Improving Fair Training under Correlation Shifts

Yuji Roh [1] Kangwook Lee [2] Steven Euijong Whang [1] Changho Suh [1]

Abstract

Model fairness is an essential element for Trustworthy AI. While many techniques for model fairness have been proposed, most of them assume that the training and deployment data distributions are identical, which is often not true in practice. In particular, when the bias between labels and sensitive groups changes, the fairness of the trained model is directly influenced and can worsen. We make two contributions for solving this problem. First, we analytically show that existing in-processing fair algorithms have fundamental limits in accuracy and group fairness. We utilize the notion of *correlation shifts* between labels and groups, which can explicitly capture the change of the above bias. Second, we propose a novel pre-processing step that samples the input data to reduce correlation shifts and thus enables the in-processing approaches to overcome their limitations. We formulate an optimization problem for adjusting the data ratio among labels and sensitive groups to reflect the shifted correlation. A key benefit of our approach lies in *decoupling* the roles of pre- and in-processing approaches: correlation adjustment via pre-processing and unfairness mitigation on the processed data via in-processing. Experiments show that our framework effectively improves existing in-processing fair algorithms w.r.t. accuracy and fairness, both on synthetic and real datasets.

1. Introduction

Model fairness is becoming indispensable in many artificial intelligence (AI) applications to prevent discrimination against specific groups such as gender, race, or age (Feldman et al., 2015; Hardt et al., 2016) or individuals (Dwork et al., 2012a). In this work, we focus on group fairness, and there are three prominent group fairness approaches: pre-processing, where training data is debiased; in-processing, where model training is tailored for fairness; and post-processing, where the trained model's output is modified to satisfy fairness – see more related works discussed in Sec. 2.

While fairness in-processing approaches are commonly used to mitigate unfairness, most of them make the limiting assumption that the training and deployment data distributions are the same (Zafar et al., 2017a; Zhang et al., 2018; Roh et al., 2021). However, the two distributions are usually different, especially in terms of data biases (Wick et al., 2019; Maity et al., 2021). For example, a recent work shows that the bias amounts likely differ between previously collected data and recently collected data (Ding et al., 2021). Moreover, when the data bias changes, the fairness and accuracy of the trained model are now unpredictable at deployment, as the above assumption is broken.

In this work, we utilize the notion of *correlation shifts* between the label y and group attribute z in the data (i.e., (y,z)-correlation shifts) to systematically address the data bias changes. Although several works have been recently proposed to investigate fair training on different types of distribution shifts, including covariate and concept shifts (Singh et al., 2021; Mishler & Dalmasso, 2022), they usually do not explicitly consider bias changes between y and z. In comparison, our (y,z)-correlation shift enables us to theoretically analyze how exactly bias changes affect fair training – see how correlation shift compares with other types of distribution shifts in Sec. 2.

For fair training under (y,z)-correlation shifts, we first 1) analyze the fundamental accuracy and fairness limits of in-processing approaches with the fixed distribution assumption using the notion of (y,z)-correlation in the data and then 2) design a novel pre-processing step to boost the performances of in-processing approaches under the (y,z)-correlation shifts. We show that existing in-processing fair algorithms are indeed limited by the training distribution and may perform poorly on the deployment distribution. In particular, a high (y,z)-correlation results in a poor accuracy-fairness tradeoff for any fair training. Therefore, as most in-processing fair algorithms assume identical training and deployment distributions, there is no guarantee their perfor-

[1] Department of Electrical Engineering, KAIST [2] Department of Electrical and Computer Engineering, University of Wisconsin-Madison. Correspondence to: Steven E. Whang <swhang@kaist.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

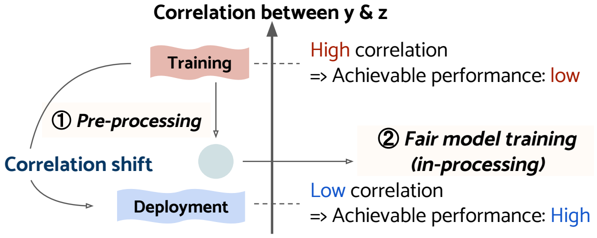

Figure 1. A high-level workflow under (y, z)-correlation shifts. The central axis represents the correlation between the label y and sensitive group attribute z. The correlation of training data is usually higher than that of deployment data. In Sec. 4, we show that the correlation determines the achievable performance of fair training. Thus, we first run our pre-processing framework and then apply existing fair algorithms on the processed data to address the correlation shift and improve the performances of fair training.

mances on the training data carry over to the deployment data. Based on the theoretical analysis, we propose a pre-processing step for reducing the shifted correlation by taking samples of (y, z)-classes. In a streaming setup, one can estimate a possible range of shifted correlations. Using this range, we solve an optimization problem that finds the new data ratio among (y, z)-classes to adjust the correlation for the shift. The new sampled data is then used as the input of any in-processing, giving it a better chance to perform well.

A key advantage of our framework is *the decoupling of pre-processing and in-processing for unfairness mitigation* where the pre-processing adjusts the correlation while the in-processing performs the rest of the unfairness mitigation, as described in Figure 1. We note that our pre-processing aims to boost the performances of in-processing approaches based on our theoretical analysis, whereas existing pre-processing approaches for fairness simply remove the bias in the data and are not designed to explicitly benefit the in-processing approaches – see Sec. 6.1 for details. Our framework thus takes the best of both worlds of pre- and in-processings where (1) pre-processing solves the data problems, and (2) in-processing performs its best on the improved data.

In our experiments, we verify our theoretical results and demonstrate how our framework outperforms state-of-the-art pre-processing and in-processing baselines. Experiments on both synthetic and real-world datasets show that our framework effectively improves the accuracy and fairness performances of the state-of-the-art in-processing approaches (Zafar et al., 2017a; Zhang et al., 2018; Roh et al., 2021) under correlation shifts. Also, our framework performs better than two-step baselines that first run an existing pre-processing approach (Kamiran & Calders, 2011) and then an in-processing approach. We also show how to reasonably estimate correlation shifts and that our framework is robust against incorrect estimates.

Summary of Contributions **(1)** We formalize the notion of (y, z)-correlation shifts, which is key to improving fair training against data bias changes. **(2)** Using the notion of (y, z)-correlation, we theoretically show that existing in-processing fair algorithms are limited by the training distribution and may perform poorly on the deployment distribution. **(3)** Based on the theoretical analysis, we propose a novel pre-processing step to boost the performances of fair in-processing approaches. **(4)** We demonstrate that our framework effectively improves the performances of the state-of-the-art fair algorithms under (y, z)-correlation shifts, even if the correlation estimates are inaccurate.

Notation & Fairness Metrics Let θ be the model weights, $x \in \mathbb{X}$ be the input feature to the model, $y \in \mathbb{Y}$ be the true label, and $\hat{y} \in \mathbb{Y}$ be the predicted label where $\hat{y}$ is a function of (x, θ). Let $z \in \mathbb{Z}$ be a sensitive group attribute, e.g., gender or race. We assume a binary setting ($\mathbb{Y} = \mathbb{Z} = \{0, 1\}$). We focus on the prominent group fairness metrics, demographic parity (DP) (Feldman et al., 2015) and equalized odds (EO) (Hardt et al., 2016), where DP is achieved when the positive prediction rates are the same for the groups (i.e., $\Pr(\hat{y}{=}1|z{=}1){=}\Pr(\hat{y}{=}1|z{=}0)$) and EO is achieved when the label-wise accuracies are the same for the groups (i.e., $\Pr(\hat{y}{=}y|y{=}y, z{=}1){=}\Pr(\hat{y}{=}y|y{=}y, z{=}0), \forall y \in \{0, 1\}$).

2. Related Work

As model fairness becomes essential for Trustworthy AI, various fairness techniques have been proposed to improve group fairness that do not discriminate specific demographics (Barocas et al., 2019). The fairness techniques can be categorized into three prominent approaches: pre-, in-, and post-processings – see their representative works in Sec. C.1. Among the three categories, in-processing approaches are widely used for fair training due to their high fairness and accuracy performances, but most of them assume that the training and deployment distributions are the same.

Recently, there is an emerging focus on fairness under data distribution shifts. In Table 1, we summarize various types of shifts into the two categories:

Table 1. Different distribution shifts.

Category	Type of Shifts	
General distribution shifts	covariate shift	$\Pr(x)$
	label shift	$\Pr(y)$
	concept shift	$\Pr(y\|x)$
Fairness specific shifts	demographic shift	$\Pr(z)$
	subpopulation shift	$\Pr(y, z)$
	corr. shift (ours)	$\Pr(z\|y)$

1. *General distribution shifts* (Singh et al., 2021; Rezaei et al., 2021; Chen et al., 2022; Mishler & Dalmasso, 2022) focus on shifts involving the input feature (x) and label (y), which are widely studied in the traditional machine learning literature. Here the bias changes between label (y) and group (z) are not explicitly considered.

2. *Fairness-specific shifts* (Maity et al., 2021; An et al., 2022; Giguere et al., 2022; Schrouff et al., 2022; Dai & Brown, 2020) handle group (z) changes, as z is especially correlated with fair training. Our work also falls into this

category. A recent study (Maity et al., 2021) theoretically analyzes the behavior of fair training under a change in bias called subpopulation shifts, where a specific group has fewer positively-labeled examples during training time compared to deployment time. Another study (Giguere et al., 2022) designs a new test method to serve a fair model under another distribution change called demographic shifts, where the subgroup distribution may change – see an *empirical comparison* with our work in Sec. B.16. A recent work (An et al., 2022) proposes a self-training-based transfer algorithm that requires specific model architecture (e.g., adversary network) to support data changes. Also, there is another work to model label bias shifts (Dai & Brown, 2020), but this work does not provide theoretical analyses and fairness algorithms. In comparison, our contribution lies in 1) formalizing the notion of (y, z)-correlation shifts, which is important for explaining with theoretical evidence the connection between the data bias changes and behaviors of fair training, 2) analyzing the fundamental limits of in-processing approaches under correlation shifts, and 3) proposing a pre-processing step based on the theoretical analysis for assisting the existing fairness approaches. In addition, our framework is general and can *support any model architecture and training procedure*. We leave more detailed comparisons with these works in Sec. C.2.

Also, compared to the traditional domain adaptation techniques that use reweighting strategies (Li et al., 2016; Gu et al., 2021; Rezaei et al., 2021), our framework is specialized to handle shifted (y, z)-correlations, which directly affect the performance limits of the fairness algorithms, as we will show in our theoretical analyses. In addition, although our approach can also be implemented as a reweighting strategy, we believe the decoupling of pre- and in-processing is especially useful from a practical standpoint where our pre-processing can be combined with any existing in-processing fairness algorithm as is and boost its performance.

Another line of research is supporting robustness in fair training, including handling noisy groups (Celis et al., 2021; Wang et al., 2020) or poisoning attacks (Mehrabi et al., 2021; Solans et al., 2020). Although this direction is not our immediate focus, we do perform preliminary experiments in Secs. B.17 and B.18 to show some potential to support noisy group attributes or poisoning attack scenarios.

In addition to group fairness, we discuss other noteworthy fairness definitions including causality-based fairness (Kilbertus et al., 2017) in Secs. C.1 and C.3.

3. Limitations of Prior Work & Problem Setup

Most fair in-processing approaches assume that the training and deployment distributions are the same, which means that they assume the same level of bias as well. However, data bias may shift over time as confirmed by recent studies (Wick et al., 2019; Maity et al., 2021; Ding et al., 2021), which means that the deployment data may actually have a different bias than the training data.

A bias change in the deployment data may have an adverse effect on a trained model's performance. For example, suppose we train a perfectly fair model on the training data while sacrificing accuracy. Here, if the bias between the label and group changes in the deployment distribution, the trained model's fairness can worsen while the accuracy still remains imperfect. The underlying problem is that the model was trained with a different bias in mind – see a detailed toy example to show this limitation in Sec. A.1. Thus, we aim to overcome this issue under the following setup.

Problem Setup Let D_{train} and D_{deploy} be the training and deployment data, respectively. We assume access to the (x, y, z) values in D_{train} and m samples of the (y, z) values in D_{deploy}. Accessing some of the new data is a common and practical scenario for detecting drifts (Lu et al., 2018) in a data stream. Here, we can detect whether there exists a bias change (shift) by comparing the (y, z) values in D_{train} and a small number of (y, z) values in D_{deploy}. Our goal is to have high fairness and accuracy even when the bias shifts.

To this end, we first **1)** theoretically analyze why the bias change affects the performances of fair training by utilizing the notion of (y, z)-correlation (**Sec. 4**) and then **2)** design a pre-processing step to address such bias changes and boost the performances of any fair in-processing approaches under the above problem setup (**Sec. 5**).

4. Fair Training under Correlation Shifts

To systematically study the effects of data bias changes, we first analyze the achievable performances of fair training via a *correlation between y and z that can be used to explicitly measure data bias regarding sensitive groups* (Sec. 4.1) and discuss the limitations of fair in-processings when this correlation shifts (Sec. 4.2). We define correlation as follows:

(y, z)-correlation Data bias can be represented via the correlation between the label y and the sensitive group attribute z, where the correlation represents a statistical relationship between two random variables. We thus define *(y, z)-correlation* using Pearson's correlation coefficient (Rodgers & Nicewander, 1988) $\rho_{\mathrm{yz}} = \frac{Cov(\mathrm{y},\mathrm{z})}{\sigma(\mathrm{y})\sigma(\mathrm{z})}$, which is known to effectively capture biases in real-world scenarios. Here, $Cov(\cdot)$ is the covariance, and $\sigma(\cdot)$ is the standard deviation. As we assume y and z are binary, we can express ρ_{yz} as follows (Cohen & Cohen, 1975):

$$\rho_{\mathrm{yz}} = \frac{\Pr(\mathrm{y}=1,\mathrm{z}=1)\Pr(\mathrm{y}=0,\mathrm{z}=0) - \Pr(\mathrm{y}=1,\mathrm{z}=0)\Pr(\mathrm{y}=0,\mathrm{z}=1)}{\sqrt{\Pr(\mathrm{y}=1)\Pr(\mathrm{y}=0)\Pr(\mathrm{z}=1)\Pr(\mathrm{z}=0)}}.$$

This notion of (y, z)-correlation plays a key role in both analyzing the achievable performances of fair training (Sec. 4) and improving fair training against data bias changes (Sec. 5). In the rest of the paper, the term *correlation* always indicates the correlation between labels (y) and groups (z).

4.1. Achievable Performance Analysis via Correlation

We identify the fundamental accuracy and fairness limits of fair training based on the (y, z)-correlation of the data. We first analyze the most common case of improving the fairness w.r.t. a single fairness metric. We then extend our analysis to the more complicated case of improving the fairness w.r.t. multiple metrics, which is important for capturing various social contexts, but has been seldom studied in the literature. In both cases, we show that the (y, z)-correlation determines the achievable performance of fair training.

CASE 1 – Fair Training w.r.t. a Single Metric When improving group fairness, fair training is known to face an accuracy-fairness tradeoff, where the accuracy is sacrificed to make the model fairer. Recently, Menon & Williamson (2018) investigate that the accuracy-fairness tradeoff w.r.t. demographic parity is affected by how much y and z are *aligned* in the data. For example, if the y and z values are identical for all examples, achieving high fairness may require low accuracy. In contrast, if y and z are randomly set, fairness and accuracy can be achieved together. The following proposition shows this previous work's result.

Proposition 1 (From Menon & Williamson (2018))**.** *(Informal) When a model is trained w.r.t. demographic parity, a high alignment between* y *and* z *leads to a worse accuracy-fairness tradeoff.*

For our purposes, we infer a similar relationship using (y, z)-correlation based on Proposition 1. The following lemma makes a connection between the (y, z)-correlation and the conditional probabilities of y given z under some conditions.

Lemma 2. *If the marginal probabilities of* y *and* z *(i.e.,* $\Pr(\mathrm{y} = y)$ *and* $\Pr(\mathrm{z} = z)$*) remain the same, the* (y, z)*-correlation* ρ_{yz} *is proportional to the difference between the conditional probabilities of* y *given different* z *values (i.e.,* $\Pr(\mathrm{y} = 1|\mathrm{z} = 1) - \Pr(\mathrm{y} = 1|\mathrm{z} = 0)$*).*

The proof for Lemma 2 can be found in Sec. A.2. By applying Lemma 2 to Proposition 1, we can infer that the (y, z)-correlation also determines the accuracy-fairness tradeoff under certain conditions. According to the previous work (Menon & Williamson, 2018), alignment is defined as how many examples in each group have a specific label. We can thus measure alignment as $\Pr(\mathrm{y} = y|\mathrm{z} = 1) + \Pr(\mathrm{y} = y'|\mathrm{z} = 0)$ where $y \neq y'$ and $y, y' \in \{0, 1\}$. Then, we can convert this term into the difference between the conditional probabilities of y given different z values, which is used in Lemma 2 – see details in Sec. A.3. As a result, we derive the following corollary, which shows that the (y, z)-correlation determines the achievable accuracy-fairness tradeoff.

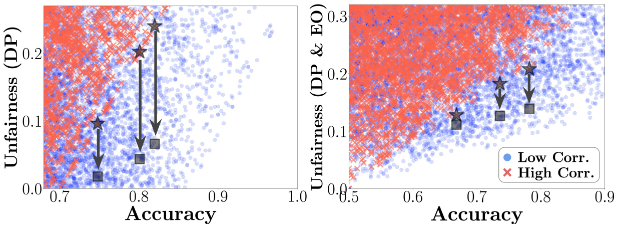

Figure 2. Simulation results of accuracy-unfairness (left: DP and right: DP & EO) performances of various classifiers on two synthetic datasets that have low and high (y, z)-correlations. Each blue dot (red cross) indicates a single classifier's performance on the low (high) correlation data. The lower the unfairness value, the better. For each dataset, we generate enough classifiers to show the full range of possible performances. As a result, low correlation enables classifiers to attain better accuracy-fairness tradeoffs (i.e., close to the bottom right), regardless of supporting single or multiple fairness metrics. Also, the optimal classifiers trained on high correlation data (black stars) have suboptimal performances on the low correlation data (black squares).

Corollary 3. *When a model is trained w.r.t. demographic parity, and the marginal probabilities of* y *and* z *remain the same, the achievable accuracy-fairness tradeoff of the model is determined by the* (y, z)*-correlation. The higher the correlation, the worse the accuracy-fairness tradeoff.*

Remark 4. *We can relax the assumption that marginal probabilities are fixed in Corollary 3. When the marginal probabilities of* y *and* z *change up to* γ_{y} *and* γ_{z}*, respectively, Corollary 3 can be generalized as follows: the achievable accuracy-fairness tradeoff is determined by* $\rho_{\mathrm{y,z}} \cdot \eta$*, where* $\rho_{\mathrm{y,z}}$ *is the* (y, z)*-correlation and* $\eta \in [\sqrt{\frac{\Pr(\mathrm{y}=1)-\gamma_{\mathrm{y}}-(\Pr(\mathrm{y}=1)+\gamma_{\mathrm{y}})^2}{\Pr(\mathrm{z}=1)+\gamma_{\mathrm{z}}-(\Pr(\mathrm{z}=1)-\gamma_{\mathrm{z}})^2}}, \sqrt{\frac{\Pr(\mathrm{y}=1)+\gamma_{\mathrm{y}}-(\Pr(\mathrm{y}=1)-\gamma_{\mathrm{y}})^2}{\Pr(\mathrm{z}=1)-\gamma_{\mathrm{z}}-(\Pr(\mathrm{z}=1)+\gamma_{\mathrm{z}})^2}}]$*. Thus, the higher the* $\rho_{\mathrm{y,z}} \cdot \eta$*, the worse the tradeoff. In our framework, we actually allow the marginal probabilities of* y *and* z *to change up to* γ_{y} *and* γ_{z} *– see details in Sec. 5.1.*

Simulation We now confirm our theoretical analysis via a simulation – see details on the setting in Sec. B.1. The left plot in Figure 2 shows the accuracy-unfairness performances of classifiers on two synthetic datasets with low and high correlations when using DP for measuring fairness (the plot on the right will be explained later). We measure unfairness where a lower value indicates better fairness (i.e., perfectly fair when 0) – see the exact metrics in Sec. 6. We generate various synthetic classifiers on the two datasets to show the full range of possible model performances. The blue dots (red crosses) are the classifiers on the dataset with low (high) correlation. As a result, low correlation results in better accuracy-fairness tradeoffs (i.e., close to the bottom right). We will discuss the black stars and squares in Sec. 4.2.

Remark 5. *We note that Corollary 3 does not necessarily apply for equalized odds (EO), which is achieved when the accuracies conditioned on the true label are the same for the groups. In theory, a perfect classifier can achieve perfect fairness w.r.t. EO, so the* (y, z)*-correlation does not deter-*

mine the limits of the accuracy-fairness tradeoff w.r.t. EO – see the detailed explanation in Sec. B.2. However, classifiers are not perfect in practice, and we empirically observe that higher (y, z)*-correlation leads to a worse accuracy-fairness tradeoff w.r.t. EO as well – see the empirical results in Sec. B.3.*

CASE 2 – Fair Training w.r.t. Multiple Metrics Beyond the single metric case, we now extend our analysis to support multiple fairness metrics together. Although supporting multiple metrics is necessary to address fairness under various social contexts, most fairness works do not address this problem. A major challenge is that group fairness metrics are known to be mutually exclusive, which means that the metrics cannot be perfectly satisfied together, unless the data is completely unbiased (Barocas et al., 2019; Kleinberg et al., 2017). Interestingly, our (y, z)-correlation provides an opportunity to support multiple metrics as much as possible in a principled fashion. We focus on supporting two metrics and leave the support of more metrics as future work.

We first analyze when a model can improve both DP and EO to get hints on the relationship between the (y, z)-correlation and improving fairness w.r.t. two metrics. Here, we focus on the ε-fairness of both metrics, which is a relaxed version of perfect fairness (Barocas et al., 2019), where ε indicates the unfairness level of the model. For example, we can define ε-DP as $|\Pr(\hat{\mathrm{y}}{=}1|\mathrm{z}{=}1) - \Pr(\hat{\mathrm{y}}{=}1|\mathrm{z}{=}0)| \leq \varepsilon$ and ε-EO as $|\Pr(\hat{\mathrm{y}}{=}y|\mathrm{y}{=}y, \mathrm{z}{=}1) - \Pr(\hat{\mathrm{y}}{=}y|\mathrm{y}{=}y, \mathrm{z}{=}0)| \leq \varepsilon$. Here, $\varepsilon \in [0, 1]$ and a lower ε means higher fairness. The next proposition shows when a model can achieve both ε-DP and ε-EO.

Proposition 6 (ε-DP & ε-EO)**.** *Let a model achieve both ε-DP and ε-EO. Then, the following inequality holds:* $\{|\Pr(\mathrm{y} = y|\mathrm{z} = z) - \Pr(\mathrm{y} = y|\mathrm{z} = z')| \cdot |\Pr(\hat{\mathrm{y}} = y|\mathrm{y} = y, \mathrm{z} = z) - \Pr(\hat{\mathrm{y}} = y|\mathrm{y} = y', \mathrm{z} = z)|\} \leq 2\varepsilon,\ y \neq y', z \neq z', y, y', z, z' \in \{0, 1\}$.

The proof for Proposition 6 can be found in Sec. A.4. Here, the achievable fairness level ε is affected by the data bias, which is represented by the difference between the conditional probabilities of y given z (i.e., $|\Pr(\mathrm{y} = y|\mathrm{z} = z) - \Pr(\mathrm{y} = y|\mathrm{z} = z')|$). For example, if the data is highly biased, a model cannot achieve high fairness (i.e., low ε) unless the second term on the left side (i.e., $|\Pr(\hat{\mathrm{y}} = y|\mathrm{y} = y', \mathrm{z} = z) - \Pr(\hat{\mathrm{y}} = y|\mathrm{y} = y, \mathrm{z} = z)|$) decreases. However, the second term is related to the accuracy, where lowering it to zero causes the model to make random predictions or predictions that are all the same. Thus, Lemma 2 and Proposition 6 give the following corollary.

Corollary 7. *When a model is trained w.r.t. both DP and EO, and the marginal probabilities of* y *and* z *do not change, the achievable fairness of the model is determined by the* (y, z)*-correlation if the accuracy does not change. The higher the correlation, the lower the achievable fairness.*

Similar to the single-metric case, we confirm the theoretical result of supporting multiple metrics via the simulation in the right plot in Figure 2. Again, we observe that low correlation enables classifiers to attain better accuracy-fairness tradeoffs. We also leave discussions for improving fairness w.r.t. another prominent metric called predictive parity (PP) (Berk et al., 2021) together with DP and EO in Secs. A.5 (ε-PP and ε-DP) and A.6 (ε-EO and ε-PP).

4.2. Effects of Correlation Shifts

As the performance ranges of fair training are different according to the data correlation, a fair classifier learned on some training data does not necessarily have optimal performance on deployment data with a different correlation. We say there is a *correlation shift* between two data distributions D_1 and D_2 when $|\rho_{\mathrm{yz}}^{D_1} - \rho_{\mathrm{yz}}^{D_2}| \neq 0$, where $\rho_{\mathrm{yz}}^{D_i}$ indicates the (y, z)-correlation of D_i. Here, the correlation *shift* reflects the bias *change* between data distributions.

We revisit our previous simulation in Figure 2 that helps to understand how the correlation shifts affect fair training. We consider a correlation shift from a training distribution with high correlation to a deployment distribution with low correlation. We start from a few optimal classifiers trained on the high-correlated data, which are denoted as black stars. These classifiers unfortunately have suboptimal performances on the low-correlation deployment data (black squares). Hence, in-processing approaches that assume the same training and deployment data distributions lose the chance of achieving better fairness and accuracy in the deployment distribution under correlation shifts.

Based on our theoretical and simulation results, we design a pre-processing approach using (y, z)-correlation as a tuning knob to give existing in-processing approaches a better opportunity to achieve maximum accuracy and fairness performances in the following section.

5. Framework

To improve the performance of fair training in the presence of (y, z)-correlation shifts, we propose a novel fair training framework that consists of 1) applying a pre-processing step for reflecting the shifted (y, z)-correlation and 2) utilizing any existing in-processing algorithms for fair training on top of the improved data. In Sec. 5.1, we explain the pre-processing step, which estimates the correlation range and solves an optimization problem for finding a new data distribution that follows the shifted correlation by adjusting the (y, z)-class ratios. In Sec. 5.2, we explain our overall training process and extensions to support similar distributions.

5.1. Pre-processing for Correlation Shifts

Conditions We design an optimization that finds the best (y, z)-class ratios given a (y, z)-correlation ρ_{yz} by using

Lemma 2, which shows that ρ_{yz} is proportional to the conditional probability difference under some assumptions. Note that the conditional probability can be written using the class weights. Let the original data ratio of each (y=y, z=z)-class be $w_{\mathrm{y}=y,\mathrm{z}=y}$. Let the new data ratio be $w'_{\mathrm{y}=y,\mathrm{z}=y}$, which is required to satisfy the shifted correlation of the deployment data. Note that $\sum_{\forall y,z} w'_{\mathrm{y}=y,\mathrm{z}=z}$=1. Also, let c be the correlation constant that is the difference between the conditional probabilities of y given z (i.e., $\Pr(\mathrm{y}{=}1|\mathrm{z}{=}1) - \Pr(\mathrm{y}{=}1|\mathrm{z}{=}0)$) in the deployment data. Let $[\alpha, \beta]$ be the range of the correlation constant c (i.e., $c \in [\alpha, \beta]$). We use a relaxed version of the assumption in Lemma 2 where the marginal probabilities of y and z can change by up to γ_{y} and γ_{z}, as discussed in Remark 4.

Part 1: Estimating Correlation Range We first find a shift range $[\alpha, \beta]$, which can be estimated or given in advance. When the estimation is needed, we consider the scenario where accessing some of the new data is possible, which is necessary for detecting data drifts (Lu et al., 2018). As stated in Sec. 3, we assume we have the (y, z) values of m samples in the deployment data. We can estimate the new correlation constant c by calculating the empirical probability $\hat{\Pr}(\mathrm{y}|\mathrm{z})$ using the given m samples. Here, we can use the maximum likelihood estimator, i.e., $\hat{\Pr}(\mathrm{y}{=}y|\mathrm{z}{=}z)$= (num. samples in (y, z))/(num. samples in z). We can also compute a confidence interval of the estimated c using standard concentration inequalities (e.g., Hoeffding's inequality) – see a formal estimation guarantee and proof in Sec. A.7.

Some key questions are 1) whether the correlation can be estimated for small amounts of new data and 2) whether our framework works well even when the estimation is incorrect. We answer them via empirical results in Secs. 6.3 and 6.4.

Part 2: Optimization With the above conditions and estimated correlation, we set a problem whose goal is to minimize the squared difference between the original and new data ratios to reduce the information loss, similar to other preprocessings (Zemel et al., 2013; Quadrianto et al., 2019):

$$\min_{w'} \ \textstyle\sum_{\forall y,z} (w_{\mathrm{y}=y,\mathrm{z}=z} - w'_{\mathrm{y}=y,\mathrm{z}=z})^2$$

$$\text{s.t.} \ \ \alpha \leq \frac{w'_{\mathrm{y}=1,\mathrm{z}=1}}{w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=0,\mathrm{z}=1}} - \frac{w'_{\mathrm{y}=1,\mathrm{z}=0}}{w'_{\mathrm{y}=1,\mathrm{z}=0} + w'_{\mathrm{y}=0,\mathrm{z}=0}} \leq \beta,$$

$$|(w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=1,\mathrm{z}=0}) - \Pr_{\text{train}}(\mathrm{y}=1)| \leq \gamma_{\mathrm{y}},$$

$$|(w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=0,\mathrm{z}=1}) - \Pr_{\text{train}}(\mathrm{z}=1)| \leq \gamma_{\mathrm{z}},$$

$$\textstyle\sum_{\forall y,z} w'_{\mathrm{y}=y,\mathrm{z}=z}{=}1, \ 0 \leq w'_{\mathrm{y}=y,\mathrm{z}=z} \leq 1, \ \forall y \in \{0,1\}, z \in \{0,1\}$$

where $\Pr_{\text{train}}(\mathrm{y}{=}1){=}w_{\mathrm{y}=1,\mathrm{z}=1}{+}w_{\mathrm{y}=1,\mathrm{z}=0}$ and $\Pr_{\text{train}}(\mathrm{z}{=}1)$ $=w_{\mathrm{y}=1,\mathrm{z}=1}{+}w_{\mathrm{y}=0,\mathrm{z}=1}$. Note that if we know the exact shifted correlation value (i.e., $\alpha{=}\beta{=}c$), the first constraint becomes an equality constraint.

The above optimization is a nonconvex quadratically constrained quadratic problem (QCQP), as the objective is quadratic, and the first constraint is nonconvex quadratic. However, the nonconvex QCQP is known to be hard to solve (d'Aspremont & Boyd, 2003). Thus, we apply the semidefinite (SDP) relaxation, which is one of the convex relaxations known to give a reasonable lower bound of the optimal value of the original problem (Park & Boyd, 2017):

$$\min_{X,x} \ \mathbf{Tr}(XP_0) + q_0^T x$$

$$\text{s.t.} \ \mathbf{Tr}(XP_\alpha) \geq 0, \ \mathbf{Tr}(XP_\beta) \leq 0,$$

$$|q_2^T x - \Pr_{\text{train}}(y=1)| \leq \gamma_{\mathrm{y}}, \ \ |q_3^T x - \Pr_{\text{train}}(z=1)| \leq \gamma_{\mathrm{z}},$$

$$q_4^T x = 1, \ \ 0 \leq x_i \leq 1 \ \forall i, \ \ \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0$$

where $\mathbf{Tr}(\cdot)$ is the trace, $X = xx^T$, $x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$ $= \begin{bmatrix} w'_{1,1} & w'_{1,0} & w'_{0,1} & w'_{0,0} \end{bmatrix}^T$, $q_0 = -2 \begin{bmatrix} w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \end{bmatrix}^T$, $q_2{=} \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T$, $q_3{=} \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T$, $q_4{=} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$, $P_0{=}\mathrm{diag}(\mathbf{1})$, and $P_\gamma = \begin{bmatrix} 0 & -\gamma/2 & 0 & (1-\gamma)/2 \\ -\gamma/2 & 0 & (-1-\gamma)/2 & 0 \\ 0 & (-1-\gamma)/2 & 0 & -\gamma/2 \\ (1-\gamma)/2 & 0 & -\gamma/2 & 0 \end{bmatrix}$ where $\gamma \in \{\alpha, \beta\}$. Conversion details are in Sec. A.9. As this relaxed problem is now convex, we can solve it using convex optimization solvers (e.g., CVXPY (Diamond & Boyd, 2016)).

Remark 8. *In distribution shift studies, it is common to assume that the training and deployment distributions differ only in specific aspects (e.g., only label dist. is changed). Without such an assumption, the training and deployment distributions can be arbitrarily far apart, and there is no way to infer the new distribution via the training samples (Huang et al., 2006). In our framework, we focus on the distribution changes of* y *and* z *and do not explicitly model the change of the input feature* x *distribution. Our framework thus works best when the* x *distribution does not change, but does not strictly require this condition unlike other previous works on fairness under different types of shifts (Maity et al., 2021; Giguere et al., 2022). We empirically show that our framework indeed performs well in a real-world scenario where the* x *distribution does shift – see Sec. 6.3.*

5.2. Overall Training

Algorithm 1 Fair Training under Correlation Shifts

```
Input: training data $D$, original ratio $w_{\mathrm{y,z}}$, $m$ samples of ($\mathrm{y}_{\text{deploy}}$,
       $\mathrm{z}_{\text{deploy}}$), confidence level of estimation $1-\delta$, thresholds $\gamma_{\mathrm{y}}$
       and $\gamma_{\mathrm{z}}$, in-processing algorithm $f$
$\alpha, \beta$ = MLE($\mathrm{y}_{\text{deploy}}, \mathrm{z}_{\text{deploy}}, 1-\delta$) or given in advance
$w'_{\mathrm{y,z}}$ = SDPsolver($w_{\mathrm{y,z}}, \alpha, \beta, \gamma_{\mathrm{y}}, \gamma_{\mathrm{z}}$)
$d_j \leftarrow w'_{\mathrm{y}=y,\mathrm{z}=z}/w_{\mathrm{y}=y,\mathrm{z}=z}, \forall j \in \mathbb{I}_{(y,z)}, \forall (y,z) \in \mathbb{Y} \times \mathbb{Z}$
d = $\{d_i\}_{i=1,\ldots,n}$
(Optional) d = MinDistChange($D, w_{\mathrm{y,z}}, w'_{\mathrm{y,z}}$)
Draw new data $D'$ from $D$ via weighted sampling w.r.t. d
$\theta \leftarrow$ initial model parameters
for each epoch do: Update $\theta$ based on $f$ on $D'$
Output: $\theta$
```

We present the overall process for fair training under correlation shifts in Algorithm 1. The algorithm includes *pre-processing* for reflecting the shifted correlation and *in-processing* for fair training on top of the improved data. During the pre-processing, we first obtain the shift range $[\alpha, \beta]$, which can be given in advance or inferred using m samples and the target confidence level of estimation $1 - \delta$ with a maximum likelihood estimator (MLE) – see a detailed algorithm in Sec. A.8. We then find the new (y, z)-class ratio $w'_{\mathrm{y,z}}$ based on the SDP relaxation of our optimization, which can be solved using convex optimization solvers (e.g., CVXPY). During the in-processing, we calculate the sample-wise weights to ensure that the sample weight sum in each (y, z)-class is $w_{\mathrm{y}=y,\mathrm{z}=z} \cdot n$, where n is the total number of samples in the original training data. Within each (y, z)-class, all samples have the same weight. We then draw new data D' from the original data via weighted sampling according to the sample-wise weights. Finally, we train a model using an in-processing fair algorithm f on D'.

In addition, using an optional step (`MinDistChange`), we can address the scenario where the pre-processing should minimally change the original training data, which is sometimes preferred in other applications (Kamiran & Calders, 2011). Details on this extension are in Sec. A.10.

6. Experiments

We perform experiments to evaluate our framework. We use logistic regression in most experiments and utilize multi-layer neural networks in Sec. B.11. We evaluate the models on separate test datasets and repeat all experiments with five different random seeds. We use CVXPY as a convex optimization solver – see detailed settings in Sec. B.4.

Fairness metrics We focus on two prominent group fairness metrics: demographic parity (DP) (Feldman et al., 2015) and equalized odds (EO) (Hardt et al., 2016). We measure the fairness disparities (i.e., unfairness) among sensitive groups as follows: *DP disparity* = $\max_{z \in \mathbb{Z}} |\Pr(\hat{\mathrm{y}} = 1|\mathrm{z} = z) - \Pr(\hat{\mathrm{y}} = 1)|$ and *EO disparity* = $\max_{z \in \mathbb{Z}, y \in \mathbb{Y}} |\Pr(\hat{\mathrm{y}} = y|\mathrm{z} = z, \mathrm{y} = y) - \Pr(\hat{\mathrm{y}} = y|\mathrm{y} = y)|$. When measuring the unfairness w.r.t. both DP and EO, we take the maximum of both disparities (i.e., $\max($*DP disp.*, *EO disp.*$)$). Note that lower disparity means better fairness.

Datasets We use a total of three datasets for training: one synthetic dataset and two real-world benchmark datasets. We generate the synthetic dataset using a method similar to Zafar et al. (2017a). The synthetic training dataset has 2,000 samples and consists of two non-sensitive attributes $(\mathrm{x}_1, \mathrm{x}_2)$, one sensitive attribute z, and one label attribute y – see details in Sec. B.4. We also utilize two real datasets: ProPublica COMPAS (Angwin et al., 2016) consists of 5,278 samples, and its labels indicate recidivism; AdultCensus (Kohavi, 1996) has 43,131 samples, and its labels indicate a person's annual income. We use gender as z. To construct test data representing the deployment distribution with shifted correlation (c_{test}), we use three methods: (1) re-sampling data within each (y, z)-class in the original test data, (2) modifying the z values while fixing the x and y distributions in the original test data (see details in Sec. B.4), and (3) utilizing a newly-collected data, where we train on AdultCensus (Kohavi, 1996), but test on a recent version of this dataset called ACSIncome (Ding et al., 2021).

Baselines We compare our framework with three types of baselines: (1) *vanilla (non-fair)* training using logistic regression, (2) *in-processing-only* training, and (3) *two-step* training that first runs an existing pre-processing algorithm and then an in-processing algorithm. For in-processing-only training, we use the following three approaches: Fairness Constraints (FC) (Zafar et al., 2017a;b), which adds an unfairness penalty term to the loss function; Adversarial Debiasing (AD) (Zhang et al., 2018), which adversarially trains a classifier with a fairness discriminator; and FairBatch (FB) (Roh et al., 2021), which adaptively adjusts batch ratios among groups to improve fairness. When we run in-processing approaches for multiple fairness metrics, we naturally extend each approach by combining the fairness constraints for different metrics – see details in Sec. B.4. For two-step training, we use a pre-processing algorithm called Reweighing (RW) (Kamiran & Calders, 2011) to debias the data before running the above in-processings, where RW balances the data amounts across groups. We thus run the in-processing algorithms on the less-biased data by RW.

Hyperparameters We consider *four scenarios of knowing the range of the correlation constant* c in the test data. **(1)** In Secs. 6.1 & 6.2, we assume that the exact c_{test} value is given in advance (i.e., $\alpha{=}\beta{=}c_{\text{test}}$). **(2)** In Secs. 6.3 & B.13, we use an *estimated* c range that is calculated using some of the new data. **(3)** In Sec. B.14, we evaluate with arbitrary c ranges (i.e., $c_{\text{test}} \pm x\%$). **(4)** In Secs. 6.4 & B.15, we assume that the correlation itself is *incorrect*. We set both γ_{y} and γ_{z} to 0.1, which is a larger value than the actual marginal probability changes in the test data. We also test for the values of 0.2 and 0.3, and the overall trends remain the same. The in-processings' hyperparameters are in Sec. B.4.

6.1. Accuracy and Fairness

We first compare the accuracy and fairness performances of our framework with baselines on the synthetic and COMPAS datasets in Tables 2 and 3 – see many more results in Appendix B, including the AdultCensus experiments (Sec. B.6), which show similar results. The test data has lower correlation than the training data. LR shows vanilla training without any fairness technique. Other baselines are clustered based on three in-processing approaches: FC, AD, and FB. For each in-processing approach X, applying our pre-processing (denoted as Ours+X) generally shows

Table 2. Performances on the synthetic test dataset w.r.t. a single metric (DP) and multiple metrics (DP & EO) – see the performances w.r.t. a single metric EO in Sec. B.5. The test datasets are constructed via re-sampling from the original distribution. The correlation constant c of the test data is 50% of that of the training data. We compare our framework with three types of baselines: (1) non-fair training: LR; (2) in-processing-only training: FC, AD, and FB; (3) two-step training: RW (Kamiran & Calders, 2011) + in-processings. In the last row, we also show the performances of an in-processing algorithm (FB) trained on the test distribution, which can be considered as the upper bounds.

	Single (DP)		Multiple (DP & EO)	
Method	Acc.	Unfair.	Acc.	Unfair.
LR	.865±.000	.173±.000	.865±.000	.173±.000
FC (Zafar et al., 2017a;b)	.778±.011	.038±.013	.853±.001	.075±.002
RW+FC	.848±.004	.079±.003	.848±.004	.082±.002
Ours+FC	.849±.002	**.034±.004**	.851±.005	**.060±.003**
AD (Zhang et al., 2018)	.762±.016	.032±.011	.821±.004	.068±.003
RW+AD	.845±.002	.087±.006	.847±.003	.084±.005
Ours+AD	.814±.011	**.017±.006**	.842±.009	**.054±.008**
FB (Roh et al., 2021)	.821±.000	.048±.000	.849±.001	.091±.005
RW+FB	.859±.002	.055±.003	.855±.001	.071±.008
Ours+FB	.836±.001	**.003±.001**	.852±.004	**.058±.001**
FB on test dist. (upper bound)	.838±.002	.003±.002	.859±.002	.058±.004

Table 3. Performances on the COMPAS test dataset. Other settings are identical to those in Table 2.

	Single (DP)		Multiple (DP & EO)	
Method	Acc.	Unfair.	Acc.	Unfair.
LR	.660±.000	.129±.000	.660±.000	.225±.000
FC (Zafar et al., 2017a;b)	.656±.004	.050±.021	.654±.007	.137±.034
RW+FC	.654±.004	.068±.039	.651±.010	.124±.031
Ours+FC	.657±.008	**.037±.021**	.652±.020	**.106±.038**
AD (Zhang et al., 2018)	.655±.003	.054±.015	.661±.005	**.111±.035**
RW+AD	.657±.005	.092±.022	.655±.007	.145±.056
Ours+AD	.650±.003	**.045±.008**	.664±.006	.117±.039
FB (Roh et al., 2021)	.647±.001	.038±.013	.650±.002	.187±.019
RW+FB	.653±.003	.094±.016	.652±.003	.197±.020
Ours+FB	.648±.004	**.027±.001**	.657±.004	**.130±.014**
FB on test dist. (upper bound)	.659±.001	.012±.008	.664±.001	.095±.014

better fairness while achieving comparable or even better accuracies, either when supporting only DP or both DP and EO. The in-processing-only baselines mostly show worse fairness and accuracy compared to applying our approach, as the in-processing-only baselines are trained with different biases from test data distribution in mind. The baselines of applying RW before in-processing generally do not achieve high fairness compared to ours. The reason is that existing pre-processing approaches like RW simply mitigate data bias as much as possible, which is not always beneficial for the in-processing. In comparison, our approach takes a more principled approach by adjusting the bias according to the correlation shift with the purpose of improving the in-processing performance. As a result, ours enables the in-processing approaches to be closer to optimal performances (last row). In Sec. B.7, we show that the pre-processed data by our algorithm is indeed *more aligned* with the true test distribution than the original training data.

We observe similar results when using the *two other test settings* explained above. One is to modify the z values while fixing the x and y distributions using the synthetic dataset (Sec. B.8). The other is to use two real-world income datasets collected in the 1990s and 2010s for training and testing, respectively, where they have different (y, z)-correlation values and also shifted x distribution (Sec. 6.3).

In addition, we enrich the empirical results by 1) presenting accuracy-fairness *tradeoff curves* of the methods (Sec. B.9) and 2) showing the effects of the optional step in Algorithm 1 (Sec. B.10), which finds possibly-different sample weights within each (y, z)-class to minimize the overall distribution change between the original and pre-processed data.

6.2. Varying the Correlation of the Test Data

We compare the algorithm performances when varying the correlation of the test data. Figure 3 shows the accuracy and fairness performances of FB and Ours+FB when the test data's correlation constant c varies from 10% to 70% of the training data's c. Interestingly, when the correlation of the test data differs significantly from the training data (i.e., close to 10%), the in-processing-only baseline (FB) shows worse fairness. We suspect that the baseline is mitigating a different type of bias than that of the test data. On the other hand, our pre-processing successfully enables the in-processing algorithm to achieve high accuracy and fairness (e.g., for a 10% test corr., the unfairness decreases from 0.108 to 0.003). In addition, we vary the correlation from 110% to 150% in Sec. B.12 where we show how our pre-processing enables in-processing to achieve high fairness.

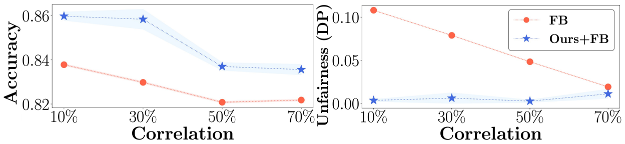

Figure 3. Performances of FB and Ours+FB on the synthetic data while varying the correlation of the test data.

6.3. Estimating Correlations and Handling x Dist. Shifts

We evaluate our method when 1) the shifted correlation is estimated from only a small amount of deployment data and 2) when the x distribution changes. We use the two real-world income datasets collected in the 1990s (i.e., AdultCensus) and 2010s (i.e., ACSIncome) for training and testing, respectively. As the two datasets have a large time gap, they naturally have different (y, z)-correlations as well as input feature x distributions (see Sec. B.13 for an analysis). Here, we assume access to only 1,000 samples ($\simeq 0.07\%$) of the 2010s data to measure the shifted correlation constant c – see the estimation performances when varying the number of samples in Sec. B.13. We compute the new c value via a maximum likelihood estimator, as described in Sec. 5.1. As a result, the estimated c is 0.140, where the true c computed on the entire new data is 0.147. Here, we can find an error range of the estimated c with a specific probability, as shown in Sec. A.7. We note that we can use the error range as $[\alpha, \beta]$. When running our pre-processing with this estimation, our framework still outperforms the in-processing baselines (Table 10). There are **two takeaways**: 1) one can reasonably estimate the correlation shift via a practical amount of new data, and 2) our method performs well even when x shifts.

6.4. Robustness Against Unknown Correlations

Finally, we evaluate our approach when the exact shifted correlation is unknown with two scenarios: 1) misspecifying the correlation in the algorithm and 2) giving a range of correlation shifts to the algorithm. **[Scenario 1]** We first consider when the shifted correlation is *incorrectly specified*. We set $\alpha{=}\beta{=}c_{\text{specified}}$, where $c_{\text{specified}}{\neq}c_{\text{test}}$. Figure 4 shows the performances of FB and Ours+FB when the true correlation of the test data is 60% of the training data's correlation. Ours improves the in-processing-only baseline's accuracy for the entire range of considered correlations and improves fairness when the specified correlation is higher than 30%. Hence, our approach is still beneficial when the estimation error is within 10%. **[Scenario 2]** In Sec. B.14, we also run our approach with *a range of correlation shifts*. Here we set the arbitrary $[\alpha, \beta]$ range to be $[c_{\text{test}}{-}x\%, c_{\text{test}}{+}x\%]$ with various x values to see the robustness of our method. As a result, there are **two takeaways**: 1) our framework successfully boosts the in-processing-only baseline performances when the $[\alpha, \beta]$ range is reasonable, and 2) even if we do not have any information about the shift, our framework performs at least as well as the in-processing-only baselines. We observe *similar results on real-world data* in Sec. B.15.

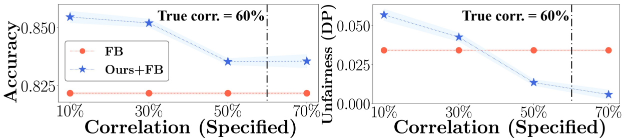

Figure 4. Performances of FB and Ours+FB on the synthetic data while varying the *specified* correlation in our algorithm to have 10% to 70% where the true correlation of the test data is 60%.

7. Conclusion

We addressed the problem of model fairness in the presence of bias changes in the data. We first formalized the notion of (y, z)-correlation for capturing bias and analyzed the performance limits of in-processing approaches in the presence of correlation shifts. We then proposed a decoupling framework where pre-processing is used to adjust the correlation, and in-processing is used for unfairness mitigation. Experiments showed how our pre-processing enables existing in-processing approaches to achieve high fairness and accuracy under correlation shifts and outperform baselines.

Acknowledgements

This work was partly supported by a Google Research Award, by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00157, Robust, Fair, Extensible Data-Centric Continual Learning and No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2018R1A5A1059921 and NRF-2022R1A2C2004382). This work was also supported in part by NSF Award DMS-2023239, NSF/Intel Partnership on Machine Learning for Wireless Networking Program under Grant No. CNS-2003129, the Understanding and Reducing Inequalities Initiative of the University of Wisconsin-Madison, Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation, and the KSEA Young Investigator Grant.

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. M. A reductions approach to fair classification. In *ICML*, pp. 60–69, 2018.

An, B., Che, Z., Ding, M., and Huang, F. Transferring fairness under distribution shifts via fair consistency regularization. In *NeurIPS*, 2022.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There's software used across the country to predict future criminals. And its biased against blacks. ProPublica, 2016.

Barocas, S., Hardt, M., and Narayanan, A. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., and Roth, A. Fairness in criminal justice risk assessments: The state of the art. *Sociol. Methods Res.*, 50(1):3–44, 2021.

Celis, L. E., Huang, L., Keswani, V., and Vishnoi, N. K. Fair classification with noisy protected attributes: A framework with provable guarantees. In *ICML*, 2021.

Chen, Y., Raab, R., Wang, J., and Liu, Y. Fairness transferability subject to bounded distribution shift. In *NeurIPS*, 2022.

Choi, K., Grover, A., Singh, T., Shu, R., and Ermon, S. Fair generative modeling via weak supervision. In *ICML*, 2020.

Chzhen, E., Denis, C., Hebiri, M., Oneto, L., and Pontil, M. Leveraging labeled and unlabeled data for consistent fair binary classification. In *NeurIPS*, pp. 12760–12770. 2019.

Cohen, J. and Cohen, P. Applied multiple regression/correlation analysis for the behavioral sciences. 1975.

Cotter, A., Jiang, H., and Sridharan, K. Two-player games for efficient non-convex constrained optimization. In *ALT*, pp. 300–332, 2019.

Dai, J. and Brown, S. M. Label bias, label shift: Fair machine learning with unreliable labels. In *NeurIPS 2020 Workshop on Consequential Decision Making in Dynamic Environments*, 2020.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring adult: New datasets for fair machine learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *NeurIPS*, 2021.

du Pin Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *NeurIPS*, pp. 3995–4004, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *ITCS*, pp. 214–226, 2012a. ISBN 978-1-4503-1115-1.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. Fairness through awareness. In *ITCS*, pp. 214–226, 2012b.

d'Aspremont, A. and Boyd, S. Relaxations and randomized methods for nonconvex qcqps. *EE392o Class Notes, Stanford University*, 1:1–16, 2003.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *SIGKDD*, pp. 259–268, 2015.

Giguere, S., Metevier, B., Brun, Y., Thomas, P. S., Niekum, S., and da Silva, B. C. Fairness guarantees under demographic shift. In *ICLR*, 2022.

Givens, C. R. and Shortt, R. M. A class of wasserstein metrics for probability distributions. *Michigan Math. J.*, 31(2):231–240, 1984.

Gu, X., Yu, X., yang, y., Sun, J., and Xu, Z. Adversarial reweighting for partial domain adaptation. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *NeurIPS*, volume 34, pp. 14860–14872, 2021.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *NeurIPS*, pp. 3315–3323, 2016.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. Correcting sample selection bias by unlabeled data. *NIPS*, 19:601–608, 2006.

Jiang, H. and Nachum, O. Identifying and correcting label bias in machine learning. In *AISTATS*, pp. 702–712, 2020.

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

Kamiran, F., Karim, A., and Zhang, X. Decision theory for discrimination-aware classification. In *IEEE ICDM*, pp. 924–929, 2012.

Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., and Schölkopf, B. Avoiding discrimination through causal reasoning. In *NeurIPS*, pp. 656–666, 2017.

Kleinberg, J. M., Mullainathan, S., and Raghavan, M. Inherent trade-offs in the fair determination of risk scores. In *ITCS*, pp. 43:1–43:23, 2017.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *SIGKDD*, pp. 202–207, 1996.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. Counterfactual fairness. In *NeurIPS*, pp. 4066–4076. 2017.

Li, S., Song, S., and Huang, G. Prediction reweighting for domain adaptation. *IEEE TNNLS*, 28(7):1682–1695, 2016.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE TKDE*, 31 (12):2346–2363, 2018.

Maity, S., Mukherjee, D., Yurochkin, M., and Sun, Y. Does enforcing fairness mitigate biases caused by subpopulation shift? In *NeurIPS*, 2021.

Mehrabi, N., Naveed, M., Morstatter, F., and Galstyan, A. Exacerbating algorithmic bias through fairness attacks. In *AAAI*, volume 35, pp. 8930–8938, 2021.

Menon, A. K. and Williamson, R. C. The cost of fairness in binary classification. In *FAT**, pp. 107–118, 2018.

Mishler, A. and Dalmasso, N. Fair when trained, unfair when deployed: Observable fairness measures are unstable in performative prediction settings. *ArXiv*, 2022.

Narayanan, A. Translation tutorial: 21 fairness definitions and their politics. In *ACM FAccT*, volume 1170, 2018.

Park, J. and Boyd, S. General heuristics for nonconvex quadratically constrained quadratic programming. *ArXiv*, 2017.

Peyré, G. and Cuturi, M. Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J. M., and Weinberger, K. Q. On fairness and calibration. In *NeurIPS*, pp. 5684–5693, 2017.

Quadrianto, N., Sharmanska, V., and Thomas, O. Discovering fair representations in the data domain. In *CVPR*, pp. 8227–8236, 2019.

Rezaei, A., Liu, A., Memarrast, O., and Ziebart, B. D. Robust fairness under covariate shift. *AAAI*, pp. 9419–9427, 2021.

Rodgers, J. L. and Nicewander, W. A. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

Roh, Y., Lee, K., Whang, S. E., and Suh, C. FR-Train: A mutual information-based approach to fair and robust training. In *ICML*, pp. 8147–8157, 2020.

Roh, Y., Lee, K., Whang, S. E., and Suh, C. FairBatch: Batch selection for model fairness. In *ICLR*, 2021.

Schrouff, J., Harris, N., Koyejo, S., Alabdulmohsin, I. M. I., Schnider, E., Opsahl-Ong, K., Brown, A., Roy, S., Mincu, D., Chen, C., Dieng, A., Natarajan, V., Liu, Y., Karthikesalingam, A., Heller, K., Chiappa, S., and D'Amour, A. N. Fairness properties do not transfer: do we have viable solutions for real-world applications? In *NeurIPS*, 2022.

Shen, Y. and Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, pp. 5739–5748, 2019.

Singh, H., Singh, R., Mhasawade, V., and Chunara, R. Fairness violations and mitigation under covariate shift. In *FAccT*, pp. 3–13, 2021.

Solans, D., Biggio, B., and Castillo, C. Poisoning attacks on algorithmic fairness. In *ECML PKDD*, pp. 162–177, 2020.

Thomas, P. S., Castro da Silva, B., Barto, A. G., Giguere, S., Brun, Y., and Brunskill, E. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004, 2019.

Wang, S., Guo, W., Narasimhan, H., Cotter, A., Gupta, M., and Jordan, M. Robust optimization for fairness with noisy protected groups. In *NeurIPS*, pp. 5190–5203, 2020.

Wick, M., panda, s., and Tristan, J.-B. Unlocking fairness: a trade-off revisited. In *NeurIPS*, 2019.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, pp. 962–970, 2017a.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, pp. 1171–1180, 2017b.

Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *ICML*, pp. 325–333, 2013.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *AIES*, pp. 335–340, 2018.

Zhang, F. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Zhang, J. and Bareinboim, E. Fairness in decision-making - the causal explanation formula. In *AAAI*, pp. 2037–2045, 2018.

Zhao, H., Coston, A., Adel, T., and Gordon, G. J. Conditional learning of fair representations. In *ICLR*, 2020.

A. Appendix – Theory

A.1. Toy Example

In fair training, the data bias reflects the relationship between a label y and group attribute z. For example, if all positive labels are in the same group, the data can be considered highly biased. Conversely, if the labels are randomly assigned to all groups, the data can be considered unbiased.

A bias change in the deployment data may have an adverse affect on a trained model's performance. Figure 5 shows a toy example that illustrates how a fair classifier's performance is affected by a bias change during deployment. Here, the bias can be expressed via correlation, and we discuss their relationship in Sec. 4. The training distribution is biased, where Group 2 has more positive labels than Group 1. On the other hand, in the deployment distribution, the bias decreases as the labels are equally distributed for each group. Suppose we train a fair classifier on the training data as shown on the left side where the DP fairness is perfect (i.e., $\Pr(\hat{y}=1)$ are the same for the groups), but the accuracy is not as a result. On the deployment data, the DP worsens while the accuracy still remains imperfect. The underlying problem is that the classifier was trained with a different bias in mind.

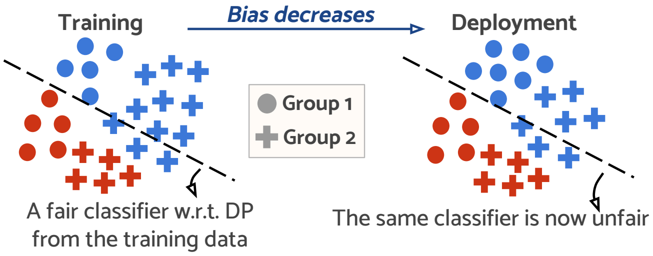
Training
Bias decreases
Deployment
● Group 1
✚ Group 2
A fair classifier w.r.t. DP from the training data
The same classifier is now unfair

Figure 5. A toy example for illustrating the impact of bias changes. The blue and red colors indicate positive and negative examples, respectively. On the left, there is a fair classifier that achieves DP on the training distribution. On the right, the deployment data has a lower bias, making the trained classifier both unfair and inaccurate.

A.2. Proof for Lemma 2

Continuing from Sec. 4.1, we provide a proof for Lemma 2.

Proof. We denote the Pearson's correlation coefficient between y and z as ρ_{yz}. By definition, $\rho_{\mathrm{yz}} = \frac{Cov(\mathrm{y},\mathrm{z})}{\sigma(\mathrm{y})\sigma(\mathrm{z})}$. For binary y and z, we can rewrite ρ_{yz} as follows (Cohen & Cohen, 1975):

$$\rho_{\mathrm{yz}} = \frac{Cov(\mathrm{y},\mathrm{z})}{\sigma(\mathrm{y})\sigma(\mathrm{z})} = \frac{\Pr(\mathrm{y}=1,\mathrm{z}=1)\Pr(\mathrm{y}=0,\mathrm{z}=0) - \Pr(\mathrm{y}=1,\mathrm{z}=0)\Pr(\mathrm{y}=0,\mathrm{z}=1)}{\sqrt{\Pr(\mathrm{y}=1)\Pr(\mathrm{y}=0)\Pr(\mathrm{z}=1)\Pr(\mathrm{z}=0)}}$$

If the marginal probabilities of y and z (i.e., $\Pr(\mathrm{y}=y)$ and $\Pr(\mathrm{z}=z)$) remain the same,

$$\begin{aligned}
\rho_{\mathrm{yz}} &\propto \Pr(\mathrm{y}=1,\mathrm{z}=1)\Pr(\mathrm{y}=0,\mathrm{z}=0) - \Pr(\mathrm{y}=1,\mathrm{z}=0)\Pr(\mathrm{y}=0,\mathrm{z}=1) \\
&\propto \frac{\Pr(\mathrm{y}=1,\mathrm{z}=1)}{\Pr(\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=0,\mathrm{z}=0)}{\Pr(\mathrm{z}=0)} - \frac{\Pr(\mathrm{y}=1,\mathrm{z}=0)}{\Pr(\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=0,\mathrm{z}=1)}{\Pr(\mathrm{z}=1)} \\
&= \Pr(\mathrm{y}=1|\mathrm{z}=1)\Pr(\mathrm{y}=0|\mathrm{z}=0) - \Pr(\mathrm{y}=1|\mathrm{z}=0)\Pr(\mathrm{y}=0|\mathrm{z}=1) \\
&= \Pr(\mathrm{y}=1|\mathrm{z}=1)(1-\Pr(\mathrm{y}=1|\mathrm{z}=0)) - \Pr(\mathrm{y}=1|\mathrm{z}=0)(1-\Pr(\mathrm{y}=1|\mathrm{z}=1)) \\
&= \Pr(\mathrm{y}=1|\mathrm{z}=1) - \Pr(\mathrm{y}=1|\mathrm{z}=0) - \Pr(\mathrm{y}=1|\mathrm{z}=1)\Pr(\mathrm{y}=1|\mathrm{z}=0) + \Pr(\mathrm{y}=1|\mathrm{z}=0)\Pr(\mathrm{y}=1|\mathrm{z}=1) \\
&= \Pr(\mathrm{y}=1|\mathrm{z}=1) - \Pr(\mathrm{y}=1|\mathrm{z}=0).
\end{aligned}$$

Therefore, if the marginal probabilities of y and z remain the same, the (y,z)-correlation ρ_{yz} is proportional to $\Pr(\mathrm{y}=1|\mathrm{z}=1) - \Pr(\mathrm{y}=1|\mathrm{z}=0)$, which is the difference between the conditional probabilities of y given different z values.

□

A.3. The Accuracy-Fairness Tradeoff When Improving Fairness w.r.t. a Single Metric

Continuing from Sec. 4.1, we show that the (y,z)-correlation determines the accuracy-fairness tradeoff under certain conditions by applying Lemma 2 to Proposition 1.

The previous work (Menon & Williamson, 2018) shows that a higher alignment between y and z leads to a worse accuracy-fairness tradeoff w.r.t. demographic parity – more details are described in Proposition 8 in Menon & Williamson (2018). According to the previous work, alignment is defined as how many examples in each group have a specific label. We can thus measure the alignment as $\Pr(\mathrm{y}=1|\mathrm{z}=1)+\Pr(\mathrm{y}=0|\mathrm{z}=0)$. Then, we can rewrite the equation as follows:

$$\begin{aligned}\text{alignment} &= \Pr(\mathrm{y}=1|\mathrm{z}=1)+\Pr(\mathrm{y}=0|\mathrm{z}=0)\\ &= \Pr(\mathrm{y}=1|\mathrm{z}=1)+(1-\Pr(\mathrm{y}=1|\mathrm{z}=0))\\ &= 1+\Pr(\mathrm{y}=1|\mathrm{z}=1)-\Pr(\mathrm{y}=1|\mathrm{z}=0).\end{aligned}$$

By applying Lemma 2 to the above result, we observe that the alignment between y and z is proportional to ((y,z)-correlation $+1$) when the marginal distributions of y and z remain the same. Therefore, when a model is trained w.r.t. demographic parity and the marginal probabilities of y and z do not change, a higher (y,z)-correlation results in a worse accuracy-fairness tradeoff.

A.4. ε-DP & ε-EO

Continuing from Sec. 4.1, we provide a proof for Proposition 6.

Proof. A model achieves both ε-DP and ε-EO when the two inequalities $|\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z)-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z')|\le\varepsilon$ (i.e., ε-DP) and $|\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=z)-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=z')|\le\varepsilon$ (i.e., ε-EO) are satisfied, where $y\neq y', y,y'\in\{0,1\}$ and $z\neq z', z,z'\in\{0,1\}$.

By combining the law of total probability and ε-DP, we can get the following inequality:

$$\begin{aligned}-\varepsilon\le{}&\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z,\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z,\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z')\le\varepsilon.\end{aligned}\tag{1}$$

Also, ε-EO can be rewritten as follows:

$$-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\le\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z,\mathrm{y}=y)\le\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y).\tag{2}$$

By substituting Eq. 2 to Eq. 1, we can get the following inequality:

$$\begin{aligned}&(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y'))\Pr(\mathrm{y}=y'|\mathrm{z}=z)+(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z')\\ &\le\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z,\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z,\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z')\\ &\le(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y'))\Pr(\mathrm{y}=y'|\mathrm{z}=z)+(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z').\end{aligned}\tag{3}$$

By subtracting Eq. 2 from Eq. 3,

$$\begin{aligned}&-\varepsilon+(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y'))\Pr(\mathrm{y}=y'|\mathrm{z}=z)+(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z')\\ &\le 0\\ &\le\varepsilon+(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y'))\Pr(\mathrm{y}=y'|\mathrm{z}=z)+(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))\Pr(\mathrm{y}=y|\mathrm{z}=z)\\ &-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y')\Pr(\mathrm{y}=y'|\mathrm{z}=z')-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y)\Pr(\mathrm{y}=y|\mathrm{z}=z').\end{aligned}$$

By rearranging the terms, we get the followings:

$$\begin{aligned}
&-2\varepsilon \le (\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y') - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))(\Pr(\mathrm{y}=y'|\mathrm{z}=z) - \Pr(\mathrm{y}=y'|\mathrm{z}=z')) \le 2\varepsilon \\
&\Rightarrow -2\varepsilon \le (\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y') - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z',\mathrm{y}=y))(1 - \Pr(\mathrm{y}=y|\mathrm{z}=z) - 1 + \Pr(\mathrm{y}=y|\mathrm{z}=z')) \le 2\varepsilon.
\end{aligned}$$

Thus, we get the following inequality, which is in Proposition 6:

$$|\Pr(\mathrm{y}=y|\mathrm{z}=z) - \Pr(\mathrm{y}=y|\mathrm{z}=z')|\,|\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=z) - \Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y',\mathrm{z}=z)| \le 2\varepsilon.$$

□

A.5. ε-PP & ε-DP

Continuing from Sec. 4.1, we consider fair training w.r.t. both predictive parity (PP) and demographic parity (DP). We can define ε-PP & ε-DP as follows:

$$\begin{gathered}
-\varepsilon \le \Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0) - \Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1) \le \varepsilon \ \text{ (i.e., } \varepsilon\text{-PP)} \\
-\varepsilon \le \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0) - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1) \le \varepsilon \ \text{ (i.e., } \varepsilon\text{-DP)}
\end{gathered}$$

Based on these definitions, we give a proposition for ε-PP and ε-DP:

Proposition 9 (ε-PP & ε-DP). *Let a model achieve both ε-PP and ε-DP. Then, the following inequality holds:*

$$\frac{|\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z) - \Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z')|}{2\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=z) + \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z) + \Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=z')} \le \varepsilon,\ \ z \ne z',\ z,z' \in \mathbb{Z},\ y \in \mathbb{Y}.$$

Proof. From ε-DP, we get

$$-\varepsilon + \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1) \le \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0) \le \varepsilon + \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1). \tag{4}$$

From ε-PP, we get

$$-\varepsilon \le \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}0)\frac{\Pr(\mathrm{z}{=}0)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}0)}\frac{\Pr(\mathrm{y}{=}y,\hat{\mathrm{y}}{=}y,\mathrm{z}{=}0)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}0)} - \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}1)\frac{\Pr(\mathrm{z}{=}1)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}1)}\frac{\Pr(\mathrm{y}{=}y,\hat{\mathrm{y}}{=}y,\mathrm{z}{=}1)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}1)} \le \varepsilon. \tag{5}$$

By substituting Eq. 4 to Eq. 5, we can get the following inequality:

$$\begin{aligned}
&\{-\varepsilon + \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\}\frac{\Pr(\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)} - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\frac{\Pr(\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)} \\
&\le \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)\frac{\Pr(\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)} - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\frac{\Pr(\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)} \qquad (6) \\
&\le \{\varepsilon + \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\}\frac{\Pr(\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)} - \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\frac{\Pr(\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}.
\end{aligned}$$

By subtracting Eq. 5 from Eq. 6,

$$\begin{aligned}
&-\varepsilon + \{-\varepsilon + \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}1)\}\frac{\Pr(\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)} - \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}1)\frac{\Pr(\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)} \\
&\le 0 \\
&\le \varepsilon + \{\varepsilon + \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}1)\}\frac{\Pr(\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)} - \Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}1)\frac{\Pr(\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\frac{\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}.
\end{aligned}$$

By rearranging the terms, we get the following inequality:

$$\begin{aligned}
&-\varepsilon \cdot \frac{\Pr(\mathrm{z}=0)\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{(\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0))^2} - \varepsilon \\
&\le \Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\{\frac{\Pr(\mathrm{z}=1)\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=1)}{(\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1))^2} - \frac{\Pr(\mathrm{z}=0)\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{(\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0))^2}\} \\
&\le \varepsilon \cdot \frac{\Pr(\mathrm{z}=0)\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y,\mathrm{z}=0)}{(\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0))^2} + \varepsilon
\end{aligned}$$

which is rewritten as follows:

$$\begin{aligned}
&-\varepsilon\cdot\frac{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}\\
&\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\{\frac{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)}-\frac{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}\}\\
&\leq\varepsilon\cdot\frac{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}.
\end{aligned}$$

By multiplying $\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}$ for all terms in the above inequality, we can get

$$-\varepsilon\leq\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)}\leq\varepsilon. \tag{7}$$

Let $A=\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)$. Since $-\varepsilon\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\leq\varepsilon$ (i.e., ε-DP), we can make another inequality from Eq. 7:

$$\begin{aligned}
&\frac{(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{A}\\
&\leq\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1)\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{A}\\
&\leq\frac{(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{A}.
\end{aligned} \tag{8}$$

By subtracting Eq. 7 and Eq. 8,

$$\begin{aligned}
&-\varepsilon+\frac{(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{A}\\
&\leq 0\\
&\leq\varepsilon+\frac{(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=1))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)-(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0))\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)}{A}.
\end{aligned}$$

Since $\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z)\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=z)=\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z)$, we can rewritten the above inequality as follows:

$$\begin{aligned}
&-\varepsilon+\frac{-\varepsilon\{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)+\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)\}+\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=1)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=0)}{A}\\
&\leq 0\\
&\leq\varepsilon+\frac{\varepsilon\{\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)+\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)\}+\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=1)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=0)}{A}.
\end{aligned}$$

Now, let $B=\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)+\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)$ and $C=\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=1)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=0)$. Then,

$$-\varepsilon+\frac{1}{A}(-\varepsilon\cdot B+C)\leq 0\leq\varepsilon+\frac{1}{A}(\varepsilon\cdot B+C).$$

By arranging the terms,

$$\begin{aligned}
-\varepsilon\cdot(1+\frac{B}{A})+\frac{C}{A}\leq 0\leq\varepsilon\cdot(1+\frac{B}{A})+\frac{C}{A}\\
\implies-\varepsilon\leq\frac{-C}{A+B}\leq\varepsilon.
\end{aligned}$$

Therefore, we can conclude

$$\frac{|\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=0)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=1)|}{2\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=0)+\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)}\leq\varepsilon.$$

If we reverse the z values in the derivation, we get the same formula with only the z value changed in the above expression. As a result, we get the following inequality, which is in Proposition 9:

$$\frac{|\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z')|}{2\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=z)+\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z)+\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=z')}\leq\varepsilon,\ \ z\neq z',\ z,z'\in\mathbb{Z},\ y\in\mathbb{Y}.$$

□

Therefore, to make $\varepsilon=0$, the numerator term $|\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z')|$ should be zero. When $|\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z)-\Pr(\mathrm{y}=y,\hat{\mathrm{y}}=y|\mathrm{z}=z')|=0$, the following is satisfied: $(\mathrm{y},\hat{\mathrm{y}})\perp\mathrm{z}$, which implies $\mathrm{y}\perp\mathrm{z}$ and $\hat{\mathrm{y}}\perp\mathrm{z}$. Here, $\mathrm{y}\perp\mathrm{z}$ indicates that $|\Pr(\mathrm{y}=y|\mathrm{z}=z)-\Pr(\mathrm{y}=y|\mathrm{z}=z')|$ is zero. As a result, perfectly satisfying PP and DP requires the data to be fully unbiased. We thus suspect that the achievable model fairness w.r.t. both PP and DP is affected by the (y,z)-correlation.

A.6. ε-EO & ε-PP

Continuing from Sec. 4.1, we consider fair training w.r.t. both equalized odds (EO) and predictive parity (PP). We can define ε-EO & ε-PP as follows:

$$-\varepsilon\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=0)-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\leq\varepsilon\ \ \text{(i.e., }\varepsilon\text{-EO)}$$
$$-\varepsilon\leq\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=0)-\Pr(\mathrm{y}=y|\hat{\mathrm{y}}=y,\mathrm{z}=1)\leq\varepsilon\ \ \text{(i.e., }\varepsilon\text{-PP)}$$

Based on these definitions, we give a proposition for ε-EO and ε-PP:

Proposition 10 (ε-EO & ε-PP)**.** *Let a model achieve both ε-EO and ε-PP. Then, the following inequality holds:*

$$\frac{\Pr(\hat{\mathrm{y}}{=}y|\mathrm{y}{=}y,\mathrm{z}{=}z')}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}z)+\Pr(\mathrm{y}{=}y,\mathrm{z}{=}z)}\cdot|\Pr(\mathrm{y}{=}y)-\Pr(\mathrm{y}{=}y|\mathrm{z}{=}z')\cdot\frac{\Pr(\hat{\mathrm{y}}{=}y)}{\Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}z')}|\leq\varepsilon,\ \ z\neq z',\ z,z'\in\mathbb{Z},\ y\in\mathbb{Y}.$$

Proof. From ε-EO, we get

$$-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=0)\leq\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1).\tag{9}$$

From ε-PP, we get

$$-\varepsilon\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=0)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\leq\varepsilon.\tag{10}$$

By substituting Eq. 9 to Eq. 10, we can get the following inequality:

$$\begin{aligned}&(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1))\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\\&\leq\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=0)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\\&\leq(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1))\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}.\end{aligned}\tag{11}$$

By subtracting Eq. 10 from Eq. 11,

$$\begin{aligned}&-\varepsilon+(-\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1))\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\\&\leq0\\&\leq\varepsilon+(\varepsilon+\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1))\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}.\end{aligned}$$

By rearranging the terms, we get the following inequality:

$$
\begin{aligned}
&-\varepsilon\cdot\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}\\
&\le \Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)\{\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\}\\
&\le \varepsilon\cdot\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}.
\end{aligned}
$$

By multiplying $\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}$ for all terms in the inequality, we can get

$$
-\varepsilon\le\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}\{\frac{\Pr(\mathrm{y}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}-\frac{\Pr(\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\}\le\varepsilon
$$

which is rewritten as follows:

$$
-\varepsilon\le\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}\{\Pr(\mathrm{y}=y,\mathrm{z}=0)-\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\cdot\Pr(\mathrm{y}=y,\mathrm{z}=1)\}\le\varepsilon.
$$

Since $\Pr(\mathrm{y}=y,\mathrm{z}=0)=\Pr(\mathrm{y}=y)-\Pr(\mathrm{y}=y,\mathrm{z}=1)$ by the total probability law,

$$
-\varepsilon\le\frac{\Pr(\hat{\mathrm{y}}{=}y|\mathrm{y}{=}y,\mathrm{z}{=}1)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}0)+\Pr(\mathrm{y}{=}y,\mathrm{z}{=}0)}\{\Pr(\mathrm{y}{=}y)-\Pr(\mathrm{y}{=}y,\mathrm{z}{=}1)-\frac{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}0)}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}1)}\cdot\Pr(\mathrm{y}{=}y,\mathrm{z}{=}1)\}\le\varepsilon.
$$

By arranging the terms,

$$
\begin{aligned}
&-\varepsilon\le\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}\{\Pr(\mathrm{y}=y)-\Pr(\mathrm{y}=y,\mathrm{z}=1)\cdot(1+\frac{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)})\}\le\varepsilon\\
&\Rightarrow-\varepsilon\le\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}\{\Pr(\mathrm{y}=y)-\Pr(\mathrm{y}=y,\mathrm{z}=1)\cdot\frac{\Pr(\hat{\mathrm{y}}=y)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}\}\le\varepsilon.
\end{aligned}
$$

Therefore, we can conclude

$$
\frac{\Pr(\hat{\mathrm{y}}=y|\mathrm{y}=y,\mathrm{z}=1)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=0)+\Pr(\mathrm{y}=y,\mathrm{z}=0)}\cdot|\Pr(\mathrm{y}=y)-\Pr(\mathrm{y}=y,\mathrm{z}=1)\cdot\frac{\Pr(\hat{\mathrm{y}}=y)}{\Pr(\hat{\mathrm{y}}=y,\mathrm{z}=1)}|\le\varepsilon.
$$

If we reverse the z values in the derivation, we get the same formula with only the z value changed in the above expression. As a result, we get the following inequality, which is in Proposition 10:

$$
\frac{\Pr(\hat{\mathrm{y}}{=}y|\mathrm{y}{=}y,\mathrm{z}{=}z')}{\Pr(\hat{\mathrm{y}}{=}y,\mathrm{z}{=}z)+\Pr(\mathrm{y}{=}y,\mathrm{z}{=}z)}\cdot|\Pr(\mathrm{y}{=}y)-\Pr(\mathrm{y}{=}y|\mathrm{z}{=}z')\cdot\frac{\Pr(\hat{\mathrm{y}}{=}y)}{\Pr(\hat{\mathrm{y}}{=}y|\mathrm{z}{=}z')}|\le\varepsilon,\ \ z\neq z',\ z,z'\in\mathbb{Z},\ y\in\mathbb{Y}.
$$

□

Therefore, $\varepsilon=0$ when $\Pr(\mathrm{y}=y)=\Pr(\mathrm{y}=y|\mathrm{z}=z')$ and $\Pr(\hat{\mathrm{y}}=y)=\Pr(\hat{\mathrm{y}}=y|\mathrm{z}=z')$ (i.e., $\mathrm{y}\perp\mathrm{z}$ and $\hat{\mathrm{y}}\perp\mathrm{z}$). Here, satisfying both $\mathrm{y}\perp\mathrm{z}$ and $\hat{\mathrm{y}}\perp\mathrm{z}$ is the sufficient condition of perfectly satisfying EO and PP. Note that $\mathrm{y}\perp\mathrm{z}$ implies that the data is unbiased. We thus suspect that the achievable model fairness w.r.t. both EO and PP is affected by the (y,z)-correlation.

A.7. Estimating Shifted Correlation Range Using Samples

Continuing from Sec. 5.1, we discuss using samples in the deployment data to estimate the shifted correlation value c and its range $[\alpha, \beta]$. To this end, we show the relationship between the number of samples and the confidence of the estimation through the following theorem.

Theorem 11. *Let n_1, n_0, n_{11}, and n_{01} be the number of samples with $(\mathrm{z}=1)$, $(\mathrm{z}=0)$, $(\mathrm{z}=1, \mathrm{y}=1)$, and $(\mathrm{z}=0, \mathrm{y}=1)$, respectively. Let $\hat{c} = \dfrac{n_{11}}{n_1} - \dfrac{n_{01}}{n_0}$. If $n_1 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$ and $n_0 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$, then $|\hat{c} - c| \leq \epsilon$ with probability $1-\delta$.*

Proof. By Hoeffding's inequality (Hoeffding, 1963), we can write the following inequality w.r.t. $\frac{n_{11}}{n_1}$:

$$\Pr\left(\left|\frac{n_{11}}{n_1} - \Pr(\mathrm{y}=1|\mathrm{z}=1)\right| \geq \frac{\epsilon}{2}\right) \leq 2\exp\left(-\frac{n_1\epsilon^2}{2}\right). \tag{12}$$

Using the condition w.r.t. n_1 in Theorem 11 (i.e., $n_1 \geq \frac{2\ln(4/\delta)}{\epsilon^2}$), we can rewrite the Hoeffding's inequality as follows:

$$\begin{aligned}
\Pr\left(\left|\frac{n_{11}}{n_1} - \Pr(\mathrm{y}=1|\mathrm{z}=1)\right| \geq \frac{\epsilon}{2}\right) &\leq 2\exp\left(-\frac{n_1\epsilon^2}{2}\right) && (13)\\
&\leq 2\exp\left(-\frac{2\ln(4/\delta)}{\epsilon^2}\frac{\epsilon^2}{2}\right) && (14)\\
&= 2\exp(\ln(\delta/4)) && (15)\\
&= \frac{\delta}{2}. && (16)
\end{aligned}$$

Similarly, the inequality w.r.t. $\frac{n_{01}}{n_0}$ can be written as follows:

$$\Pr\left(\left|\frac{n_{01}}{n_0} - \Pr(\mathrm{y}=1|\mathrm{z}=0)\right| \geq \frac{\epsilon}{2}\right) \leq 2\exp\left(-\frac{n_0\epsilon^2}{2}\right) \leq \frac{\delta}{2}. \tag{17}$$

These inequalities show that if $n_1 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$ and $n_0 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$, then $|\dfrac{n_{11}}{n_1} - \Pr(\mathrm{y}{=}1|\mathrm{z}{=}1)| \leq \epsilon/2$ and $|\dfrac{n_{01}}{n_0} - \Pr(\mathrm{y}{=}1|\mathrm{z}{=}0)| \leq \epsilon/2$ each with probability at least $1-\delta/2$.

Therefore, if $n_1 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$ and $n_0 \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$, then $|\hat{c} - c| = |(\dfrac{n_{11}}{n_1} - \dfrac{n_{01}}{n_0}) - (\Pr(\mathrm{y}{=}1|\mathrm{z}{=}1) - \Pr(\mathrm{y}{=}1|\mathrm{z}{=}0))| \leq \epsilon/2 + \epsilon/2 \leq \epsilon$ with probability at least $1-\delta$. □

By using Theorem 11, we can calculate how many samples are required to achieve a specific error range with a specific probability. For example, if we aim to have an error range $c - 0.1 \leq \hat{c} \leq c + 0.1$ with 0.95 probability (i.e., $\epsilon = 0.1$ and $\delta = 0.05$), we need $n_1 \geq 876$ and $n_0 \geq 876$ samples to satisfy the condition $n \geq \dfrac{2\ln(4/\delta)}{\epsilon^2}$. Also, when we measure $\hat{c}$ using m $(= n_1 + n_0)$ samples, we can construct the shift range $[\alpha, \beta]$ with a specific probability $1-\delta$ – see details in the following section (Sec. A.8).

A.8. Maximum Likelihood Estimation in Our Scenario

Continuing from Sec. 5.1, we explain the details of the maximum likelihood estimation (MLE) in Algorithm 1. During the estimation, we use Theorem 11 in Sec. A.7, which explains how to get the confidence interval of the estimation using m samples with a specific probability $(1-\delta)$. Note that the confidence level $(1-\delta)$ is a hyperparameter (e.g., $(1-\delta) = 0.9$).

Algorithm 2 shows the overall steps in the estimation. We first calculate the estimated correlation constant c (i.e., $\hat{c}$). Then, we find the error bound ϵ by using the conditions in Theorem 11 (i.e., $\epsilon = \sqrt{\frac{2}{n}\ln(\frac{4}{\delta})}$). As a result, we can construct the shift range $[\alpha, \beta] = [\hat{c} - \epsilon, \hat{c} + \epsilon]$ with a probability $1-\delta$.

Algorithm 2 MLE

```
Input: $m$ samples of $(\mathrm{y}_{\text{deploy}}, \mathrm{z}_{\text{deploy}})$, confidence level of estimation $1-\delta$
$\hat{\Pr}(\mathrm{y}=y|\mathrm{z}=z) = \frac{(\text{num. of samples with } (\mathrm{y}=y, \mathrm{y}=z))}{(\text{num. of samples with } (\mathrm{y}=z))}$
$\hat{c} = \frac{\hat{\Pr}(\mathrm{y}=1|\mathrm{z}=1)}{\hat{\Pr}(\mathrm{z}=1)} - \frac{\hat{\Pr}(\mathrm{y}=1|\mathrm{z}=0)}{\hat{\Pr}(\mathrm{z}=0)}$
$n_1 = $ num. of samples with $(\mathrm{z}=1)$
$n_0 = $ num. of samples with $(\mathrm{z}=0)$
$n = \min(n_1, n_0)$
$\epsilon = \sqrt{\frac{2}{n}\ln\left(\frac{4}{\delta}\right)}$
$\alpha = \hat{c} - \epsilon$
$\beta = \hat{c} + \epsilon$
Output: $\alpha$, $\beta$
```

A.9. Semidefinite Relaxation

Continuing from Sec. 5.1, we provide details of the semidefinite relaxation in our optimization.

Recall our original optimization as follows:

$$
\begin{aligned}
&\min_{w'} \sum_{\forall y,z} (w_{\mathrm{y}=y,\mathrm{z}=z} - w'_{\mathrm{y}=y,\mathrm{z}=z})^2 \\
&\text{s.t. } \alpha \le \frac{w'_{\mathrm{y}=1,\mathrm{z}=1}}{w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=0,\mathrm{z}=1}} - \frac{w'_{\mathrm{y}=1,\mathrm{z}=0}}{w'_{\mathrm{y}=1,\mathrm{z}=0} + w'_{\mathrm{y}=0,\mathrm{z}=0}} \le \beta, \\
&|(w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=1,\mathrm{z}=0}) - \Pr_{\text{train}}(\mathrm{y}=1)| \le \gamma_{\mathrm{y}}, \\
&|(w'_{\mathrm{y}=1,\mathrm{z}=1} + w'_{\mathrm{y}=0,\mathrm{z}=1}) - \Pr_{\text{train}}(\mathrm{z}=1)| \le \gamma_{\mathrm{z}}, \\
&\sum_{\forall y,z} w'_{\mathrm{y}=y,\mathrm{z}=z} = 1,\ \ 0 \le w'_{\mathrm{y}=y,\mathrm{z}=z} \le 1,\ \ \forall y \in \{0,1\}, z \in \{0,1\}
\end{aligned}
$$

where $\Pr_{\text{train}}(\mathrm{y}=1) = w_{\mathrm{y}=1,\mathrm{z}=1} + w_{\mathrm{y}=1,\mathrm{z}=0}$ and $\Pr_{\text{train}}(\mathrm{z}=1) = w_{\mathrm{y}=1,\mathrm{z}=1} + w_{\mathrm{y}=0,\mathrm{z}=1}$.

As the above optimization is a non-convex quadratically constrained quadratic problem (non-convex QCQP), we now apply the semidefinite relaxation (SDP relaxation) (Park & Boyd, 2017). We first rewrite the above optimization using matrices:

$$
\begin{aligned}
&\min_{x}\ x^T P_0 x + q_0^T x \\
&\text{s.t. } x^T P_\alpha x \ge 0,\ x^T P_\beta x \le 0, \\
&|q_2^T x - \Pr_{\text{train}}(y=1)| \le \gamma_{\mathrm{y}}, \\
&|q_3^T x - \Pr_{\text{train}}(z=1)| \le \gamma_{\mathrm{z}} \\
&q_4^T x = 1,\ \ 0 \le x_i \le 1\ \ \forall i
\end{aligned}
$$

where $x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} w'_{1,1} & w'_{1,0} & w'_{0,1} & w'_{0,0} \end{bmatrix}^T$, $q_0 = -2\begin{bmatrix} w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \end{bmatrix}^T$, $q_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}^T$, $q_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T$, $q_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$, $P_0 = \mathrm{diag}(\mathbf{1})$, $P_\alpha = \begin{bmatrix} 0 & -\alpha/2 & 0 & (1-\alpha)/2 \\ -\alpha/2 & 0 & (-1-\alpha)/2 & 0 \\ 0 & (-1-\alpha)/2 & 0 & -\alpha/2 \\ (1-\alpha)/2 & 0 & -\alpha/2 & 0 \end{bmatrix}$, and $P_\beta = \begin{bmatrix} 0 & -\beta/2 & 0 & (1-\beta)/2 \\ -\beta/2 & 0 & (-1-\beta)/2 & 0 \\ 0 & (-1-\beta)/2 & 0 & -\beta/2 \\ (1-\beta)/2 & 0 & -\beta/2 & 0 \end{bmatrix}$.

As $x^T P x = \mathbf{Tr}(P(xx^T))$, we can get the following optimization:

$$\begin{aligned}\min_{X,x}\ & \mathbf{Tr}(XP_0) + q_0^T x \\ \text{s.t. } & \mathbf{Tr}(XP_\alpha) \geq 0,\ \mathbf{Tr}(XP_\beta) \leq 0 \\ & |q_2^T x - \Pr_{\text{train}}(y=1)| \leq \gamma_{\text{y}}, \\ & |q_3^T x - \Pr_{\text{train}}(z=1)| \leq \gamma_{\text{z}} \\ & q_4^T x = 1,\ 0 \leq x_i \leq 1\ \forall i, \\ & X = xx^T\end{aligned}$$

where $\mathbf{Tr}(\cdot)$ is the trace and $X = xx^T$.

In the above optimization, the last constraint is non-convex. Thus, we relax the non-convex constraint $X = xx^T$ into $X - xx^T \succeq 0$, which is convex, and then use a Schur complement (Zhang, 2006) to get the final SDP form:

$$\begin{aligned}\min_{X,x}\ & \mathbf{Tr}(XP_0) + q_0^T x \\ \text{s.t. } & \mathbf{Tr}(XP_\alpha) \geq 0,\ \mathbf{Tr}(XP_\beta) \leq 0 \\ & |q_2^T x - \Pr_{\text{train}}(y=1)| \leq \gamma_{\text{y}}, \\ & |q_3^T x - \Pr_{\text{train}}(z=1)| \leq \gamma_{\text{z}} \\ & q_4^T x = 1,\ 0 \leq x_i \leq 1\ \forall i, \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0.\end{aligned}$$

When we solve the above SDP relaxation problem using convex optimization solvers, we set the 5×5-matrix A as the variable so as to indicate $\begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix}$. Then, we get the solution $x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$ by taking the first four elements of the last vector in the resulting matrix A.

A.10. Optional Step for Minimizing Overall Distribution Change

Continuing from Sec. 5.2, we explain the details on the optional step in Algorithm 1. We can use the optional step (`MinDistChange`) to find non-uniform data sample weights within each (y, z)-class that minimizes the overall distribution change in terms of the Wasserstein distance (Givens & Shortt, 1984), rather than using identical weights. In the optional step, we first divide the examples in each (y=y, z=z)-class into two sets {y=y, z=z, t=0} and {y=y, z=z, t=1} with equal numbers, where t is a specific feature that can be a criterion for dividing the examples. For example, we can find a median of a non-sensitive attribute x_1 and set t = 0 for an example if the x_1 value of the example is lower than the median of x_1. We set t = 1 otherwise. We then make a candidate set of partial weights $w'_{\text{y}=y,\text{z}=z,\text{t}=t}$ on each (y=y, z=z, t=t)-class, where $\sum w'_{\text{y}=y,\text{z}=z,\text{t}=t} = w'_{\text{y}=y,\text{z}=z}$. Note that we can extend t beyond the binary setting to find more detailed partial weights. For each partial weight candidate, we calculate the candidate sample-wise weights ($\mathbf{d}_{\text{tmp}}$) to ensure that the sample weight sum in each (y=y, z=z, t=t)-class is $w_{\text{y}=y,\text{z}=z,\text{t}=t} \cdot n$, where n is the total number of samples in the original training data. Then, we draw new data $\tilde{D}$ from the original training data D via weighted sampling according to $\mathbf{d}_{\text{tmp}}$. Then, we calculate the Wasserstein distance between $\tilde{D}$ and D via an optimal transport technique (Peyré & Cuturi, 2019). As a result, the algorithm returns the final sample-wise weights ($\mathbf{d}_{\text{min}}$) that result in the closest distribution from the original data.

Algorithm 3 MinDistChange

```
Input: train data $D$, original ratio $w_{\mathrm{y,z}}$, new ratio $w'_{\mathrm{y,z}}$
In each (y=$y$, z=$z$)-class, divide the examples into two sets $\{$y=$y$, z=$z$, t=0$\}$ and $\{$y=$y$, z=$z$, t=1$\}$ with equal numbers
  partials = $[0, 1/m, 2/m, ..., 1]$
  $\mathrm{W}_{\mathrm{y}=y,\mathrm{z}=z} \leftarrow []$,  $\forall(y, z) \in \mathbb{Y} \times \mathbb{Z}$
  for each (y=$y$, z=$z$)-class do
      for p in partials do
          $w'_{\mathrm{y}=y,\mathrm{z}=z,\mathrm{t}=0} = w'_{\mathrm{y}=y,\mathrm{z}=z} \cdot p$
          $w'_{\mathrm{y}=y,\mathrm{z}=z,\mathrm{t}=1} = w'_{\mathrm{y}=y,\mathrm{z}=z} \cdot (1-p)$
          Append $[w'_{\mathrm{y}=y,\mathrm{z}=z,\mathrm{t}=0}, w'_{\mathrm{y}=y,\mathrm{z}=z,\mathrm{t}=1}]$ to $\mathrm{W}_{\mathrm{y}=y,\mathrm{z}=z}$
candidates $\leftarrow \mathrm{W}_{\mathrm{y}=1,\mathrm{z}=1} \times \mathrm{W}_{\mathrm{y}=1,\mathrm{z}=0} \times \mathrm{W}_{\mathrm{y}=0,\mathrm{z}=1} \times \mathrm{W}_{\mathrm{y}=0,\mathrm{z}=0}$
  minD $\leftarrow$ an initial large value
  for partial-weight in candidates do
      $d_j \leftarrow w'_{\mathrm{y}=y,\mathrm{z}=z,\mathrm{t}=t}/(w_{\mathrm{y}=y,\mathrm{z}=z} \cdot 0.5), \forall j \in \mathbb{I}_{(y,z,t)}, \forall(y,z,t) \in \mathbb{Y} \times \mathbb{Z} \times \{0,1\}$
      $\mathbf{d}_{\mathrm{tmp}}$ = $\{d_i\}_{i=1,\ldots,n}$
      Draw data $\tilde{D}$ from $D$ via weighted sampling w.r.t. $\mathbf{d}_{\mathrm{tmp}}$
      wassD $\leftarrow$ Calculate the Wasserstein distance between $\tilde{D}$ and $D$ via optimal transport
      if wassD < minD then
          $\mathbf{d}_{\mathrm{min}}$ = $\mathbf{d}_{\mathrm{tmp}}$
          minD = wassD
Output : $\mathbf{d}_{\mathrm{min}}$
```

B. Appendix – Experiments

B.1. Simulation Settings

Continuing from Sec. 4.1, we explain the details of the simulation. The goal of the simulation is to confirm our theoretical observations by generating various synthetic classifiers that show the full range of possible model performances. To this end, we generate 1000 data samples, where each sample has y and z features. We first set y and z to have a specific correlation. We then generate various synthetic classifiers to have different predicted labels ŷ by varying the probability $\Pr(\hat{\mathrm{y}} = 1)$ in each (y, z)-class, regardless of the other input features. Note that we do not train actual models (e.g., logistic regression, SVM). For each synthetic classifier, we measure the accuracy and fairness performances based on y, z, and the classifier's ŷ. We repeat the above procedures while varying y and z to have different correlations. As a result, we get the simulation results in Figure 2 by plotting all the classifier accuracy and fairness performances within each (y, z)-correlation.

B.2. Theoretical Limits for Equalized Odds

Continuing from Sec. 4.1, we explain why the (y, z)-correlation does not determine the limits of the accuracy-fairness tradeoff w.r.t. equalized odds (EO). Here, we consider an ideal case in terms of accuracy, where a perfectly-accurate classifier can be obtained for the given data. Under this setting, let there be a perfectly accurate classifier A (i.e., $\Pr(\hat{\mathrm{y}} = \mathrm{y}) = 1$). Since classifier A predicts all samples correctly, $\Pr(\hat{\mathrm{y}} = y | \mathrm{y} = r, \mathrm{z} = z)$ is also 1 for all y and z, which indicates perfect EO (i.e., $\Pr(\hat{\mathrm{y}} = y | \mathrm{y} = r, \mathrm{z} = 0) = \Pr(\hat{\mathrm{y}} = y | \mathrm{y} = r, \mathrm{z} = 1)$ for all y). In other words, achieving both perfect accuracy and perfect fairness w.r.t. EO is always possible, which means that the limit of the accuracy-fairness tradeoff w.r.t. EO is always the same, regardless of the data distribution, including the (y, z)-correlation. Thus, the (y, z)-correlation does not determine the *limits* of the accuracy-fairness tradeoff w.r.t. EO.

However, classifiers are not perfect *in practice*, and we empirically observe that higher (y, z)-correlation leads to a worse accuracy-fairness tradeoff w.r.t. EO in the following section.

B.3. Empirical Observations for Equalized Odds

Continuing from Sec. 4.1, we perform an experiment to observe that higher (y, z)-correlation leads to a worse accuracy-fairness tradeoff w.r.t. equalized odds (EO) in practice. Figure 6 shows the accuracy-unfairness performances of the logistic regression model trained using FairBatch (Roh et al., 2021) to improve EO on two synthetic datasets with low and high (y, z)-correlations. As a result, low correlation enables classifiers to attain better accuracy-fairness tradeoffs (i.e., close to the bottom right) w.r.t. EO. This experiment shows that the (y, z)-correlation indeed affects the accuracy-fairness tradeoff w.r.t. EO in practice.

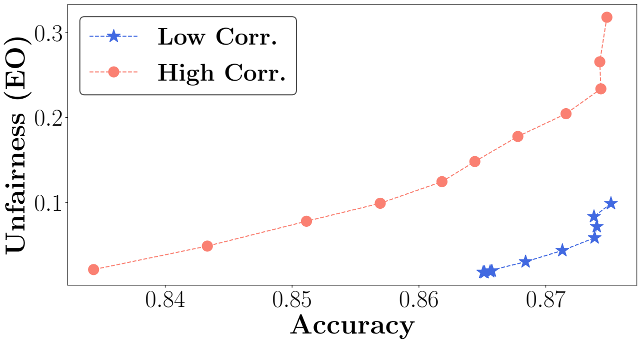

Figure 6. Accuracy-unfairness performances of fair training with FairBatch (Roh et al., 2021) on two synthetic data with different (y, z)-correlations. We measure fairness w.r.t. equalized odds (EO).

B.4. Other Experimental Settings

Continuing from Sec. 6, we provide more details on the experimental settings. The batch sizes of the synthetic, COMPAS, and AdultCensus datasets are 100, 200, and 2,000, respectively. For the synthetic dataset, we use 2,000 samples for the training dataset and 1,000 samples for the test dataset. For the real datasets, we split the entire data into 4:1 for the training and test datasets. We set the learning rate to 0.0005. Our experiments are performed using PyTorch on a Linux server with Intel Xeon Silver 4210R CPUs and NVIDIA Quadro RTX 8000 GPUs.

We generate the synthetic training dataset with 2,000 samples and consists of two non-sensitive attributes $(\mathrm{x}_1, \mathrm{x}_2)$, one sensitive attribute z, and one label attribute y. Each sample $(\mathrm{x}_1, \mathrm{x}_2, \mathrm{y})$ is drawn from the following Gaussian distributions: $(\mathrm{x}_1, \mathrm{x}_2)|\mathrm{y} = 0 \sim \mathcal{N}([-2; -2], [10, 1; 1, 3])$ and $(\mathrm{x}_1, \mathrm{x}_2)|\mathrm{y} = 1 \sim \mathcal{N}([2; 2], [5, 1; 1, 5])$. For the sensitive attribute z, we generate a biased distribution: $\Pr(\mathrm{z} = 1) = \Pr((\mathrm{x}'_1, \mathrm{x}'_2)|\mathrm{y} = 1)/[\Pr((\mathrm{x}'_1, \mathrm{x}'_2)|\mathrm{y} = 0) + \Pr((\mathrm{x}'_1, \mathrm{x}'_2)|\mathrm{y} = 1)]$ where $(\mathrm{x}'_1, \mathrm{x}'_2) = (\mathrm{x}_1 \cos(\pi/4) - \mathrm{x}_2 \sin(\pi/4), \mathrm{x}_1 \sin(\pi/4) + \mathrm{x}_2 \cos(\pi/4))$.

When we construct the synthetic test dataset by modifying the z values while fixing the x and y distributions in the original test data, we change the k value in $(\mathrm{x}'_1, \mathrm{x}'_2) = (\mathrm{x}_1 \cos(\pi/k) - \mathrm{x}_2 \sin(\pi/k), \mathrm{x}_1 \sin(\pi/k) + \mathrm{x}_2 \cos(\pi/k))$.

For all in-processing algorithms, we start from a candidate set and use cross-validation on the (pre-processed) training data to choose the hyperparameters that result in the best fairness while having an accuracy that best aligns with other results.

To support multiple fairness metrics, we naturally extend each in-processing approach by combining the fairness constraints for different metrics via a tuning knob that adjusts the importance between each metric. Here, the fairness constraints are implemented differently in each algorithm. Fairness Constraints (FC) (Zafar et al., 2017a) adds each unfairness penalty term to the loss function. Thus, we extend FC by adding multiple penalty terms and adjust the importance between each penalty term by a tuning knob. Adversarial Debiasing (AD) (Zhang et al., 2018) utilizes a discriminator of each fairness metric for the adversarial training. Thus, we extend AD by adding multiple fairness discriminators and adjust the importance between each discriminator by adding a tuning knob in the classifier's loss function. FairBatch (FB) (Roh et al., 2021) solves a bilevel optimization that has an objective for minimizing a fairness disparity according to each fairness metric. Thus, we extend FB by minimizing the maximum of fairness disparities (e.g., $\max(DPdisp., EOdisp.)$) and adjust the importance between each disparity using a tuning knob.

B.5. Accuracy and Fairness – EO only

Continuing from Sec. 6.1, we show additional results on the synthetic dataset. Here, we train the fairness algorithms w.r.t. a single fairness metric EO. As a result, Table 4 shows consistent results to those in Table 2, where our approach improves the fairness performances of the in-processing-only baselines.

Table 4. Performances on the synthetic test dataset w.r.t. a single metric EO. Other experimental settings are identical to those in Table 2.

	Single (EO)	
Method	Acc.	Unfair.
LR	.866 ± .001	.101 ± .001
FC	.865 ± .003	.062 ± .001
Ours+FC	.856 ± .004	.039 ± .005
AD	.854 ± .003	.043 ± .002
Ours+AD	.849 ± .004	.032 ± .007
FB	.854 ± .001	.043 ± .000
Ours+FB	.851 ± .009	.039 ± .017

B.6. Accuracy and Fairness – AdultCensus

Continuing from Sec. 6.1, we show the results on the AdultCensus dataset. Table 5 shows the accuracy and fairness performances of the algorithms on the AdultCensus test dataset w.r.t. a single metric (DP) and multiple metrics (DP & EO). Other setting are identical to Table 2. We observe consistent results where our framework improves accuracy and fairness of the in-processing-only baselines and also shows better fairness than the two-step baselines using RW.

Table 5. Performances on the AdultCensus test dataset. Other experimental settings are identical to those in Table 2.

	Single (DP)		Multiple (DP & EO)	
Method	Acc.	Unfair.	Acc.	Unfair.
LR	.824 ± .001	.074 ± .002	.824 ± .001	.074 ± .002
FC	.805 ± .013	.021 ± .004	.807 ± .013	.078 ± .017
RW+FC	.810 ± .004	.020 ± .006	.810 ± .008	.048 ± .013
Ours+FC	.820 ± .003	**.005** ± **.002**	.805 ± .014	**.033** ± **.013**
AD	.777 ± .022	**.007** ± **.004**	.806 ± .011	.052 ± .004
RW+AD	.808 ± .010	.025 ± .007	.800 ± .015	.035 ± .003
Ours+AD	.803 ± .009	**.007** ± **.003**	.805 ± .009	**.033** ± **.003**
FB	.825 ± .002	.017 ± .001	.826 ± .001	.049 ± .001
RW+FB	.818 ± .008	.020 ± .005	.817 ± .008	.071 ± .010
Ours+FB	.824 ± .001	**.008** ± **.003**	.826 ± .003	**.037** ± **.012**

B.7. How the Pre-processed Data Aligns with the True Test Data

Continuing from Sec. 6.1, we also show that the pre-processed data distribution (D_{pre}) by our algorithm is more aligned with the true test distribution (D_{test}) compared to the original training distribution (D_{train}), in terms of (y, z)-correlation and Wasserstein distance. We calculate the second-order Wasserstein distance via an optimal transport technique (Peyré & Cuturi, 2019) on the synthetic data. We experiment on three degrees on shifts and make two observations in Table 6: 1) the correlations of D_{pre} and D_{test} are indeed more similar relative to D_{train} and 2) the Wasserstein distance between D_{pre} and D_{test} is lower than between D_{train} and D_{test}. Both observations confirm that the reweighed data aligns well with the test data.

As our method improves the alignment between D_{pre} and D_{test}, the in-processing algorithms trained on D_{pre} (i.e., ours + in-processing algorithms) show high accuracy and fairness performances on D_{test}, as shown in Sec. 6.

Table 6. Alignment between data distributions. We use the synthetic data.

	Level of correlation shift in c_{test}		
	Severe	Normal	No shift
c_{train} (correlation in D_{train})	0.3591	0.3591	0.3591
c_{pre} (correlation in D_{pre})	0.0359	0.1796	0.3590
c_{test} (correlation in D_{test})	0.0360	0.1800	0.3591
Wass. dist. between D_{train} and D_{test}	0.7966	0.6807	0.4399
Wass. dist. between D_{pre} and D_{test}	0.6257	0.6022	0.4399

B.8. Accuracy and Fairness – Other Test Data Construction: Synthetic Data

Continuing from Sec. 6.1, we compare the algorithm performances when using a different method to construct the test dataset of the synthetic-data experiment. Table 7 shows the accuracy and fairness performances on the synthetic dataset when constructing the test dataset by modifying z directly.

We again generate the synthetic dataset using a method similar to Zafar et al. (2017a). The synthetic dataset consists of two non-sensitive attributes $(\mathrm{x}_1, \mathrm{x}_2)$, one sensitive attribute z, and one label attribute y. Each sample $(\mathrm{x}_1, \mathrm{x}_2, \mathrm{y})$ is drawn from the following Gaussian distributions: $(\mathrm{x}_1, \mathrm{x}_2)|\mathrm{y} = 0 \sim \mathcal{N}([-2;-2],[10,1;1,3])$ and $(\mathrm{x}_1, \mathrm{x}_2)|\mathrm{y} = 1 \sim \mathcal{N}([2;2],[5,1;1,5])$. For the sensitive attribute z, we generate a biased distribution: $\Pr(\mathrm{z} = 1) = \Pr((\mathrm{x}_1', \mathrm{x}_2')|\mathrm{y} = 1)/[\Pr((\mathrm{x}_1', \mathrm{x}_2')|\mathrm{y} = 0) + \Pr((\mathrm{x}_1', \mathrm{x}_2')|\mathrm{y} = 1)]$ where $(\mathrm{x}_1', \mathrm{x}_2') = (\mathrm{x}_1 \cos(\pi/k) - \mathrm{x}_2 \sin(\pi/k), \mathrm{x}_1 \sin(\pi/k) + \mathrm{x}_2 \cos(\pi/k))$. For the training dataset, we set k to 4. We then change the k value to generate the test dataset with 50% of the training data correlation.

As a result, we observe consistent results where our framework improves the accuracy and fairness performances of the in-processing-only baselines and generally shows better fairness than the two-step baselines using RW.

Table 7. Performances on the synthetic test dataset. The test dataset is constructed via modifying the z values of the original distribution. Other settings are identical to those in Table 2.

	Single (DP)		Multiple (DP & EO)	
Method	Acc.	Unfair.	Acc.	Unfair.
LR	.871 ± .000	.138 ± .000	.871 ± .000	.138 ± .000
FC	.805 ± .006	.040 ± .005	.830 ± .001	.119 ± .003
RW+FC	.847 ± .005	.047 ± .007	.852 ± .005	.053 ± .004
Ours+FC	.831 ± .003	**.005** ± **.004**	.854 ± .004	**.052** ± **.002**
AD	.792 ± .012	.048 ± .009	.815 ± .008	.133 ± .006
RW+AD	.836 ± .010	.042 ± .008	.849 ± .004	**.054** ± **.004**
Ours+AD	.820 ± .006	**.005** ± **.003**	.847 ± .002	.057 ± .007
FB	.804 ± .001	.051 ± .002	.831 ± .001	.128 ± .002
RW+FB	.844 ± .004	.036 ± .003	.853 ± .005	.057 ± .011
Ours+FB	.824 ± .002	**.015** ± **.002**	.854 ± .003	**.054** ± **.003**

B.9. Accuracy and Fairness – Tradeoff Curves

Continuing from Sec. 6.1, we provide the accuracy-fairness disparity (DP) trade-off curves of FairBatch (FB) and Ours+FB on the the synthetic data in Figure 7. Our framework shows a better accuracy-fairness tradeoff compared to the in-processing approach (FB). FB shows an "inversed" curve, as it is trained with wrong biases in mind.

B.10. Using the Optional Step

Continuing from Sec. 6.1, we show the results of when using the optional step using `MinDistChange` in Algorithm 1. This step finds possibly-different data sample weights within each (y, z)-class to minimize the overall distribution change between the training and pre-processed data. Figure 8 shows the Wasserstein distances between the original training data and pre-processed data when our algorithm uses either the basic or optional step, and the correlation constant c ranges from 10%

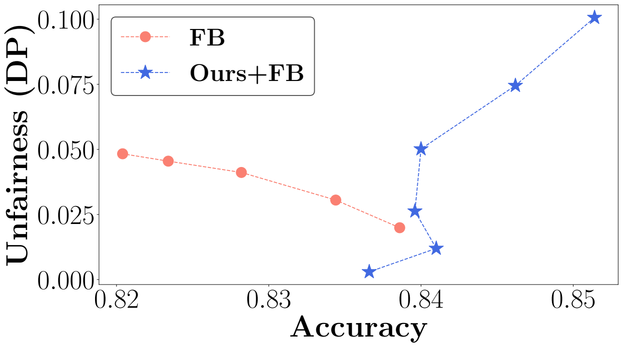

Figure 7. Accuracy-fairness disparity trade-off curves of FB and Ours+FB on the synthetic data.

to 70% of the training data's correlation. As a result, the optional step generally reduces the Wasserstein distance between the training and pre-processed data distributions, especially for smaller c values. In addition, Table 8 shows the performances on the synthetic test dataset when our algorithm uses either the basic step (Ours) or the optional step (Ours+Optional). As a result, when using the new data from the optional step as an input of the in-processing approaches, the final classifier's accuracy and fairness are not sacrificed much compared to using data from the basic step, while minimally changing the data distribution.

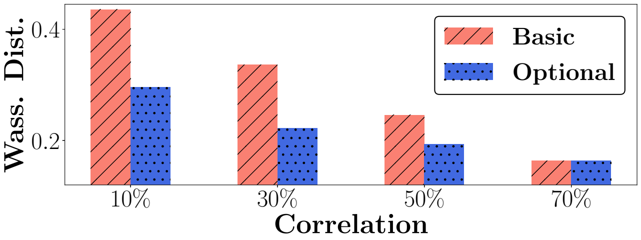

Figure 8. Wasserstein distances between the original training data and pre-processed data when our algorithm uses either the basic step or the optional step, while varying the target correlation to have 10% to 70% correlation of the training data.

Table 8. Performances on the synthetic test dataset. Other settings are identical to those in Table 2.

	Single (DP)	
Method	Acc.	Unfair.
Ours+FC	.849 ± .002	.034 ± .004
Ours+Optional+FC	.830 ± .003	.027 ± .005
Ours+AD	.814 ± .011	.017 ± .006
Ours+Optional+AD	.808 ± .007	.014 ± .006
Ours+FB	.836 ± .001	.003 ± .001
Ours+Optional+FB	.842 ± .002	.010 ± .004

B.11. Accuracy and Fairness – Neural Networks

Continuing from Sec. 6, we show the results on the synthetic dataset when using a multi-layer neural network for all algorithms. Here, the neural network consists of three linear layers connected by the rectified linear unit (ReLU) activation function. In Table 9, we observe consistent results to those in Table 2, where our pre-processing is indeed effective in improving the performances of in-processing-only baselines. This result supports that our approach can work for any ML model, including the multi-layer neural network and logistic regression.

B.12. Varying the Correlation of the Test Data: More Biases

Continuing from Sec. 6.2, we vary the correlation of the test data to have more biases than the training data. Figure 9 shows the accuracy and fairness performances of FB and Ours+FB, while varying the correlation constant c of the test data up to 150%. When the correlation increases more than 100% of that of the training data, the in-processing-only

Table 9. Performances on the synthetic test dataset w.r.t. a single metric (DP). We use a multi-layer neural network (NN) for all algorithms. Other experimental settings are identical to those in Table 2.

	Single (DP)	
Method	Acc.	Unfair.
FC	.865 ± .003	.101 ± .012
Ours+FC	.855 ± .003	.070 ± .006
AD	.866 ± .001	.100 ± .009
Ours+AD	.860 ± .003	.080 ± .010
FB	.795 ± .025	.065 ± .036
Ours+FB	.835 ± .006	.016 ± .017

baseline (FB) cannot improve fairness because the training data does not capture the bias level in the test data. Hence, the in-processing-only baseline cannot be used in applications that require high fairness. On the other hand, our pre-processing enables the in-processing approach to achieve the high fairness that it may need, with some accuracy degradation.

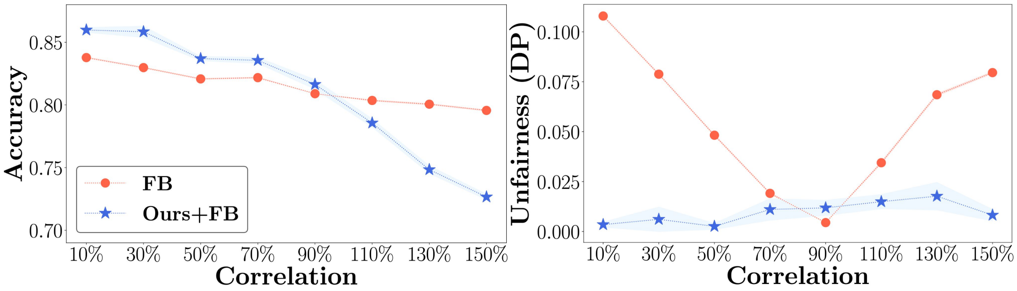

Figure 9. Performances of FB and Ours+FB on the synthetic data while varying the correlation of the test data to have 10% to 150% correlation of the training data.

B.13. Accuracy and Fairness – Other Test Data Construction: Real Data

Continuing from Sec. 6.3, we compare the algorithm performances when using a different method to construct the test dataset of the real-data experiment. Table 10 shows the accuracy and fairness performances when using the two income datasets collected in the 1990s (Kohavi, 1996) (i.e., AdultCensus dataset used in Sec. B.6) and 2010s (Ding et al., 2021) (i.e., ACSIncome dataset) for training and testing, respectively. Here, the 1990s and 2010s datasets have the different (y, z)-correlation values. In addition, the input feature x distribution also slightly shifted, as shown in Figure 10.

In this real-world scenario, we first estimate the shifted correlation constant c and range $[\alpha, \beta]$ by accessing some of the 2010s (deployment) data. Here, we show how the number of samples affects the shift range $[\alpha, \beta]$ using Theorem 11. Figure 11 shows the estimated correlation values when varying the number of samples used in the estimation. The blue dots indicate the estimated correlation constant c values that are inferred by the maximum likelihood estimation (MLE). The bars show the error ranges of each estimation with 90% probability, which can be calculated using Theorem 11 – see details in Sec. A.7. As discussed in Theorem 11, a larger number of samples yields a smaller range. In addition, although the range is large when the number of samples decreases (e.g., m=100), the point estimate value for c is close to the true value.

We then run our pre-processing by setting $[\alpha, \beta]$ with the error range of the estimation. We choose the values when the number of samples is 1,000 ($\simeq$ 0.07%), as it has a reasonable error range with a suitable number of samples. As a result, we observe consistent results in Table 10 where our framework improves the accuracy and fairness performances of the in-processing-only baselines.

B.14. Setting the Correlation Range to $c_{\text{test}} \pm x\%$

Continuing from Sec. 6.4, we evaluate our approach when $[\alpha, \beta] = [c_{\text{test}} - x\%, c_{\text{test}} + x\%]$. Figure 12 shows the accuracy and fairness performances of FB and Ours+FB, while varying the correlation constant c of the test data. Here, we set the *specified* correlation range in our algorithm to $[\alpha, \beta] = [c_{\text{test}} - x\%, c_{\text{test}} + x\%]$, where $x \in \{10, 50, 100\}$. When $x = 10\%$,

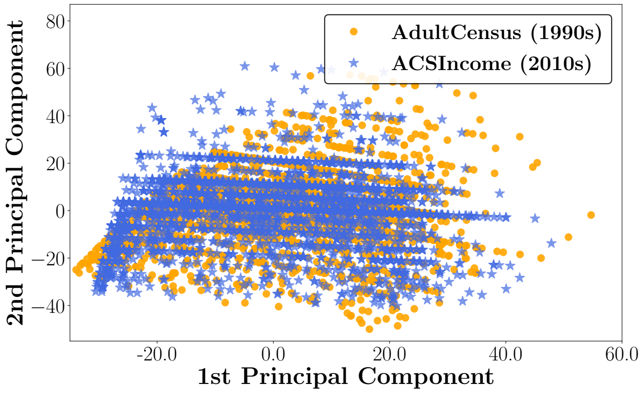

Figure 10. Principal component analysis (PCA) results on the input feature x of the two US income datasets: AdultCensus (Kohavi, 1996) and ACSIncome (Ding et al., 2021). Here, the distributions of the principal components on input feature x are different in the two datasets.

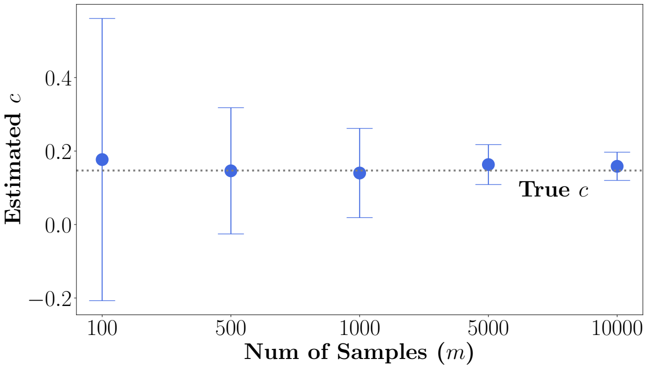

Figure 11. Estimating the correlation of the ACSIncome (deployment) data when varying the number of samples used in the estimation. The blue dots indicate the estimated correlation constant c values that are inferred by the maximum likelihood estimation (MLE). The bars indicate the error ranges of each estimation with 90% probability, which can be calculated using Theorem 11. The error ranges can be used as $[\alpha, \beta]$ in our pre-processing step. Here, the true c of ACSIncome is 0.147.

Table 10. Model performances on the ACSIncome test data (Ding et al., 2021) collected in the 2010s. The models are trained on the AdultCensus data (Kohavi, 1996) collected in the 1990s. Both datasets have labels that indicate each person's annual income. In our pre-processing, we use the estimated correlation range $[\alpha, \beta]$ that is inferred by using 1,000 ($\simeq$ 0.07%) samples of the ACSIncome data.

	Single (DP)	
Method	Acc.	Unfair.
FC	.646 ± .013	.104 ± .015
Ours+FC	.665 ± .010	.079 ± .011
AD	.637 ± .031	.137 ± .037
Ours+AD	.638 ± .023	.051 ± .048
FB	.643 ± .007	.128 ± .021
Ours+FB	.694 ± .002	.068 ± .006
FB on test dist. (upper bound)	.704 ± .007	.044 ± .035

we observe similar trends as in Figure 3, where our framework improves the accuracy and fairness performances of the in-processing-only baselines. When $x = 100\%$ (i.e., the worst-case setting of $[\alpha, \beta]$), the accuracy and fairness performances of our framework converge to the in-processing-only baselines. When $x = 50\%$, the performances of our framework are in between those of when $x = 10\%$ and $x = 100\%$. There are two takeaways: 1) our framework successfully boosts the in-processing-only baseline performances when the $[\alpha, \beta]$ range is reasonable, and 2) even if we do not have any information about the correlation shift, our framework performs at least as well as the in-processing-only baselines.

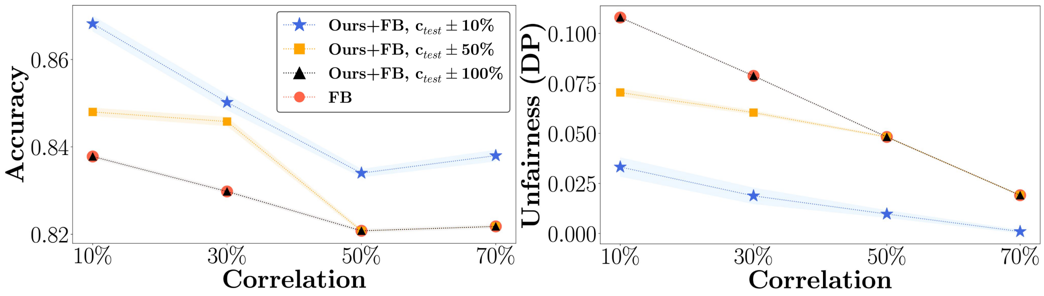

Figure 12. Performances of FB and Ours+FB on the synthetic data while varying the correlation of the test data c_{test} to have 10% to 70% correlation of the training data. We set the *specified* correlation range in our algorithm to $[\alpha, \beta] = [c_{\text{test}} - x\%, c_{\text{test}} + x\%]$, where $x \in \{10, 50, 100\}$.

B.15. Robust Against Misspecified Correlations: Real Data

Continuing from Sec. 6.4, we evaluate our approach when misspecifying the correlation in our algorithm on the real-world COMPAS dataset (Angwin et al., 2016). Same with Sec. 6.4, we set $\alpha = \beta = c_{\text{specified}}$, where $c_{\text{specified}} \neq c_{\text{test}}$. Figure 13 shows the accuracy and fairness performances of FB and Ours+FB when the true correlation of the test data is 60% of the training data's correlation. While achieving similar or higher accuracies compared to the in-processing-only baseline for the entire range of considered correlations, our framework improves the fairness of the in-processing-only baseline (FB) when the specified correlation is higher than 30%. This result shows that our approach is still beneficial when the estimation error is reasonable (say ±10%). We also note that when the specified correlation is far from the true value, our framework may not improve the performances of the baseline.

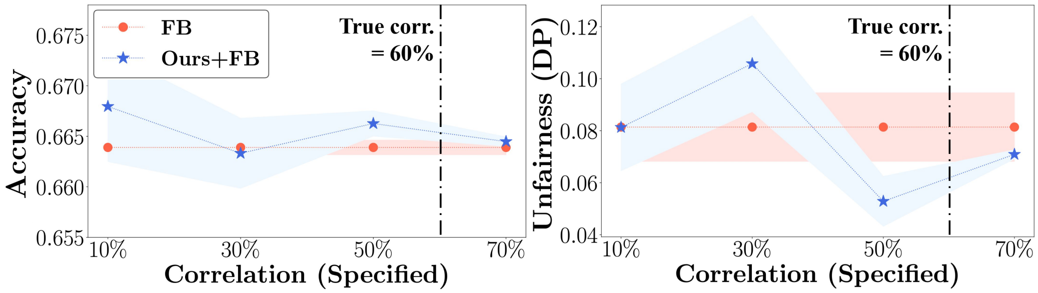

Figure 13. Performances of FB and Ours+FB on the COMPAS data while varying the *specified* correlation in our algorithm to have 10% to 70% where the true correlation of the test data is 60%.

B.16. Empirical Comparison with Giguere et al. (2022)

Continuing from Sec. 2, we add a new baseline called Shifty (Giguere et al., 2022), which focuses on the distribution shift most relevant to ours. Shifty first trains candidate models on the training data and selects only the models showing high fairness in the shifted deployment data. To give a favorable condition to Shifty, we assume that Shifty knows the exact test distribution. Table 11 shows the accuracy and fairness performances of the in-processing-only baseline FairBatch, Shifty, and our framework w.r.t. a single metric (DP) and multiple metrics (DP & EO) in the synthetic and COMPAS datasets used in Tables 2 and 3. As a result, both ours and Shifty improve the fairness of the in-processing-only baseline, but ours shows better fairness than Shifty while achieving similar or higher accuracy. The reason is that Shifty is selecting the final model among the candidates that were already trained on the original training data, whereas ours trains a new model on the improved (pre-processed) data.

B.17. Supporting Noisy Group Attributes

Continuing from Sec. 2, we evaluate the performance of our method on increasingly noisy data and observe that our method can potentially be extended to the noisy group scenario.

We perform a new experiment, where 10–50% of the group information in the training data is randomly flipped. Table 12

Table 11. Performances on the synthetic and COMPAS test datasets. We compare in-processing-only baseline FairBatch (Roh et al., 2021), Shifty (Giguere et al., 2022), and our framework. Other settings are identical to those in Table 2.

	Synthetic				COMPAS			
	Single (DP)		Multiple (DP & EO)		Single (DP)		Multiple (DP & EO)	
Method	Acc.	Unfair.	Acc.	Unfair.	Acc.	Unfair.	Acc.	Unfair.
FB (Roh et al., 2021)	.821 ± .000	.048 ± .000	.849 ± .001	.091 ± .005	.647 ± .001	.038 ± .013	.650 ± .002	.187 ± .019
FB + Shifty	**.838 ± .001**	.024 ± .008	.844 ± .003	.063 ± .008	.647 ± .001	.029 ± .001	.649 ± .002	.162 ± .025
Ours+FB	.836 ± .001	**.003 ± .001**	**.852 ± .004**	**.058 ± .001**	**.648 ± .004**	**.027 ± .001**	**.657 ± .004**	**.130 ± .014**

shows the accuracy and fairness performances of the in-processing-only baseline FairBatch (FB) and our framework (Ours+FB) w.r.t. demographic parity (DP). Here, FB cannot achieve high fairness performance (i.e., low DP disp.), as the group distribution in the training data is different from that in the test data. In comparison, our framework enables FB to achieve relatively high fairness at the expense of some accuracy degradation.

We note that these results are preliminary and only show that our method has some potential to support noisy group attributes, and we believe there is plenty of room for improvement. In particular, we believe our method can be further extended with other robust training methods like (Shen & Sanghavi, 2019).

Table 12. Performances on the synthetic test dataset when the group information in the training data is randomly flipped. We train the algorithms w.r.t. demographic parity (DP).

	10% flipping		30% flipping		50% flipping	
Method	Acc.	Unfair. (DP)	Acc.	Unfair. (DP)	Acc.	Unfair. (DP)
FB	0.818	0.087	0.864	0.165	0.883	0.250
Ours + FB	0.808	0.069	0.823	0.101	0.838	0.142

B.18. Supporting Poisoning Attack Scenario

Continuing from Sec. 2, we evaluate the performance of our method on poisoned data and observe that our method can potentially be extended to the poisoning attack scenario.

We perform a new experiment using one of the poisoning attack methods for fair training (Roh et al., 2020), where 10% of the training labels of a specific group are flipped so as to maximize the accuracy degradation. Table 13 shows the accuracy and fairness performances of the in-processing-only baseline FairBatch (FB) and our framework w.r.t. demographic parity (DP) on the poisoned data; we also report FB's performance on the clean data, which can be considered as the upper-bound performance. Here, when achieving similar fairness, our framework improves the accuracy of FB on the poisoned data. We suspect that, while our pre-processing reduces the correlation shift between training and deployment data, the effect of the poisoning attack is also mitigated.

We note that these results are preliminary and only show that our method has some potential to support the poisoning attack scenario, and we believe there is plenty of room for improvement.

Table 13. Performances on the synthetic test dataset when 10% of the training labels are poisoned (Roh et al., 2020). We train the algorithms w.r.t. demographic parity (DP).

Method	Acc.	Unfair. (DP)
FB	0.699	0.026
Ours + FB	0.721	0.024
FB on the clean data (upper bound)	0.773	0.023

C. Appendix – More Related Work

C.1. Traditional Model Fairness

As model fairness becomes essential for Trustworthy AI, various fairness definitions have been proposed to reflect legal and social requirements (Narayanan, 2018). Among the definitions, we focus on group fairness, which aims to not discriminate specific groups. There are three prominent group fairness metrics: demographic parity (Feldman et al., 2015), equalized odds (Hardt et al., 2016), and predictive parity (Berk et al., 2021). To support the fairness metrics, various fairness techniques have been proposed, which can be categorized into three prominent approaches: (1) pre-processing approaches (Kamiran & Calders, 2011; Zemel et al., 2013; Feldman et al., 2015; du Pin Calmon et al., 2017; Choi et al., 2020; Jiang & Nachum, 2020), which debias, reweight, or generate training data, (2) in-processing approaches (Zafar et al., 2017a;b; Agarwal et al., 2018; Zhang et al., 2018; Cotter et al., 2019; Roh et al., 2020; 2021), which modify model training itself for fairness, and (3) post-processing approaches (Kamiran et al., 2012; Hardt et al., 2016; Pleiss et al., 2017; Chzhen et al., 2019), which manipulate only the model outputs without changing the training inside. Among the three categories, in-processing approaches are widely used for unfairness mitigation, but most of them assume that the training and deployment data distributions are the same.

Another line of research is to support multiple fairness metrics (Thomas et al., 2019; Zhao et al., 2020). Thomas et al. (2019) proposes a fairness testing framework that can support multiple metrics. Zhao et al. (2020) shows that EO can be achieved while preserving the original DP. In comparison, we analyze when a model can achieve both ε-DP and ε-EO.

Beyond group fairness, there are other noteworthy fairness definitions including individual fairness (Dwork et al., 2012b), which aims to give similar predictions to similar individuals, and causality-based fairness (Kilbertus et al., 2017; Kusner et al., 2017; Zhang & Bareinboim, 2018), which aims to improve fairness by understanding the causal relationship between attributes. Extending our analysis and framework to these definitions is an interesting future work.

C.2. Fairness under Data Distribution Shifts

Continuing from Sec. 2, we further compare our work with the previous studies on data distribution shifts. The following paragraphs contain detailed discussions of the two categories of distribution shifts: general distribution shifts and fairness-specific shifts.

Among the general distribution shifts (i.e., covariate, label, and concept shifts in Table 1), the concept shift is the most relevant definition to the correlation shift. As shown in Table 1 of Sec. 2, the correlation and concept shifts focus on $\Pr(\mathrm{z}|\mathrm{y})$ and $\Pr(\mathrm{y}|\mathrm{x})$, respectively. As the concept shifts consider the distribution changes of the label (y) and input feature (x), this type of shift can implicitly describe the correlation shifts when the input feature contains the group attribute (z). However, a unique characteristic of the correlation shifts is to explicitly capture the bias changes between y and z, where z is especially relevant to fair training. Thus, the notion of correlation shifts enables us to analyze the behavior of fair training when the bias changes.

In addition, the key difference between the correlation shift and the other fairness-specific shifts (subpopulation and demographic shifts) is that the other shifts are defined under specific assumptions on the data distribution, which are not required in our correlation shift definition. Here are the assumptions in the subpopulation and demographic shifts:

- The subpopulation shift (Maity et al., 2021) assumes that the loss expectations w.r.t. input feature X and label Y of training and deployment distributions are the same (i.e., $E_{\text{train}}[(h(\mathrm{x}), \mathrm{y})|\mathrm{z} = z] = E_{\text{test}}[(h(\mathrm{x}), \mathrm{y})|\mathrm{z} = z]$, where E indicates the expectation). Thus, as discussed in the related work section, a major example of subpopulation shifts is when a specific group has fewer positively-labeled examples during training time compared to deployment time, while the distributions of the x and y attributes remain the same.
- Similarly, the demographic shift (Giguere et al., 2022) assumes that the joint probabilities of x and y on the training and deployment distributions are identical (i.e., $\Pr_{\text{train}}(\mathrm{x} = x, \mathrm{y} = y|\mathrm{z} = z) = \Pr_{\text{test}}(\mathrm{x} = x, \mathrm{y} = y|\mathrm{z} = z)$). This assumption is similar to the assumption of the subpopulation shift, but the difference is whether the loss values of the model are explicitly considered or not.

The above assumptions are used in the theoretical analyses of the previous works (Maity et al., 2021; Giguere et al., 2022). In comparison, our theoretical analyses in Sec. 4 of using correlation shifts are not limited by any additional assumptions, and we show that the achievable performances of fair training are determined by the (y, z)-correlation.

C.3. Connection to Causality-based Fairness

Continuing from Sec. 2, we discuss how our work can be connected with causality-based fairness. To this end, we give a concrete example to show the correlation shifts when the sensitive group attribute itself is a confounder, one of the important roles of attributes in causality-based fairness.

We consider a car insurance example, where the goal is to predict future accident rates from the past driving record: suppose that input feature x is the past driving record, label y is the future accident rate, and sensitive attribute z is the race.

Then, the structural causality graph between them would look like: z → (x, y) and x → y, i.e., z being a confounder (Figure 14). The arrows from z to x and y could be due to (a) different socialization and driving behavior formation process that partially depends on racial identity and (b) another unobserved mediator such as where they live in, etc. In this example where z is a confounder, it is clear that z and y are correlated.

We expect this confounding effect to change over time. This will make the sensitive attribute z either a stronger or weaker confounding factor, which impacts the correlation between y and z.

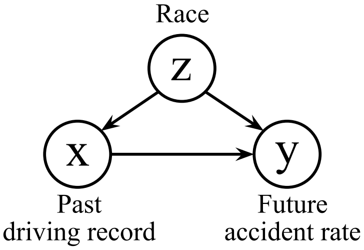

Figure 14. A causality graph for the car insurance example in Sec. C.3.