Toward Unified Data and Algorithm Fairness via Adversarial Data Augmentation and Adaptive Model Fine-tuning

Yanfu Zhang[1], Runxue Bao[1], Jian Pei[2], and Heng Huang[1]
[1]*Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, United States*
[2]*Department of Computer Science, Duke University, Durham, NC, United States*
yaz91@pitt.edu, baorunxue@gmail.com, jpei@cs.sfu.ca, heng.huang@pitt.edu

***Abstract*—There is some recent research interest in algorithmic fairness for biased data. There are a variety of pre-, in-, and post-processing methods designed for this problem. However, these methods are exclusively targeting data unfairness and algorithmic unfairness. In this paper, we propose a novel intra-processing method to broaden the application scenario of fairness methods, which can simultaneously address the two bias sources. Since training modern deep models from scratch is expensive due to the enormous training data and the complicated structures, we propose an augmentation and fine-tuning framework. First, we design an adversarial attack to generate weighted samples disentangled with the protected attribute. Next, we identify the fair sub-structure in the biased model and fine-tune the model via weight reactivation. At last, we provide an optional joint training scheme for the augmentation and the fine-tuning. Our method can be combined with a variety of fairness measures. We benchmark our method and some related baselines to show the advantage and the scalability. Experimental results on several standard datasets demonstrate that our approach can effectively learn fair augmentation and achieve superior results to the state-of-the-art baselines. Our method also generalizes well to different types of data.**

***Index Terms*—fairness, augmentation, fine-tuning**

I. Introduction

Recently, high-stakes decision-making urges trustworthy machine learning models. For example, data might be corrupted and ML models can also be biased just as human decision-makers [1]–[3]. Algorithmic fairness is gaining growing interest to address this problem. Many works are attempting to achieve fairness commitments for classification models [4]–[9]. Some works try to address a substantial source of the bias, i.e., the dataset itself. Alternatively, many methods try to rectify bias that manifests in models during training, which can be categorized into pre-, in-, or post-processing frameworks. Although these methods achieve great success in many tasks, there are some scenarios preventing their application due to the rapid growth of the size of modern machine learning problems. For example, it is common to adopt some pre-trained backbone models for related tasks, e.g., transformer models for computer vision and BERT for text analysis. In real-world applications, the models are usually trained with the accumulation of data, and there are potential data distribution may vary with time. In these cases, pre-processing and in-processing methods are expensive since they require retraining from scratch each time, and state-of-the-art models may require thousands of GPU hours. Post-processing methods sometimes cannot fully use the models since they are viewed as black boxes.

Intra-processing algorithms have emerged to address these problems. An intra-processing approach has access to a pre-trained model and a dataset typically differing from the biased training dataset. It outputs a debiased model typically by altering the biased model, e.g., updating or augmenting the weights. However, existing intra-processing methods usually overlook some preference of the distribution of data representation concerning fairness, and the fine-tuning cannot fully make use of the network structure. This paper proposes a novel and flexible intra-processing framework to address these limitations. Our contributions are summarized as follows,

- We propose to adjust the biased model by a two-step framework. First, we generate augmented data using adversarial attacks from a potentially biased validation set. Then we fine-tune the unfair model using the augmented dataset. We have more control over the behaviors in the two stages, which correspond to data fairness and algorithmic fairness, respectively.
- We propose a fairness attack method. Compared to the standard adversarial attack, our approach use a global attack to disentangle the protected attribute from the data representations and assign some sample weights to the augmented data, to indicate the powerful fairness attack instead of the general robustness attack.
- We propose identifying the fair sub-structure in the pre-trained model and reactivating the corrupted weights in the fine-tuning, motivated by the over-parameterization property of deep neural networks. Moreover, we discuss a reward-guided joint training scheme for the data augmentation and the model fine-tuning.
- We experimentally demonstrate that our algorithm outperforms state-of-the-art intra-processing baselines, and our approach generalizes well to various settings, e.g., tabular and vision datasets. We also conduct extensive experiments to show the difference between intra-processing and the rest processing methods. The ablation study validates the

This work was partially supported by NSF IIS 1838627, 1837956, 1956002, 2211492, CNS 2213701, CCF 2217003, DBI 2225775.

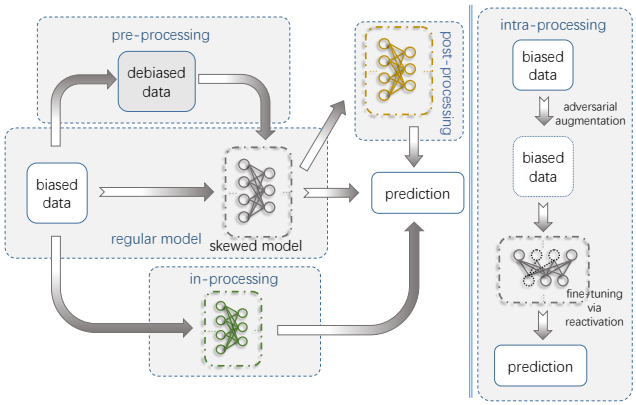

Fig. 1: Comparison of pre-, in-, post-, and intra-processing.

effectiveness of our approach.

II. Methodology

1) Problem Formulation: Our task is to adjust an unfair model using a validation dataset. Formally, $\mathcal{D} = \{(X_i, Y_i)\}$ denotes a dataset, where X_i is a data point containing one binary protected attribute A, and Y_i is the label. $f_\theta : \mathbb{R}^d \to [0, 1]$ is an unfair neural network with weights θ (we will drop θ when it is clear from context). $\hat{\mathcal{Y}} = \{f(X_i)|(X_i, Y_i)\}$ is the prediction. $\rho(\mathcal{Y}, \hat{\mathcal{Y}})$ denotes the performance of f, and we use balanced accuracy in this paper. Specifically, we assume f is l layers feed-forward neural network, and its i^{th} layer is $f^{(i)}$. We denote $f = f^{(l)} \circ f'$, so that the first $l-1$ layers $f' = f^{(l-1)} \circ \cdots \circ f^{(1)}$ can be viewed as an encoder to compute data representations. $\mu(\mathcal{D}, \hat{\mathcal{Y}}, A) \in [0, 1]$ is a bias measure. One typically chooses an appropriate definition of the fairness measure depending on the applications, which we will discuss later.

Since there is usually some trade-off between the performance ρ and the bias μ, we want to decrease the bias μ without significantly sacrifices the performance ρ. A common practice is to maximize the model performance subject to some predetermined tolerance ϵ to the bias, and we have the objective function,

$$\Phi_{\mu,\rho,\epsilon}(\mathcal{D}, \hat{\mathcal{Y}}, A) = \begin{cases} \rho & \text{if } \mu < \epsilon \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

An intra-processing algorithm takes in the validation dataset $\mathcal{D}_{val}$ and a trained model f_θ and outputs a finetuned $f_{\theta'}$ with weights θ' via optimizing the objective $\phi_{\mu,\rho,\epsilon}$. Note that the difference between intra-processing algorithms and pre-, in-, and post- methods makes these methods useful for different problem settings because these paradigms have different access to the data and model, i.e., pre- methods mainly consider the data, in- mainly consider the model training, and the post- sometimes cannot access the model details.

Now we describe the fairness measures used in this work. Following the above notations, we first define the true positive and false positive rates as,

$$\begin{aligned} TPR_{A=a}(\mathcal{D}, \hat{\mathcal{Y}}) &= \frac{|\{i|\hat{Y}_i = Y_i = 1, a_i = a\}|}{|\{i|\hat{Y}_i = Y_i = 1\}|} \\ &= P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = a, Y_i = 1), \quad (2) \\ FPR_{A=a}(\mathcal{D}, \hat{\mathcal{Y}}) &= \frac{|\{i|\hat{Y}_i = 1, Y_i = 0, a_i = a\}|}{|\{i|\hat{Y}_i = 1, Y_i = 0\}|} \\ &= P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = a, Y_i = 0). \quad (3) \end{aligned}$$

Next, we describe the fairness measures used in this paper. *Statistical Parity Difference (SPD)*,

$$\begin{aligned} SPD(\mathcal{D}, \hat{\mathcal{Y}}, A) = P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = 0) \\ - P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = 1). \end{aligned} \tag{4}$$

Equal opportunity difference (EOD) ,

$$EOD(\mathcal{D}, \hat{\mathcal{Y}}, A) = TPR_{A=0}(\mathcal{D}, \hat{\mathcal{Y}}) - TPR_{A=1}(\mathcal{D}, \hat{\mathcal{Y}}). \tag{5}$$

Average Odds Difference (AOD),

$$\begin{aligned} AOD(\mathcal{D}, \hat{\mathcal{Y}}, A) = \frac{1}{2}\Big(\Big(FPR_{A=0}(\mathcal{D}, \hat{\mathcal{Y}}) - FPR_{A=1}(\mathcal{D}, \hat{\mathcal{Y}})\Big) \\ \Big(TPR_{A=0}(\mathcal{D}, \hat{\mathcal{Y}}) - TPR_{A=1}(\mathcal{D}, \hat{\mathcal{Y}})\Big)\Big). \end{aligned} \tag{6}$$

2) Adversarial Data Augmentation: As an intra-processing method, we will use the validation data and the model structure simultaneously. Here validation data refers to a few data points for fine-tuning without breaking the test integrity. In this section, we start describing our method from a balanced validation set, then discuss an extension to an biased dataset in the end.

For balanced data, we assume $\mathcal{D}_{val}$ is balanced concerning the labels and the protected attribute. We augment the validation data via generating adversarial perturbations for the following reasons. First, data augmentation helps boost model accuracy and robustness, whose trade-off is the core problem for algorithmic fairness. Second, the idea of adversarial training has been successfully employed in some related fairness algorithms, for example, distributionally robust optimization for individual fairness (which can be viewed as an in-processing method). We explicitly consider the adversarial data augmentation as a separate step in our intra-processing approach, which allows more control on the behavior of the adversarial training.

However, our problem is different from the standard adversarial attack setting in two senses: first, fairness requirements imply some data distribution property; second, the augmentation needs to consider both classification and bias. We will detail the two challenges and our solutions below.

Global attack to disentangle protected attribute: We observe that algorithmic fairness has an implicit requirement for the distribution of data representations compared to standard classification. Figure. 2 gives an example where representations entangled with protected attribute values may lead to unfairness. We propose considering a global attack instead of an instance-wise attack to generate augmented data to address this

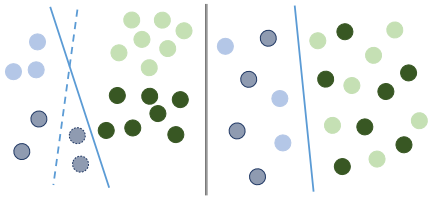

Fig. 2: For each data point, color blue/green denotes label and the hue denotes the protected attribute. Solid line is the fair classifier, and dashed line is biased. Left: data presentation entangled with protected attribute may lead to biased classifier. Right: mixing data representations can avoid the problem.

problem. We define a global attack as an adversarial sample generated from an instance and some data instances with the same labels but mixed protected attributes values, which avoid the potential entanglement between an instance-wise attack and the protected attributes.

In detail, for an instance of interest X_i, we draw a mini-batch $\mathcal{D}_{X_i} = X_j$. Particularly, the samples in the mini-batch have the same label as X_i, and the attribute values are different from X_i with probability p_{flip}. p_{flip} is a hyper-parameter. We consider the following objective function,

$$\ell_{adj} = \sum_{X_j \in D_{X_i}} ||f'(X_i) - f'(X_j)||_2^2, \tag{7}$$

here $f'(X)$ is the second-last layer output of the unfair model f, which can be viewed as the data representation. It is worthy of noting that Eq. (7) is related to some deep metric learning losses. For example, assume we have a pseudo negative X_i^-, and a margin m, the triplet loss is,

$$\begin{aligned}\ell_{triplet} =& \mathbb{E}\left[(||f'(X_i) - f'(X_j)||_2^2 - \right. \\ & \left. ||f'(X_i) - f'(X_i^-)||_2^2 + m)_+\right], \end{aligned} \tag{8}$$

alternatively, we can specify two margin term m_1 and m_2 to obtain the triplet loss,

$$\begin{aligned}\ell_{margin} = & \mathbb{E}\left[((||f'(X_i) - f'(X_j)||_2^2 - m_1 + m_2)_+ \right. \\ & \left. -(-||f'(X_i) - f'(X_i^-)||_2^2 + m_1 + m_2)_+\right], \end{aligned} \tag{9}$$

Although the loss term is different, it is easy to verify that the gradient w.r.t. X_i takes the same form. Since we will use multi-step projected gradient descent to compute the adversarial attack, our loss will share some property of deep metric learning: attacking the loss will reduce the performance of models by increasing the distance between positive pairs because the ultimate goal of metric learning is to pull the positive pairs together while pushing the negative pairs apart. In other words, we obtain some augmented data as a mixup of the data with the same labels but different protected attribute values, which blurs the protected attribute and leads to a fair representation.

We consider two types of perturbation: ℓ_2 attack, which perturbs a sample by its normalized gradient; ℓ_∞ attack, where the sample is perturbed to its gradient direction. Some previous works have shown the effectiveness of these random perturbations.

Assigning weights for unfair data: The standard attack problem usually consider a single objective, e.g., the classification accuracy. Our problem additionally involves the data bias. To be specific, we consider three types of augmented data,

- Those augmented data contribute to the general model robustness but less specific to fairness. In detail, $\Phi_{\mu,\rho,\epsilon}(\mathcal{D}, \hat{\mathcal{Y}}, A)$ are affected while $\mu(\mathcal{D}, \hat{\mathcal{Y}}, A)$ are stable.
- Those augmented data contribute specific to fairness, for example, when the false positive of augmented data is much higher than that of the real data. In detail, $\mu(\mathcal{D}, \hat{\mathcal{Y}}, A)$ are affected while $\Phi_{\mu,\rho,\epsilon}(\mathcal{D}, \hat{\mathcal{Y}}, A)$ are stable.
- Both model performance and bias are affected.

Since we focus on algorithmic fairness, we cannot directly use the model objective function. Some re-arrangement is necessary to emphasize the second and the third types of augmented data. To address this problem, we not only generate augmented data but also assign weights to these data. We define the weights as,

$$\max\{1, \lambda(\hat{\mu} - \epsilon + \epsilon_m * (\hat{Y}_{a=0} - \hat{Y}_{a=1})^2) + 1\}, \tag{10}$$

here $\hat{Y}_{a=0}$ is the prediction of the augmented data point with protected attribute set to 0, and $\hat{Y}_{a=1}$ is the vise versa. λ and ϵ_m are hyper-parameters. $\hat{\mu}$ is the bias proxy computed from the mini-batch so that the data points causing unfairness (i.e., larger than the margin $\hat{\mu} - \epsilon$) have larger weights than general augmented data. While fine-tuning the model, we use stochastic gradient-based methods, and the mini-batches are sampled w.r.t. to these weights.

At last, we discuss how to extend the above adversarial data augmentation when a balanced validation dataset is not available. Since our approach constructs mini-batch with awareness of the labels and protected attributes, we can still generate the adversarial attack samples using the same procedure for a single data point of interest. One feasible method is to randomly sample the data points in the validation set and with replacement. This statistical bootstrapping approach can create a balanced augmentation dataset.

3) Fine-tuning Model via Weights Reactivation: Now we have the augmented dataset $\mathcal{D}'_{val}$. Instead of training from scratch, we adjust the unfair model via fine-tuning against $\mathcal{D}'_{val}$. Modern deep neural networks are over-parameterized. Many works have shown that there are some redundant sub-structures inside a model regarding their contribution to the model performance, e.g., carefully removing a large number of channels or layer shortcuts [10]–[14] usually will not affect the model performance significantly. Moreover, deep neural networks are known for that they can memorize samples with random labels [15]. Motivated by these findings, we propose a weight reactivation method instead of directly fine-tuning the unfair network. First, we freeze the unfair network and assign a mask network to the weights. Then we learn the mask networks to identify those corrupted weights. Next, we keep the valuable weights and reinitialize the corrupted weights

(which is the weights reactivation step). Finally, we fine-tune the reactivated network for some steps.

Formally, let M be a binary mask, which has the same size of θ. We first initialize all entries in M with 1 and construct a masked network $f(X; \theta \odot M)$, which is identical to f in the prediction ability. We then identify the sub-networks causing unfairness with θ frozen via solving the following problem,

$$\min_{M \in \{0,1\}^N} \mathcal{L}_{ft}(f(X; \theta \odot M), y), \quad s.t. \ \|M\|_0/N \leq \tau. \tag{11}$$

here $\mathcal{L}_{ft}$ is the fine-tune loss, τ is a threshold, and $1 - \tau$ of the weights are identified as makes little contribution to algorithmic fairness. By solving the problem in Eq. (11), we can have the optimal mask M^* and the corresponding weights $\theta_M = M^* \odot \theta$, which is a fair sub-structure inside the unfair model.

Directly solving the problem in Eq. 11 is quite hard because of the constraint of L_0 norm. To overcome this difficulty, we reparameterize masks with continuous values using a continuous variable m and recover it using sigmoid. To enable gradient calculation, we can use straight through estimator (STE) [16]. To make the optimization easier, we can change the problem to the following form:

$$\min_{m} \mathcal{L}_{ft}(f(x; \mathcal{W} \odot M), y) + \beta \mathcal{R}(\|M\|_0/N, k), \tag{12}$$

Where β is a coefficient parameter, $\mathcal{R}$ is a regularization term to push $\|M\|_0/N$ to a pre-defined threshold τ. By using this regularization term, the sparsity of all weights is counted together. The optimization of binary masks is then more flexible than using the same sparsity rate for all layers. We choose $\mathcal{R}(\|M\|_0/N, \tau) = \log(\max(\|M\|_0/N, \tau)/\tau)$. After we obtain the mask θ_M, we can obtain the corresponding f_{θ_M}. We no longer need the mask network in the finetune. Instead, we reactivate the zero weights in f_{θ_M} and finetune f_{θ_M} using the weighted objective function $\mathcal{L}_{ft} = \mathbb{E}\left[w\mathcal{L}_{pred}(\hat{(y)}, y)\right]$, where w is the weight of the augmented data point, and $\mathcal{L}_{pred}$ is a generic loss function, e.g., binary cross-entropy.

4) Joint Augmentation and Fine-tuning: Our data augmentation and fine-tuning algorithm consider data fairness and algorithmic fairness, respectively. A naive pipeline first generates the augmented data using different attack types then fine-tune the model. Alternatively, some works show that combining the two isolated processes is sometimes beneficial. We further introduce a reward mechanism to explore the combinations of augmentations efficiently. Assume we have K attack types, we define $P = [p_i]$ to be the probability that the i^{th} attack type is selected, and initial $p_i = \frac{1}{K}$. We generate the attack for each mini-batch according to P using and compute ℓ_i for different attack types. We update P by rewarding powerful attacks, $P_i = \min\{1, (1+\gamma)P_i\}$, $P_j = \max\{0, P_j - \frac{\gamma P_i}{K-1}\}$, $\forall j \neq i$. We alternate the augmentation using updated P and the fine-tuning using the running augmentation. In practice, we can accelerate the training by only running the pruning part.

TABLE I: Computational results on CIFAR-10S benchmark. Since the bias tolerance is 0.05, some approaches are not considered fair. Our method has the best accuracy under the fairness constraint.

	accuracy	bias
Baseline	0.892 ± 0.004	0.080
Uni.Conf. [17]	0.842 ± 0.011	0.097
Adv.Debias [18]	0.841 ± 0.011	0.099
Dom.Disc. [19]	0.904 ± 0.049	0.043
Dom.Ind. [20]	0.920 ± 0.009	0.005
RndPert [21]	0.913 ± 0.021	0.048
LayerwiseOpt [21]	0.898 ± 0.016	0.043
Adv.Ft [21]	0.917 ± 0.018	0.051
Proposed^{unif}	0.919 ± 0.010	0.018
Proposed^{ind}	0.914 ± 0.007	0.033
Proposed	0.926 ± 0.012	0.009
Proposed^{joint}	0.935 ± 0.019	0.014

TABLE II: The performance of the baseline model and our approach for CIFAR-10S benchmark under different bias level.

Bias level	Method	Accuracy
80	Baseline	0.935
	Proposed	0.946
	Proposed^{joint}	0.944
90	Baseline	0.917
	Proposed	0.933
	Proposed^{joint}	0.941
99	Baseline	0.894
	Proposed	0.914
	Proposed^{joint}	0.921

TABLE III: Computational results on CelebA dataset. The results are based on five runs and the mean Bias column indicates the unfair models.

	accuracy	bias
Baseline	0.53 ± 0.00	> 0.05
ROC [22]	0.53 ± 0.01	< 0.05
EqOdds [23]	0.98 ± 0.00	> 0.05
CalibEqOdds [24]	0.51 ± 0.01	< 0.05
RndPert	0.56 ± 0.03	> 0.05
LayerwiseOpt	0.52 ± 0.02	< 0.05
Adv.Ft	0.91 ± 0.00	< 0.05
Proposed	0.93 ± 0.00	< 0.05
Proposed^{joint}	0.96 ± 0.01	< 0.05

III. Results

In this section, we empirically evaluate our approach. We conduct the experiments on two representative data forms, image, and tabular data. The results demonstrate that our method achieves comparable or superior fairness compared to related fair algorithms, and we also show the difference between intra-processing methods and the rest approaches. We also include the ablation studies and discuss some properties of our method in the benchmark.

1) Image Data Classification: We consider two image datasets, CIFAR-10 Skewed and CelebA. CIFAR-10 Skewed is a synthesized dataset serving as a benchmark for comparing intra-processing methods and the related schemes. We also include the necessary ablation study using this benchmark. CelebA is a real-world dataset to further verify the advantage of our approach compared to other state-of-the-art methods. We detail the construction of the two datasets and the experimental evaluation in the following.

We use the CIFAR-10 Skewed (CIFAR-10S) benchmark [20] to show the effectiveness of the intra-processing scheme compared to the rest processing schemes. CIFAR-10S is based on CIFAR-10 [25], a dataset with 50,000 32×32 images evenly distributed between 10 object classes. In CIFAR-10S, each of the ten original classes is subdivided into two new domain subclasses, corresponding to color and grayscale domains within that class. Per class, the 5,000 training images are split 95% to 5% between the two domains; five classes are 95% color, and five classes are 95% grayscale. The total number of images allocated to each domain is thus balanced. We create two copies of the standard CIFAR-10 test set for testing: one in color (COLOR) and one in grayscale (GRAY). These two datasets are considered separately, and only the 10-way classification decision boundary is relevant. The CelebA dataset [26] is a popular image dataset used in computer science research. In this experiments we choose two models [20], [27]. One predicts whether or not the person is young, and the other predicts whether the person is smiling. We set the protected attribute to Fitzpatrick skin tones in the range $4-6$, as in [28]. We label the attributes and use the same pre-training setting following [21]. For both datasets we use a ResNet-18 [29] pretrained on ImageNet from the PyTorch library as the initial model. Table. III summarizes the results. Proposed and Proposed^{joint} are our method without and the jointly training, respectively. Besides, we include two variants for the ablation study. Proposed^{unif} omits the weighted sampling for the augmented data. Proposed^{ind} omit the protected attribute in mini-batch sampling. Table. III summarizes the results.

2) Tabular Data Classification: Besides image datasets, we also consider three widely-used tabular binary classification datasets from AIF360 [30] to show that our approach can generalize to different application scenarios. Each dataset contains at least one protected feature. For all experiments, we follow the settings in [21]. The results are obtained by averaging the fairness metrics on the test sets based on ten random initialization. Table IV summarizes the results on the Adult dataset. Table V summarizes the results on the COMPAS dataset. Table VI summarizes the results on the Bank dataset. For all datasets, we follow [21] and use a feed-forward neural network with ten fully-connected layers of size 32. A BatchNorm layer follows each fully-connected layer. We use a dropout fraction of 0.2. For more details, please refer to [21]. The rest of the settings are similar to the image tasks.

3) Discussions: In this experiment, we have several observations. For the CIFAR-10S benchmark, a perfect unbiased model trained on clean data has a performance of 95.4%. Table. I shows that our method works are superior to all the baselines significantly. Table. II further highlights the performance evolution w.r.t. bias, and we can find that for extremely high bias (i.e., 99%), our method still performs well. Our approach can achieve nearly perfect fairness when the bias is moderately high (i.e., 80%). We also notice that the model accuracy is relatively stable w.r.t. initialization. However, the model bias usually has a larger perturbation which can be future work. We notice that the algorithmic design for specific fairness criteria cannot generalize to different scenarios. These results are consistent with the observation that many group fairness constraints are intrinsically incompatible so that trade-offs between them shall be considered [31]. On the contrary, the adversarial framework is more flexible, and different fairness objectives can share the same processing. Our approach usually has better-balanced accuracy and comparable (i.e., no statistical significance) bias than the state-of-the-art intra-processing baselines. This result indicates that our approach dominates the baselines Pareto-optimally. Proposed^{unif} and Proposed^{ind} are ablated versions. The performance gap between the ablated version and the proposed full algorithm demonstrates the effectiveness of our design. On the other hand, the advantage of joint training is more dependent on the tasks. We notice that, in general, Proposed^{joint} works better than Proposed for image data. Tabular data have a different structure compared to image data, and the model is usually simpler, which benefits less from the joint training.

IV. Conclusion

In this paper, we propose a novel intra-processing fairness framework. Our framework includes two steps. First, we augment the available data to reduce the potential data level bias. Second, we identify the fair sub-structure in the biased model and fine-tune the reactivated model to obtain the algorithmic fairness. We benchmark the performance of the intra-processing method and show the effectiveness of our design. Extensive experiments demonstrate that our approach is suitable for various application scenarios and has a comparable performance w.r.t. state-of-the-art methods.

References

[1] J. Dastin, "Amazon scraps secret ai recruiting tool that showed bias against women," *Reuters*, 2018.

[2] J. Li, J. Pei, and H. Huang, "Communication-efficient robust federated learning with noisy labels," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 914–924.

[3] N. Vigdor, "Apple card investigated after gender discrimination complaints," *The New York Times*, 2019.

[4] M. Gwilliam *et al.*, "Rethinking common assumptions to mitigate racial bias in face recognition datasets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4123–4132.

[5] M. Yurochkin *et al.*, "Training individually fair ml models with sensitive subspace robustness," in *International Conference on Learning Representations*, 2019.

[6] Y. Zhang, S. Gao, and H. Huang, "Recover fair deep classification models via altering pre-trained structure," in *European Conference on Computer Vision*. Springer, 2022.

[7] A. Bower, L. Niss, Y. Sun, and A. Vargo, "Debiasing representations by removing unwanted variation due to protected attributes," *arXiv preprint arXiv:1807.00461*, 2018.

TABLE IV: Computational results on Adult dataset. We use AOD and SPD as the fairness measure and race and sex as the protected attribute. − indicates that the bias is out of bound so that the accuracy cannot be accepted.

	AOD-sex		SPD-race		SPD-sex	
	accuracy	bias	accuracy	bias	accuracy	bias
Baseline	−	0.175 ± 0.016	−	0.178 ± 0.013	−	0.341 ± 0.018
ROC [22]	−	0.052 ± 0.009	−	0.050 ± 0.006	−	0.052 ± 0.007
EqOdds [23]	−	0.081 ± 0.018	0.51	0.000 ± 0.001	0.66	0.026 ± 0.007
CalibEqOdds [24]	−	0.299 ± 0.020	−	0.178 ± 0.019	−	0.221 ± 0.012
Adv.Debias [18]	0.81	0.008 ± 0.011	0.65	0.042 ± 0.008	0.60	0.036 ± 0.002
RndPert	0.73	0.044 ± 0.009	0.64	0.051 ± 0.001	0.59	0.042 ± 0.003
LayerwiseOpt	0.62	0.024 ± 0.010	0.63	0.041 ± 0.010	0.58	0.038 ± 0.004
Adv.Ft	0.61	0.032 ± 0.009	0.61	0.033 ± 0.011	0.55	0.044 ± 0.004
Proposed	0.77	0.049 ± 0.011	0.64	0.042 ± 0.008	0.60	0.043 ± 0.002
Proposed^{joint}	0.65	0.036 ± 0.013	0.61	0.028 ± 0.012	0.57	0.036 ± 0.003

TABLE V: Results on COMPAS dataset. We use EOD as the fairness measure and sex as the protected attribute.

	accuracy	bias
Baseline	−	0.152 ± 0.147
ROC [22]	0.50	0.013 ± 0.028
EqOdds [23]	0.51	0.011 ± 0.009
CalibEqOdds [24]	0.36	0.023 ± 0.029
Adv.Debias [18]	0.62	0.081 ± 0.109
RndPert	−	0.084 ± 0.016
LayerwiseOpt	0.52	0.039 ± 0.043
Adv.Ft	0.59	0.036 ± 0.017
Proposed	0.61	0.031 ± 0.014
Proposed^{joint}	0.62	0.042 ± 0.011

TABLE VI: Computational results on Bank dataset. We use SPD as the fairness measure and age as the protected attribute.

	accuracy	bias
Baseline	−	0.191 ± 0.049
ROC [22]	−	0.078 ± 0.056
EqOdds [23]	0.51	0.001 ± 0.000
CalibEqOdds [24]	−	0.164 ± 0.013
Adv.Debias [18]	−	0.107 ± 0.086
RndPert	−	0.113 ± 0.022
LayerwiseOpt	−	0.058 ± 0.039
Adv.Ft	0.53	0.050 ± 0.031
Proposed	0.53	0.049 ± 0.027
Proposed^{joint}	0.52	0.050 ± 0.030

[8] Y. Zhang, L. Luo, and H. Huang, "Unified fairness from data to learning algorithm," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1499–1504.

[9] M. Kim *et al.*, "Fairness through computationally-bounded awareness," in *Advances in Neural Information Processing Systems*, 2018, pp. 4842–4852.

[10] G. Huang *et al.*, "Condensenet: An efficient densenet using learned group convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2752–2761.

[11] R. Bao *et al.*, "Doubly sparse asynchronous learning for stochastic composite optimization," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, 2022, pp. 1916–1922.

[12] F. Yang, M. Cisse, and O. O. Koyejo, "Fairness with overlapping groups; a probabilistic perspective," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[13] R. Bao and Others, "Fast oscar and owl regression via safe screening rules," in *International Conference on Machine Learning*, 2020.

[14] Y. Zhang, S. Gao, and H. Huang, "Exploration and estimation for model compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 487–496.

[15] D. Arpit, S. Jastrzkebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 233–242.

[16] Y. Bengio and othersothers, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[17] M. Alvi *et al.*, "Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[18] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[19] J. Zhao *et al.*, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," *arXiv preprint arXiv:1707.09457*, 2017.

[20] Z. Wang *et al.*, "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8919–8928.

[21] Y. Savani, C. White, and N. S. Govindarajulu, "Intra-processing methods for debiasing neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2798–2810, 2020.

[22] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 924–929.

[23] M. Hardt and Others, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016.

[24] G. Pleiss *et al.*, "On fairness and calibration," *Advances in neural information processing systems*, vol. 30, 2017.

[25] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[26] Z. Liu *et al.*, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[27] E. Denton, B. Hutchinson, M. Mitchell, and T. Gebru, "Detecting bias with generative counterfactual face attribute augmentation," 2019.

[28] B. Wilson, J. Hoffman, and J. Morgenstern, "Predictive inequity in object detection," *arXiv preprint arXiv:1902.11097*, 2019.

[29] K. He and zheng, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] R. K. Bellamy *et al.*, "Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," *arXiv preprint arXiv:1810.01943*, 2018.

[31] "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.