FairIF: Boosting Fairness in Deep Learning via Influence Functions with Validation Set Sensitive Attributes

Haonan Wang*
haonan.wang@u.nus.edu
National University of Singapore
Singapore

Ziwei Wu*
ziweiwu2@illinois.edu
University of Illinois at Urbana-Champaign
USA

Jingrui He
jingrui@illinois.edu
University of Illinois at Urbana-Champaign
USA

ABSTRACT

Empirical loss minimization during machine learning training can inadvertently introduce bias, stemming from discrimination and societal prejudices present in the data. To address the shortcomings of traditional fair machine learning methods—which often rely on sensitive information of training data or mandate significant model alterations—we present FairIF, a unique two-stage training framework. Distinctly, FairIF enhances fairness by recalibrating training sample weights using the influence function. Notably, it employs sensitive information from a validation set, rather than the training set, to determine these weights. This approach accommodates situations with missing or inaccessible sensitive training data. Our FairIF ensures fairness across demographic groups by retraining models on the reweighted data. It stands out by offering a plug-and-play solution, obviating the need for changes in model architecture or the loss function. We demonstrate that the fairness performance of FairIF is guaranteed during testing with only a minimal impact on classification performance. Additionally, we analyze that our framework adeptly addresses issues like group size disparities, distribution shifts, and class size discrepancies. Empirical evaluations on three synthetic and five real-world datasets across six model architectures confirm FairIF's efficiency and scalability. The experimental results indicate superior fairness-utility trade-offs compared to other methods, regardless of bias types or architectural variations. Moreover, the adaptability of FairIF to utilize pretrained models for subsequent tasks and its capability to rectify unfairness originating during the pretraining phase are further validated through our experiments.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**.

1 INTRODUCTION

In automatic, high-stake decision-making systems, the use of machine learning techniques is commonplace. In spite of the effectiveness of these machine learning techniques, recent works [21] have uncovered algorithmic discrimination across demographic groups in real-world applications, which raises severe fairness concerns [16]. In response, there has been a flurry of research on fairness in machine learning [12, 36], with a primary emphasis on proposing formal notions of fairness [22, 56] and "de-biasing" techniques to achieve these goals [2, 17].

The issue of indirect discrimination in deep learning algorithms has garnered substantial research interest because of its profound implications for output fairness. Addressing it directly, such as removing sensitive attributes during training, is inadequate to guarantee equality due to its intricate causes [11]. In order to ensure that the system is not biased against some sensitive features, previous methods either heavily rely on the sensitive information in the training data to construct training objective functions [15, 57], or add additional modules to the original model to ensure fairness and balance in the predictions despite the presence of potentially biased data [1, 58]. In general, the vast majority of works on fairness assume that sensitive information, such as gender or race, is contained in the training set and that the model's design is completely accessible and modifiable [30, 35]. However, in many scenarios, it is difficult to gather or use sensitive information for decision-making due to privacy or legal restrictions [45]. Moreover, in numerous real-world applications [48], the developed models rigorously adhere to state-of-the-art architectures to achieve the desired performance. Modifying complex designs, *e.g.,* introducing an additional Multi-Layer Perceptron (MLP) branch for fairness, will degrade the performance or introduce extra complication into the model tuning. Therefore, algorithms that rely solely on sensitive data or require substantial modifications to target methods are typically difficult to implement in practice. In this study, we pose the following research question:

Can we design a practical method which can yield models with better fairness performance when we cannot modify the architecture of the target model or have access to a great amount of sensitive information?

We provide an affirmative answer to the above question and introduce the FairIF model, a mechanism that enhances fairness in models by reweighting training samples. Distinctively, FairIF preserves the model's original architecture, facilitating its seamless integration with a broad spectrum of machine learning models optimized by gradient descent. This framework promotes enhanced fairness metrics including equality of opportunity, odds, and accuracy. Crucially, FairIF doesn't mandate the inclusion of sensitive data within the training samples—only a modest validation set annotated with group labels is needed. The methodology unfolds in two phases. Initially, the Influence Function (IF) [28] is employed

*Both authors contributed equally to this research.

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635844

to quantify sample influence, *i.e.*, the change in model prediction if a training sample is up- (down-)weighted by a specific amount. Subsequently, sample weights are optimized to ensure that the given model achieves performance equality across all groups of the validation set. In the succeeding phase, the model undergoes retraining on the weighted dataset, addressing fairness concerns while maintaining inherent performance. Furthermore, an analysis underscores that the performance gap between the original and retrained models remains constrained, and the fairness performance is guaranteed when tested.

The experiments are threefold. First, on synthetic CI-MNIST dataset [39], by independently generating three different types of bias, (1) difference group sizes, (2) difference class distributions within each group, and (3) difference class sizes, FairIF successfully mitigates the unfairness caused by different group sizes, class distributions, and class sizes. This shows that the method can handle a variety of biases in a controlled environment. Second, with commonly used deep neural networks on three real-world fairness datasets and two image datasets [26, 34], FairIF improves the values of multiple fairness metrics while maintaining or even enhancing model accuracy. This performance is superior to five previous state-of-the-art methods. This illustrates that FairIF not only scales well but also surpasses other methods in handling complex, uncontrolled, real-world data. Third, even when used with different pretrained models, FairIF manages to alleviate the unfairness within these models without detriment to their performance. This highlights the adaptability and versatility of FairIF across different model architectures and initializations. Additionally, we empirically analyze the role of the validation set in FairIF. The result indicates that FairIF is able to achieve desired performance with only a small amount of validation set, which makes the proposed method feasible and desirable for real-world applications in which sensitive information is hard to collect. Besides, to further study how our method achieves fairness, we also examine what examples are reweighted by FairIF. Examining the specific examples reweighted by FairIF provides one way to better understand the intermediate process, which is critical for trustworthiness and further improvement of the method.

2 RELATED WORK

Influence Function. Originating in 1970s statistics, the influence function was designed to measure a model's dependence on specific training samples. It was later incorporated into machine learning, notably aiding in interpreting predictions by assessing each sample's impact [28]. Numerous extensions followed, such as Barshan et al.'s work focusing on local influence relative to global effects [5], and Basu et al.'s exploration of the collective influence of large training groups [8, 27]. Other studies accelerated inference for over-parameterized neural networks [10, 20]. Of note, existing influence function methods assume the Hessian matrix's positive definiteness. Many dynamically compute this before model convergence, risking inaccuracies. Our FairIF method awaits full model convergence, ensuring more accurate and time-efficient computations.

Fairness-aware machine learning. Fair machine learning aims to counteract biases in automated systems. In classification, group fairness demands consistent classification error across protected-attribute groups [22, 56]. Existing solutions often involve extensive model and training modifications [52, 58, 61] or rely on sensitive data [18, 47]. Some address cases without the sensitive attribute [30, 53] while others use sample reweighing [25, 29], which can be costly due to frequent retraining. From model repair, methods have been proposed leveraging counterfactual distributions [46] or sample influence estimation [42]. While some, like Wang et al., employed the Influence Function for instance-level fairness constraints [47], others used it to compute sample weights to bridge fairness gaps [32]. However, such methods often need entire training sets' sensitive data or entail computationally demanding steps, such as computing the Hessian matrix per sample and using a solver for linear programming problems, making them unsuitable for large-scale datasets in deep learning models [32, 47]. Our FairIF method stands out, ensuring fairness using only group annotations on a small validation set, without changing the model's architecture. This introduces a fresh avenue for bolstering model fairness.

3 PRELIMINARY

3.1 Notation

We consider the setting where each sample consists of an input $x \in \mathcal{X}$, a label $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the input and output space respectively, and each example has a corresponding sensitive attribute $s \in \mathcal{S}$. For simplicity's sake, assume that $\mathcal{S} = \{0, 1\}$. Let K denotes the number of classes, $[K] := \{1, 2, .., K\}$. We mainly focus on the classification problem and denote the classifier $h(x) = \arg\max_{i \in [K]} f_\theta^i(x)$, where $f_\theta \in \mathbb{R}^K$ is a neural network parameterized by $\theta \in \Theta$. We denote the number of parameters as P, then $\Theta \subseteq \mathbb{R}^P$. Denote the training data set of size n as $\mathcal{D} = \{z_1, z_2, ..., z_n\}$, where $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, and the validation data set (with sensitive attributes) of size m as $\mathcal{D}^s = \{z_1^s, z_2^s, ..., z_m^s\}$, where $z_j^s = (x_j, y_j, s_j) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{S}$. In this work, we are interested in the setting where the sensitive attribute s is not available for the training set, as collecting large amount of data with the sensitive attributes is typically expensive and at the risk of privacy leakage [53]. Instead, we assume that we have access to a small validation set with annotations of the sensitive attribute. The standard training procedure minimizes the empirical risk $\mathcal{L}(\mathcal{D}, \theta)$,

$$\theta^\star = \arg\min_{\theta \in \Theta} \mathcal{L}(\mathcal{D}, \theta) = \sum_{i=1}^{n} \frac{1}{n} \ell(z_i, \theta), \tag{1}$$

where $\ell(z_i, \theta) : \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathbb{R}^+$ is the loss function (*e.g.*,cross entropy loss function) for z_i. Our goal is to learn a fair neural network $f_{\theta_\epsilon^\star}$ parameterized by $\theta_\epsilon^\star \in \Theta$, through minimizing the weighted loss $\mathcal{L}_\epsilon$:

$$\theta_\epsilon^\star = \arg\min_{\theta \in \Theta} \mathcal{L}_\epsilon(\mathcal{D}, \theta) = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} (\frac{1}{n} + \epsilon_i)\, \ell(z_i, \theta), \tag{2}$$

where $\epsilon = [\epsilon_1, \epsilon_2, ..., \epsilon_n]^\top \in \mathbb{R}^n$ is a reweight vector of the training samples. Next, we introduce the fairness notions used in this work. For notation convenience, we denote the true positive rate on group 1 (sensitive attribute $s = 1$) as $\text{TPR}^{(1)} = \mathbb{P}(h = 1 \mid s = 1, y = 1)$ where h is the classifer's prediction, and the true positive rate on group 0 (sensitive attribute $s = 0$) as $\text{TPR}^{(0)} = \mathbb{P}(h = 1 \mid s = 0, y = 1)$. Similarly, we denote the true negative rate on group 1

as $\text{TNR}^{(1)} = \mathbb{P}(h = 0 \mid s = 1, y = 0)$ and the true negative rate on group 0 as $\text{TNR}^{(0)} = \mathbb{P}(h = 0 \mid s = 0, y = 0)$.

Accuracy Equality requires the classification system to have equal misclassification rates across sensitive groups [56]: $\mathbb{P}(h(x) = y|s = 0) = \mathbb{P}(h(x) = y|s = 1)$. Then, we define the *Accuracy Difference (AD)* as:

$$AD = |\mathbb{P}(h(x) = y|s = 0) - \mathbb{P}(h(x) = y|s = 1)|. \tag{3}$$

Equal Odds is defined as $\mathbb{P}(h = 1 \mid s = 1, y = y') = \mathbb{P}(h = 1 \mid s = 0, y = y'), \forall y' \in \{0, 1\}$. It is sometimes also referred to as disparate mistreatment, aiming to equalize the *true positive* and *false positive rates* for a (binary-) classifier [22]. Following [37, 53], we define *Average Odds Difference (AOD)* as:

$$AOD = \frac{1}{2}\left[|\text{TPR}^{(1)} - \text{TPR}^{(0)}| + |\text{TNR}^{(1)} - \text{TNR}^{(0)}|\right]. \tag{4}$$

Equal Opportunity is weaker than Equal Odds, but it typically allows for stronger utility [22]: $\mathbb{P}(h = 1 \mid s = 1, y = 1) = \mathbb{P}(h = 1 \mid s = 0, y = 1)$. Also, following [37, 53], we define *Equality of Opportunity Difference (EOD)* as:

$$EOD = |\text{TPR}^{(1)} - \text{TPR}^{(0)}|. \tag{5}$$

3.2 Influence Function

The method of Influence Function (IF) [28] aims at approximating how the minimizer of the loss function $\theta^\star$ would change if we were to reweight the i-th training example. The key idea is to make a first-order approximation of change in $\theta^\star$ around $\epsilon_i = 0$ with Taylor expansion. Specifically, if the i-th training sample is upweighted by a small ϵ_i, then the perturbed risk minimizer $\theta^\star_{\epsilon_i}$ becomes:

$$\theta^\star_{\epsilon_i} \triangleq \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell\left(z_i, \theta\right) + \epsilon_i\, \ell\left(z_i, \theta\right). \tag{6}$$

The change of model parameters due to the introduction of the weight ϵ_i is:

$$\theta^\star_{\epsilon_i} - \theta^\star \approx \frac{d\theta^\star_{\epsilon_i}}{d\epsilon_i}\Big|_{\epsilon_i=0}\epsilon_i = \underline{-H^{-1}_{\theta^\star}\nabla_\theta \ell(z_i, \theta^\star)}\epsilon_i = \underline{\mathcal{I}_{param}(z_i)}\epsilon_i, \tag{7}$$

where $H_{\theta^\star} = \frac{1}{n}\sum_{i=1}^{n}\nabla^2_\theta \ell\left(z_i, \theta^\star\right)$ is the Hessian of the objective at $\theta^\star$, and $\mathcal{I}_{param}(z_i) \in \mathbb{R}^P$ denotes the influence of sample z_i on the model parameters. Following the assumption adopted in previous works [5, 27, 28, 44], $H_{\theta^\star}$ is positive definite. Note, this assumption is relatively weak, under the condition that $\theta^\star$ is the minimizer of the loss function. Similarly, the change in loss can be approximated [5] as:

$$\begin{aligned}\ell\left(z, \theta^\star_{\epsilon_i}\right) - \ell\left(z, \theta^\star\right) &\approx \frac{d\ell\left(z, \theta^\star_{\epsilon_i}\right)}{d\epsilon_i}\epsilon_i \\ &= \underline{-\nabla_\theta \ell\left(z, \theta^\star\right)^\top H^{-1}_{\theta^\star}\nabla_\theta \ell(z_i, \theta^\star)}\epsilon_i \\ &= \underline{\mathcal{I}_{loss}(z_i, z)}\epsilon_i,\end{aligned} \tag{8}$$

where the term $\mathcal{I}_{loss}(z_i, z) \in \mathbb{R}$ represents the influence of sample z_i on the loss computed over sample z.

4 METHOD

4.1 Correcting Discrepancy by Sample Reweighting

The foundational insight underpinning FairIF is that the influence of a training sample on the model prediction can be estimated through the influence function. This approach has been theoretically proven to offer precise estimations for linear models and has been empirically validated as effective in real-world applications [19, 27, 28, 60]. Specifically, For any continuously differentiable functions $F(\mathcal{D}, \theta) \in \mathbb{R}$, $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$, the change of F with respect to ϵ_i, the sample weight of z_i, can be computed based on the influence function:

$$\begin{aligned}&F(\mathcal{D}, \theta^\star_{\epsilon_i}) - F(\mathcal{D}, \theta^\star) \\ &\approx \frac{1}{|\mathcal{D}|}\frac{d\sum_j^{|\mathcal{D}|}\left(F(\{z_j\}, \theta^\star_{\epsilon_i}) - F(\{z_j\}, \theta^\star)\right)}{d\theta^\star_{\epsilon_i}}\frac{d\theta^\star_{\epsilon_i}}{d\epsilon_i}\epsilon_i \\ &= -\frac{1}{|\mathcal{D}|}\sum_j^{|\mathcal{D}|}\nabla_\theta F(\{z_j\}, \theta^\star)^\top H^{-1}_{\theta^\star}\nabla_\theta \ell(z_i, \theta^\star)\epsilon_i.\end{aligned} \tag{9}$$

The Equation (9) can be regarded as an extension of the conclusion of Equation (8). To apply the conclusion of Equation (9) to the fairness setting, we define the function F as a general metric measuring fairness. Denote the two different groups of the validation set $\mathcal{D}^s$ as $\mathcal{D}^0$ and $\mathcal{D}^1$ respectively, where for any $z^s_j = (x_j, y_j, s_j) \in \mathcal{D}^0, s_j = 0$ and $z_{j'} = (x_{j'}, y_{j'}, s_{j'}) \in \mathcal{D}^1, s_{j'} = 1$. Then, for the two groups, the change of function F caused by permuting sample weights are,

$$F(\mathcal{D}^0, \theta^\star_\epsilon) - F(\mathcal{D}^0, \theta^\star) = -\frac{1}{|\mathcal{D}^0|}\sum_{z_i\in\mathcal{D}, z_j\in\mathcal{D}^0}\nabla_\theta F(z_j)^\top H^{-1}_{\theta^\star}\nabla_\theta \ell(z_i, \theta^\star)\epsilon_i, \tag{10}$$

and,

$$F(\mathcal{D}^1, \theta^\star_\epsilon) - F(\mathcal{D}^1, \theta^\star) = -\frac{1}{|\mathcal{D}^1|}\sum_{z_i\in\mathcal{D}, z_{j'}\in\mathcal{D}^1}\nabla_\theta F(z_{j'})^\top H^{-1}_{\theta^\star}\nabla_\theta \ell(z_i, \theta^\star)\epsilon_i. \tag{11}$$

Then, the goal of achieving fairness is to achieve equalized performance F over the two groups, i.e., solving $\epsilon^\star$ to satisfy the following equation,

$$F(\mathcal{D}^0, \theta^\star_{\epsilon^\star}) - F(\mathcal{D}^1, \theta^\star_{\epsilon^\star}) = 0. \tag{12}$$

Combining Equations (10)(11)(12), we have:

$$\begin{aligned}&F(\mathcal{D}^0, \theta^\star_{\epsilon^\star}) - F(\mathcal{D}^1, \theta^\star_{\epsilon^\star}) - \left(F(\mathcal{D}^0, \theta^\star) - F(\mathcal{D}^1, \theta^\star)\right) \\ &= \Big(\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'}\in\mathcal{D}^1}\nabla_\theta F(z_{j'}) - \frac{1}{|\mathcal{D}^0|}\sum_{z_j\in\mathcal{D}^0}\nabla_\theta F(z_j)\Big)^\top H^{-1}_{\theta^\star}\sum_{z_i\in\mathcal{D}}\nabla_\theta \ell(z_i, \theta^\star)\epsilon^\star_i,\end{aligned}$$

Notably, the second term $F(\mathcal{D}^0, \theta^\star) - F(\mathcal{D}^1, \theta^\star)$ denotes the metric discrepancy of unweighted model across two demographic groups. We denote this empirically measurable discrepancy under metric F as $\text{diff}(\mathcal{D}^s, F, \theta^\star)$. Then we have the following equation,

$$\begin{aligned}F(\mathcal{D}^0, \theta^\star_{\epsilon^\star}) - F(\mathcal{D}^1, \theta^\star_{\epsilon^\star}) = \text{diff}(\mathcal{D}^s, F, \theta^\star) + \Big(&\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'}\in\mathcal{D}^1}\nabla_\theta F(z_{j'}) \\ &- \frac{1}{|\mathcal{D}^0|}\sum_{z_j\in\mathcal{D}^0}\nabla_\theta F(z_j)\Big)^\top H^{-1}_{\theta^\star}\sum_{z_i\in\mathcal{D}}\nabla_\theta \ell(z_i, \theta^\star)\epsilon^\star_i,\end{aligned}$$

In order to find $\epsilon^\star$ making $F(\mathcal{D}^0, \theta^\star_{\epsilon^\star}) - F(\mathcal{D}^1, \theta^\star_{\epsilon^\star}) = 0$, we introduce the following optimization problem,

$$\epsilon^\star = \arg\min_{\epsilon}\Big[\text{diff}(\mathcal{D}^s, F, \theta^\star) + \Big(\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'}\in\mathcal{D}^1}\nabla_\theta F(z_{j'}) - \frac{1}{|\mathcal{D}^0|}\sum_{z_j\in\mathcal{D}^0}\nabla_\theta F(z_j)\Big)^\top H_{\theta^\star}^{-1}\sum_{z_i\in\mathcal{D}}\nabla_\theta \ell(z_i,\theta^\star)\epsilon_i\Big]^2. \tag{13}$$

In the subsequent section, we will leverage this conclusion to address practical fairness challenges.

4.2 FairIF: Achieving Fairness through Influence Function Reweighting

We now present FairIF, a straightforward two-stage methodology that eliminates the need for group annotations within the training set or alterations to the original models. Initially, we determine the sample weights using the influence function to ensure a balanced TPR and TNR performance across varied groups. Subsequently, in the second stage, we train the final model utilizing the reweighted training samples.

Stage One. To address discrepancies across three fairness notions—AD, AOD, and AOE—FairIF aims to equalize True Positive Rate (TPR) and True Negative Rate (TNR) between two groups by adjusting the sample weights, denoted as ϵ. Section 5.1 offers an in-depth analysis of metric selection, highlighting that equalizing TPR and TNR serves as an effective fairness objective and can alleviate disparities under the aforementioned notions. However, since TPR and TNR are non-differentiable, they impede the use of Equation (13) to determine ϵ. To circumvent this, we adopt the gumbel softmax technique [24] to approximate and render TPR and TNR differentiable. These approximations are represented as F_{TPR} and F_{TNR} respectively. Then, the objective can be defined as:

$$\begin{aligned}\min_{\epsilon}\Big[&\text{diff}(\mathcal{D}^s, F_{TPR}, \theta^\star) + \Big(\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'}\in\mathcal{D}^1}\nabla_\theta F_{TPR}(z_{j'}) - \frac{1}{|\mathcal{D}^0|}\sum_{z_j\in\mathcal{D}^0}\nabla_\theta F_{TPR}(z_j)\Big)^\top H_{\theta^\star}^{-1}\sum_{z_i\in\mathcal{D}}\nabla_\theta \ell(z_i,\theta^\star)\epsilon_i\Big]^2 \\ &+\Big[\text{diff}(\mathcal{D}^s, F_{TNR}, \theta^\star) + \Big(\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'}\in\mathcal{D}^1}\nabla_\theta F_{TNR}(z_{j'}) - \frac{1}{|\mathcal{D}^0|}\sum_{z_j\in\mathcal{D}^0}\nabla_\theta F_{TNR}(z_j)\Big)^\top H_{\theta^\star}^{-1}\sum_{z_i\in\mathcal{D}}\nabla_\theta \ell(z_i,\theta^\star)\epsilon_i\Big]^2 + \lambda\|\epsilon\|_2,\end{aligned} \tag{14}$$

The first and second terms aim at balancing the TPR and TNR between groups with ϵ. And the last one is the regularization with weight factor $\lambda \in \mathbb{R}_+$, which enforces the weights close to zero as assumed by the influence function [28].

Stage Two. Next, we train the final model $f_{\theta^\star_{\epsilon^\star}}$ by reweighting the training samples with $\epsilon^\star$. The weighted loss is,

$$\mathcal{L}_{\epsilon^\star}(\mathcal{D},\theta) = \sum_{i=1}^{n}\Big(\frac{1}{n} + \epsilon_i^\star\Big)\, l\,(z_i,\theta)\,. \tag{15}$$

We present the detailed FairIF training algorithm in Appendix B, Algorithm 1. Initially, the model is rigorously trained using standard empirical risk until reaching convergence, resulting in the parameter $\theta^\star$. Subsequent to this, we calculate the influence function and determine the appropriate weights to achieve equality under metrics TPR and TNR across groups in the validation set. In the final step, we train the model with reweighted data. Note, differently from previous methods [40, 44] computing influence function on the fly, in our method, the influence function is computed after the model is converged, which not only saves the computation time of influence function, but also delivers a more accurate estimation. We leave the computation details of the influence function in Appendix C.

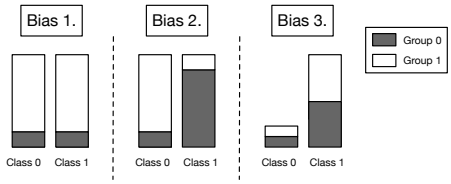

Figure 1: Illustration of three types of bias.

While Equation (14) effectively reduces disparity on the validation set, largely due to overparameterization [59], its applicability to the testing phase and potential impact on classification performance remain questions. Subsequent sections, Section 5 and 6, address these concerns. Specifically, Section 5 demonstrates bounded disparities between validation and testing fairness metrics, and the task performance guarantee on the test set. Section 6, meanwhile, empirically evaluates FairIF across eight datasets and six models.

5 ANALYSIS

This section details FairIF's characteristics, focusing on minimizing True Positive Rate (TPR) and True Negative Rate (TNR) discrepancies between groups (Section 5.1). This approach addresses three fairness notions: AD, AOE, and AOD. By exploring the root causes of these discrepancies, we demonstrate that effectively minimizing the disparity in TPR and TNR can alleviate these fairness concerns. The rationale presented solidifies the optimization objective chosen by FairIF. Section 5.2 examines how fairness in the validation phase, achieved through optimized sample weights, extends to the testing phase. This is analyzed using TPR and TNR disparities and the Rademacher complexity. In Section 5.3, we showcase the difference in classification accuracy between the original and the retrained models remains minimal when the model converges.

5.1 Mitigating Disparity under Different Notions

In this section, we delve into an analytical examination underscoring the significance of balancing True Positive Rates (TPR) and True Negative Rates (TNR) across varied groups. The TPR and TNR equalization is pivotal in alleviating disparities in line with several fairness definitions. Consistent with the findings of prior research [39, 55], the ingrained bias in the models predominantly emerges from the following sources within the training data: 1. Variances in group sizes; 2. Discrepancies in class distribution within individual groups, often termed as the group distribution shift; 3. Inequalities in class sizes. To offer a clearer perspective on these bias forms, refer to Figure 1 where we have depicted a visualization of these three distinct bias categories. With analysis of the relation between the three fairness notions and the TPR and TNR metrics, we derive the following proposition:

Proposition 1. *Under the existence of three types of data bias, if the model prediction satisfies equalized TPR and TNR, i.e.,* $TPR^0 -$

$TPR^1 = 0$ and $TNR^0 - TNR^1 = 0$, then the three notions of fairness can be achieved, $AD = AOD = EOD = 0$.

Proof sketch. Given the definitions of Average Odds Difference (AOD) and Equality of Opportunity Difference (EOD) detailed in Section 3.1, it becomes straightforward to deduce that both AOD and EOD would be zero if the TPR and TNR are harmonized between two distinct groups. Let's delve into the Accuracy Difference (AD). It can be expressed as:

$$AD = |\alpha TPR^{(0)} - \beta TPR^{(1)} + (1-\alpha)TNR^{(0)} - (1-\beta)TNR^{(1)}|,$$

Here, $\mathbb{P}(y = 1|s = 0)$ and $\mathbb{P}(y = 1|s = 1)$ are represented by the variables α and β, respectively. Further dissecting the three forms of bias on a case-by-case basis, it becomes evident that AD either mirrors or is constrained by the combination of differences in TPR and TNR across these scenarios. The comprehensive proof of Proposition 1 has been catalogued for reference in Appendix A.

The aforementioned proposition demonstrates that by harmonizing the TPR and TNR across two groups, we can simultaneously achieve three pivotal notions of fairness, namely: Accuracy Equality, Equal Odds, and Equal Opportunity. This rationale supports our decision to employ a metric that combines both TPR and TNR within FairIF.

5.2 Fairness Guarantee on Test Data

In the preceding section, we show that the metric combining TPR and TNR serves as an effective measure of fairness, and our proposed method aspires to ensure equality in TPR and TNR across distinct groups. In this section, our focus shifts to substantiating that the fairness performance remains consistent during testing. Before delving into the fairness assurances offered by FairIF, it's essential to familiarize ourselves with the concept of Rademacher complexity[6], which measures the learnability of function classes. The Rademacher complexity for a function class is defined as below:

Definition 1. *Given a space Z, and a set of i.i.d. examples $S = \{z_1, z_2, ..., z_m\} \subseteq Z$, for a function class $\mathcal{F}$ where each function $r : Z \to \mathbb{R}$, the empirical Rademacher complexity of $\mathcal{F}$ is given by:*

$$\widehat{\mathrm{Rad}}_S(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{r\in\mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^{m} \sigma_i r(z_i) \right) \right] \tag{16}$$

Here, $\sigma_1, ..., \sigma_m$ are independent random variables uniformly drawn from $\{-1, 1\}$.

In what follows, we establish the bound for TPR disparity during test. A parallel argument can be made for TNR. As indicated in Section 3, the classifier is denoted by $h(x)$. However, diverging from prior notation for the sake of derivation convenience in this section, we modify the binary output of the classifier to range in $\{-1, 1\}$. Its corresponding space, $\mathcal{H}$, encompasses hypotheses with values in $\{-1, 1\}$. In order to align with the definition of Rademacher complexity, we represent the TPR disparity between two groups on dataset $\mathcal{D}$ as $r_h(\mathcal{D})$. Here, the TPR difference r_h acts as a function dependent on the classifier function h. Consequently, the function class $\mathcal{F}$ is articulated as:

$$\mathcal{F} = L(\mathcal{H}) \triangleq \left\{ r_h(\mathcal{D}) \to \left| \frac{1}{|\mathcal{D}^{1,y=1}|} \sum_{(x,y=1)\in\mathcal{D}^1} \mathbb{1}_{\{h(x)=1\}} - \frac{1}{|\mathcal{D}^{0,y=1}|} \sum_{(x,y=1)\in\mathcal{D}^0} \mathbb{1}_{\{h(x)=1\}} \right| : h \in \mathcal{H} \right\}.$$

where $\mathcal{D}^{1,y=1}$ represent the positive samples from group 1 within $\mathcal{D}$, while $\mathcal{D}^{0,y=1}$ signifies the equivalent for group 0. From this definition, it's clear that the range of $\mathcal{F}$ is bounded within $[0, 1]$.

For simplicity in notation, we use S, rather than $\mathcal{D}^S$, to represent the validation set utilized by our method for sample weight computation. And the size of sample set S is m. Then, let's define $R(h) = \mathbb{E}_{\mathcal{D}^{\text{test}}}[r_h(\mathcal{D}^{\text{test}})]$ as the TPR fairness risk of the classifier h on the test set. Meanwhile, $\hat{R}_S(h) = \frac{1}{m}\sum_{z_i\in S} r_h(z_i)$ represents the empirical fairness risk of the classifier h on the sample set S. Note, samples within S and $\mathcal{D}^{\text{test}}$ are i.i.d. and originate from the same data distribution. Based on the property of Rademacher complexity [43], for any $\delta > 0$, with probability at least $1 - \delta$ over S:

$$\forall h \in \mathcal{H}: \quad R(h) \le \hat{R}_S(h) + 2 * \underline{\widehat{Rad}_S(L(\mathcal{H}))} + 3\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{17}$$

Before we discuss the details of each terms of the previous inequality, let's denote S_1^+ and S_0^+ as the positive samples in group 1 and group 0 in the sample set S respectively. Then for the second term on the right-hand side (RHS) of Equation (17), we can express it in terms of $\widehat{\mathrm{Rad}}_S(\mathcal{H})$:

$$\begin{aligned}
&\widehat{\mathrm{Rad}}_S(L(\mathcal{H})) = \widehat{\mathrm{Rad}}_S(\mathcal{F}) \\
=&\mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{|S_1^+|} \sum_{x_i\in S_1^+} \sigma_i \frac{1+h(x_i)}{2} - \frac{1}{|S_0^+|} \sum_{x_i\in S_0^+} \sigma_i \frac{1+h(x_i)}{2} \right] \\
\le&\mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{|S_1^+|} \sum_{x_i\in S_1^+} \sigma_i \frac{1+h(x_i)}{2} \right] + \mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{|S_0^+|} \sum_{x_i\in S_0^+} \sigma_i \frac{1+h(x_i)}{2} \right] \\
=&\mathbb{E}_\sigma \left[\frac{1}{2|S_1^+|} \sum_{i=1}^{|S_1^+|} \sigma_i + \frac{1}{2} \sup_{h\in\mathcal{H}} \frac{1}{|S_1^+|} \sum_{x_i\in S_1^+} \sigma_i h(x_i) \right] \\
&+ \mathbb{E}_\sigma \left[\frac{1}{2|S_0^+|} \sum_{i=1}^{|S_0^+|} \sigma_i + \frac{1}{2} \sup_{h\in\mathcal{H}} \frac{1}{|S_0^+|} \sum_{x_i\in S_0^+} \sigma_i h(x_i) \right] \\
=&\frac{1}{2}\mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{|S_1^+|} \sum_{x_i\in S_1^+} \sigma_i h(x_i) \right] + \frac{1}{2}\mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{|S_0^+|} \sum_{x_i\in S_0^+} \sigma_i h(x_i) \right] = \underline{\widehat{\mathrm{Rad}}_S(\mathcal{H})}
\end{aligned}$$

Our theoretical analysis elucidates key insights from Equation (17). Specifically, the second term on the right-hand side primarily reflects the Rademacher complexity of the model. This complexity is intricately connected to the family of neural networks to which the model belongs, and is constrained by moderate assumptions [51]. Concurrently, due to the power of over-parameterization [59], the first term, $\hat{R}_S(h)$, approaches zero (the discrepancy on the validation set can be optimized to nearly vanish). The third term's magnitude correlates with the validation set size m, and as this size expands, our bound tightens. It's noteworthy that for current deep learning datasets, a validation set, even if relatively small compared to the training set, is often ample. Empirical validations of this claim are further explored in Appendix G.

5.3 Accuracy Guarantee on Test Data

In this subsection, our primary objective is to understand the model's task performance on the test dataset. To achieve this, we represent the test loss, using $\theta_\epsilon^\star$ for each sample, leveraging the first-order Taylor approximation:

$$\ell\left(z_{\text{test}}, \theta_\epsilon^\star\right) = \ell\left(z_{\text{test}}, \theta^\star\right) + \mathcal{I}_{\text{loss}}\ \left(z_{\text{test}}, \mathcal{D}\right)\epsilon + O\left(\|\epsilon\|^2\right). \quad (18)$$

For many benchmark datasets, $\|\epsilon\|^2 \leq \frac{c}{n^2}$ holds [54], where c represents the data quantity for which $\epsilon_i \neq 0$. Following the Influence Function methodology in [28], we omit the term $O\left(\|\epsilon\|^2\right)$ in Equation (18). Expanding this further to encompass the test loss over the test set $\mathcal{D}^{\text{test}}$, we derive:

$$\begin{aligned}
\mathcal{L}(\mathcal{D}^{\text{test}}, \theta_\epsilon^\star) - \mathcal{L}\left(\mathcal{D}^{\text{test}}, \theta^\star\right) &\approx \mathbb{E}_{z_{\text{test}} \in \mathcal{D}^{\text{test}}}\left[\mathcal{I}_{\text{loss}}\left(z_{\text{test}}, \mathcal{D}\right)\right]\epsilon \\
&= \left[-\nabla_\theta \mathcal{L}\left(\mathcal{D}^{\text{test}}, \theta^\star\right)^\top\right]\left[\sum_{z_i \in \mathcal{D}} H_{\theta^\star}^{-1} \nabla_\theta \ell\left(z_i, \theta^\star\right)\right]\epsilon \\
&\leq \left\|\nabla_\theta \mathcal{L}(\mathcal{D}^{\text{test}}, \theta^\star)\right\|_2 \left\|\sum_{z_i \in \mathcal{D}} \mathcal{I}_{\text{param}}\ (z_i)\right\|_2 \|\epsilon\|_2 \\
&\leq \left\|\nabla_\theta \mathcal{L}(\mathcal{D}^{\text{test}}, \theta^\star)\right\|_2 \cdot \gamma \cdot \|\epsilon\|_2
\end{aligned} \quad (19)$$

The concluding inequality assumes a positive real number γ exists such that $\left\|\sum_{z_i \in \mathcal{D}} \mathcal{I}_{\text{param}}\ (z_i)\right\|_2 \leq \gamma$. Even though this assumption aligns with earlier works [54], it's crucial to note that $\left\|\sum_{z_i \in \mathcal{D}} \mathcal{I}_{\text{param}}\ (z_i)\right\|_2 \leq \sum_{z_i \in \mathcal{D}} \left\|\mathcal{I}_{\text{param}}\ (z_i)\right\|_2$. This term signifies the influence of each training sample on the test loss. Given that empirically introducing or excluding individual one training sample only marginally impacts the test loss, the assumption of γ is validated. Moreover, with the objective pushing $\|\epsilon\|_2$ towards zero, variations in test performance between our fair model and the original unweighted model are constrained.

6 EXPERIMENTS

In our experiments, we initially underscore the effectiveness of FairIF in mitigating fairness concerns arising from three distinct bias types. This is illustrated using the synthetic dataset, CI-MNIST (Section 6.1). Subsequently, we compare FairIF against five baseline approaches, effectively illustrating the method's performance and scalability on real-world datasets (Section 6.2). We further showcase the empirical results obtained when integrating FairIF with pre-existing models (Section 6.3). Finally, we examine the data instances that undergo reweighting in the training set to achieve fairness, assessing their alignment with human preference (Section 6.4). We leave the empirically investigation of the size of validation set to FairIF's operation in Appendix G.

Regarding datasets employed, our experiments leverage three variations of the synthetic image dataset **CI-MNIST**[39]. Additionally, three real-world tabular datasets: **Adult**[4], **German**[4], and **COMPAS**[14], along with two genuine image datasets, **CelebA** [34] and **FairFace**[26], are integrated into our study. For the comparative baselines, we utilize the following approaches: **CFair**[61], **DOMIND**[50], **ARL**[30], **FairSMOTE**[13], and **Influence** [32]. We leave the details about datasets, baselines, scalable implementation of influence function, and configuration details in appendix C, D, E, and F.

6.1 Synthetic Experiments

The synthetic dataset CI-MNIST [39] is a variant of the MNIST dataset. In this dataset, each input image x has a label $y \in \{0, 1\}$ indicating odd or even respectively, and the background color, blue or red, is the sensitive attribute. CI-MNIST offers manual control over different types of bias in the dataset, such as the number of samples in each group and class, and the group distribution over the class. To examine the effectiveness of FairIF under different types of bias, we independently introduce the three major types of bias analyzed in Section 5 into the dataset. We compare FairIF with the five other baselines on three CI-MNIST variants based on three models: a multilayer perception (MLP), a convolutional network (CNN) and a LeNet[31]. The experimental results are shown in Table 1, Table 2 and Table 3 respectively. We report the results of the epochs with the best task accuracy performance. We provide the visualization of three different types of data bias in Figure 1.

Bias 1: Group Size Discrepancy. We set 15% of images in the two classes as red and the remaining as blue. The distributions over classes of each group are the same, and the number of images in each class is also the same. In this setting, the group of red background is under-representative.

Bias 2: Group Distribution Shift. We keep the amount of data within each class and each group to be the same, but set 85% of images in class 0 with blue background and 15% of images in class 1 with blue background. In this case, the group distributions over the classes are different.

Bias 3: Class Size Discrepancy. We set group distributions and the total amount of data within each group to be the same. And the amount of data in class 1 is 25% of class 0.

As shown in Table 1, 2 and 3, compared with the original model trained by ERM, FairIF can achieve lower discrepancies under three different notions of fairness, which demonstrates that FairIF can mitigate the fairness issue caused by the three different types of bias. In the meantime, we notice that FairIF mostly achieves higher accuracy than the original model under different biases, indicating the spurious correlation problem is alleviated by our sample reweighting mechanism. Also, we observe that the Influence baseline often achieves a high fairness level but at an unsatisfactory sacrifice of task performance, although we have conducted a hyperparameter grid search for the best fairness-utility trade-off. Further, comparing FairIF with all other state-of-the-art methods, we find that FairIF mostly has the top two performance on all the metrics regardless of the model and the dataset. This shows FairIF can achieve better fairness-utility trade-offs, even compared with methods directly using all the group information (e.g., FairSMOTE and DOMIND) and additional adversarial models (e.g., CFair and ARL).

6.2 Real-world Experiments

In this study, we evaluate FairIF in comparison to five baseline models, utilizing three distinct tabular datasets: Adult [4], German [4], and COMPAS [14]. To mitigate the risk of overfitting on these datasets, we follow the previous works [32, 47] and adopt the logistic regression for testing. Results can be found in Table 4. Furthermore, to demonstrate the effectiveness and scalability of our proposed method with large datasets and advanced models, we further tests on image datasets CelebA [34] and FairFace [26] were conducted. For fairness, all baseline models were adapted using

Table 1: (MLP) Comparison of FairIF with baselines on CI-MNIST with different types of bias. The Original row shows the performances of the model trained with ERM. For AD, AOD and EOD, smaller values indicate better fairness performance. Bold font is used to highlight the best result and the underscores are for the second-best result.

	Group Size Discrepancy				Group Distribution Shift				Class Size Discrepancy			
Models	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD
Original	97.55	1.104	0.013	0.011	98.19	0.256	0.065	0.056	97.27	0.818	0.010	0.012
CFair	97.09	1.745	0.018	0.026	97.49	**0.060**	0.041	0.034	96.09	0.798	0.009	0.011
DOMIND	92.96	1.162	0.020	0.031	97.23	0.696	0.074	0.047	96.64	0.798	0.027	0.051
ARL	96.95	1.424	0.014	0.017	97.62	0.374	0.085	0.070	96.54	0.335	0.008	0.015
FairSMOTE	96.18	1.031	0.014	0.016	97.36	0.292	0.054	0.038	96.25	0.310	**0.006**	**0.010**
Influence	97.40	**0.708**	**0.007**	0.010	97.11	0.760	**0.037**	**0.009**	97.25	0.443	0.010	0.019
FairIF	**98.01**	1.012	0.011	**0.006**	**98.49**	0.223	0.053	0.031	**97.71**	**0.108**	0.008	0.011

Table 2: (CNN) Comparison of FairIF with baselines on CI-MNIST with different types of bias. Bold font highlights the best result and the underscores are for the second-best result.

	Group Size Discrepancy				Group Distribution Shift				Class Size Discrepancy			
Models	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD
Original	98.46	1.090	0.011	0.011	98.81	0.577	0.043	0.030	98.65	0.488	0.006	0.008
CFair	98.62	**0.233**	**0.002**	**0.001**	97.17	2.755	0.007	0.008	98.37	**0.095**	**0.002**	**0.003**
DOMIND	**98.97**	0.779	0.008	0.015	98.88	0.248	0.007	0.004	**98.92**	0.437	0.006	0.009
ARL	98.35	0.999	0.010	0.012	98.69	0.599	0.049	0.028	98.43	0.482	0.004	**0.003**
FairSMOTE	97.94	0.541	0.007	0.008	98.58	0.241	0.009	0.006	97.92	0.383	0.005	0.004
Influence	98.13	0.468	0.006	0.003	98.42	0.235	0.049	0.037	97.99	0.475	0.005	**0.003**
FairIF	98.69	0.405	0.006	0.007	**98.94**	**0.234**	**0.004**	**0.003**	98.80	0.317	0.005	0.004

Table 3: (LeNet) Comparison of FairIF with baselines on CI-MNIST with different types of bias. Bold font highlights the best result and the underscores are for the second-best result.

	Group Size Discrepancy				Group Distribution Shift				Class Size Discrepancy			
Models	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD
Original	99.11	0.464	0.005	0.006	99.17	0.153	0.026	0.018	98.81	0.152	**0.001**	**0.001**
CFair	99.07	**0.204**	0.005	0.007	99.16	0.118	0.020	**0.012**	98.65	0.127	0.003	0.004
DOMIND	99.12	0.318	0.004	0.007	**99.54**	0.058	0.021	0.014	99.12	0.177	0.004	0.006
ARL	99.10	0.278	**0.003**	0.004	98.86	0.172	0.034	0.028	98.69	0.266	0.003	0.004
FairSMOTE	98.96	0.226	0.004	0.006	99.03	0.091	0.022	0.021	98.74	0.118	0.003	0.004
Influence	98.91	0.287	**0.003**	**0.001**	98.67	0.168	0.042	0.030	99.05	**0.064**	0.004	0.004
FairIF	**99.13**	0.375	0.004	0.003	99.42	**0.042**	**0.018**	0.013	**99.25**	0.088	0.002	0.003

Table 4: Comparison of FairIF with baselines on three real-world tabular datasets. Bold font is used for the best values.

	Adult				German				COMPAS			
Methods	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD
Original	69.76	17.40	0.302	0.303	**75.50**	15.45	0.083	0.041	**68.09**	2.09	0.183	0.220
CFair	**70.78**	16.97	0.312	0.345	69.50	11.01	0.018	**0.020**	66.29	0.29	0.399	0.455
DOMIND	68.51	15.92	0.222	**0.237**	69.00	10.26	0.072	0.103	67.89	0.36	0.174	0.240
ARL	69.49	20.53	0.373	0.388	74.50	12.16	0.027	0.036	64.30	0.65	0.197	0.269
FairSMOTE	68.43	16.67	0.268	0.271	64.12	9.92	0.053	0.082	66.14	1.42	0.198	0.242
Influence	68.79	15.85	0.269	0.268	72.01	13.25	0.056	0.107	67.32	1.01	0.164	**0.163**
FairIF	68.97	**15.34**	**0.170**	0.265	74.68	**8.68**	**0.017**	0.024	67.53	**0.28**	**0.156**	0.218

Table 5: Comparison of FairIF with baselines on two real-world image datasets. Bold font is used for the best values.

	FairFace				CelebA			
Methods	Acc(%)	AD(%)	AOD	EOD	Acc(%)	AD(%)	AOD	EOD
Original	86.68	7.23	0.0721	0.0610	**95.41**	4.54	0.2485	0.4975
CFair	**86.72**	5.12	0.0509	0.0358	95.11	4.23	0.1954	0.3520
DOMIND	86.69	4.86	0.0496	0.0172	94.88	4.18	0.1864	0.3475
ARL	86.59	5.59	0.0078	0.0125	95.08	4.31	0.2185	0.4108
FairSMOTE	86.42	5.86	0.0488	0.0267	95.26	4.42	0.2256	0.3872
FairIF	86.63	**4.14**	**0.0048**	**0.0105**	95.37	**3.81**	**0.1220**	**0.3145**

Table 6: Performance of FairIF with Pretrained Models. The relative change caused by using FairIF is presented in percentage. Smaller values for AD, AOD, and EOD indicate larger discrepancy mitigation.

	FairFace				CelebA			
Models	Acc	AD	AOD	EOD	Acc	AD	AOD	EOD
+ResNet-18	+0.40%	-35.81%	-48.81%	-66.87%	+0.07%	-23.73%	-37.08%	-23.51%
+ResNet-34	-0.17%	-21.05%	-77.96%	-77.76%	-0.21%	-9.66%	-25.78%	-7.80%
+ResNet-50	-0.01%	-9.46%	-37.21%	-49.90%	-0.34%	-29.02%	-28.22%	-9.76%

ResNet-18 as a feature extractor. It's worth noting that results for the Influence baseline on image datasets were omitted due to prohibitive computational demands. These findings are presented in Table 5. Hyperparameters for all methodologies were tuned based on validation set performance.

As shown in Table 4 and 5, we find that just a small amount of group information on the validation set can allow FairIF to achieve low performance discrepancies between demographic groups. In the meantime, the accuracy performance of FairIF is very close to the original method, which supports our theoretical guarantees. Compared with the Influence baseline, FairIF often achieves a similar or better fairness level while maintaining better task performance on the three tabular datasets. And FairIF is compatible with deep neural networks on large-scale image datasets while the Influence baseline is not. Furthermore, even comparing with methods that directly use group annotation on the training set, e.g. CFair, DOMIND and FairSMOTE, FairIF can still most often outperform them in both accuracy and fairness metrics regardless of the backbone model, which proves its efficacy and scalability.

6.3 Debiasing Pretrained Models

As observed in previous research [49], the pretrained models deliver discriminative results across different demographic groups, which might be caused by the pretraining procedure and the pretraining dataset collected from social networks, international online newspapers, and web searches. The proposed method FairIF yields fair models through changing the sample weights, which makes FairIF a promising approach to remove the discrimination encoded in the pretrained parameters. Note, for methods requiring a modification of the network structure, the power of pretraining usually cannot be fully utilized. In this section, we aim to answer the question of how FairIF mitigates the fairness issue of the pretrained models. We evaluate FairIF with three commonly used pretrained models, ResNet-18, ResNet-34, and ResNet-50 [23]. We finetune and evaluate these pretrained models on FairFace and CelebA and compare them with their counterparts trained with FairIF. The relative changes caused by using FairIF are presented in percentage in Table 6. With three different pretrained models on the two datasets, FairIF consistently mitigates the discrepancies of three different notions of fairness without hurting the accuracy performance.

6.4 Sample Weight Study

We now probe into how FairIF achieves low performance discrepancies across different groups with the same or higher accuracy. In order to perform this analysis, we use the group annotations on the training data to closely examine what examples are up-weighted or down-weighted in the training set. Note, we don't use the group information in the training stage. We show the samples mostly upweights and downweights by FairIF in Figure 2. For the CelebA dataset, we observe that FairIF tends to give more weight to examples of men with blond hair and white hair, while it decreases the weight of examples of females with blond hair. The result is as expected, the group (Male, Blond Hair) is the minority and the group (Female, Blond Hair) is the majority. For the FairFace dataset, FairIF tends to upweight the samples of people with dark skin and downweight the samples of people with white skin. This is aligned with our expectation because the accuracy for group $s = 1$ (white) is higher than the group $s = 0$ (black). By upweighting the minority group and downweighting the majority group, the performance tends to balance. Note that FairIF computes different weights for different samples, and it is thus smarter than simply equalizing the weights of the samples from different groups in different classes in other reweighting methods such as FairSMOTE.

(a) Samples from the CelebA dataset

(b) Samples from the FairFace dataset

Figure 2: Examples of the reweighting done by FairIF. In each figure, mostly up-weighted samples are presented in the first row, and mostly down-weighted are in the second.

7 CONCLUSION

Empirical loss minimization in training machine learning models can unintentionally amplify inherent discrimination and societal biases. Recognizing this, we presented FairIF, a novel two-stage training framework. Distinct from methods that rely heavily on sensitive training data or demand major model alterations, FairIF re-trains on a weighted dataset. These weights, derived using the influence function, ensure uniform model performance across diverse groups. The unique selling point of FairIF is its adaptability: it integrates seamlessly with models using stochastic gradient descent without altering the training algorithm, requiring only group annotations from a small validation set. Theoretically, we showcased that the performance delta between the reweighted-data model and the original optimal one remains finite, and fairness discrepancies during testing across groups are limited. By addressing these discrepancies, FairIF adeptly tackles disparities stemming from group and class size variations, as well as distribution shifts. Empirical assessments on synthetic datasets underscore FairIF's capability in producing models that better balance fairness and utility. Tests on real-world datasets further vouch for its efficiency and scalability. Additionally, our exploration with pretrained models demonstrates FairIF's prowess in leveraging their strengths for subsequent tasks, all the while rectifying fairness issues from their training stages.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, IIS-2137468, and IIS-2002540. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. 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*. AAAI Press, 2412–2420. https://doi.org/10.1609/aaai.v33i01.33012412

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. 2018. A Reductions Approach to Fair Classification. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 60–69. http://proceedings.mlr.press/v80/agarwal18a.html

[3] Naman Agarwal, Brian Bullins, and Elad Hazan. 2016. Second-order stochastic optimization in linear time. *stat* 1050 (2016), 15.

[4] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.

[5] Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. 2020. RelatIF: Identifying Explanatory Training Samples via Relative Influence. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy] (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 1899–1909. http://proceedings.mlr.press/v108/barshan20a.html

[6] Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. 2002. Localized Rademacher Complexities. In *Computational Learning Theory, 15th Annual Conference on Computational Learning Theory, COLT 2002, Sydney, Australia, July 8-10, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2375)*, Jyrki Kivinen and Robert H. Sloan (Eds.). Springer, 44–58. https://doi.org/10.1007/3-540-45435-7_4

[7] Samyadeep Basu, Phillip Pope, and Soheil Feizi. 2021. Influence Functions in Deep Learning Are Fragile. In *Proc. of ICLR*. OpenReview.net. https://openreview.net/forum?id=xHKVVHGDOEk

[8] Samyadeep Basu, Xuchen You, and Soheil Feizi. 2020. On Second-Order Group Influence Functions for Black-Box Predictions. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 715–724. http://proceedings.mlr.press/v119/basu20b.html

[9] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *ArXiv preprint* abs/1810.01943 (2018). https://arxiv.org/abs/1810.01943

[10] Zalán Borsos, Mojmir Mutny, and Andreas Krause. 2020. Coresets via Bilevel Optimization for Continual Learning and Streaming. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/aa2a77371374094fe9e0bc1de3f94ed9-Abstract.html

[11] Alessandro Castelnovo, Riccardo Crupi, Greta Greco, Daniele Regoli, Ilaria Giuseppina Penco, and Andrea Claudio Cosentini. 2022. A clarification of the nuances in the fairness metrics landscape. *Scientific Reports* 12, 1 (2022), 4209.

[12] Simon Caton and Christian Haas. 2020. Fairness in machine learning: A survey. *ArXiv preprint* abs/2010.04053 (2020). https://arxiv.org/abs/2010.04053

[13] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in Machine Learning Software: Why? How? What to do?. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

[14] William Dieterich, Christina Mendoza, and Tim Brennan. 2016. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc* 7, 7.4 (2016), 1.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[16] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *FAccT*. PMLR, 172–186.

[17] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, Longbing Cao, Chengqi Zhang, Thorsten Joachims, Geoffrey I. Webb, Dragos D. Margineantu, and Graham Williams (Eds.). ACM, 259–268. https://doi.org/10.1145/2783258.2783311

[18] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. 2018. Beyond Distributive Fairness in Algorithmic Decision Making: Feature Selection for Procedurally Fair Learning. In *Proc. of AAAI*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 51–60. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16523

[19] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying Large Language Model Generalization with Influence Functions. arXiv:2308.03296 [cs.LG]

[20] Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. 2021. FastIF: Scalable Influence Functions for Efficient Model Interpretation and Debugging. In *Proc. of EMNLP*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 10333–10350. https://doi.org/10.18653/v1/2021.emnlp-main.808

[21] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic Bias: From Discrimination Discovery to Fairness-aware Data Mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 2125–2126. https://doi.org/10.1145/2939672.2945386

[22] Moritz Hardt, Eric Price, and Nati Srebro. 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*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 3315–3323. https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[24] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proc. of ICLR*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[25] Heinrich Jiang and Ofir Nachum. 2020. Identifying and Correcting Label Bias in Machine Learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy] (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 702–712. http://proceedings.mlr.press/v108/jiang20a.html

[26] Kimmo Karkkainen and Jungseock Joo. 2021. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1548–1558.

[27] Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. 2019. On the Accuracy of Influence Functions for Measuring Group Effects. 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*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5255–5265. https://proceedings.neurips.cc/paper/2019/hash/a78482ce76496fcf49085f2190e675b4-Abstract.html

[28] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1885–1894. http://proceedings.mlr.press/v70/koh17a.html

[29] Emmanouil Krasanakis, Eleftherios Spyromitros Xioufis, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2018. Adaptive Sensitive Reweighting to Mitigate Bias in Fairness-aware Classification. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 853–862. https://doi.org/10.1145/3178876.3186133

[30] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. 2020. Fairness without Demographics through Adversarially Reweighted Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/07fc15c9d169ee48573edd749d25945d-Abstract.html

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[32] Peizhao Li and Hongfu Liu. 2022. Achieving Fairness at No Utility Cost via Data Reweighing with Influence. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 12917–12930. https://proceedings.mlr.press/v162/li22p.html

[33] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just Train Twice: Improving Group Robustness without Training Group Information. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6781–6792. http://proceedings.mlr.press/v139/liu21f.html

[34] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 3730–3738. https://doi.org/10.1109/ICCV.2015.425

[35] David Madras, Elliot Creager, Toniann Pitassi, and Richard S. Zemel. 2018. Learning Adversarially Fair and Transferable Representations. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 3381–3390. http://proceedings.mlr.press/v80/madras18a.html

[36] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[37] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Rishabh Iyer. 2021. BiFair: Training Fair Models with Bilevel Optimization. *ArXiv preprint* abs/2106.04757 (2021). https://arxiv.org/abs/2106.04757

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[39] Charan Reddy, Deepak Sharma, Soroush Mehri, Adriana Romero, Samira Shabanian, and Sina Honari. 2021. Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics. (2021).

[40] Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. 2020. Not All Unlabeled Data are Equal: Learning to Weight Data in Semi-supervised Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/f7ac67a9aa8d255282de7d11391e1b69-Abstract.html

[41] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally Robust Neural Networks. In *Proc. of ICLR*. OpenReview.net. https://openreview.net/forum?id=ryxGuJrFvS

[42] Prasanna Sattigeri, Soumya Ghosh, Inkit Padhi, Pierre Dognin, and Kush R Varshney. 2022. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems* 35 (2022), 35894–35906.

[43] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[44] Stefano Teso, Andrea Bontempelli, Fausto Giunchiglia, and Andrea Passerini. 2021. Interactive Label Cleaning with Example-based Explanations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 12966–12977. https://proceedings.neurips.cc/paper/2021/hash/6c349155b122aa8ad5c877007e05f24f-Abstract.html

[45] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.

[46] Hao Wang, Berk Ustun, Flavio P Calmon, and SEAS Harvard. 2018. Avoiding disparate impact with counterfactual distributions. In *NeurIPS Workshop on Ethical, Social and Governance Issues in AI*.

[47] Jialu Wang, Xin Eric Wang, and Yang Liu. 2022. Understanding Instance-Level Impact of Fairness Constraints. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 23114–23130. https://proceedings.mlr.press/v162/wang22ac.html

[48] Mei Wang and Weihong Deng. 2021. Deep face recognition: A survey. *Neurocomputing* 429 (2021), 215–244.

[49] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. 2019. Racial Faces in the Wild: Reducing Racial Bias by Information Maximization Adaptation Network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 692–702. https://doi.org/10.1109/ICCV.2019.00078

[50] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. 2020. Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 8916–8925. https://doi.org/10.1109/CVPR42600.2020.00894

[51] Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. 2018. On the margin theory of feedforward neural networks. (2018).

[52] Ziwei Wu and Jingrui He. 2022. Fairness-aware Model-agnostic Positive and Unlabeled Learning. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 1698–1708.

[53] Shen Yan, Hsien-Te Kao, and Emilio Ferrara. 2020. Fair Class Balancing: Enhancing Model Fairness without Observing Sensitive Attributes. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1715–1724. https://doi.org/10.1145/3340531.3411980

[54] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. 2023. Dataset Pruning: Reducing Training Data by Examining Generalization Influence. In *The Eleventh International Conference on Learning Representations*.

[55] Zhe Yu. 2021. Fair Balance: Mitigating Machine Learning Bias Against Multiple Protected Attributes With Data Balancing. *ArXiv preprint* abs/2107.08310 (2021). https://arxiv.org/abs/2107.08310

[56] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1171–1180. https://doi.org/10.1145/3038912.3052660

[57] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 962–970. http://proceedings.mlr.press/v54/zafar17a.html

[58] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.

[59] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (2021), 107–115.

[60] Jieyu Zhang, Haonan Wang, Cheng-Yu Hsieh, and Alexander Ratner. 2022. Understanding Programmatic Weak Supervision via Source-aware Influence Function. *ArXiv preprint* abs/2205.12879 (2022). https://arxiv.org/abs/2205.12879

[61] Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J. Gordon. 2020. Conditional Learning of Fair Representations. In *Proc. of ICLR*. OpenReview.net. https://openreview.net/forum?id=Hkekl0NFPr

A APPENDIX: PROOF OF PROPOSITION

Proof. For Equal Odds and Equal Opportunity, recall that they are defined as:

$$AOD = \frac{1}{2}\left[|\text{TPR}^{(1)} - \text{TPR}^{(0)}| + |\text{TNR}^{(1)} - \text{TNR}^{(0)}|\right].$$
$$EOD = |\text{TPR}^{(1)} - \text{TPR}^{(0)}|.$$

When TPR and TNR are equalized between the two groups, AOD and EOD are both 0, and thus Equal Odds and Equal Opportunity are achieved.

For Accuracy Equality, denoting $\mathbb{P}(y = 1|s = 0)$ and $\mathbb{P}(y = 1|s = 1)$ as α and β respectively, we can rewrite AD into:

$$AD = |\alpha TPR^{(0)} - \beta TPR^{(1)} + (1-\alpha)TNR^{(0)} - (1-\beta)TNR^{(1)}|.$$

Next we discuss how equalized TPR and TNR helps achieve Accuracy Equality with the presence of the three types of bias.

Bias 1: Group Size Discrepancy. When the two groups have different group sizes but the same class distribution and the same class size, we have $\alpha = \beta$, and thus

$$\begin{aligned} AD=& |\alpha TPR^{(0)} - \alpha TPR^{(1)} + (1-\alpha)TNR^{(0)} - (1-\alpha)TNR^{(1)}| \\ \leq& \alpha|TPR^{(0)} - TPR^{(1)}| + (1-\alpha)|TNR^{(0)} - TNR^{(1)}|. \end{aligned}$$

Bias 2: Group Distribution Shift. When the two groups have different class distributions but the same group size and class size, we have $\alpha = 1 - \beta$. And without loss of generality, we have $\alpha \leq 0.5 \leq \beta$ (as shown in the middle figure in Figure 1). Then we can get,

$$\begin{aligned} AD =& |\alpha TPR^{(0)} - (1-\alpha)TPR^{(1)} + (1-\alpha)TNR^{(0)} - \alpha TNR^{(1)}| \\ \leq& \alpha|TPR^{(0)} - TPR^{(1)}| + \alpha|TNR^{(0)} - TNR^{(1)}| \\ &+(1-2\alpha)|TPR^{(1)} - TNR^{(0)}|. \end{aligned}$$

The group 0 and group 1 dominate two different classes with same proportion. In the example, Figure 1, 85% of data in class 0 belongs to group 0; and 85% of data in class 1 belongs to group 1. We assume the difficulty of fitting two classes are same, then $TPR^{(1)} \approx TNR^{(0)}$. We further have,

$$AD \leq \alpha|TPR^{(0)} - TPR^{(1)}| + \alpha|TNR^{(0)} - TNR^{(1)}|.$$

Bias 3: Class Size Discrepancy. When the two classes have different sizes but the two groups have the same size and class distributions, we have $\alpha = \beta$, and thus

$$\begin{aligned} AD=& |\alpha TPR^{(0)} - \alpha TPR^{(1)} + (1-\alpha)TNR^{(0)} - (1-\alpha)TNR^{(1)}| \\ \leq& \alpha|TPR^{(0)} - TPR^{(1)}| + (1-\alpha)|TNR^{(0)} - TNR^{(1)}|. \end{aligned}$$

B APPENDIX: FAIRIF ALGORITHM

The model initially undergoes training using standard empirical risk until it converges, resulting in the parameter $\theta^\star$. Following this, we compute the influence function and assign appropriate weights to ensure equal True Positive Rate (TPR) and True Negative Rate (TNR) across different groups within the validation set. Unlike previous methods such as those by Ren et al. [40] and Teso et al. [44], which calculate the influence function dynamically, our approach computes it only after the model has converged. This strategy not only reduces the computational time required for the influence function but also yields a more precise estimation.

Algorithm 1: FairIF.

Input: training set $\mathcal{D}$, validation set $\mathcal{D}^s$, model f and initial paramter θ, hyperparameters λ.
- Stage one: Balancing Influence
1. Train f_θ on $\mathcal{D}$ via ERM until converge to obtain $\theta^\star$.
2. Compute $\sum_{z_i \in \mathcal{D}} \nabla_\theta \ell(z_i, \theta^\star)$ and $H^{-1}_{\theta^\star}$ with stochastic estimation.
3. Compute performance differences $\text{diff}(\mathcal{D}^s, F_{TPR}, \theta^\star)$, $\text{diff}(\mathcal{D}^s, F_{TNR}, \theta^\star)$ over $\mathcal{D}^s$, and averaged gradient $\frac{1}{|\mathcal{D}^0|}\sum_{z_j \in \mathcal{D}^0} \nabla_\theta F_{TNR}(z_j)$, $\frac{1}{|\mathcal{D}^1|}\sum_{z_{j'} \in \mathcal{D}^1} \nabla_\theta F_{TNR}(z_{j'})$.
4. Obtain the balancing weight vector $\epsilon^\star$ through optimizing the objective, Equ. (14).
- Stage two: Reweighting
5. Train f_θ on $\mathcal{D}$ with the loss in Equ. (15) to obtain $\theta^\star_{\epsilon^\star}$.
return Final model $f_{\theta^\star_{\epsilon^\star}}$.

C APPENDIX: COMPUTATION DETAILS OF INFLUENCE FUNCTION

Computation details of Influence Function. As shown in Equation (9), the estimation of influence score requires the computation of the inverse hessian. The size of hessian matrix is propotional to the number of model parameters, thus directly computing the inversion of a hessian matrix, *i.e.*$\mathbf{H}^{-1}_{\theta^\star}$, is prohibitive. As described in the previous work [28], there are two different ways to efficiently compute $\nabla_\theta F(\{z_j\}, \theta^\star)^\top H^{-1}_{\theta^\star} \nabla_\theta \ell(z_i, \theta^\star)$. The first technique, Conjugate Gradients (CG), is a standard transformation of matrix inversion into an optimization problem. But, as an optimization problem, CG is slow for large dataset. The second method is LiSSA (Linear time Stochastic Second-Order Algorithm) method [3]. LiSSA is a stochastic estimation, which only samples a single point per iteration and results in significant speedups. Besides, LiSSA provides an unbiased estimation of the Hessian-vector product through implicitly computing it with a mini-batch of samples. As demonstrated in the previous works [7], the stochastic method is efficient and relatively accurate for sample-wise influence estimation. In this work, we employ the second method and the computation of Hessian-vector products (HVPs) can be summarized as:

- Step 1. Let $v := \sum_{z_i \in \mathcal{D}} \nabla_\theta l(z_i)$, and initialize the inverse HVP estimation $\mathbf{H}^{-1}_{0,\theta^\star} v = v$.
- Step 2. For $i \in \{1, 2, \ldots, J\}$, recursively compute the inverse HVP estimation using a batch size B of randomly sampled a data point $z_{i'}$, $\mathbf{H}^{-1}_{i,\theta^\star} v = v + \left(I - \nabla^2_\theta l(z_i)\right) \mathbf{H}^{-1}_{i-1,\theta^\star} v$, where J is a sufficiently large integer so that the above quantity converges.
- Step 3. Repeat Step 1-2 T times independently, and return the averaged inverse HVP estimations.

D APPENDIX: BASELINES

In the experiments, we employ the following baselines:

- **CFair** [61]. This adversarial approach aims to minimize balanced error rates over the target variable and protected attributes to achieve accuracy equality and equal odds.

- **DOMIND** [50]. This is a domain-independent training scheme that learns a shared feature representation with an ensemble of classifiers for different domains.
- **ARL** [30]. An adversarial optimization strategy that uses computationally identifiable errors to improve worst-case performance over unobserved protected groups.
- **FairSMOTE** [13]. A pre-processing technique that balances internal distributions to ensure equal representation in both positive and negative classes based on the sensitive attribute.
- **Influence** [32]. A reweighting strategy that uses a linear programming solver to compute the weights which perfectly bridge the fairness gap.

E APPENDIX: DATASET DESCRIPTION

CI-MNIST. The Correlated and Imbalanced MNIST (CI-MNIST) is firstly proposed by [39] to evaluate the bias-mitigation approaches in challenging setups and be capable of controlling different dataset configurations. The label $y \in \{0, 1\}$ indicates whether image x is odd or even. And the group attribute $s \in \{0, 1\}$ denotes the background color is blue or red. The original dataset assumes that there is a clean and balanced set for test. In this work, we make the distribution of train set and test set to be consistent. As described in Section 6.1, three different types of bias are independently introduced. The data statistics of them are presented in the Table 7. We keep the train/valid/test splits as the original setup [39].

	Odd (y=0)		Even (y=1)	
	Blue (s=0)	**Red (s=1)**	**Blue (s=0)**	**Red (s=1)**
Different Group Size	30245	5337	29257	5161
Group Dist. Shift	30245	5337	5163	29255
Different Class Size	17791	17791	4303	4301

Table 7: CI-MNIST Data Statistics.

Adult. Each instance in the Adult dataset[4] describes an adult with 114 attributes, e.g., gender, education level, age, etc, from the 1994 US Census. We use gender as the sensitive attribute ($s = 0$ for female and $s = 1$ for male), and the task is to predict whether his/her income is larger than or equal to 50K/year. The data statistics are presented in Table 8. We use the train/valid/test splits from the commonly used API aif360 [9].

German. The task in the German dataset[4] is to classify people as having good or bad credit risks by features related to the economical situation, with gender as the sensitive attribute restricted to female (s=1) and male (s=1). The data statistics are presented in Table 8.

COMPAS. The task in the COMPAS[14] dataset is to predict recidivism from someone's criminal history, jail and prison time, demographics, and COMPAS risk scores, with race as the protected sensitive attribute restricted to black (s=0) and white defendants (s=1). The data statistics are also presented in Table 8.

CelebA. The CelebA celebrity face dataset is proposed by [34]. Follow the task setup of [41] in which the label y is set to be the Blond Hair attribute, and the spurious attribute s is set to be the Male attribute: being female spurious correlates with having blond hair. The minority groups are (blond, male) and the majority groups are (blond, female). We use the standard train/valid/test splits from [41] in the main experiment (Section 6.2).

	Negative (y=0)		Positive (y=1)	
	s=0	**s=1**	**s=0**	**s=1**
Adult	13026	20988	1669	9539
German	201	499	109	191
COMPAS	1514	1281	1661	822

Table 8: Tabular Data Statistics.

	Not Blond Hair (y=0)		Blond Hair (y=1)	
	Female (s=0)	**Male (s=1)**	**Female (s=0)**	**Male (s=1)**
CelebA	89931	28234	82685	1749

Table 9: CelebA Data Statistics.

FairFace. FairFace is proposed by [26]. The face image dataset is balanced on race, gender and age. In our work, we take the gender prediction as the task and denote the $y = 1$ as *Male* and $y = 0$ as *Female*. And the sensitive attribute s s set to be the race, where $s = 0$ denotes *Black* and $s = 1$ denotes the *White*. We keep the train/valid/test splits as the original setup [26].

	Female (y=0)		Male (y=1)	
	Black (s=0)	**White (s=1)**	**Black (s=0)**	**White (s=1)**
FairFace	6894	6895	8789	9823

Table 10: FairFace Data Statistics.

F APPENDIX: IMPLEMENTATION DETAILS

In this section, we elucidate the architectures and hyperparameters chosen for each methodology.

For the three variations of CI-MNIST:

- MLP: Employs a single hidden layer with a dimension of 64.
- CNN: Comprises two convolutional layers, each with filters measuring 5×5.
- LeNet: Features three convolutional layers.

Across these datasets, we maintained a consistent learning rate of 0.0002, set λ to 0.1, and undertook training over 500 epochs.

For the datasets Adult, German, and COMPAS:

- We set the logistic regression's learning rate to 0.001 and designated the training epoch count as 250.

For the CelebA and FairFace datasets:

- We adopted the PyTorch [38] versions of ResNet-18, 34, and 50 [23].
- Guided by hyperparameter recommendations from [33], we locked the learning rate at 0.0002 without any learning rate scheduling, set λ to 0.1, and capped training epochs at 50. The final layer's hidden dimension was set to 128.

For FairIF's second stage, we mirrored the training configuration of the first stage. Notably, in Section 6.3, our models begin with weights pretrained on ImageNet. All our experiments were executed using four Tesla V100 SXM2 GPUs, supported by a 12-core CPU operating at 2.2GHz.

G APPENDIX: SIZE OF THE VALIDATION SET

In Sections 6.1 and 6.2, we employed standard validation sets for each dataset, leveraging their cost-effectiveness due to their size being 5-10 times smaller than training sets. To explore if FairIF could enhance performance with even smaller validation sets—thus reducing group annotation costs—we tested it on the FairFace and CelebA datasets with validation set sizes of 100%, 50%, 25%, and 10%. Adjusting sample weights and tuning FairIF based on them, Table 11 indicates that the fairness-utility trade-off of FairIF is optimal with 50% or full validation sets, given the refined influence estimations from more group data. The extremely small validation (10% of original validation set) led to a dip in accuracy, emphasizing the importance of a reasonable validation set for effective parameter tuning.

Table 11: Effect of Validation Data on FairIF. The Orig. row indicates the performance of the model trained with ERM.

	FairFace				CelebA			
	Acc(%)	**AD(%)**	**AOD**	**EOD**	**Acc(%)**	**AD(%)**	**AOD**	**EOD**
Orig.	86.68	7.23	0.0721	0.0610	95.41	4.54	0.2485	0.4975
Full	86.63	4.14	0.0048	0.0105	95.37	3.81	0.1220	0.3145
50%	86.17	5.01	0.0073	0.0152	95.59	4.14	0.1491	0.3981
25%	86.91	5.63	0.0089	0.0132	95.31	4.85	0.1873	0.3748
10%	85.49	6.44	0.0149	0.0298	93.17	5.19	0.2219	0.4753