ProxiMix: Enhancing Fairness with Proximity Samples in Subgroups

Jingyu Hu[1], Jun Hong[2], Mengnan Du[3] and Weiru Liu[1]

[1] *University of Bristol, Beacon House, Queens Rd, Bristol, UK*
[2] *University of the West of England, Coldharbour Ln, Stoke Gifford, Bristol, UK*
[3] *New Jersey Institute of Technology, 323 Dr Martin Luther King Jr Blvd, Newark, USA*

Abstract

Many bias mitigation methods have been developed for addressing fairness issues in machine learning. We have found that using linear mixup alone, a data augmentation technique, for bias mitigation, can still retain biases present in dataset labels. Research presented in this paper aims to address this issue by proposing a novel pre-processing strategy in which both an existing mixup method and our new bias mitigation algorithm can be utilized to improve the generation of labels of augmented samples, hence being proximity aware. Specifically, we propose ProxiMix which keeps both pairwise and proximity relationships for fairer data augmentation. We have conducted thorough experiments with three datasets, three ML models, and different hyperparameters settings. Our experimental results show the effectiveness of ProxiMix from both fairness of predictions and fairness of recourse perspectives.

Keywords

Group Fairness, Bias Mitigations, Mixup, Data Augmentation

1. Introduction

Machine learning has been used as an effective decision-making aid in more and more fields. However, concerns have been raised about the potential unjust or biased predictions by models, which can harm individual and societal values [1]. Most popular ML models are considered black-box, making it difficult to understand their internal decision-making processes. To address this issue, there is a growing focus on achieving fair and trustworthy ML by developing explainable and interpretable techniques [2, 3, 4], auditing models to detect hidden bias [5, 6], as well as mitigating the spotted bias [7, 8].

Among various mitigation methods, mixup-based methods have attracted increasing attention from the community. Mixup [9] is a data augmentation method that linearly interpolates two samples to generate synthesized data for model generalization. Some research investigates the combination of mixup with subgroup analysis for addressing fairness issues in datasets, applying it as an augmentation strategy in preprocessing [10] or a loss regularization term in training [11]. However, one limitation of Mixup is that if the original labels in the dataset are biased, this bias can persist in the labels of mixed samples. The generated data labels can introduce additional bias to models.

AEQUITAS 2024: Workshop on Fairness and Bias in AI | co-located with ECAI 2024, Santiago de Compostela, Spain

ym21669@bristol.ac.uk (J. Hu); jun.hong@uwe.ac.uk (J. Hong); mengnan.du@njit.edu (M. Du); weiru.liu@bristol.ac.uk (W. Liu)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

To bridge the research gap, in this work, we propose ProxiMix to address the issue of biased labels in pre-processing for bias mitigation. Motivated by the relabeling the discrimination method [12], which assigns labels to instances based on their K-nearest neighbors to ensure that similar individuals have similar labels, our proposed approach adds proximity samples for re-auditing mixed labels to mitigate potential bias in mixup. The intuition is that compared with focusing on pairwise labels, considering the labels of proximity samples as latent label relationships can reduce the probability of generating biased labels. We have conducted experiments to compare the existing pairwise mixup with the proposed proximity-aware mixup on multiple models and datasets. The results show that our ProxiMix achieves higher fairness, particularly when the original labels in the dataset are highly biased.

Our main contributions can be summarised as follows: (1) We propose a new bias mitigation algorithm to address the label bias retainment issue in the current mixup method; (2) Subgroup preference analysis: we explore how different subgroups perform during the sampling process; (3) Trade-off analysis: we explore the tradeoff between using our proximity-based strategy and the traditional mixup; (4) Validation: we validate the effectiveness of our method using prediction-based metrics and the cost of counterfactual explanations from an XAI perspective.

2. Related Work

The fairness problem can be divided into individual and group levels. Individual fairness measures the bias by checking if similar predictions can be made for similar individuals. Group fairness compares the treatments of fairness in unprivileged and privileged groups. Fairness is achieved when the treatments are equal between groups. Prediction-based fairness and recourse-based fairness are two perspectives for evaluating model fairness. In this paper, we focus on group fairness in machine learning.

Fairness of Prediction Outcomes Most fairness metrics are based on predicted outcomes. Demographic Parity (DP) [13] based metrics use predicted outcomes to assess whether different demographic groups are equally favored by the model. It aims at having equal proportions of positive outcomes across subgroups. The DP difference between groups is called Statistical Parity Difference (SP), and DP ratio between groups is called Disparate Impact (DI). In addition to depending on predictions only, there are some fairness metrics [14] that consider both predicted and actual outcomes. Equality of Opportunity (EO) measures the True Positive Rate (TPR) of subgroups. Equalized odds (Eodds) compares both True Positive Rate (TPR) and False Positive Rate (FPR) of each groups.

Fairness of Recourse Another recent research trend is to apply Explainable Artificial Intelligence (XAI) methods to address fairness issues. One of the key components in this area is counterfactual explanation (CE), sometimes also called as algorithm recourse. CE focuses on explaining why a particular outcome occurred instead of an alternative plausible outcome. [15, 16]. Recourse refers to identifying the closest counterfactuals that could alter the result with minimal feature changes. Several algorithms have been developed to generate such counterfactual explanations for machine learning models [17, 18, 19]. The concept of fairness of recourse are proposed by [20] and defined as the disparity of the mean cost to achieve the desirable recourse among the unprivileged subgroups. [6, 21] propose metrics based on the cost of counterfactual explanation to measure fairness performance across subgroups. Predictive Counterfactual Fairness (PreCoF) [22] utilises CEs to detect the underlying patterns for the discrimination in the model.

Bias Mitigation Methods Bias mitigation methods can be categorized into three stages: pre-processing, in-processing, and post-processing [8, 23]. Pre-processing mitigations aim to reduce bias by modifying and creating a fairer training dataset [24, 25, 26]. In-processing mitigation occurs during training by adding regularization terms and constraints to models [11, 27]. Mitigations in the post-processing stage like calibration are applied after a model has been successfully trained [21, 28]. Both pre-processing and post-processing-based methods are model-agnostic as they occur before and after the model training.

Over-sampling in the pre-processing stage refers to changing the distribution of the training dataset by adding more samples. Duplicating instances of the unprivileged group is one straightforward strategy [29, 30]. [31, 32] generate synthetic samples around the unprivileged group to mitigate bias. MixSG [10] takes both the privileged and unprivileged groups into consideration when synthesizing new data using mixup, but the potential bias in generated labels has not been discussed yet.

3. Preliminaries and Problem Statement

Notations Given the dataset $D = \{(X, Y, Z)\}_{i=1}^{N}$ with N samples, where X is a set of features space, and each feature x in X has a set of values in d_{x_i}, label $Y \in \mathcal{Y} := \{0, 1\}$, and a sensitive attribute $Z \in \mathcal{Z} := \{0, 1\}$. The dataset is divided into training set D_{train} and test set D_{test}. We use D_{train} to fit a classifier model $f : X \to Y$ and D_{test} to assess the model's prediction and fairness performance. Fairness is measured by the model's performance on the difference between subgroups identified by Z. We define the unprivileged/minority group when Z=0, and Z=1 is the privileged/majority group.

Mixup Strategy in Fairness Mixup [9] is a data augmentation technique that involves blending pairs of samples to create new synthetic training examples. The premise of mixup is that linear combinations of features will result in the same linear combinations of target labels. Thus, mixup applies stochastic linear combinations to samples $S_0(x_0, y_0)$, $S_1(x_1, y_1)$ to generate a new sample $\tilde{S}(\tilde{x}, \tilde{y})$, with random parameters λ drawn from a Beta distribution.

$$\tilde{x} = \lambda * x_0 + (1 - \lambda) * x_1, \quad \text{where } x_0, x_1 \text{ are input vectors} \tag{1}$$

$$\tilde{y} = \lambda * y_0 + (1 - \lambda) * y_1, \quad \text{where } y_0, y_1 \text{ are target labels} \tag{2}$$

To address fairness concerns, previous research has explored the practice of sampling S_0 and S_1 from different subgroups, applying this step to both pre-processing stage like mixSG [10] and in-processing stage like fairMixup [11] as bias mitigation methods.

Bias Persist After Mixup The premise of mixup lies in the linear relationship between features and labels. The challenge here is if the original labels in the dataset are biased, the labels of mixed samples can retain this bias. The newly generated biased samples can impact the fairness of the trained model.

Table 1
Simplified Sample Examples of Individual Income Prediction

Sample	Gender	Capital Gain	Capital Loss	Occupation	Age	Income
M1	Male	8200	0	Officer	34	**>50K (1)**
M2	Male	7800	-100	Officer	35	**>50K (1)**
F1	Female	7800	-200	Sales	23	**<=50K (0)**
F2	Female	8200	0	Officer	34	**<=50K (0)**

Here is a toy example. Table 1 presents simplified instances of individual income predictions by the ML model. The predicted label Y indicates whether an individual is high-income ($> 50K$) or low-income ($\leq 50K$). The features X used for prediction include *Age*, *Occupation*, *Gender*, *CapitalGain*, and *CapitalLoss*. *Gende*r is considered as the sensitive attribute Z, dividing the data into subgroups. Here, we consider the female subgroup as unprivileged.

The table shows individual features of male samples ($M1$ and $M2$) and the female sample ($F2$) are remarkably similar (Officer with similar *CapitalGain* and *Age*), but with different income labels. This shows initial bias that female and male groups are treated unequally.

We follow the mixSG method to select one sample from one subgroup and another from the other subgroup to generate $(\tilde{x}, \tilde{y})$. Assume we have chosen one sample $F2$ from the female subgroup, $F2$ will be randomly paired with either $M1$ or $M2$ from the male subgroup. If the mixture ratio λ of the female sample $F2$ is over 50%, we say the mixed sample S_{FM} is female. Otherwise, S_{FM} is male.

When the random $\lambda = 0.8$, S_{FM} will be a female sample. And the label y_{FM} of the mixed female sample will primarily depend on the label from female $F2$, meaning that both combinations of $F2$ with $M1$ or $M2$ will have a high probability of low income ($\leq 50K$). Though individual features of high-income men (M1 and M2) and low-income women (F2) are remarkably similar (Officer with similar capital gain and age), mixed label still indicates a tendency toward lower incomes for female. If $\lambda = 0.2$, the mixed sample will be most depend on the label from the male sample and the generated sample becomes male with high income. The labels of mixed samples are heavily influenced by gender. Considering the initial bias in the dataset, new samples generated by mixup can deepen gender bias against unprivileged groups, causing the model to be more likely to predict male samples as high-income and female samples as low-income under similar conditions.

4. Methodology and Experiment Design

To address the issue of possible biased label for mix-up, we proposed a method called ProxiMix for improvments. It synthesizes $D_{\text{new}} = \{(X, Y, Z)\}_{j=1}^{K}$ from D_{train} with the consideration of both pairwise and proximity samples, to reduce dataset bias. Fitting the model with fairer dataset $D'_{train} = D_{train} \cup D_{\text{new}}$ is expected to improve its fairness performance.

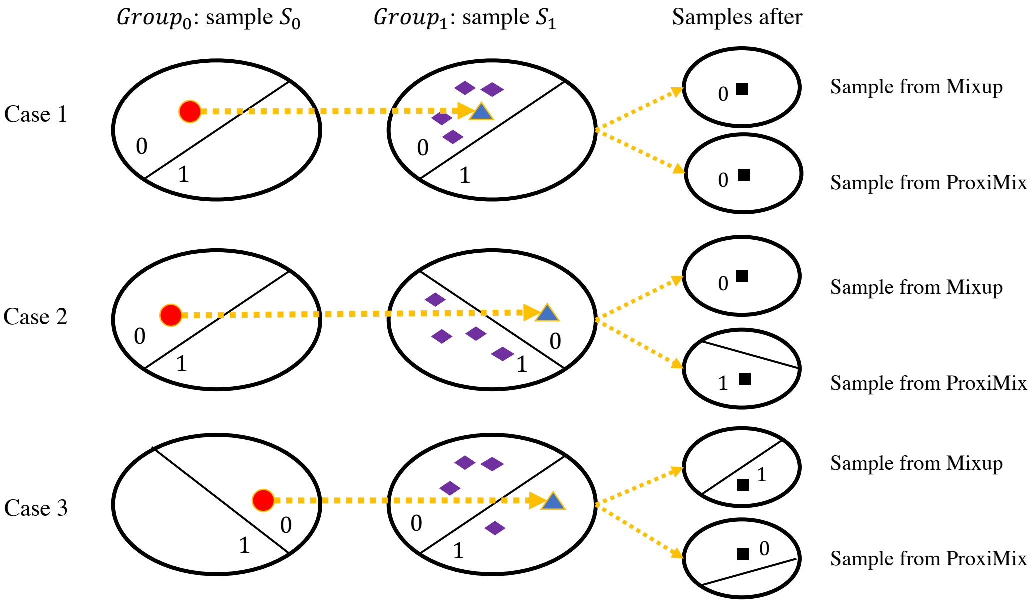

Figure 1: A comparison between proximity-based mixup and linear mixup. The red circle represents S_0, the blue triangle represents S_1, the purple diamonds represent the proximity set D_p, and the black square indicates the samples after mixing up. Here, we consider the particular case for case three where labels of most proximity samples are opposite to S_1. The mixing ratio is set to 0.5.

4.1. ProxiMix Algorithm

The Importance of Proximity Awareness Given a sample S_0 from group $D_{\text{train}}(Z = 0)$, and another sample S_1 from $D_{\text{train}}(Z = 1)$, the proximity samples set of S_1 is defined as $D_p = \{S_{p_0}, S_{p_1}, ..., S_{p_m}\}$. The label value of each sample can be either 0 or 1. We illustrate three cases when mixing up two samples S_0 and S_1: **(1) Case 1:** Labels of S_0, S_1 and all of their proximity samples are the same. **(2) Case 2:** Labels of S_0 and S_1 are the same, but there exist different labels among proximity samples D_p. **(3) Case 3:** Labels of S_0 and S_1 are different. Figure 1 presents these three cases.

In Case 1, linear mixing and proximity yield the same results because there are no impurities between the two samples. In Case 2, both samples S_0 and S_1 have the same label. This implies that direct mixing will result in all labels becoming 0 regardless of the mixing ratio. This approach ignores the samples from 1 in between and can potentially introduce bias when predicting subgroups with the 1 label. In Case 3, the mixed label depends on the mixing rate when using mixup directly. Specifically, the mixed label becomes 1 when the mixing rate exceeds 0.5. However, we can see in the example that the majority of the proximity samples D_p between 0 and 1 belong to 0. It suggests that the probability of being classified as 0 should be higher. Considering the proportion of proximity labels can enhance the probability of being classified as 0.

ProxiMix Algorithm Design ProxiMix consists of two parts: we first introduce proximity-based mixed label Y_{sim} and then combine Y_{sim} with Y_λ from the existing mixup [10] using d-adjusted balancing degree.

As discussed above, the current mixup approach does not account for potential biases in labels. Our proposal aims to determine the mixed label by considering the proportions of labels in proximity samples. Specifically, when mixing two samples, S_0 and S_1, we calculate their Euclidean distance with their one-hot encoded features[1], denoted as $P_{dis} = ||x_0 - x_1||$, to measure their proximity. Then, we select all the samples that are within the P_{dis} distance from S_0 to form a potential proximity samples set ProxiSet. The final mixed label for S_0 and S_1 is assigned based on the label with the larger proportion within the $S_0 \cup \mathit{ProxiSet}$.

Let's look back at the toy example: $\mathit{NewSet} = \{F2, M1, M2\}$ when we want to mix $F2$ with either $M1$ or $M2$. Two-thirds of the labels in the *NewSet* is high income, so that the proximity-based mixed Y_{sim} is high income.

We combine our proximity-based Y_{Sim} with Y_λ from the current mixup to form the new definition of mixed $\tilde{Y}$, achieved by calculating $d * Y_\lambda + (1-d) * Y_{Sim}$, where d is a balancing degree between 0 and 1. The algorithm pseudocode is described in Algorithm 1.

Algorithm 1 ProxiMix Algorithm

```
Input $S_0(x_0, y_0, z_0) \sim D_{\text{train}}(Z=0), S_1(x_1, y_1, z_1) \sim D_{\text{train}}(Z=1)$
procedure ProxiMix($S_0, S_1, D_{\text{train}}, d$)
    procedure Proximity-Based-Mixed($S_0, S_1, D_{\text{train}}$)
        $\mathit{ProxiSet} = []$ .
        $P_{dis} = ||x_0 - x_1||$
        for each sample $S_i(x_i, y_i, z_i)$ in $D_{\text{train}}(Z=1)$ do
            $P_{cur} = ||x_i - x_0||$
            if $P_{cur} \leq P_{dis}$ then
                Add $S_i$ to $\mathit{ProxiSet}$.
            end if
        end for
        $\mathit{NewSet} = S_0 \cup \mathit{ProxiSet}$
        $Y_{Sim} = \mathit{Label\_Counts}(Y \in \mathit{NewSet}) / \mathit{size}(\mathit{NewSet})$
    end procedure
    procedure Lambda-Based-Mix($S_0, S_1$)
        $\lambda = \text{Beta}(\alpha, \alpha)$
        $Y_\lambda = \lambda * y_0 + (1-\lambda) * y_1$
    end procedure
    $\tilde{Y} = d * Y_\lambda + (1-d) * Y_{Sim}, d \in [0,1]$
    Return $\tilde{Y}$
end procedure
```

[1] scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

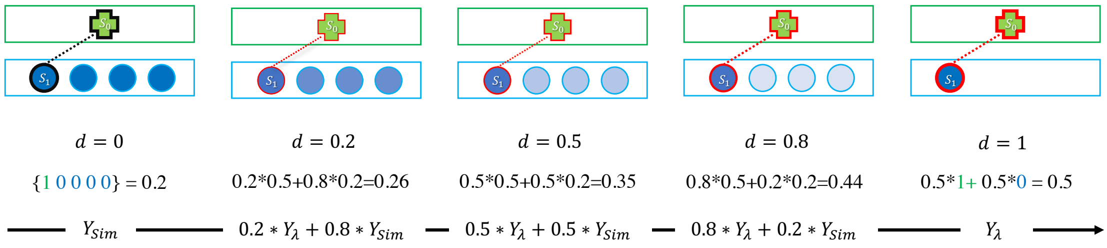

Figure 2: An Example of ProxiMix with Balancing $d = [1, 0.8, 0.5, 0.2, 0]$, $\lambda = 0.5$

Fig. 2 shows an example of how ProxiMix works. Samples are categorized into two subgroups, green and blue, based on their colors. The shape of each sample represents its label: circles for label 0, and plus-signs for label 1. Specifically, the green circle (S_0) and the blue plus-sign (S_1) are two samples selected for ProxiMix. The new label of the mixed samples changes with different values of the balancing parameter *d*. The varying shades of blue samples represent the impact degree of Y_{sim}, while the thickness of the red lines between S_0 and S_1 represents the strength of Y_λ. The black line indicates no consideration for Y_λ. For $d = 1$, it employs the original mixup Y_λ; for $d = 0$, it utilizes our proximity-based Y_{Sim} exclusively; and it combines the two for values in between. We will discuss how different *d* impact the model performance in Section 5.2.

Accelerating Calculation of ProxiMix in Practice Our core idea is to introduce proximity samples' label set *ProxiSet* as a reference when performing label mixup. To enhance computational efficiency, we find *ProxiSet* first in practice. Our implementation is as follows: (1) Given a randomly selected sample S_0 from $D_{train}(Z = z)$, we first find its *ProxiSet* from $D_{train}(Z = \neg z)$. *ProxiSet* contains K samples that are proximal to S_0; (2) Then, we treat each sample in *ProxiSet* as S_1 and sequentially mix it with S_0, following the 'furthest-first' rule. It means the mixing begins with the sample in *ProxiSet* that is furthest from S_0. After each mix, we remove the used sample from *ProxiSet*; (3) Repeat this process K/M times until the desired M new samples are generated. The generated samples are merged to D_{train} as training samples for classification model.

4.2. Experiment Setting

Fig. 3 presents the overall workflow of our experiment. The parameter balancing degree *d* in our mixup algorithm is tested with values ranging from 0 to 1, in increments of 0.1. The proximity samples for each round are set to 25. we consider proximity when there are at least 5 neighbors to ensure credibility. The mixing ratio λ is randomly generated from the Beta(1,1) distribution.

Datasets The experiment is conducted on three datasets for classification problems: (1) Adult income dataset [33]: predicting whether a person's annual income exceeds 50K (high/low-income); (2) Law school dataset [34]: predicting whether a person's in law school will fail/pass the exam; (3) Credit default dataset [35]: predicting whether a person's credit payment will be on-time/overdue.

Models Three models including logistic regression (LogReg), decision trees (DT) and multi-

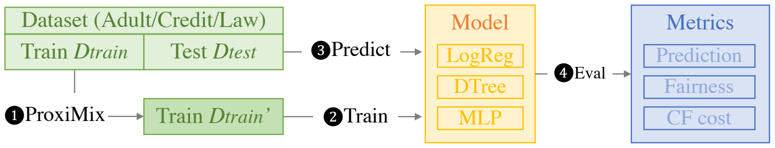

Figure 3: The Experiment Workflow

layer perceptron (MLP) are tested. All implementations are based on scikit-learn[2]. The maximum depth is 7 in the decision tree. We use a three-layer MLP with 128 neurons in the ith hidden layer, 'rule' as the activation function, and a maximum of 1500 iterations. The random seed is set to 42 for reproducible results.

Metrics Prediction performance metrics are based on True Positive (TP), False Positive (FP),False Negative (FN),True Negative (TN) in the confusion matrix. The equations of Precision, Recall, and F1-score are as follows. Recall is also called True Positive Rate.

$$Precision = \frac{TP}{TP+FP}; Recall = \frac{TP}{TP+FN}; F1-score = \frac{2\cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

The following equations calculate the True Positive Rate (TPR) and False Positive Rate (FPR) in the subgroup where the sensitive attribute $Z = z$.

$$TPR_z = \frac{TP}{TP+FN}; FPR_z = \frac{FP}{FP+TN} where(D(Z=z)) \tag{4}$$

Fairness performance is evaluated by Demographic Parity (DP) and Equalized Odds (Eodds) between subgroups[3]. We define the label for the unprivileged group as z_0 and for the privileged group as z_1. Their difference and ratio between DP_{z_0} and DP_{z_1} are noted as $\Delta DP, DP\%$.

$$DP_{z_0} = \mathrm{P}\left(f(x)=1 \mid Z=z_0\right); DP_{Z_1} = \mathrm{P}\left(f(x)=1 \mid Z=z_1\right) \tag{5}$$

$$\Delta DP = DP_{z_1} - DP_{z_0}; DP\% = \frac{DP_{z_0}}{DP_{z_1}} \tag{6}$$

Eodds difference $\Delta Eodds$ is defined as the greater one of TPR and FPR across subgroups, and eodds ratio $Eodds\%$ is the smaller metrics of TPR and FPR ratio.

$$\Delta Eodds = Max(TPR_{z_1} - TPR_{z_0}, FPR_{z_1} - FPR_{z_0}) \tag{7}$$

$$Eodds\% = Min\left(\frac{TPR_{z_0}}{TPR_{z_1}}, \frac{FPR_{z_0}}{FPR_{z_1}}\right) \tag{8}$$

Counterfactual explanation cost [21] is also assessed across subgroups to examine fairness from an XAI perspective. Given a classification model f, the counterfactual explanation of a sample $d_s \in D$ can be denoted as $d_{cf} = CF(d_s, f)$. The cost of a counterfactual explanation

[2] scikit-learn.org/
[3] fairlearn.org/

Table 2
Four different subgroup combinations for sampling in Adult, Law, Credit datasets

	C1 (z,y)	C2 (z,y)	C3 (z,y)	C4 (z,y)
Adult	female, low-income	female, high-income	male, low-income	male, high-income
Law	female, failed	female, passed	male, failed	male, passed
Credit	female, on-time	female, overdue	male, on-time	male, overdue
	C1'($\bar{y}$)	C1'($\bar{y}$)	C3'($\bar{y}$)	C3'($\bar{y}$)
Adult	male group	male group	female group	female group
Law	male group	male group	female group	female group
Credit	male group	male group	female group	female group

is the distance between d_s and d_{cf}. In this way, we can compute the counterfactual cost for each sample in dataset D. The average costs of counterfactuals across different groups can be considered as a measure of fairness: with the cost gap between groups (e.g., females and males) increasing, the model's unfairness also grows. Our evaluation follows the implementation of counterfactual explanation cost package[4], and specifically, we opt counterfactual explanations cost without constraints as metrics.

5. Results

In section 5.1, we fix the balancing degree d of ProxiMix and examined the impact of different sampling modes for subgroups on the outcomes. In section 5.2, we fix the sampling mode and explore the impact of different balancing degrees d on the results. To ensure the consistency of findings, Section 5.3 assesses the effectiveness of ProxiMix from the counterfactual cost perspective.

5.1. Sampling Mode Preferences in ProxiMix with Fixed Balancing Degree

ProxiMix is built on the mixup concept, which involves continuously selecting and mixing two samples to generate new data. To identify which combinations of samples had a more positive impact on the model's performance, we divide the dataset into different subgroups and sample from them.

There are four subgroups with considerations on both labels and values of a single sensitive feature in Z. The first sample selected from each group $D_{train}(Y = y, Z = z)$ is notated as $C1, C3$, the second sample selected from the subgroup $D_{train}(Y = \bar{y})$ which has the opposite sensitive label is notated as $C1'$,$C2'$,$C3'$,$C4'$, respectively. In Table 2, $C1$ is sampled from <female, low-income> subgroup in the Adult dataset, from the <female, failed> subgroup in Law dataset, and from the <female, on-time> subgroup from the Credit dataset respectively. $C1'$ refers to the sample selected from the male group in the adult, law and credit datasets. All sampling combinations are listed in Table 2. We denote the sample derived from ProxiMix with different sampling combination modes as $C_i \odot C_j$, where $C_i \in \{C1, C2, C3, C4\}, C_j \in \{C1', C3'\}$.

Table 3 presents models performance using ProxiMix under four sampling combinations $C_i \odot C_j$ and compares it with performance without any augmentation (baseline).

[4] github.com/HammerLabML/ModelAgnosticGroupFairnessCounterfactuals/

Table 3
Prediction (F1 score) and Fairness (DP%) Performance Comparison across Different Sampling Subgroups in Adult and Law School Datasets (d=0.5, LogReg stands for logistic regression, and DT represents the decision tree.

Dataset	Adult Income				Law School			
Model	LogReg		DT		LogReg		DT	
	F1 Score	DP%	F1 Score	DP%	F1 Score	DP%	F1 Score	DP%
Baseline	0.7791	0.2892	0.7782	0.2847	0.6408	0.9856	0.6146	**0.9935**
$C1 \odot C1'$	0.7758	0.2439	0.7749	0.2792	0.6680	0.9261	0.6336	0.9831
$C2 \odot C1'$	0.7820	**0.4730**	0.7729	**0.3698**	0.6279	**0.9948**	0.6428	0.9925
$C3 \odot C3'$	0.7705	0.2625	0.7721	0.2971	0.6696	0.9619	0.6309	0.9837
$C4 \odot C3'$	0.7884	0.2889	0.7780	0.2988	0.6251	0.9840	0.6369	0.9921

In the adult dataset, we found that different subgroup sampling combinations have different impacts on ProxiMix performance. The $C2 \odot C1'$ (augmenting high-income female) significantly improves the fairness performance of both decision tree and logistic regression models. In contrast, $C1 \odot C1'$(augmenting low-income female) degrades the fairness of both models, suggesting it introduces extra bias to the underrepresented group. This implies that focusing on underrepresented labels in the unprivileged group when generating samples (such as high income) can greatly improve fairness performance.

In the Law dataset, nearly all mixup methods enhance model prediction performance, but only marginally improve fairness. This is because fairness performance DP% without any augmentation already exceeds 90%, indicating the minimal bias in the model. Therefore, the improvement potential is limited.

Overall, ProxiMix enhances fairness when a model displays significant bias. Also, the choice of the subgroup for sampling during mixup is important: some enhance fairness, while others can even worsen it.

5.2. The Impact of Balancing Degree in ProxiMix

In the above section we have discussed the different sampling strategies with a balanced mixup ($d = 0.5$). This section explores how different d in ProxiMix can impact model performance. Here, we fix strategy $C_i \odot C_j$ while changing balance degree d.

Fig. 4 illustrates the impact of data augmentation on model fairness in the Credit dataset, under $C1 \odot C1'$ and $C3 \odot C3'$ strategies, with different degree d. The trend shows most combinations positively affect a model fairness, with an optimal d that maximizes fairness improvements. The best performance is achieved at *d=0.7* for the $C1 \odot C1'$ strategy, while for $C3 \odot C3'$, the optimal performance is reached at *d=0.2*.

Similar patterns are observed in the adult dataset (Fig.5): the impact of different values of d on the model also shows a trend. Specifically, data generated with the $C2 \odot C2'$ strategy shows the better improvement in model fairness when d ranges from *0.2* to *0.5*.

We noticed the best fairness DP% and Eodds% occurs at $d = 1$ under $C4 \odot C3'$. However, both TPR of female and male groups decline when d exceeds *0.5*. [36] mentions a similar scenario and suggests to consider both relative and absolute values in fairness performance. To have a further investigation of their performance in absolute values, Table 4 presents the model's

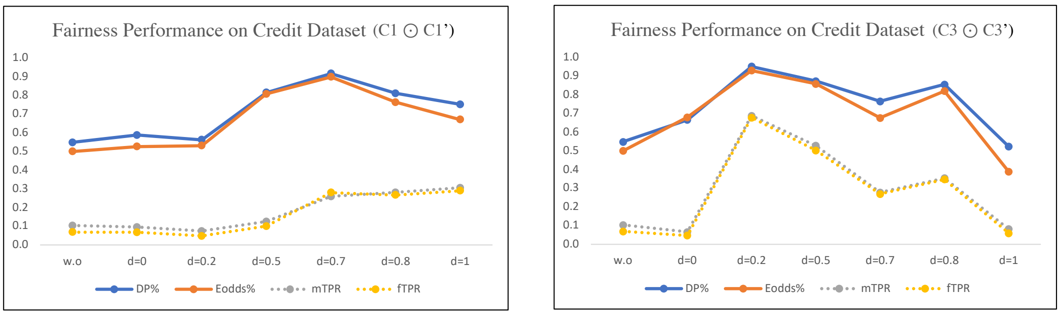

Figure 4: The fairness performance changes under different balancing degree d in Credit Default dataset under MLP model (fTPR: TPR in female group, mTPF: TPR in male group, $d = [0, 0.2, 0.5, 0.7, 0.8, 1]$, refer to Appendix B.1 for detailed results)

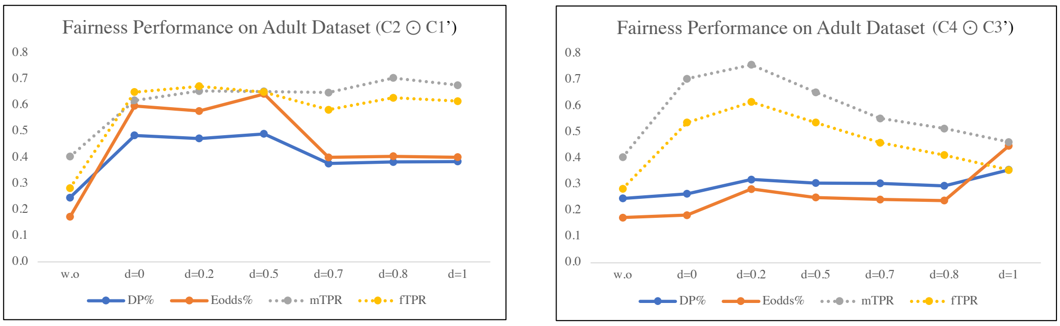

Figure 5: The fairness performance changes under different balancing degree d in the Adult dataset under MLP model (fTPR: TPR in female group, mTPF: TPR in male group, d = *[0, 0.2, 0.5, 0.7, 0.8, 1]*, refer to Appendix B.2 for detailed results)

Table 4
Subgroup-Level Performance Comparison on the Adult Dataset using $C4 \odot C3'$ Sample (F-score refers performance on whole dataset, mF-score presents the F-score in male group, fF-score is the F-score in female group)

	F1 score	mF1 score	fF1 score	mTPR	fTPR	DP%	Eodds%
Baseline	0.7003	0.6895	0.6833	0.4038	0.2831	0.2464	0.1730
d=0	**0.7976**	**0.7851**	**0.7898**	0.7046	0.5368	0.2637	0.1822
d=0.2	0.7857	0.7719	0.7776	**0.7579**	**0.6158**	0.3188	0.2821
d=0.5	0.7841	0.7717	0.7804	0.6523	0.5368	0.3052	0.2499
d=0.7	0.7651	0.7531	0.7633	0.5527	0.4596	0.3041	0.2420
d=0.8	0.7529	0.7421	0.7453	0.5135	0.4118	0.2946	0.2381
d=1	0.7198	0.7132	0.6927	0.4619	0.3548	**0.3558**	**0.4471**

performance across different subgroups. We can see the model trained with data augmentation in the *0* to *0.5* range, although having lower fairness metrics compared to $d = 1$, shows an absolute improvement in model performance. Therefore, we conclude the optimal balancing d for $C4 \odot C3'$ strategy is *0.2*.

Table 5
Counterfactual explanations cost comparison on the Adult dataset with Decision Tree across female(F) and male(M) subgroups with different balancing degree $d = [0, 0.5, 1]$.

Strategy	Baseline	$C1 \odot C1'$			$C2 \odot C1'$			$C3 \odot C3'$			$C4 \odot C3'$		
d	N/A	0	0.5	1	0	0.5	1	0	0.5	1	0	0.5	1
M_{avg}	1.0049	1.5920	1.2988	1.2401	0.4484	0.7440	0.7817	1.1813	1.2762	1.1320	1.0904	0.5551	0.8681
M_{std}	0.7893	1.1671	1.1786	1.1822	0.5500	0.4778	0.9267	1.2770	1.0981	1.1641	0.9260	0.8140	1.2145
F_{avg}	1.0550	1.6986	1.3794	1.3447	0.4291	0.7738	0.7821	1.2001	1.3633	1.1906	1.0666	0.5221	0.7813
F_{std}	0.8887	1.3167	1.2271	1.3758	0.5715	0.5184	1.0165	1.5087	1.3018	1.2127	0.9648	0.7750	1.1617
Δavg	0.0500	0.1066	0.0806	0.1047	0.0193	0.0298	**0.0004**	0.0187	0.0871	0.0586	0.0238	0.0330	0.0868
Δ std	0.0994	0.1497	0.0485	0.1936	**0.0215**	0.0406	0.0899	0.2317	0.2036	0.0486	0.0387	0.0391	0.0528

Table 6
Counterfactual Explanations cost comparison on Law dataset with Decision Tree across female(F) and male(M) subgroups with different balancing degree d

Strategy	Baseline	$C1 \odot C1'$			$C2 \odot C1'$			$C3 \odot C3'$			$C4 \odot C3'$		
d	N/A	0	0.5	1	0	0.5	1	0	0.5	1	0	0.5	1
M_{avg}	0.8671	0.8168	0.7392	0.9506	1.2589	0.7909	0.8026	0.8089	0.8415	0.6298	1.0218	0.8243	0.6968
M_{std}	0.9075	0.7896	0.8382	0.9877	1.2913	0.8120	0.8901	0.8039	0.9753	0.6942	1.0415	0.9347	0.6902
F_{avg}	0.9289	0.9444	0.8134	1.0363	1.4756	0.8351	0.8688	0.9384	0.9278	0.7033	1.1637	0.8763	0.6768
F_{std}	0.9929	0.9233	0.8645	1.0259	1.5175	0.8293	0.9724	0.9269	1.0245	0.7914	1.1162	1.0108	0.6904
Δavg	0.0618	0.1276	0.0742	0.0857	0.2167	0.0442	0.0662	0.1295	0.0863	0.0735	0.1419	0.0519	**0.0200**
Δ std	0.0854	0.1336	0.0263	0.0382	0.2262	0.0173	0.0823	0.1230	0.0492	0.0971	0.0747	0.0761	**0.0001**

5.3. Counterfactual Cost across Different Groups

We now evaluate the effectiveness of our algorithm from the XAI perspective, and the results are consistent with the above observations. First, we calculate the average (avg) and standard deviation (std) of the counterfactual cost across female (F) and male (M) subgroups. Then, we compare the cost gaps between the two groups. A smaller gap indicates fairer counterfactual explanations within different groups. In the Adult dataset, $C2 \odot C1'$ remains to show more significant bias mitigation performance. In the Law school dataset, as we have disscussed above, the improvment is limited because the bias in the original dataset is not significant.

6. Conclusion

This paper proposes a new debiasing algorithm called ProxiMix. It extends the mixup technique by considering labels from proximity samples in the subgroup to mitigate potential bias in the preprocessing stage. Our experiments evaluated the performance of ProxiMix with different sampling combinations and balancing degrees. The results prove that adding proximity-based labels improves fairness performance, and there exists optimal balancing degree for achieving the most significant enhancement. These observations were further supported by the experimental results on the cost comparison of counterfactual explanations. In future work, we plan to extent ProxiMix to multi-class tasks and consider intersectional fairness.

Acknowledgments

This work is funded by Doctoral Training Partnership Studentship of Engineering and Physical Sciences Research Council (EPSRC-DTP, EP/W524414/1/2894964).

References

[1] O. A. Osoba, W. Welser IV, W. Welser, An intelligence in our image: The risks of bias and errors in artificial intelligence, Rand Corporation, 2017.

[2] N. Burkart, M. F. Huber, A survey on the explainability of supervised machine learning, Journal of Artificial Intelligence Research 70 (2021) 245–317.

[3] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information fusion 58 (2020) 82–115.

[4] G. Ciatto, F. Sabbatini, A. Agiollo, M. Magnini, A. Omicini, Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review, ACM Computing Surveys 56 (2024) 1–35.

[5] I. D. Raji, J. Buolamwini, Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 429–435.

[6] L. Kavouras, K. Tsopelas, G. Giannopoulos, D. Sacharidis, E. Psaroudaki, N. Theologitis, D. Rontogiannis, D. Fotakis, I. Emiris, Fairness aware counterfactuals for subgroups, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[7] U. Gohar, L. Cheng, A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges, arXiv preprint arXiv:2305.06969 (2023).

[8] M. Hort, Z. Chen, J. M. Zhang, F. Sarro, M. Harman, Bia mitigation for machine learning classifiers: A comprehensive survey, arXiv preprint arXiv:2207.07068 (2022).

[9] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[10] M. Navarro, C. Little, G. I. Allen, S. Segarra, Data augmentation via subgroup mixup for improving fairness, arXiv preprint arXiv:2309.07110 (2023).

[11] C.-Y. Chuang, Y. Mroueh, Fair mixup: Fairness via interpolation, arXiv preprint arXiv:2103.06503 (2021).

[12] B. T. Luong, S. Ruggieri, F. Turini, K-nn as an implementation of situation testing for discrimination discovery and prevention, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 502–510.

[13] M. B. Zafar, I. Valera, M. Gomez Rodriguez, K. P. Gummadi, Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 1171–1180.

[14] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Advances in neural information processing systems 29 (2016).

[15] B. G. Buchanan, E. H. Shortliffe, Rule based expert systems: the mycin experiments

of the stanford heuristic programming project (the Addison-Wesley series in artificial intelligence), Addison-Wesley Longman Publishing Co., Inc., 1984.

[16] S. Gregor, I. Benbasat, Explanations from intelligent systems: Theoretical foundations and implications for practice, MIS quarterly (1999) 497–530.

[17] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 conference on fairness, accountability, and transparency, 2020, pp. 607–617.

[18] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harv. JL Tech. 31 (2017) 841.

[19] D. Brughmans, P. Leyman, D. Martens, Nice: an algorithm for nearest instance counterfactual explanations, Data Mining and Knowledge Discovery (2023) 1–39.

[20] V. Gupta, P. Nokhiz, C. D. Roy, S. Venkatasubramanian, Equalizing recourse across groups, arXiv preprint arXiv:1909.03166 (2019).

[21] A. Artelt, B. Hammer, Explain it in the same way!–model-agnostic group fairness of counterfactual explanations, arXiv preprint arXiv:2211.14858 (2022).

[22] S. Goethals, D. Martens, T. Calders, Precof: counterfactual explanations for fairness, Machine Learning (2023) 1–32.

[23] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 329–338.

[24] F. Kamiran, T. Calders, Classifying without discriminating, in: 2009 2nd international conference on computer, control and communication, IEEE, 2009, pp. 1–6.

[25] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp. 259–268.

[26] H. Sun, K. Wu, T. Wang, W. H. Wang, Towards fair and robust classification, in: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), IEEE, 2022, pp. 356–376.

[27] F. Kamiran, T. Calders, M. Pechenizkiy, Discrimination aware decision tree learning, in: 2010 IEEE international conference on data mining, IEEE, 2010, pp. 869–874.

[28] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, K. Q. Weinberger, On fairness and calibration, Advances in neural information processing systems 30 (2017).

[29] J. J. Amend, S. Spurlock, Improving machine learning fairness with sampling and adversarial learning, J. Comput. Sci. Coll 36 (2021) 14–23.

[30] A. Morano, Bias mitigation for automated decision making systems, Politecnico di Torino (2020).

[31] D. Dablain, B. Krawczyk, N. Chawla, Towards a holistic view of bias in machine learning: Bridging algorithmic fairness and imbalanced learning, arXiv preprint arXiv:2207.06084 (2022).

[32] J. Chakraborty, S. Majumder, T. Menzies, Bias in machine learning software: Why? how? what to do?, CoRR (2021).

[33] R. Kohavi, et al., Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid., in: Kdd, volume 96, 1996, pp. 202–207.

[34] K. Xivuri, H. Twinomurinzi, A systematic review of fairness in artificial intelligence

algorithms, in: Responsible AI and Analytics for an Ethical and Inclusive Digitized Society: 20th IFIP WG 6.11 Conference on e-Business, e-Services and e-Society, I3E 2021, Galway, Ireland, September 1–3, 2021, Proceedings 20, Springer, 2021, pp. 271–284.

[35] I.-C. Yeh, C.-h. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Expert systems with applications 36 (2009) 2473–2480.

[36] G. Maheshwari, A. Bellet, P. Denis, M. Keller, Fair without leveling down: A new intersectional fairness definition, in: EMNLP 2023-The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[37] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, E. Ntoutsi, A survey on datasets for fairness-aware machine learning, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 12 (2022) e1452.

A. Appendices: Dataset Description

A.1. Adult Income Dataset

The Adult Income dataset is also known as the Census Income dataset. Its documentation [5] provides a detailed description of 14 features in the dataset. We omitted some features, such as ‘fnlwgt’, and the final features we used after data cleaning are as follows.

Table 7
The Adult Dataset Descriptions

Feature Name	Value Type	Description
Sex (sensitive features)	Categories	Gender of the persom, eg. Male, Female
Workclass	Categories	Type of employment, eg. Private, Self-employed
Age	Continuous	Age of the person
Education	Categories	Highest level of education, eg. Bachelors, Some-college
Education-num	Continuous	Education level of the person
Marital-Status	Categories	Marital status of the persom, eg. Single, Married
Occupation	Categories	Occupation of the persom, eg. Tech-support, Sales
Relationship	Categories	Role in the family, eg. Not-in-family, Own-child
Capital-Gain	Continuous	Capital gains of the persom
Capital-Loss	Continuous	Capital loss of the persom
Hours-Per-Week	Continuous	Hours worked per week
Race	Categories	Race of the person, eg. White, Other
Salary (ground truth Y)	Categories	Whether annual income exceeds $50K$

[5] https://www.cs.toronto.edu/~delve/data/adult/adultDetail.html

A.2. Law School Dataset

The Law School dataset contains admission records for law schools. We followed the description provided in [37] and the data cleaning pipeline in [21], extracting the following features for the experiment.

Table 8
The Law School Dataset Descriptions

Feature Name	Value Types	Description
gender (sensitive feature)	Categories	Gender
race	Categories	Race
decile1	Continuous	The decile in the school given his grades in Year 1
decile3	Continuous	The decile in the school given his grades in Year 3
lsat	Continuous	LSAT score
ugpa	Continuous	Undergraduate GPA.
zfygpa	Continuous	The first year Law school GPA
zgpa	Continuous	The cumulative law school GPA.
fulltime	Categories	Work full-time or part-time
fam_inc	Continuous	Family income
pass_bar (ground truth Y)	Categories	Whether passed the bar exam.

A.3. Credit Default Dataset

The Credit Default dataset, also known as the credit card clients dataset, explores default payments on credit cards. Followings are the features and descriptions.

Table 9
The Credit Default Dataset Descriptions

Attribute	Value Types	Description
SEX(sensitive feature)	Categories	Gender
EDUCATION	Categories	Highest education
AGE	Continuous	Age
LIMIT_BAL	Continuous	Amount of given credit
PAY_i ($i \in \{1, 2, 3, 4, 5, 6\}$)	Continuous	Repayment status for ith month
BILL_AMT_i ($i \in \{1, 2, 3, 4, 5, 6\}$)	Continuous	Amount of bill statement for ith month
PAY_AMT_i ($i \in \{1, 2, 3, 4, 5, 6\}$)	Continuous	Amount of previous payment for ith month
Default_Payment(ground truth Y)	Categories	Whether default payment or not next month

B. Appendices: Results

B.1. ProxiMix in Credit Default Dataset with MLP model

Table 10
The prediction and fairness performance under different balancing degree d in Credit Default dataset with MLP model (Acc: accuracy, F1: F1-score, m: performance in male subgroup, f: performance in female subgroup)

d	Strategy	Acc.	F1	ΔDP	DP%	$\Delta Eodds$	Eodds%	mF1	mTPR	mFPR	fF1	fTPR	fFPR
d=0	$C1 \odot C1'$	0.7829	0.5075	0.0170	0.5885	0.0278	0.5263	0.5131	0.0953	0.0243	0.5022	0.0675	0.0128
d=0.2	$C1 \odot C1'$	0.7802	0.4906	0.0147	0.5625	0.0259	0.5318	0.4964	0.0741	0.0210	0.4853	0.0482	0.0112
d=0.5	$C1 \odot C1'$	0.7708	0.5217	0.0119	0.8156	0.0239	0.8082	0.5266	0.1247	0.0453	0.5174	0.1008	0.0396
d=0.7	$C1 \odot C1'$	0.6877	0.5393	0.0181	0.9171	0.0205	0.8994	0.5404	0.2600	0.1817	0.5386	0.2805	0.2020
d=0.8	$C1 \odot C1'$	0.7574	0.5922	0.0303	0.8111	0.0289	0.7643	0.5853	0.2812	0.1227	0.5963	0.2673	0.0938
d=1	$C1 \odot C1'$	0.7569	0.6006	0.0446	0.7523	0.0463	0.6712	0.5880	0.3059	0.1408	0.6087	0.2901	0.0945
	Baseline	0.7781	0.5076	0.0233	0.5485	0.0352	0.5000	0.5147	0.1035	0.0354	0.5008	0.0684	0.0177
d=0	$C2 \odot C1'$	0.5277	0.5013	0.0623	0.8937	0.0661	0.8807	0.4884	0.6894	0.5540	0.5083	0.6599	0.4879
d=0.2	$C2 \odot C1'$	0.4959	0.4775	0.0358	0.9408	0.0477	0.9177	0.5026	0.6894	0.5319	0.4611	0.7020	0.5796
d=0.5	$C2 \odot C1'$	0.5599	0.5262	0.0080	0.9844	0.0145	0.9695	0.5376	0.6565	0.4604	0.5185	0.6670	0.4749
d=0.7	$C2 \odot C1'$	0.6998	0.6079	0.0333	0.8942	0.0306	0.8816	0.6032	0.4941	0.2588	0.6105	0.4829	0.2281
d=0.8	$C2 \odot C1'$	0.7047	0.5520	0.0134	0.9332	0.0173	0.9037	0.5542	0.2659	0.1622	0.5504	0.2787	0.1795
d=1	$C2 \odot C1'$	0.6466	0.5563	0.0200	0.9410	0.0179	0.9416	0.5559	0.4471	0.3063	0.5560	0.4382	0.2884
	Baseline	0.7781	0.5076	0.0233	0.5485	0.0352	0.5000	0.5147	0.1035	0.0354	0.5008	0.0684	0.0177
d=0	$C3 \odot C3'$	0.7757	0.4847	0.0114	0.6661	0.0194	0.6794	0.4877	0.0659	0.0243	0.4816	0.0465	0.0165
d=0.2	$C3 \odot C3'$	0.4284	0.4192	0.0322	0.9516	0.0460	0.9305	0.4478	0.6882	0.6156	0.4004	0.6784	0.6615
d=0.5	$C3 \odot C3'$	0.6440	0.5691	0.0495	0.8736	0.0492	0.8589	0.5618	0.5282	0.3491	0.5730	0.5022	0.2998
d=0.7	$C3 \odot C3'$	0.7641	0.5974	0.0371	0.7649	0.0389	0.6761	0.5849	0.2776	0.1202	0.6057	0.2691	0.0812
d=0.8	$C3 \odot C3'$	0.7203	0.5889	0.0339	0.8551	0.0352	0.8203	0.5795	0.3541	0.1961	0.5947	0.3471	0.1608
d=1	$C3 \odot C3'$	0.7822	0.4988	0.0179	0.5233	0.0225	0.3886	0.5013	0.0812	0.0240	0.4958	0.0587	0.0093
	Baseline	0.7781	0.5076	0.0233	0.5485	0.0352	0.5000	0.5147	0.1035	0.0354	0.5008	0.0684	0.0177
d=0	$C4 \odot C3'$	0.7749	0.5701	0.0379	0.6603	0.0361	0.5572	0.5650	0.2071	0.0815	0.5724	0.1797	0.0454
d=0.2	$C4 \odot C3'$	0.7747	0.5970	0.0460	0.6773	0.0460	0.6061	0.5961	0.2765	0.1006	0.5960	0.2305	0.0610
d=0.5	$C4 \odot C3'$	0.7844	0.5997	0.0310	0.7378	0.0330	0.6813	0.6002	0.2565	0.0748	0.5982	0.2235	0.0510
d=0.7	$C4 \odot C3'$	0.2873	0.2741	0.0112	0.9878	0.0149	0.9835	0.2935	0.9518	0.8931	0.2614	0.9571	0.9081
d=0.8	$C4 \odot C3'$	0.7806	0.5457	0.0302	0.6129	0.0344	0.5211	0.5482	0.1588	0.0527	0.5424	0.1245	0.0275
d=1	$C4 \odot C3'$	0.7824	0.4918	0.0166	0.4915	0.0259	0.3813	0.4970	0.0741	0.0195	0.4868	0.0482	0.0074
	Baseline	0.7781	0.5076	0.0233	0.5485	0.0352	0.5000	0.5147	0.1035	0.0354	0.5008	0.0684	0.0177

B.2. ProxiMix in Adult Income Dataset with MLP model

Table 11
The prediction and fairness performance under different balancing degree d in the Adult dataset with MLP model (Acc: accuracy, F1: F1-score, m: performance in male subgroup, f: performance in female subgroup)

d	Strategy	F1	ΔDP	DP%	$\Delta Eodds$	Eodds%	mF1	mTPR	mFPR	fF1	fTPR	fFPR
d=0	$C1 \odot C1'$	0.7302	0.1653	0.1935	0.2128	0.1240	0.7238	0.4922	0.0793	0.6818	0.2794	0.0098
d=0.2	$C1 \odot C1'$	0.7834	0.1942	0.2905	0.1565	0.2571	0.7747	0.6455	0.1112	0.7618	0.4890	0.0286
d=0.5	$C1 \odot C1'$	0.7431	0.1427	0.2532	0.1374	0.1759	0.7342	0.4885	0.0611	0.7222	0.3511	0.0107
d=0.7	$C1 \odot C1'$	0.7823	0.1814	0.2864	0.1349	0.2291	0.7723	0.6165	0.0958	0.7693	0.4816	0.0220
d=0.8	$C1 \odot C1'$	0.7731	0.1637	0.2554	0.1547	0.1768	0.7652	0.5628	0.0698	0.7499	0.4081	0.0123
d=1	$C1 \odot C1'$	0.7723	0.2026	0.2385	0.1920	0.1635	0.7622	0.6185	0.1119	0.7492	0.4265	0.0183
	Baseline	0.7003	0.1238	0.2464	0.1207	0.1730	0.6895	0.4038	0.0595	0.6833	0.2831	0.0103
d=0	$C2 \odot C1'$	0.7806	0.1325	0.4847	0.0398	0.5979	0.7711	0.6188	0.0991	0.7801	0.6507	0.0592
d=0.2	$C2 \odot C1'$	0.7901	0.1442	0.4729	0.0450	0.5785	0.7819	0.6550	0.1067	0.7843	0.6728	0.0617
d=0.5	$C2 \odot C1'$	0.7834	0.1402	0.4906	0.0392	0.6437	0.7787	0.6529	0.1101	0.7641	0.6507	0.0709
d=0.7	$C2 \odot C1'$	0.7886	0.1689	0.3773	0.0658	0.4007	0.7795	0.6485	0.1061	0.7793	0.5827	0.0425
d=0.8	$C2 \odot C1'$	0.7924	0.1902	0.3832	0.0804	0.4046	0.7837	0.7046	0.1351	0.7787	0.6287	0.0547
d=1	$C2 \odot C1'$	0.7891	0.1800	0.3846	0.0744	0.4011	0.7793	0.6769	0.1243	0.7809	0.6158	0.0499
	Baseline	0.7003	0.1238	0.2464	0.1207	0.1730	0.6895	0.4038	0.0595	0.6833	0.2831	0.0103
d=0	$C3 \odot C3'$	0.7832	0.1997	0.3582	0.1149	0.3713	0.7747	0.6958	0.1429	0.7631	0.5809	0.0531
d=0.2	$C3 \odot C3'$	0.7626	0.1339	0.3412	0.0676	0.3369	0.7520	0.5253	0.0624	0.7597	0.4577	0.0210
d=0.5	$C3 \odot C3'$	0.7729	0.1646	0.2658	0.1466	0.1975	0.7645	0.5675	0.0741	0.7525	0.4210	0.0146
d=0.7	$C3 \odot C3'$	0.7911	0.2052	0.2884	0.1493	0.2360	0.7806	0.6732	0.1202	0.7775	0.5239	0.0284
d=0.8	$C3 \odot C3'$	0.7763	0.1912	0.2167	0.2198	0.1373	0.7700	0.6003	0.0883	0.7359	0.3805	0.0121
d=1	$C3 \odot C3'$	0.7573	0.1410	0.2634	0.1346	0.1936	0.7496	0.5078	0.0531	0.7350	0.3732	0.0103
	Baseline	0.7003	0.1238	0.2464	0.1207	0.1730	0.6895	0.4038	0.0595	0.6833	0.2831	0.0103
d=0	$C4 \odot C3'$	0.7976	0.2260	0.2637	0.1678	0.1822	0.7851	0.7046	0.1330	0.7898	0.5368	0.0242
d=0.2	$C4 \odot C3'$	0.7857	0.2447	0.3188	0.1421	0.2821	0.7719	0.7579	0.1848	0.7776	0.6158	0.0521
d=0.5	$C4 \odot C3'$	0.7841	0.1958	0.3052	0.1155	0.2499	0.7717	0.6523	0.1199	0.7804	0.5368	0.0300
d=0.7	$C4 \odot C3'$	0.7651	0.1559	0.3041	0.0931	0.2420	0.7531	0.5527	0.0803	0.7633	0.4596	0.0194
d=0.8	$C4 \odot C3'$	0.7529	0.1437	0.2946	0.1017	0.2381	0.7421	0.5135	0.0682	0.7453	0.4118	0.0162
d=1	$C4 \odot C3'$	0.7198	0.1226	0.3558	0.1071	0.4471	0.7132	0.4619	0.0716	0.6927	0.3548	0.0320
	Baseline	0.7003	0.1238	0.2464	0.1207	0.1730	0.6895	0.4038	0.0595	0.6833	0.2831	0.0103