 if and only if two random variables x and y are independent to each other.

According to Lemma 1, there is an equivalence between statistical parity and zero mutual information.

Lemma 1. (Equivalence between statistical parity and zero mutual information [12, 36]). Statistical parity requires a sensitive attribute to be statistically independent to the learning results, which is equivalent to zero mutual information. Mathematically, given a learning outcome  $\tilde{\mathbf{y}}$  and the sensitive attribute s, we have

$$\underbrace{p_{\tilde{\mathbf{y}}|s} = p_{\tilde{\mathbf{y}}}}_{tatistical\ parity} \Leftrightarrow p_{\tilde{\mathbf{y}},s} = p_{\tilde{\mathbf{y}}}p_s \Leftrightarrow \underbrace{I(\tilde{\mathbf{y}};s) = 0}_{zero\ mutual\ information}$$

PROOF. Omitted for brevity.

#### 2.2 Multi-Group Fairness Problem

In order to generalize Lemma 1 from a single sensitive attribute to a set of sensitive attributes  $S = \{s^{(1)}, \ldots, s^{(k)}\}$ , we first introduce the concept of vectorized sensitive attribute  $\mathbf{s}$  given S. We define the vectorized sensitive attribute  $\mathbf{s} = [s^{(1)}, \ldots, s^{(k)}]$  as a multi-dimensional random variable where each element of  $\mathbf{s}$  represents the corresponding sensitive attribute in S (e.g.,  $\mathbf{s}[i] = s^{(i)}$  is the i-th sensitive attribute). Based on that, we have the following equivalence. For notational simplicity, we denote  $I(\tilde{\mathbf{y}}; s^{(1)}, \ldots, s^{(k)})$ ,  $p_{\tilde{\mathbf{v}}, S^{(1)}, \ldots, S^{(k)}}$  and  $p_{S^{(1)}, \ldots, S^{(k)}}$  with  $I(\tilde{\mathbf{y}}; \mathbf{s})$ ,  $p_{\tilde{\mathbf{y}}, \mathbf{s}}$  and  $p_{\mathbf{s}}$ , respectively.

$$p_{\tilde{\mathbf{y}}|\mathbf{s}} = p_{\tilde{\mathbf{y}}} \Leftrightarrow p_{\tilde{\mathbf{y}},\mathbf{s}} = p_{\tilde{\mathbf{y}}}p_{\mathbf{s}} \Leftrightarrow I(\tilde{\mathbf{y}};\mathbf{s}) = 0$$
 (1)

Based on Eq. (1), we formally define the problem of multi-group fairness in machine learning as a mutual information minimization problem, which is summarized as follows.

PROBLEM 1. MULTIFAIR: Multi-group fairness in machine learning.

**Input:** (1) a set of k sensitive attributes  $S = \{s^{(1)}, s^{(2)}, \dots, s^{(k)}\};$  (2) a set of n data points  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{s}_i, y_i) | i = 1, \dots, n\}$  where  $\mathbf{x}_i$  is the feature vector of the i-th data point,  $y_i$  is its corresponding label and  $\mathbf{s}_i = [s_i^{(1)}, \dots, s_i^{(k)}]$  describes the vectorized sensitive attributes on S of the i-th data point (with  $s_i^{(j)}$  being the corresponding attribute value of the j-th sensitive attribute  $s^{(j)}$ ); and (3) a learning algorithm represented by  $l(\mathbf{x}; \mathbf{s}; y; \tilde{\mathbf{y}}; \theta)$ , where l is the loss function,  $\tilde{\mathbf{y}} = \operatorname{argmin}_{\tilde{\mathbf{y}}} l(\mathbf{x}; \mathbf{s}; y; \tilde{\mathbf{y}}; \theta)$  is the learning outcome on the input data with  $\theta$  being model parameters.

**Output:** a set of revised learning outcomes  $\{\tilde{\mathbf{y}}^*\}$  which minimizes (1) the empirical risk  $\mathbb{E}_{(\mathbf{x},\mathbf{s},y)\sim\mathcal{D}}[l(\mathbf{x};\mathbf{s};y;\tilde{\mathbf{y}};\theta)]$  and (2) the expectation of mutual information between the learning outcomes and the sensitive attributes  $\mathbb{E}_{(\mathbf{x},\mathbf{s},y)\sim\mathcal{D}}[I(\tilde{\mathbf{y}};\mathbf{s})]$ .

**Remark:** a byproduct of MULTIFAIR is that the statistical parity can also be achieved on any subset of sensitive attributes included in S, which is summarized in Lemma 2. This could be particularly useful in that the algorithm administrator does not need to re-train the model in order to obtain fair learning results if s/he is only interested in a subset of available sensitive attributes.

Lemma 2. Consider statistical parity as the fairness notion. Given a learning outcome  $\tilde{\mathbf{y}}$ , a set of k sensitive attributes  $\mathcal{S} = \{s^{(1)}, \ldots, s^{(k)}\}$  and the vectorized sensitive attribute  $\mathbf{s} = [s^{(1)}, \ldots, s^{(k)}]$ . If  $\tilde{\mathbf{y}}$  is fair with respect to  $\mathbf{s}$ , then  $\tilde{\mathbf{y}}$  is fair with respect to any vectorized sensitive attribute  $\mathbf{s}_{sub}$  induced from the subset of sensitive attributes  $\mathcal{S}_{sub} \subseteq \mathcal{S} = \{s^{(1)}, \ldots, s^{(k)}\}$ .

PROOF. By Eq. (1), if  $\tilde{\mathbf{y}}$  is fair with respect to  $\mathbf{s} = [s^{(1)}, \dots, s^{(k)}]$ , we have  $p_{\tilde{\mathbf{y}}, \mathbf{s}} = p_{\tilde{\mathbf{y}}} p_{\mathbf{s}}$ . Let  $\mathcal{S}^{C}_{\mathrm{sub}} = \mathcal{S} \setminus \mathcal{S}_{\mathrm{sub}}$ , where  $\setminus$  denotes set minus. Then for an arbitrary subset of sensitive attribute  $\mathcal{S}_{\mathrm{sub}} \subseteq \mathcal{S}$ , we take marginal over all elements in  $\mathcal{S}^{C}_{\mathrm{sub}}$  and get

$$\int_{\forall s \in S_{\text{sub}}^{C}} p_{\tilde{\mathbf{y}},s} = \int_{\forall s \in S_{\text{sub}}^{C}} p_{\tilde{\mathbf{y}}} p_{s}$$

$$\int_{\forall s \in S_{\text{sub}}^{C}} p_{\tilde{\mathbf{y}},s} = p_{\tilde{\mathbf{y}}} \int_{\forall s \in S_{\text{sub}}^{C}} p_{s}$$

$$p_{\tilde{\mathbf{y}},s_{\text{sub}}} = p_{\tilde{\mathbf{y}}} p_{s_{\text{sub}}}$$
(2)

which implies that statistical parity is satisfied between  $\tilde{y}$  and  $s_{\text{sub}}$ 

#### 3 PROPOSED METHOD

In this section, we present a generic end-to-end algorithmic framework, named MULTIFAIR, for multi-group fairness in machine learning. We first formulate the problem as a mutual information minimization problem, and then present a variational representation of mutual information. Based on that, we present the MULTIFAIR framework to solve the optimization problem, followed by discussions on generalizations and variants of our proposed framework.

#### 3.1 Objective Function

Given a dataset  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{s}_i, y_i) | i = 1, \dots, n\}$ , the multi-group fairness in machine learning (Problem 1) can be naturally formulated as minimizing the following objective function,

$$J = \mathbb{E}_{(\mathbf{x}, \mathbf{s}, y) \sim \mathcal{D}} \left[ l(\mathbf{x}; \mathbf{s}; y; \tilde{\mathbf{y}}; \theta) + \alpha I(\tilde{\mathbf{y}}; \mathbf{s}) \right]$$
(3)

where l is a task-specific loss function for a learning task,  $\theta$  is the model parameters for the corresponding learning task,  $\tilde{\mathbf{y}}$  is the learning outcome and  $\alpha>0$  is the regularization hyperparameter. An example of loss function l is the negative log likelihood shown below.

$$l(\mathbf{x}; \mathbf{s}; y; \tilde{\mathbf{y}}; \theta) = -\log \tilde{\mathbf{y}}[y]$$

where y is the class label and  $\tilde{\mathbf{y}}$  denotes the probabilities of being classfied into the corresponding class.

To optimize the above objective function, a key challenge lies in optimizing the mutual information between the learning outcome and the vectorized sensitive feature  $I(\tilde{y}; s)$ . Inspired by the seminal work of Belghazi et al. [4], a natural choice would be to apply off-the-shelf mutual information estimation methods for highdimensional data. Examples include MINE [4], Deep Infomax [15] and CCMI [24], which estimate mutual information by parameterizing neural networks to maximize tight lower bounds of mutual information. However, in a mutual information minimization problem like Eq. (3), it is often counter-intuitive to maximize a lower bound of mutual information. Though one could still maximize the objective function of these estimators to estimate the mutual information and use such estimation to guide the optimization of Eq. (3) as a minimax game, it is hindered by two hurdles. First, it requires learning a well-trained estimator to estimate the mutual information during each epoch of optimizing Eq. (3). Second, if the estimator is not initialized with proper parameter settings, mutual information may be poorly estimated, which could further result in failing to find a good saddle point in such a minimax game.

# 3.2 Variational Representation of Mutual Information

In this paper, we take a different strategy from MINE and other similar methods by deriving a variational representation of mutual information  $I(\tilde{\mathbf{y}};\mathbf{s})$ . Our variational representation leverages a variational distribution of the vectorized sensitive feature  $\mathbf{s}$  given the learning outcome  $\tilde{\mathbf{y}}$ , which is summarized in Lemma 3.

Lemma 3. Suppose the joint distribution of the learning outcome  $\tilde{\mathbf{y}}$  and the vectorized sensitive feature  $\mathbf{s}$  is  $p_{\tilde{\mathbf{y}},\mathbf{s}}$  and the marginal distributions of  $\tilde{\mathbf{y}}$  and  $\mathbf{s}$  are  $p_{\tilde{\mathbf{y}}}$  and  $p_{\mathbf{s}}$ , respectively. Mutual information  $I(\tilde{\mathbf{y}},\mathbf{s})$  between  $\tilde{\mathbf{y}}$  and  $\mathbf{s}$  has the following variational form.

$$I(\tilde{\mathbf{y}}; \mathbf{s}) = H(\mathbf{s}) + \mathbb{E}_{(\tilde{\mathbf{y}}, \mathbf{s}) \sim p_{\tilde{\mathbf{y}}, \mathbf{s}}} \left[ \log q_{\mathbf{s} | \tilde{\mathbf{y}}} \right] + \mathbb{E}_{(\tilde{\mathbf{y}}, \mathbf{s}) \sim p_{\tilde{\mathbf{y}}, \mathbf{s}}} \left[ \log \frac{p_{\tilde{\mathbf{y}}, \mathbf{s}}}{p_{\tilde{\mathbf{y}}} q_{\mathbf{s} | \tilde{\mathbf{y}}}} \right]$$

where  $q_{s|\tilde{y}}$  is the conditional variational distribution of s given  $\tilde{y}$ .

PROOF. From the definition of mutual information, we have

$$I(\tilde{\mathbf{y}}; \mathbf{s}) = H(\mathbf{s}) - H(\mathbf{s}|\tilde{\mathbf{y}}) \tag{4}$$

where  $H(\mathbf{s}|\tilde{\mathbf{y}})$  is the conditional entropy of  $\mathbf{s}$  given  $\tilde{\mathbf{y}}$ .

Then, we rewrite the conditional entropy as follows.

$$H(\mathbf{s}|\tilde{\mathbf{y}}) = \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ -\log p_{\mathbf{s}|\tilde{\mathbf{y}}} \right]$$

$$= \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ -\log q_{\mathbf{s}|\tilde{\mathbf{y}}} \right] - \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ \log \frac{p_{\mathbf{s}|\tilde{\mathbf{y}}}}{q_{\mathbf{s}|\tilde{\mathbf{y}}}} \right]$$

$$= \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ -\log q_{\mathbf{s}|\tilde{\mathbf{y}}} \right] - \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ \log \frac{p_{\tilde{\mathbf{y}}}p_{\mathbf{s}|\tilde{\mathbf{y}}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}} \right]$$

$$= \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ -\log q_{\mathbf{s}|\tilde{\mathbf{y}}} \right] - \mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s}) \sim p_{\tilde{\mathbf{y}},\mathbf{s}}} \left[ \log \frac{p_{\tilde{\mathbf{y}},\mathbf{s}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}} \right]$$

$$(5)$$

We complete the proof by combining Eq. (4) and Eq. (5) together.

Next, we minimize the variational representation shown in Lemma 3, which contains three terms: (1) the entropy H(s), (2) the expectation of log likelihood  $\mathbb{E}_{(\tilde{y},s)\sim p_{\tilde{y},s}}[\log q_{s}|\tilde{y}]$  and (3) the expectation of log density ratio  $\mathbb{E}_{(\tilde{\mathbf{y}},\mathbf{s})\sim p_{\tilde{\mathbf{y}},\mathbf{s}}}\left[\log\frac{p_{\tilde{\mathbf{y}},\mathbf{s}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}}\right]$ . For the first term  $H(\mathbf{s})$ , we assume it to be a constant term, which can be ignored in the optimization stage. The rationale behind our assumption is that, in most (if not all) use cases, the vectorized sensitive feature s relates to the demographic information of an individual (e.g., gender, race, marital status, etc.), which should remain unchanged during the learning process. Then the remaining key challenges lie in (C1) calculating  $\log q_{\mathbf{s}|\tilde{\mathbf{y}}}$  and (C2) estimating  $\log \frac{p_{\tilde{\mathbf{y}},\mathbf{s}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}}$ . The intuition of C1 and C2 is that we strive to find a learning outcome  $\tilde{\mathbf{v}}$  such that (1)  $\tilde{\mathbf{v}}$  fails to predict the vectorized sensitive feature s (refers to C1), while (2) making it hard to distinguish if the vectorized sensitive feature s is generated from the variational distribution or sampled from the original distribution (refers to C2).

C1 – Calculating  $\log q_{\mathbf{s}|\tilde{\mathbf{y}}}$ . It can be naturally formulated as a prediction problem, where the input is the learning outcome  $\tilde{\mathbf{y}}$  and the output is the probability of  $\mathbf{s}$  being predicted. To solve it, we parameterize a decoder  $f(\tilde{\mathbf{y}};\mathbf{s};\mathbf{W})$  (e.g., a neural network) as a sensitive feature predictor to 'reconstruct' the sensitive feature  $\mathbf{s}$ , where  $\mathbf{W}$  denotes the learnable parameters in the decoder.

$$\log q_{\mathbf{s}|\tilde{\mathbf{y}}} = \log f(\tilde{\mathbf{y}}; \mathbf{s}; \mathbf{W}) \tag{6}$$

For categorical sensitive attribute,  $\log q_{\mathbf{s}|\tilde{\mathbf{v}}}$  refers to the log likelihood of classifying  $\tilde{\boldsymbol{y}}$  into label  $\boldsymbol{s},$  which can be interpreted as the negative of cross-entropy loss of the decoder  $f(\tilde{y}; s; W)$ . Moreover, if s contains multiple categorical sensitive attributes, solving Eq. (6) requires solving a multi-label classification problem, which itself is not trivial to solve. In this case, we further reduce it to a singlelabel problem by applying a mapping function *map()* to map the multi-hot encoding s into a one-hot encoding  $\hat{s}$  (i.e.,  $\hat{s} = map(s)$ ). C2 – **Estimating**  $\log \frac{p_{\tilde{y},s}}{p_{\tilde{y}}q_{s|\tilde{y}}}$ . In practice, calculating  $p_{\tilde{y},s}$  and  $p_{\tilde{y}}q_{s|\tilde{y}}$ individually is hard since the underlying distributions  $p_{\tilde{\mathbf{v}},s}$  and  $p_{\tilde{\mathbf{v}}}$ are often unknown. Recall that our goal is to estimate the log of the ratio between these two joint distributions. Therefore, we estimate it through density ratio estimation, where the numerator  $p_{\tilde{\mathbf{v}},\mathbf{s}}$  denotes the original joint distribution of the learning outcome  $\tilde{y}$  and groundtruth vectorized sensitive feature s, and the denominator  $p_{\tilde{\mathbf{v}}}q_{\mathbf{s}|\tilde{\mathbf{v}}}$ denotes the joint distribution of the learning outcome  $\tilde{y}$  and the vectorized sensitive feature  $\tilde{\mathbf{s}}$  generated from the learning outcome using the aforementioned decoder.

We further reduce this density ratio estimation problem to a class probability estimation problem, which was originally developed in [6] for solving a different problem (i.e., the classification problem with the input distribution and the test distribution differing arbitrarily). The core idea is that, given a pair of learning outcome and vectorized sensitive feature, we want to predict whether it is drawn from the original joint distribution or from the joint distribution inferred by the decoder. We label each pair of learning outcome and ground-truth vectorized sensitive feature  $(\tilde{\mathbf{y}}, \mathbf{s})$  with a positive label (c=1) and each pair of learning outcome and generated vectorized sensitive feature  $(\tilde{\mathbf{y}}, \tilde{\mathbf{s}})$  with a negative label (c=-1). After that, we rewrite the probability densities as

$$p_{\tilde{\mathbf{y}},\mathbf{s}} = \Pr[c = 1|\tilde{\mathbf{y}},\mathbf{s}]$$
  $p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}} = \Pr[c = -1|\tilde{\mathbf{y}},\mathbf{s}]$ 

Then the density ratio can be further rewritten as

$$\log \frac{p_{\tilde{\mathbf{y}},\mathbf{s}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}} = \log \frac{\Pr[c=1|\tilde{\mathbf{y}},\mathbf{s}]}{\Pr[c=-1|\tilde{\mathbf{y}},\mathbf{s}]} = \log \frac{\Pr[c=1|\tilde{\mathbf{y}},\mathbf{s}]}{1-\Pr[c=1|\tilde{\mathbf{y}},\mathbf{s}]}$$

$$= \log it(\Pr[c=1|\tilde{\mathbf{y}},\mathbf{s}])$$
(7)

Furthermore, if we model  $\Pr[c = 1 | \tilde{\mathbf{y}}, \mathbf{s}]$  using logistic regression (i.e.,  $\Pr[c = 1 | \tilde{\mathbf{y}}, \mathbf{s}] = logistic(\tilde{\mathbf{y}}, \mathbf{s})$ ), Eq. (7) is reduced to a simple linear function as

$$\log \frac{p_{\tilde{\mathbf{y}},\mathbf{s}}}{p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}} = logit(logistic(\tilde{\mathbf{y}},\mathbf{s})) = \mathbf{w}_1^T \tilde{\mathbf{y}} + \mathbf{w}_2^T \mathbf{s}$$
(8)

where both  $\mathbf{w}_1$  and  $\mathbf{w}_2$  are learnable parameters.

Putting everything together, we rewrite the objective function to be minimized in Eq. (3) as the following form.

$$J = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \left[ l(\mathbf{x}; \mathbf{s}; \mathbf{y}; \tilde{\mathbf{y}}; \theta) + \alpha \log q_{\mathbf{s}|\tilde{\mathbf{y}}} \right]$$
  
+  $\alpha \mathbb{E}_{\{(\tilde{\mathbf{y}}, \mathbf{s}) \sim p_{\tilde{\mathbf{y}}, \mathbf{s}}\} \cup \{(\tilde{\mathbf{y}}, \mathbf{s}) \sim p_{\tilde{\mathbf{y}}} q_{\mathbf{s}|\tilde{\mathbf{y}}}\}} \left[ \mathbf{w}_{1}^{T} \tilde{\mathbf{y}} + \mathbf{w}_{2}^{T} \mathbf{s} \right]$  (9)

where  $p_{\tilde{\mathbf{y}},\mathbf{s}}$  is the joint distribution of the learning outcome  $\tilde{\mathbf{y}}$  and ground-truth vectorized sensitive feature  $\mathbf{s}$ ,  $p_{\tilde{\mathbf{y}}}q_{\mathbf{s}|\tilde{\mathbf{y}}}$  is the joint distribution of the learning outcome  $\tilde{\mathbf{y}}$  and predicted vectorized sensitive feature  $\mathbf{s}$ .

#### 3.3 MultiFair: Overall Framework

Based on the objective function (Eq. (9)), we propose a generic end-to-end framework to solve the multi-group fairness problem. A general overview of the model architecture is shown in Fig. 2. Our proposed model contains four main modules, including (1) feature extractor, (2) target predictor, (3) sensitive feature predictor and (4) a density ratio estimator. In principle, as long as each module is differentiable, the proposed framework can be trained by any gradient-based optimizer through backpropagation [29].

The general workflow of our proposed MULTIFAIR framework is as follows. The pseudocode of MULTIFAIR is presented in Appendix.

- The non-sensitive features and sensitive features (optional) are passed into a feature extractor to extract the learning outcomes;
- 2. The learning outcomes will be fed into a target predictor to predict the targets for a certain downstream task (i.e.,  $l(\mathbf{x}; \mathbf{s}; y; \tilde{\mathbf{y}}; \theta)$  in Eq. (9));
- 3. The learning outcomes will be passed into the sensitive feature predictor to 'reconstruct' the vectorized sensitive features (i.e.,  $\log q_{\rm s|\tilde{v}}$  in Eq. (9));

4. Together with the learning outcomes and the ground-truth vectorized sensitive features, the predicted vectorized sensitive features will be used to estimate the density ratio between the original distribution and the variational distribution (i.e., w<sub>1</sub><sup>T</sup> ỹ + w<sub>2</sub><sup>T</sup> s in Eq. (9)).

Given a data point with categorical sensitive attribute(s), the predicted vectorized sensitive feature s is usually denoted as a one-hot vector. However, learning a one-hot vector is a difficult problem due to the discrete nature of vector elements, which makes the computation non-differentiable. To resolve this issue, we approximate such one-hot encoding by Gumbel-Softmax [16], which can be calculated as follows.

$$\mathbf{s}[i] = \frac{\exp(\left[\log(\mathbf{o_s}[i]) + g_i\right]/\tau)}{\sum_{j=1}^{n_s} \exp(\left[\log(\mathbf{o_s}[j]) + g_j\right]/\tau)}$$

where  $\mathbf{o_s}$  is the output of the sensitive feature predictor,  $n_s$  is the dimension of  $\mathbf{s}, g_1, \ldots, g_{n_s}$  are i.i.d points drawn from Gumbel(0, 1) distribution, and  $\tau$  is the softmax temperature. As  $\tau \to \infty$ , the Gumbel-Softmax samples are uniformly distributed; while as  $\tau \to 0$ , the Gumbel-Softmax distribution converges to a one-hot categorical distribution. In our framework, we start with a relatively high temperature and then anneal it during epochs of training.

#### 3.4 MultiFair: Generalizations and Variants

The proposed Multifair is able to be generalized in multiple aspects. Due to the space limitation, we only give some brief examples here, each of which could be a future direction in applying our proposed framework.

- A Relationship to adversarial debiasing. Adversarial debiasing framework [37] consists of two components: (1) a predictor that predicts the class membership probabilities using given data and (2) an adversary that takes the output of the predictor to predict the sensitive attribute of given data. The framework is optimized to minimize the loss function of the predictor while maximizing the loss function of the adversary. In MULTIFAIR, if we merge feature extractor and target predictor to one single module and remove the density ratio estimator, our framework will degenerate to the adversarial debiasing method.
- **B Relationship to Information Bottleneck.** If we set the loss function l in Eq. (3) as the negative mutual information  $-I(\tilde{\mathbf{y}};y)$ , Eq. (3) becomes the information bottleneck method [31]. Then the goal becomes to learn  $\tilde{\mathbf{y}}$  that depends on the vectorized sensitive attribute  $\mathbf{s}$  minimally and ground truth y maximally.
- C Ensuring equalized odds and equal opportunity. Analogous to the relationship between mutual information and statistical parity, ensuring equalized odds and equal opportunity can be formulated as conditional mutual information minimization problem [12]. Equalized odds is equivalent to conditional mutual information conditioned on the ground-truth label of each data point. Similarly, equal opportunity only considers data points with positive label, i.e., conditioned on y=1. To adapt Multifair into equalized odds and equal opportunity, we only need minor modifications. To be specific, in sensitive feature predictor and density ratio estimator, we calculate the corresponding log likelihoods and density ratios conditioned on the labels of input data. Then, we can minimize the corresponding conditional mutual information instead.

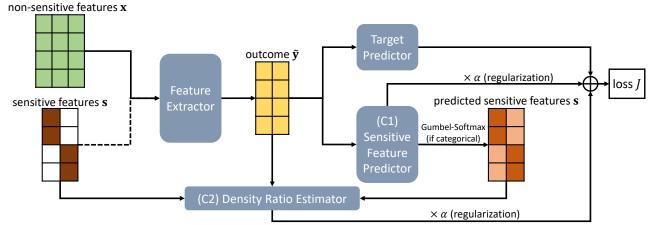


Figure 2: A General overview of our proposed MULTIFAIR framework. The dashed line between sensitive feature s and feature extractor means that sensitive features can be optionally passed into feature extractor as the input.

D - Fairness for continuous-valued sensitive features. Most existing works in fair machine learning only consider categorical sensitive attribute (e.g., gender, race). Our proposed MultiFair framework could be generalized to continuous-valued features as mutual information supports continuous-valued random variables. This advantage could empower our framework to work in even more application scenarios. For example, in image classification, we can classify images without the impact of certain image patches (e.g., patches that relate to individual's skin color). However, a major difficulty lies in modeling the variational distribution of sensitive attribute given the learning outcomes extracted from feature extractor. A potential resolution of this issue could be utilizing a generative model (e.g., VAEs [19]) as the sensitive feature predictor. E - Fairness for non-i.i.d graph data. For fair graph mining tasks, given a graph G = (A, X) where A is the adjacency matrix and X is the node feature matrix, we can use graph convolutional layer(s) as a feature extractor with the weight of the last layer to be identity matrix I and no nonlinear activation in the last graph convolution layer, in order to extract node representations. The reason for such a specific architecture in the last graph convolution layer is as follows. In general, a graph convolutional layer consists of two operations: feature aggregation and feature transformation

$$Z = f_{aggregate}(A; X) = AX$$
  $H = f_{transform}(Z; W) = \sigma(ZW)$ 

where **W** is learnable parameters and  $\sigma$  is usually a nonlinear activation. The last layer in the original GCN [20] is simply softmax(AXW), which can be viewed as a general multi-class logistic regression on the aggregated feature **Z** = **AX** (i.e., softmax(ZW)).

F – Fairness beyond classification. Note that Multifair does not have specific restrictions on the architecture of the feature extractor, target predictor or sensitive target predictor, which empowers it to handle many different types of downstream tasks by selecting the proper architecture for each module. For example, if an analyst aims to learn fair representations with respect to gender for recommendation, s/he can set the feature extractor to be a multi-layer perceptron (MLP) for learning outcome extraction, the target predictor layer to be a MLP that predicts a rating and minimizes the mean squared error (MSE) between the predicted rating and ground-truth rating, and the sensitive target predictor

to be another MLP with softmax to predict the gender based on extracted embedding.

#### 4 EXPERIMENTAL EVALUATION

In this section, we conduct experimental evaluations. All experiments are designed to answer the following questions:

**RQ1.** How does the fairness constraint impact the learning performance?

RQ2. How effective is our proposed method in mitigating bias?

#### 4.1 Experimental Settings

**A – Datasets.** We test the proposed method on three commonly-used datasets in fair machine learning research. The statistics of these datasets are summarized in Table 2.

Table 2: Statistics of datasets.

Datasets	# Samples	# Attributes	# Classes
COMPAS	6,172	52	2
Adult Income	45,222	14	2
Dutch Census	60,420	11	2

**B** – **Baseline Methods**. We compare the proposed method with several baseline methods, including *Learning Fair Representations* (*LFR*) [36], *Disparate Impact (DI)* [11], *Adversarial Debiasing (Adversarial)* [37] and *GerryFair* [17]. Detailed description of each baseline method is provided in the Appendix.

C – **Metrics.** To answer **RQ1**, we measure the performance of classification using micro F1 score (Micro F1) and macro F1 score (Macro F1). To answer **RQ2**, we measure to what extent the bias is reduced by relative bias reduction (Reduction) on average statistical imparity (Imparity). The relative bias reduction measures the relative decrease of the imparity of the debiased outcomes *Imparity*<sub>debiased</sub> to the imparity of vanilla outcomes (i.e., outcomes without fairness consideration) *Imparity*<sub>vanilla</sub>. It is computed mathematically as

$$Reduction = 1 - \frac{Imparity_{debiased}}{Imparity_{vanilla}}$$

with the average statistical imparity (Imparity) defined as *Imparity* =  $avg(|Pr(\hat{y} = c|\mathbf{x} \in g_1) - Pr(\hat{y} = c|\mathbf{x} \in g_2)|)$  for any class label c

Method	gender		race		gender & race	
	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction
Vanilla	0.9741/0.9740	0.0000	0.9741/0.9740	0.0000	0.9741/0.9740	0.0000
LFR	0.5389/0.3502	1.0000	N/A	N/A	N/A	N/A
DI	0.9741/0.9740	0.0208	0.9741/0.9740	0.0085	0.9741/0.9740	-0.1100
Adversarial	0.5746/0.5090	-0.4304	0.5746/0.5090	0.1168	0.5746/0.5090	-0.0697
GerryFair	0.9741/0.9740	0.0000	0.9465/0.9464	0.2431	0.9489/0.9487	-0.0991
MULTIFAIR (Ours)	0.8898/0.8879	0.1580	0.9392/0.9389	0.0138	0.9368/0.9363	0.0433

Table 3: Debiasing results on COMPAS dataset. Higher is better for all columns.

Table 4: Debiasing results on Adult Income dataset. Higher is better for all columns.

Method	gender		race		gender & race	
	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction
Vanilla	0.8314/0.7515	0.0000	0.8314/0.7515	0.0000	0.8314/0.7515	0.0000
LFR	0.7473/0.4277	1.0000	N/A	N/A	N/A	N/A
DI	0.8296/0.7476	0.0034	0.8305/0.7592	-0.2514	0.8266/0.7530	-0.2824
Adversarial	0.7502/0.4458	0.9508	0.7502/0.4458	0.8051	0.7502/0.4458	0.8472
GerryFair	0.8162/0.7285	-0.1458	0.8223/0.7417	-0.8160	0.8157/0.7239	-0.0077
MULTIFAIR (Ours)	0.8222/0.7318	0.0712	0.8216/0.7181	0.1093	0.8229/0.7280	0.0680

Table 5: Debiasing results on Dutch Census dataset. Higher is better for all columns.

Method	gender		marital status		gender & marital status	
	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction	Micro/Macro F1	Reduction
Vanilla	0.8343/0.8339	0.0000	0.8343/0.8339	0.0000	0.8343/0.8339	0.0000
LFR	0.5607/0.4499	0.8768	N/A	N/A	N/A	N/A
DI	0.8334/0.8327	0.0837	0.8346/0.8337	0.0095	0.8304/0.8293	0.1701
Adversarial	0.5769/0.5373	0.7738	0.5769/0.5373	0.2316	0.5769/0.5373	0.5190
GerryFair	0.8215/0.8193	0.2327	0.8259/0.8236	0.4216	0.8215/0.8193	0.0965
MultiFair (Ours)	0.8315/0.8308	0.2512	0.8304/0.8294	0.0475	0.8282/0.8277	0.1813

and any pair of two different demographic groups  $g_1$  and  $g_2$ . Note that relative bias reduction defined above can be negative if the debiased learning outcome contains more biases than the vanilla learning outcome.

More details on the experimental settings are provided in Appendix, including descriptions of datasets, data preprocessing procedures, descriptions of baseline methods, experimental protocol, model architectures, parameter settings, necessary information regarding reproducibility and additional experimental results.

#### 4.2 Main Results

We test our proposed framework, as well as baseline methods, in three different settings: debiasing binary sensitive attribute (i.e., gender for all three datasets), debiasing non-binary sensitive attribute (i.e., race for *COMPAS* and *Adult Income*, marital status for *Dutch Census*) and debiasing multiple sensitive attributes (i.e., gender & race for *COMPAS* and *Adult Income*, gender & marital status for *Dutch Census*). For each dataset and each setting, we report the results of all methods with the highest micro and macro F1 scores. This is because the algorithm administrators are often more concerned with maximizing the utility of classification algorithms. The results of *LFR* in debiasing non-binary sensitive attribute and multiple sensitive attributes are absent since it only handles binary sensitive attribute by design.

The effectiveness results of MultiFair and baseline methods on COMPAS, Adult Income and Dutch Census datasets are shown in Tables 3, 4 and 5, respectively. We provide additional results on the trade-off between micro F1 score and average statistical imparity in Appendix. From the tables, we observe that our method can mitigate bias (i.e., Reduction) effectively and consistently with a small degree of sacrifice to the vanilla classification performance (i.e., Micro/Macro F1). In addition, though LFR achieves more bias reduction, its classification performance is severely reduced by predicting all data samples as negative samples. Likewise, Adversarial achieves more bias reduction at the cost of sacrificing greatly on classification accuracy. Compared with the vanilla method, DI and GerryFair actually amplify, rather than reduce, the bias in many cases (i.e., negative reduction in tables). All in all, our proposed method achieves the best balance in reducing the bias and maintaining the classification accuracy in most cases.

#### 4.3 Ablation Study

Let  $T = \mathbb{E} \big[ l(\mathbf{x}, \mathbf{s}; y; \tilde{\mathbf{y}}; \theta) \big]$  be the empirical loss of target predictor,  $S = \alpha \mathbb{E} \big[ \log q_{\mathbf{s} | \tilde{\mathbf{y}}} \big]$  be the empirical loss of sensitive feature predictor and  $D = \alpha \mathbb{E} \big[ \mathbf{w}_1^T \tilde{\mathbf{y}} + \mathbf{w}_2^T \mathbf{s} \big]$  be the empirical loss of density ratio estimator, objective function of Multifair (Eq. (9)) can be written as J = T + S + D. To evaluate the effectiveness on optimizing the proposed variational representation of mutual information, we compare with two variants of objective function, i.e., T + S and

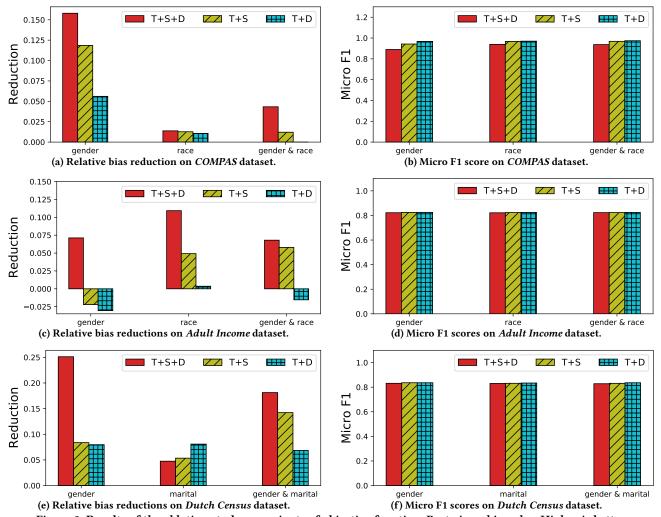


Figure 3: Results of the ablation study on variants of objective function. Best viewed in color. Higher is better.

T+D, on the same datasets and the same set of sensitive attributes as in Section 4.2. Initialization settings are kept the same among all compared objective functions (i.e., T+S+D, T+S and T+D). The results of the ablation study are shown in Figure 3. From the figure, we observe that our objective function (i.e., T+S+D) can mitigate more bias than the other two variants (i.e., T+S and T+D) in most cases. This implies that our proposed variational representation can better model the dependence between the learning outcomes and the vectorized sensitive features.

#### 5 RELATED WORK

In this section, we briefly review related literature from the following two perspectives: (1) group fairness in machine learning and (2) mutual information estimation.

A – Group fairness in machine learning aims to ensure statistical-based fairness notions across the entire populations. It has been extensively studied in many application domains, including credit scoring [11], recidivism [9], healthcare [36], recommender systems [34] and natural language processing [38]. Zemel et al. [36] use a regularized approach to learn embeddings that maximize the

parity between majority and minority groups for mapping the input data to the intermediate prototypes. Feldman et al. [11] learn a debiased input data distribution by linearly interpolating the original input distribution with a fair input distribution which is induced by maximizing the surrogate balanced error rates. Zhang et al. [37] propose an adversarial debiasing framework, which could debias when Bayes optimal is achieved by jointly learning a predictor for classification task and an adversary to predict the sensitive attributes from the learning results. Bose et al. [22] also use an adversarial training technique to encode fair representation by adding an adversary to reconstruct the input data using the learned representations together with the sensitive attribute. However, their proposed framework could only debias multiple distinct sensitive attributes instead of multiple sensitive attributes simultaneously. Kearns et al. [17] further consider fairness among the subgroups of the minority group by proposing a learner-auditor framework to play the zero-sum game through fictitious play. Different from [17], MULTIFAIR directly optimizes statistical parity through mutual information minimization instead of optimizing the self-defined surrogate 'fairness violation' functions using game-theoretic method. Adeli et al. [1] propose BR-Net, which is a convolutional neural network-based model that removes statistical dependence by minimizing Pearson's correlation. Nevertheless, BR-Net only removes the linear dependence through adversarial training whereas our proposed Multifair removes both linear and nonlinear dependence directly. In addition to statistical parity and disparate impact, Hardt et al. [14] propose another widely-used fairness notion named equal opportunity, which aims to ensure equal true positive rate between majority and minority groups.

**B** – Mutual information estimation for high-dimensional data has been made possible in recent decades by analyzing variational bounds of mutual information with machine learning techniques. Regarding variational upper bound of mutual information, Kingma et al. [19] and Rezende et al. [28] almost concurrently propose Variational Auto-Encoders (VAEs) by utilizing inference networks to maximize the Evidence Lower BOund (ELBO) objective, which could minimize a variational upper bound of mutual information conceptually. Variational lower bounds of mutual information have been extensively studied recently. Barber et al. [3] propose a variational lower bound of mutual information by introducing a variational approximation of the conditional distribution and maximize the mutual information through moment matching. Belghazi et al. [4] propose Mutual Information Neural Estimation (MINE) to estimate mutual information of two random variables by maximizing Donsker-Varadhan representation of Kullback-Leibler (KL) divergence [8] using neural networks. In [4], MINE-f, a variant of MINE, is proposed to maximize the variational estimation of f-divergence introduced by Nguyen et al. [25]. The same variational representation of f-divergence has been applied to other generative models like f-GAN [26]. Based on MINE [4], Mukherjee et al. [24] further propose a classifier-based neural estimator for conditional mutual information named CCMI. In addition, van den Oord et al. [27] propose another widely used bound of mutual information named infoNCE based on noise contrastive estimation (NCE) [13]. Hjelm et al. [15] propose Deep Infomax (DIM) to maximize the mutual information between global representation and local regions of the input, which has been further generalized to graph-structured data by Veličković et al. [32].

#### 6 CONCLUSION

In this paper, we study the problem of multi-group fairness in machine learning, where we aim to simultaneously debias the learning results with respect to multiple sensitive attributes. We formally define the multi-group fairness problem by measuring the dependence between the learning results and multiple sensitive attributes as the mutual information between learning results and a joint attribute formed by these sensitive attributes. Based on that, we formulate it as an optimization problem and further propose a generic end-toend framework, which can effectively minimize mutual information between the learning results and the joint attribute through its variational representation. We perform fair classification on three real-world datasets with the consideration of categorical sensitive attributes. The empirical evaluation results demonstrate that our proposed framework can effectively debias the classification results with respect to one or more sensitive attribute(s) with little sacrifice to the classification accuracy. Our framework is generalizable to

different settings beyond the scope of fair classification with categorical sensitive attributes in our experimental evaluation. In the future, we will investigate our framework in other learning tasks (e.g., recommendation) and its effectiveness in mitigating bias for continuous-valued sensitive attributes (e.g., age, income).

#### REFERENCES

- Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. 2019. Representation Learning with Statistical Independence to Mitigate Bias. arXiv preprint arXiv:1910.03676 (2019).
- [2] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. 2018. Interpretable Machine Learning in Healthcare. In BCB.
- [3] David Barber and Felix V Agakov. 2003. The IM Algorithm: A Variational Approach to Information Maximization. In NIPS.
- [4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual Information Neural Estimation. In ICML.
- [5] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2017. Fairness in Criminal Justice Risk Assessments: The State of the Art. arXiv preprint arXiv:1703.09207 (2017).
- [6] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative Learning under Covariate Shift. Journal of Machine Learning Research (2009).
- [7] Avishek Bose and William Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In ICML.
- [8] Monroe D Donsker and SR Srinivasa Varadhan. 1983. Asymptotic Evaluation of Certain Markov Process Expectations for Large Time. IV. Communications on Pure and Applied Mathematics (1983).
- [9] Julia Dressel and Hany Farid. 2018. The Accuracy, Fairness, and Limits of Predicting Recidivism. Science Advances (2018).
- [10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In ITCS.
- [11] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In KDD.
- [12] AmirEmad Ghassami, Sajad Khodadadian, and Negar Kiyavash. 2018. Fairness in Supervised Learning: An Information Theoretic Approach. In ISIT.
- [13] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-Contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models. In AISTATS.
- [14] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In NIPS.
- [15] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning Deep Representations by Mutual Information Estimation and Maximization. In ICLR.
- [16] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In ICLR.
- [17] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In ICML.
- [18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In ICLR.
- [19] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In ICLR.
- [20] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.
- [21] Cuicui Luo, Desheng Wu, and Dexiang Wu. 2017. A Deep Learning Approach for Credit Scoring using credit Default Swaps. Engineering Applications of Artificial Intelligence (2017).
- [22] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning Adversarially Fair and Transferable Representations. In ICML.
- [23] Scott B Morris and Russell E Lobsenz. 2000. Significance Tests and Confidence Intervals for the Adverse Impact Ratio. Personnel Psychology (2000).
- [24] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. 2020. CCMI: Classifier based Conditional Mutual Information Estimation. In UAI.
- [25] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. 2010. Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. IEEE Transactions on Information Theory (2010).
- [26] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In NIPS.
- [27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. arXiv preprint arXiv:1807.03748 (2018).
- [28] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In ICML.
- [29] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning Representations by Back-Propagating Errors. Nature (1986).
- [30] Claude E Shannon. 1948. A Mathematical Theory of Communication. The Bell System Technical Journal (1948).

- [31] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The Information Bottleneck Method. arXiv preprint physics/0004057 (2000).
- [32] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. In ICLR.
- [33] Austin Waters and Risto Miikkulainen. 2014. Grade: Machine Learning Support for Graduate Admissions. AI Magazine (2014).
- [34] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In NIPS.
- [35] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In AISTATS.
- [36] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning Fair Representations. In *ICML*.
- [37] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating Unwanted Biases with Adversarial Learning. In AIES.
- [38] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In EMNLP.

#### REPRODUCIBILITY

#### A - Pseudocode of MultiFair

```
Algorithm 1: Multifair
    Input :Training set \mathcal{D}_{train} = \{(\mathbf{x}_i, \mathbf{s}_i, y_i) | i = 1, ..., n_{train}\},
                  test set \mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, \mathbf{s}_i, y_i) | i = 1, \dots, n_{\text{test}}\},
                  regularization parameter \alpha, early stopping
                  condition c, maximum number of epochs
                  epoch<sub>max</sub>;
    Output: A set of debiased learning outcomes
                  \mathcal{Y} = {\tilde{\mathbf{y}}_i | i = 1, \dots, n_{\text{test}}}.
 1 initialize the feature extractor FE;
<sup>2</sup> initialize the target predictor TP and its loss function l_{TP};
3 initialize the sensitive feature predictor SFP and its loss
      function as the log likelihood l_{SFP}(\tilde{\mathbf{y}}; \mathbf{s}) = \log q_{\mathbf{s}|\tilde{\mathbf{v}}};
4 initialize \mathbf{w}_1 and \mathbf{w}_2 for density ratio estimation;
5 initialize the gradient-based optimizer OPT;
    // training
 6 for epoch = 1 \rightarrow \text{epoch}_{\text{max}} do
 7
          for each batch of training data \mathcal{B} \subseteq \mathcal{D}_{train} do
                initialize loss J = 0;
 8
                for each data point (x, s, y) \in \mathcal{B} do
                      get the learning outcome \tilde{\mathbf{y}} = FE(\mathbf{x}; \mathbf{s});
10
                      accumulate target predictor's loss
11
                        J = J + l_{TP}(\tilde{\mathbf{y}}; y);
                      predict sensitive feature \tilde{\mathbf{s}} = SFP(\tilde{\mathbf{y}});
12
                      accumulate sensitive feature predictor's loss
13
                       J = J + \alpha l_{SFP}(\tilde{\mathbf{y}}; \mathbf{s});
                      if s is one-hot then
14
                       \tilde{\mathbf{s}} = Gumbel-Softmax(\tilde{\mathbf{s}});
15
                      accumulate the estimated density ratio
16
                      J = J + \alpha \mathbf{w}_1^T \tilde{\mathbf{y}} + \alpha (\mathbf{w}_2^T \mathbf{s} + \mathbf{w}_2^T \tilde{\mathbf{s}})/2;
                calculate the empirical loss J = J/|\mathcal{B}|;
17
                update all learnable parameters by OPT(\nabla J);
18
          if c is satisfied then
19
                stop training;
    // test
21 get debiased learning outcomes
      \mathcal{Y} = {\tilde{\mathbf{y}}_i = FE(\mathbf{x}_i; \mathbf{s}_i) | (\mathbf{x}_i, \mathbf{s}_i, y) \in \mathcal{D}_{\text{test}}, i = 1, \dots, n_{\text{test}}};
22 return \mathcal{Y};
```

#### **B** – Dataset Descriptions

For all datasets, we randomly split them into 80% training set and 20% test set. A description of each dataset is shown below.

COMPAS dataset contains in total of 6,172 criminal defendants in Broward County, Florida. Each defendant is described by 52 attributes used by the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm for scoring their likelihood of reoffending crimes in the following 2 years. The goal is to determine whether a criminal defendant will reoffend in the next 2 years.

- Adult Income dataset contains in total of 45,222 individuals. Each individual is described by 14 attributes that relate to his/her personal demographic information, including gender, race, education, marital status, etc. The goal is to predict whether a person can earn a salary over \$50,000 a year.
- Dutch Census dataset contains in total of 60,420 individuals.
   Each individual is described by 11 attributes that relate to his/her demographic and economic information to predict whether s/he has a prestigious occupation.

#### C - Procedures for Data Preprocessing

COMPAS Dataset. We first remove data samples whose duration between screening date and arrest date is within the range of [-30, 30]. As for the features used in our experiments, we remove duplicate features and features related to date, case number and descriptions of criminal charge. In addition, we manually create a new feature by calculating the length of stay in jail for each criminal and then quantize them into three bins: length less than 1 week, length larger than 1 week but less than 3 months and length larger than 3 months. Similarly, for features related to count of criminal charges, we quantize them into three bins: count equals to 0, count within 1 to 3, and count more than 3. The rest of preprocessing procedures are as follows: (1) continuous-valued features are kept as is; (2) binary discrete-valued features are transformed into a single boolean value and (3) non-binary discrete-valued feature are transformed into one-hot encoding.

Adult Income and Dutch Census Datasets. We use all features included in the original dataset and remove data samples with missing values. As for detailed preprocessing procedures, we follow similar procedures as processing COMPAS dataset. We keep each continuous-valued feature as it is and transform the nonbinary discrete-valued feature into one-hot encoding. For binary discrete-valued feature, we transform it into a single boolean value for further use. After these procedures, the feature dimension of Adult Income dataset is transformed from 14 to 104 and the feature dimension of Dutch Census dataset is transformed from 11 to 73.

#### **D** – Descriptions of Baseline Methods

**Learning Fair Representations (LFR)** [36] is an algorithm that ensures group and individual fairness through learning a set of fair prototype representations. Each data sample is first mapped to a prototype, which is used to predict fair outcome. We use the implementation provided in IBM AIF360<sup>5</sup>. As for parameter settings, we use the default number of prototypes as described in the implementation provided by IBM AIF360, and find the best hyperparameters using the same grid search strategy as described in [36], i.e.,  $A_x = 0.01$ ,  $A_y$  is found from the set  $\{0.1, 0.5, 1, 5, 10\}$  and  $A_z$  is searched in  $\{0, 0.1, 0.5, 1, 5, 10\}$ .

**Disparate Impact (DI)** [11] ensures disparate impact by interpolating the original data distribution with an unbiased distribution. We choose this method since disparate impact is analogously similar to statistical parity, both of which ensures a small discrepancy between acceptance rates for majority and minority groups. For fair comparison, we set the linear interpolation coefficient, which

<sup>&</sup>lt;sup>5</sup>https://aif360.mybluemix.net/

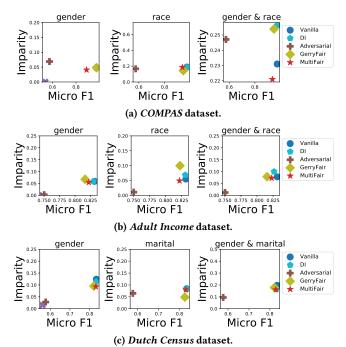


Figure 4: Trade-off between micro F1 score and average statistical imparity. Best viewed in color. Red star represents MULTIFAIR. The closer to bottom right, the better trade-off between micro F1 score and average statistical imparity. Bias is amplified by an algorithm if its corresponding point is located above the blue dot (i.e., Vanilla).

is referred to as  $\lambda$  in [11], such that the interpolation ratios of [11] and ours are the same, i.e.,  $\frac{1-\lambda}{\lambda} = \frac{1}{\alpha}$ .

Adversarial Debiasing (Adversarial) [37] ensures statistical parity by introducing an adversary to predict the sensitive attribute using the predicted outcome obtained from a predictor, where the predictor and adversary can be flexibly chosen by the algorithm administrator. Since its official source code is not available, we implement the model using the same machine configurations as our proposed framework. The original paper set both the predictor and the adversary as the logistic regression classifier. For fair comparison, we switch (1) the predictor to feature extractor and target predictor in our proposed framework and (2) the adversary to sensitive feature predictor in our framework. We also set the same learning rate as our framework.

**GerryFair** [17] ensures subgroup fairness through fictitious play by formulating the fair learning process as a two-player zero-sum game between a learner and an auditor. Objectives for both the learner and the auditor are formulated as empirical risk of costsensitive classification. As for parameter settings, since the relationship between  $\alpha$  in Multifair and parameters of *GerryFair* is

unclear, we use the default parameters provided in the officially released source code, i.e., C=15,  $\gamma=0.01$  and maximum number of iterations is set to 10.

## **E – Detailed Experimental Protocol and Model Architectures**

The learning task we consider is fair classification with respect to categorical sensitive attribute(s). For all datasets, we take both non-sensitive features and sensitive features as input to the feature extractor.

As for the detailed model architecture, for *Adult Income* and *Dutch Census* datasets, the feature extractor is a one-layer MLP with hidden dimension 32 to extract the embeddings; the target predictor contains one hidden layer that calculates the log likelihood of predicting class label using the extracted embeddings; and the sensitive feature predictor is similar to the target predictor that leverages one hidden layer to calculate the log likelihood of predicting the vectorized sensitive feature using the extracted embeddings. For *COMPAS* dataset, we set the feature extractor to be a two-layer MLP with hidden dimension 32 in each layer, while keeping the target predictor and the sensitive feature predictor to be the same as it is for *Adult Income* and *Dutch Census* datasets.

#### F - Parameter Settings and Repeatability

For all datasets, we set the regularizatio in parameter  $\alpha=0.1.$  The number of epochs for training is set to 100 with a patience of 5 for early stopping. Weight decay is set to 0.01. We tune the learning rate as 0.001 for DI and 0.0001 for Adversarial and our method. All learnable model parameters are optimized with Adam optimizer [18]. The starting temperature for Gumbel-Softmax is set to 1 and is divided by 2 every 50 epochs for annealing. To reduce randomness and enhance reproducibility, we run 5 different intializations with random seed from 0 to 4.

#### **G** - Machine Configurations

All experiments are performed on a Windows PC with i7-9700K CPU, 32GB RAM. All three datasets are publicly available online. Codes are programmed in Python 3.8 with PyTorch 1.7.0. Models (i.e., MULTIFAIR and baseline methods) are trained on CPU only. We will release the source code upon the publication of the paper.

### H – Trade-off between Micro F1 Score and Average Statistical Imparity

The results of trade-off between micro F1 score (Micro F1) and average statistical imparity (Imparity) is shown in Figure 4. From the figure, we can observe that, compared with other baseline methods, our method achieves the best trade-off between preserving classification accuracy and reducing bias (i.e., being closer to the bottom right corner in Figure 4) in most cases.