Recover Fair Deep Classification Models via Altering Pre-trained Structure

Yanfu Zhang, Shangqian Gao, and Heng Huang

Department of Electrical and Computer Engineering, University of Pittsburgh

{yaz91,shg84,heng.huang}@pitt.edu

Abstract. There have been growing interest in algorithmic fairness for biased data. Although various pre-, in-, and post-processing methods are designed to address this problem, new learning paradigms designed for fair deep models are still necessary. Modern computer vision tasks usually involve large generic models and fine-tuning concerning a specific task. Training modern deep models from scratch is expensive considering the enormous training data and the complicated structures. The recently emerged intra-processing methods are designed to debias pre-trained large models. However, existing techniques stress fine-tuning more, but the deep network structure is less leveraged. This paper proposes a novel intra-processing method to improve model fairness by altering the deep network structure. We find that the unfairness of deep models are usually caused by a small portion of sub-modules, which can be uncovered using the proposed differential framework. We can further employ several strategies to modify the corrupted sub-modules inside the unfair pre-trained structure to build a fair counterpart. We experimentally verify our findings and demonstrate that the reconstructed fair models can make fair classification and achieve superior results to the state-of-the-art baselines. We conduct extensive experiments to evaluate the different strategies. The results also show that our method has good scalability when applied to a variety of fairness measures and different data types.

Keywords: fairness, model pruning

1 Introduction

Recently, machine learning models have been increasing in usage in different applications. However, evidence shows that ML models can be biased just as human decision-makers, resulting in serious problems in some high-stakes decision-making, such as awarding loans, deciding probationers' risks, or detecting fraud. The bias in machine decision has two main sources, the intrinsic algorithm design, and the flawed training data collection. For example, people found some gender bias in Amazon's resume screening tool [11] and the credit limits of Apple Card [39]. Algorithmic fairness is gaining growing interest to alleviate this issue. Usually, some features used for the decision-making data indicate the underprivileged groups in the population. The classifiers learned with algorithmic fair-awareness are insensitive to these features, *a.k.a.* protected attributes. For example, a machine learning model is usually trained w.r.t. award loans based on user profiles, where gender and ethnicity are protected attributes. Algorithmic fairness prevents decision-making from being associated with gender or ethnicity. To fit into different application scenario, researchers proposed various definitions for algorithmic fairness, including individual fairness [13], demographic parity [8], equal odds and equal opportunities [21], disparate treatment, impact, and mistreatment [49]. Among the works attempting to achieve fairness commitments for classification models, some 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, some scenarios prevent their application. Due to the rapid growth of the size of modern machine learning problems, it is common to adopt some pre-trained backbone models and fine-tune them for some specific tasks. In real-world applications, the models are usually trained with the accumulation of data, and the 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.

Recently, 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 by updating or augmenting the weights.

However, existing intra-processing methods are usually designed for general machine learning models, thus cannot fully use the deep network structure. This paper proposes a novel intra-processing framework to address this limitation—we alter the structure of an unfair model to recover a fair counterpart. Our contributions are summarized as follows,

- We propose a conjecture that the unfairness of a deep network is caused by only a small portion of its sub-modules. We design a differentiable scheme to identify those model weights corrupted by unfairness. We verify our findings empirically and show that the percentage of corrupted weights is quite low.
- We propose several strategies to reconstruct the fair classification model by modifying the corrupted networks. First, we can obtain a slim network by removing the unfair weights and applying off-the-shelf intra-process methods. Second, we can graft informative filters into the corrupted weights. And third, we can refine the slim network via network augmentation.
- We experimentally verified our findings and demonstrated 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 conducted extensive experiments to study the corrupted weights and different model altering strategies.

2 Related Works

2.1 Fairness in Machine Learning

At a high level, algorithmic fairness can be mathematically defined by the group or individual, and various formal definitions of fairness have been proposed. Individually fair models [13] are based on the intuition that similar users deserve similar treatments. They map input metric spaces to output metric spaces, where individual fairness is defined as Lipschitz continuity of the models. Individual fairness has a preferable property that the Lipschitz continuity naturally implies statistical parity between subgroups of the population. On the other hand, group fairness (sometimes referred to as statistical parity) considers the invariance of the machine learning models on the protected non-overlapping subsets. Group fairness sometimes makes the computation simpler than individual fairness since it is compliant with statistical analysis. The core research problem for fairness is to identify the sources of unfairness and design the corresponding solutions. Imbalanced data concerning the protected attributes usually lead to an unfair model, i.e. the unfairness from the data. To address this problem, some works, including BUPT-Balancedface/RFW [40] and Fairface [27], try to build balanced data. Alternatively, some recent research finds that sometimes data imbalance doesn't necessarily lead to unfairness [17], which makes the problem more intriguing. Meanwhile, some methods aims at learning a fair classifier on top of the biased data [48,53,34,6,28]. Many debiasing algorithms can be split into three categories based on the processing of data and model. Pre-processing methods directly change the data. In-processing approaches train machine learning models tailored to making fairer prediction. Post-processing techniques refine the potentially biased predictions outputted by a fixed model. Our method is an intra-processing method [37] identifying the critical parts of the models causing the unfairness, while previous related methods promote the fairness via finetuning using partial knowledge of the models.

2.2 Fine-tuning Over-parameterized Network

Modern deep neural networks typically achieve higher performance from larger model scales. The model scales mainly come from two sources: the layer width and the network topology. Recent research shows that both the network weights and the network topology have redundancy concerning the model utility. For example, [22] benefits from huge convolution filter numbers. Some works show that carefully dropping part of network weights makes only a slight performance decrease [18], even when a substantial amount of the weights are removed. Many pruning methods are proposed to identify what weights are redundant. Magnitude pruning [20] removes weights with small norm values. Lottery ticket hypothesis (LTH) [15] shows that a sparse sub-network exists at the initialization time that can reach the performance of the full model. Moreover, the weight drop can be conducted at a different level, e.g., channel-wise [31,52,16] and weight-wise [20]. On the other hand, people found that long-distance connections can improve the model performance [24]. However, there is some inefficiency because the deeper layers consider the early features as “obsolete” ones and ignore them while learning new representations. CondenseNet [45] and ShuffleNet [51] alleviate this inefficiency through strategically pruning redundant connections and exponentially discarding cross-layer connections, respectively. One explanation for this phenomenon is that a deep neural network can be viewed as a large ensembled model, and only some sub-structures play a vital role in prediction performance. This argument is empirically supported by some network refinement methods [19,38]. Part of our framework is motivated by this body of research. Instead of refining special sub-structures, one can also train potential sub-networks with different sizes and use larger networks to help the training of smaller ones. A typical case is slimmable neural networks [47], which train sub-networks with different widths at the same time. On top of slimmable networks, universally slimmable networks [46] proposed enhanced training techniques that distill knowledge from larger sub-networks (including the full model) to smaller sub-networks. Recently, network augmentation [7] put small models into large models to improve the training of the small model.

Of note, our work is not directly related to the fair differentiable neural architecture search. FairDARTS [10] and FairNas [9] define Expectation Fairness (EF) and Strict Fairness (SF) to alleviate supernet bias and avoid the unfair advantage of skip connections for residual modules. They use the terminology “fairness” totally different from our paper (our work considers the classification fairness).

3 Methodology

In this section we first briefly recap the problem formulation of the algorithmic fairness for the intra-processing scenario to make our paper self-contained. Next, we describe our method to discover the candidate sub-modules for the model structure alteration. At last, we discuss several strategies to reconstruct a fair model with awareness of the corrupted sub-modules.

3.1 Problem Formulation

Intra-processing debiasing 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 fine-tuned $f_{\theta'}$ 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.

Fairness Measures Now we describe the fairness measures used in this work. We first define the true positive and false positive rates as,

$$TPR_{A=a}(\mathcal{D}, \hat{\mathcal{Y}}) = \frac{|\{i|\hat{Y}_i = Y_i = 1, a_i = a\}|}{|\{i|\hat{Y}_i = Y_i = 1\}|} = P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = a, Y_i = 1), \tag{2}$$

$$FPR_{A=a}(\mathcal{D}, \hat{\mathcal{Y}}) = \frac{|\{i|\hat{Y}_i = 1, Y_i = 0, a_i = a\}|}{|\{i|\hat{Y}_i = 1, Y_i = 0\}|} = P_{(X_i,Y_i)\in\mathcal{D}}(\hat{Y}_i = 1|a_i = a, Y_i = 0). \tag{3}$$

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

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

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),

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

3.2 Finding Corrupted Sub-modules

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. Note that we only use the validation data of limited size instead of the complete training data, and the reasons are two-folded. First, fine-tuning the unfair model is sometimes prohibitive considering the size of the full training data, particularly when the model is also large. Second, the complete training data are usually biased in the protected attribute distribution. It is more viable to collect a small and unbiased validation set. For the second point, one implicitly assumes that primarily training on an imbalanced dataset is inherently disadvantageous. However, whether the assumption is valid in some application scenarios is still an open question. For example, in face recognition, training on only African faces induced less bias than training on a balanced distribution of faces and distributions biased to include more African faces produced more equitable models, and adding more images of existing identities to a dataset in place of adding new identities can lead to accuracy boosts across racial categories [17]. We believe that this assumption deserves further investigation concerning the specific tasks. In this paper, we deal with this issue in a conservative manner and focus on a balanced validation set. Specifically, this section assumes that we have the unfair model f and the balanced validation data $\mathcal{D}_{val}$. In the following, we describe our method to discover the sub-modules making substantial contribution to unfairness.

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 [25,44] usually will not affect the model performance significantly. Moreover, deep neural networks are known for that they can memorize samples with random labels [3], and show some properties of ensembled models, e.g., dropout, as a frequently used technique, functions similar as bagging [42]. Motivated by these findings, we make a conjecture that different modules in a deep network make different contributions to the model prediction and fairness. With the help of the validation dataset, we can discover the influential weights leading to unfairness. Specifically, we freeze the unfair network and assign a mask network to the weights. Then we learn the mask networks to identify those corrupted weights. We will empirically verify this conjecture in the experiments. The model alteration will focus on these corrupted weights, which will be detailed in the next section.

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,

$$\min_{M \in \{0,1\}^N} \ \mathcal{L}_{ft}(f(X; \theta \odot M), y), \quad s.t. \ \|M\|_0 / N \leq \tau. \tag{7}$$

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. (7), 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.

Solving the problem in Eq. 7 directly is difficult because of the constraint of L_0 norm. To overcome this difficulty, we reparameterize masks with continuous

Algorithm 1: Discover Model Weights Corrupted by Unfairness

```
Input: validation data $\mathcal{D}_{val}$, original classifier $f$, epochs $E$, parameters $\tau$, $\beta$
Output: pruned mode $f_{ft}$
freeze $\theta$ in $f$, initialize $m$;
for $e := 1$ to $E$ do
    shuffle($\mathcal{D}_{val}$)
    for a mini-batch $(x, y)$ in $\mathcal{D}_{val}$ do
        obtain masked $f_M$ using $m$ and $\epsilon_m \sim \mathcal{N}(0, \min(1, \max(0.05, 0.5m)))$;
        compute gradients for $m$ w.r.t. (9) and update it with ADAM;
    end
end
```

values, which becomes,

$$M = \begin{cases} 1 \text{ if sigmoid}(\hat{m}) \geq 0.5 \\ 0 \text{ if sigmoid}(\hat{m}) < 0.5 \end{cases} \tag{8}$$

here $\hat{m} = m + \epsilon_m$, which use the reparameterization trick. m is the learnable relaxed mask, ϵ_m is randomly drawn Gaussian noise whose variance is adaptive w.r.t. m. We use ϵ_m to avoid the discovery stuck in bad local minimum. Eq. (8) is still not differentiable, to enable gradient calculation, we can use straight through estimator (STE) [5], and the gradients of m can be calculated by: $\frac{\partial M}{\partial m} = \frac{\partial M}{\partial \text{sigmoid}(m)} \frac{\partial \text{sigmoid}(m)}{\partial m}$.

Although binary masks are differentiable, the problem in Eq. 7 is in constraint form. To make the optimization easier, we can change the problem to the following form:

$$\min_{m} \ \mathcal{L}_{ft}(f(x; \mathcal{W} \odot M), y) + \beta \mathcal{R}(\|M\|_0 / N, k), \tag{9}$$

Where β is a coefficient parameter, $\mathcal{R}$ is a regularization term to push $\|M\|_0/N$ to a pre-defined threshold τ. Applying this regularization term will count the sparsity of all weights together. The optimization of binary masks is then more flexible than using the same sparsity rate for all layers. We choose $\mathcal{R}(\|M\|_0/N, \tau) = \log(\max(\|M\|_0/N, \tau)/\tau)$, instead of commonly used regression losses, like MAE or MSE, because both of them can not reach desired sparsity when τ is small (for example, $\tau = 0.01$).

After we obtain the mask θ_M, we can obtain the corresponding f_{θ_M}. In the next part, we will discuss several strategies to refine the fair classification model on top of f_{θ_M}. The full algorithm is described in Algorithm. 1.

3.3 Strategies to Recover the Fair Classification Model

A straightforward method to reconstruct the fair model is to remove all corrupted weights and only use the left structures. In this case, we no longer need

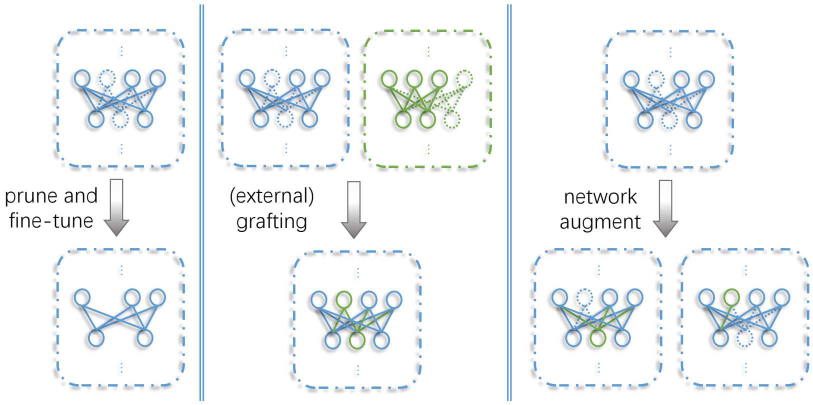

Fig. 1: Illustration of three strategies. The corrupted sub-modules are denoted by dashed circles and lines. The prune and fine-tune strategy builds a pruned model via removing all corrupted weights. The grafting strategy build a model via grafting the unfair weights using two independently discoveries (denoted by different color). The network augmentation strategy maintains the unfair model and refine the augmented fair subnetwork.

the mask network. Rather, f_{θ_M} can be viewed as a fair subnetwork inside the unfair model, which is related to model compression to some degree. However, the fair subnetwork may lose some generalization ability to the classification problem. This phenomenon is also confirmed in the model pruning literature—the prediction accuracy decrease is usually non-negligible and grows with the pruning rate. Besides, training the pruned network from scratch instead of fine-tuning may lead to worse performance. In this section, we describe several strategies to reconstruct a fair model on top of the fair subnetwork. Figure. 1 illustrates the three strategies.

Adversarial Fine-tuning Pruned Model A straightforward process to refine the fair subnetwork is to adjust the pruned f_{θ_M} using off-the-shelf in-processing methods. More specifically, we remove all zero weights in f_{θ_M} and run the adversarial fine-tuning [37]. Since the unfair full model can be processed similarly, we can compare the fine-tuning results between the fair subnetwork and the unfair full model to demonstrate the necessity of pruning for a better performance—in other words, verify our conjecture on the existence of the fair subnetwork. We will show the results in the experiments.

Network Grafting Network grafting is a learning paradigm related to pruning. Model pruning attempts to improve the model efficiency via removing unimportant filters. Alternatively, network grafting aims to improve the representation capability of deep neural networks via grafting the external information into unimportant filters. We adopt the terminologies in filter grafting [33,38] and interpret the corrupted weights as the rootstocks. We re-activate these corrupted weights via grafting, i.e., replacing the corrupted weight values using some scions. There are three types of scions: noise, internal filters, and external filters. One can follow the filter grafting [33] for grafting the corrupted weights using noise and internal filters as scions. Here the noise scions are Gaussian noise having a larger ell_1 norm than the corrupted weights. The internal filters are the weights in the fair subnetwork with the largest ell_1 norms. For the external filters, we propose a variant. We independently discover two fair subnetworks from the unfair model. This step is feasible because the fair subnetwork pruning is to solve an integer problem using approximate methods, which usually results in different seeds. We then execute layer-level grafting. Since the location of the corrupted weights for the two subnetworks are generally different, they can learn mutual information from each other.

Network Augmentation Training large neural networks usually uses regularization techniques (e.g., data augmentation, dropout) to overcome over-fitting. Some recent work [7] observes that these techniques might hurt the performance of tiny neural networks. Instead of augmenting the data, one should augment the model to avoid the under-fitting. This strategy maintains a large network, in which the tiny network is a subnetwork. Augmented nets are sampled from the large network and fine-tuned against the augmented loss function $\mathcal{L}_{aug} = \mathcal{L}(\theta_M) + \sum_i \alpha_i \mathcal{L}([\theta_M, \theta_i])$, where $[\theta_M, \theta_1]$ represents an augmented model that contains the tiny neural network and α_i is the scaling hyper-parameters. For our problem, the construction of the large augmented network is of particular convenience—we can directly use the unfair full model for that purpose. As such, we can sample the augmented net i containing the fair subnetwork from the full unfair model and update the unfair model i progressively using gradient-based optimization methods.

4 Results

We conduct the experiments on representative image and tabular datasets. The results demonstrate that our method achieves comparable or superior fairness compared to related fair algorithms.

4.1 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.

Table 1: 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. [2]	0.842 ± 0.011	0.097
Adv.Debias [50]	0.841 ± 0.011	0.099
Dom.Disc. [54]	0.904 ± 0.049	0.043
Dom.Ind. [41]	0.920 ± 0.009	0.005
RndPert [37]	0.913 ± 0.021	0.048
LayerwiseOpt [37]	0.898 ± 0.016	0.043
Adv.Ft [37]	0.917 ± 0.018	0.051
Prune + AdvFt	0.920 ± 0.014	0.033
NoiseGraft	0.909 ± 0.011	0.045
InterGraft	0.918 ± 0.013	0.013
ExtGraft	0.927 ± 0.009	0.028
NetAug	0.931 ± 0.016	0.014

Table 2: The performance of the baseline model and our approach for CIFAR-10S benchmark under different bias level.

Bias level	Baseline	Strategy		
		Adv.Ft	ExtGraft	NetAug
80%	0.935	0.941	0.944	0.946
90%	0.917	0.937	0.945	0.943
99%	0.894	0.916	0.912	0.915

Table 3: 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 [26]	0.53 ± 0.01	< 0.05
EqOdds [21]	0.98 ± 0.00	> 0.05
CalibEqOdds [36]	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
Prune + AdvFt	0.93 ± 0.00	< 0.05
ExtGraft	0.94 ± 0.00	< 0.05
NetAug	0.94 ± 0.01	< 0.05

Data Description: We use the ***CIFAR-10 Skewed*** (CIFAR-10S) benchmark [41] to show the effectiveness of the intra-processing scheme compared to the rest processing schemes. CIFAR-10S is based on CIFAR-10 [30], a dataset with 50,000 32×32 images evenly distributed between 10 object classes. In CIFAR-

Table 4: 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	
	accuracy	bias	accuracy	bias
Baseline	~~0.86~~	0.175 ± 0.016	~~0.86~~	0.178 ± 0.013
ROC [26]	~~0.79~~	0.052 ± 0.009	0.71	0.050 ± 0.006
EqOdds [21]	~~0.66~~	0.081 ± 0.018	0.51	0.000 ± 0.001
CalibEqOdds [36]	~~0.84~~	0.299 ± 0.020	~~0.75~~	0.178 ± 0.019
Adv.Debias [50]	0.81	0.008 ± 0.011	0.65	0.042 ± 0.008
RndPert	0.73	0.044 ± 0.009	0.64	0.051 ± 0.001
LayerwiseOpt	0.62	0.024 ± 0.010	0.63	0.041 ± 0.010
Adv.Ft	0.61	0.032 ± 0.009	0.61	0.033 ± 0.011
Prune + AdvFt	0.77	0.049 ± 0.011	0.64	0.042 ± 0.008
ExtGraft	0.65	0.036 ± 0.013	0.61	0.028 ± 0.012
NetAug	0.65	0.036 ± 0.013	0.61	0.028 ± 0.012

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 and one in grayscale. These two datasets are considered separately, and only the 10-way classification decision boundary is relevant. The ***CelebA*** dataset [32] consists of over 200,000 images of celebrity headshots, along with binary attributes, but some binary categorization of attributes such as gender, hair color, and age does not reflect true human diversity and is problematic [41,12]. In this experiments we choose two models. 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 [14] in the range $4-6$ following [43], and label the attributes and use the same pre-training setting following [37].

Comparative Methods: For Cifar-10S, we consider the best-performing in the benchmarking [41] including uniform confusion loss [2], Adversarial Debiasing [50], prior shift inference [54], and domain-independent training [41]. We also include three intra-processing methods in [37], Random Perturbation, Layer-wise Optimization, and Adversarial Fine-tuning. For all methods, we use the standard 10-way classifier, following [41]. For CelebA, we focus on the comparison of the proposed method with the biased baseline model and several related methods, including the reject option classification post-processing algorithm [26], which is designed to minimize statistical parity difference; the equalized odds post-processing algorithm [21] for minimizing equal opportunity differences; the Calibrated equalized odds post-processing algorithm [36] for equal opportunity dif-

Table 5: Results on COMPAS dataset.

	accuracy	bias
Baseline	~~0.85~~	0.152 ± 0.147
ROC [26]	0.50	0.013 ± 0.028
EqOdds [21]	0.51	0.011 ± 0.009
CalibEqOdds [36]	0.36	0.023 ± 0.029
Adv.Debias [50]	~~0.62~~	0.081 ± 0.109
RndPert	~~0.68~~	0.084 ± 0.016
LayerwiseOpt	0.52	0.039 ± 0.043
Adv.Ft	0.59	0.036 ± 0.017
Prune + AdvFt	0.61	0.035 ± 0.014
ExtGraft	0.61	0.044 ± 0.011
NetAug	0.60	0.041 ± 0.012

ferences. We also consider Adversarial Debiasing [50] which is an in-processing method using the adversarial critic to predict the protected attribute and highly related to the intra-processing methods.

Our setting: Prune + Adv.Ft is adversarial fine-tuning strategy. We fine-tune the model 90 epochs with 5 warmpup steps and cosine annealing learning rate scheduler, using ADAM optimizer with starting learning rate 0.01. Noise-, Inter-, and ExtGraft are grafting using noise, internal scions, and external scions, respectively. For ExtGraft, we consider two independent discovery and always use the first run as the final model the since the two perform close to each other. The model is trained for 100 epochs. NetAug is network augmentation. We use one augmentation per epoch, and train the network 100 epochs, and the scaling parameter is 1. We choose a small diversity factor of 0.05, since the prune rate is low. For both dataset we use a ResNet-18 [22] pretrained on ImageNet from the PyTorch library [35] as the initial model. We set $E = 80$, $\beta = 5$, and $\tau = 0.1$.

4.2 Additional Results on Tabular Data Classification

Besides image datasets, we also consider two widely-used tabular binary classification datasets from AIF360 [4] to show that our approach can generalize to different application scenarios. Each dataset contains at least one protected feature. For all datasets, we follow [37] 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 [37]. The rest of the settings are similar to the image tasks. The results are obtained by averaging the fairness metrics on the test sets based on ten random initialization.

Income Prediction: The Adult dataset [29] is from the Census Bureau, and the task is to predict whether a given adult makes more than $\$50,000$ a year based on attributes such as education, hours of work per week, etc., for approximately $45,000$ individuals. In this experiment, *gender* (male or female) is used as the binary protected attribute. The computational results are presented in Table 4.

Recidivism Prediction: Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) is a commercial tool to assess a criminal defendant's likelihood to re-offend. The task is to predict the recidivism risk based on the features for defendants, including the criminal history, jail and prison time, and demographics. In this experiment, *gender* (male or female) are used as binary protected attributes and EOD is the fairness measure.. The computational results are presented in Table 5.

4.3 Discussions

In this experiment, we have several observations. Table 1 shows that our method works are superior to all the baselines significantly. Table 2 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. 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 [1]. 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.

We notice that the three recovering strategies show some performance difference particularly for vision dataset. The Prune+AdvFt strategy consistently outperforms the related baselines using the full model, which verifies the existence of the fair subnetwork. The best-performing grafting strategy is via external grafting. NetAug usually performs similar to the grafting strategy. For tabular data, there is no statistical difference between the naive strategy (i.e., Prune+Adv.Ft) and the complex strategies (i.e., grafting and NetAug). It should be mentioned that although grafting and NetAug sometimes yield better fair models, the training time is significantly longer than the Prune+Adv.Ft strategy. In most cases, the Grafting strategy and the NetAug strategy have comparable performance. We recommend the Grafting strategy for general purpose model debiasing considering the model and training complexity. For large-scale fairness problems, we expect NetAug is of some use since the network augmentation technique is designed to avoid under-fitting for tiny models. However, there still lacks such a benchmark to our best knowledge.

Table 6: Pruned results guided by different reconstructed fair models. For all entries, the value denotes accuracy/bias/final pruning rate (if applicable).

Ratio	Base	Teacher Strategy		
		Adv.Ft	ExtGraft	NetAug
80%	0.912/0.054	0.907/0.040/15%	0.905/0.048/14%	0.900/0.036/17%
90%	0.901/0.072	0.898/0.041/15%	0.899/0.056/16%	0.893/0.044/12%
99%	0.879/0.115	0.874/0.066/13%	0.891/0.082/15%	0.878/0.063/14%

4.4 Further Study: Fair Subnetworks in Unfair Models

The empirical results provide some evidence on the existence of fair subnetwork in a unfair model. A further question is whether a completely unfair model can be adjust to become fair? To answer it, we consider the following problem,

$$\min_{m} \; H(f_t^u(x, \mathcal{W} \odot M), f^r(x)) + \gamma H(f_{t-1}^u(x), f_t^u(x)), \tag{10}$$

where f^u is an unfair model, and f^r is a reconstructed fair model. t is the epoch. $H(\cdot)$ is the entropy function. The first term can be regarded as the knowledge distillation loss [23] with temperature 1.0. The second term is consistency regularization between epochs. We let the unfair model mimic the outputs of the reconstructed fair model instead of the fair sub-networks, since the fair subnetwork is sub-optimal before we apply the intra-process. During pruning, we fix model weights and only update the binary mask and ignore the pruning rate constraint in this problem to expand the possible search space of robust subnetworks. Table. 6 summarizes the results on CIFAR-10S. We alter the ratio of color and grayscale images, and compute the pruned models concerning the reconstructed fair models using different strategies. The results show that simply removing the corrupted weights is adequate to obtain a fair model with acceptable accuracy. We also notice that the pruning rate is merely around 12% $\sim$ 17%, even for a high imbalance ratio. This result indicate that most weights in a nonrobust deep neural network are robust or at least insensitive to unfairness.

5 Conclusion

In this paper, we propose a novel intra-processing fairness framework. Our framework includes two steps. First, we discover the fair sub-structure using model pruning techniques. Second, we propose several strategies to reconstruct the fair deep classification model. 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.

Acknowledgement

This work was partially supported by NSF IIS 1845666, 1852606, 1838627, 1837956, 1956002, 2217003.

References

1. Inherent trade-offs in the fair determination of risk scores. arXiv preprint arXiv:1609.05807 (2016)
2. Alvi, M., 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. pp. 0–0 (2018)
3. Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 233–242. JMLR. org (2017)
4. Bellamy, R.K., et al.: Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arXiv preprint arXiv:1810.01943 (2018)
5. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
6. Bower, A., Niss, L., Sun, Y., Vargo, A.: Debiasing representations by removing unwanted variation due to protected attributes. arXiv preprint arXiv:1807.00461 (2018)
7. Cai, H., Gan, C., Lin, J., Han, S.: Network augmentation for tiny deep learning. arXiv preprint arXiv:2110.08890 (2021)
8. Calders, T., Kamiran, F., Pechenizkiy, M.: Building classifiers with independency constraints. In: 2009 IEEE International Conference on Data Mining Workshops. pp. 13–18. IEEE (2009)
9. Chu, X., Zhang, B., Xu, R.: Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12239–12248 (2021)
10. Chu, X., Zhou, T., Zhang, B., Li, J.: Fair darts: Eliminating unfair advantages in differentiable architecture search. In: European conference on computer vision. pp. 465–480. Springer (2020)
11. Dastin, J.: Amazon scraps secret ai recruiting tool that showed bias against women. Reuters (2018)
12. Denton, E., Hutchinson, B., Mitchell, M., Gebru, T.: Detecting bias with generative counterfactual face attribute augmentation (2019)
13. Dwork, C., et al.: Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference. pp. 214–226 (2012)
14. Fitzpatrick, T.B.: The validity and practicality of sun-reactive skin types i through vi. Archives of dermatology **124**(6), 869–871 (1988)
15. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=rJl-b3RcF7`
16. Ganjdanesh, A., Gao, S., Huang, H.: Interpretations steered network pruning via amortized inferred saliency maps. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
17. Gwilliam, M., et al.: Rethinking common assumptions to mitigate racial bias in face recognition datasets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4123–4132 (2021)
18. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)

19. Han, S., Pool, J., Narang, S., Mao, H., Tang, S., Elsen, E., Catanzaro, B., Tran, J., Dally, W.J.: Dsd: regularizing deep neural networks with dense-sparse-dense training flow (2016)
20. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems **28** (2015)
21. Hardt, M., Others: Equality of opportunity in supervised learning. In: Advances in neural information processing systems (2016)
22. He, K., zheng: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
24. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
25. Huang, G., et al.: Condensenet: An efficient densenet using learned group convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2752–2761 (2018)
26. Kamiran, F., Karim, A., Zhang, X.: Decision theory for discrimination-aware classification. In: 2012 IEEE 12th International Conference on Data Mining. pp. 924–929. IEEE (2012)
27. Kärkkäinen, K., Joo, J.: Fairface: Face attribute dataset for balanced race, gender, and age. arXiv preprint arXiv:1908.04913 (2019)
28. Kim, M., et al.: Fairness through computationally-bounded awareness. In: Advances in Neural Information Processing Systems. pp. 4842–4852 (2018)
29. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: Kdd. vol. 96, pp. 202–207 (1996)
30. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
31. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. International Conference on Learning Representations (ICLR) (2017)
32. Liu, Z., et al.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
33. Meng, F., Cheng, H., Li, K., Xu, Z., Ji, R., Sun, X., Lu, G.: Filter grafting for deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6599–6607 (2020)
34. Mukherjee, D., et al.: Two simple ways to learn individual fairness metrics from data. arXiv preprint arXiv:2006.11439 (2020)
35. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
36. Pleiss, G., et al.: On fairness and calibration. Advances in neural information processing systems **30** (2017)
37. Savani, Y., White, C., Govindarajulu, N.S.: Intra-processing methods for debiasing neural networks. Advances in Neural Information Processing Systems **33**, 2798–2810 (2020)
38. Shen, C., Wang, X., Yin, Y., Song, J., Luo, S., Song, M.: Progressive network grafting for few-shot knowledge distillation. arXiv preprint arXiv:2012.04915 (2020)
39. Vigdor, N.: Apple card investigated after gender discrimination complaints. The New York Times (2019)

40. Wang, M., et al.: Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In: Proceedings of the ieee/cvf international conference on computer vision. pp. 692–702 (2019)
41. Wang, Z., Qinami, K., Karakozis, I.C., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8919–8928 (2020)
42. Warde-Farley, D., Goodfellow, I.J., Courville, A., Bengio, Y.: An empirical analysis of dropout in piecewise linear networks. arXiv preprint arXiv:1312.6197 (2013)
43. Wilson, B., Hoffman, J., Morgenstern, J.: Predictive inequity in object detection. arXiv preprint arXiv:1902.11097 (2019)
44. Yang, F., Cisse, M., Koyejo, O.O.: Fairness with overlapping groups; a probabilistic perspective. Advances in Neural Information Processing Systems **33** (2020)
45. Yang, L., et al.: Condensenet v2: Sparse feature reactivation for deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3569–3578 (2021)
46. Yu, J., Huang, T.S.: Universally slimmable networks and improved training techniques. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1803–1811 (2019)
47. Yu, J., Yang, L., Xu, N., Yang, J., Huang, T.: Slimmable neural networks. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=H1gMCsAqY7`
48. Yurochkin, M., et al.: Training individually fair ml models with sensitive subspace robustness. In: International Conference on Learning Representations (2019)
49. Zafar, M.B., et al.: Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In: Proceedings of the 26th international conference on world wide web. pp. 1171–1180 (2017)
50. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340 (2018)
51. Zhang, X., zheng: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
52. Zhang, Y., Gao, S., Huang, H.: Exploration and estimation for model compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 487–496 (2021)
53. Zhang, Y., Luo, L., Huang, H.: Unified fairness from data to learning algorithm. In: 2021 IEEE International Conference on Data Mining (ICDM). pp. 1499–1504. IEEE (2021)
54. Zhao, J., et al.: Men also like shopping: Reducing gender bias amplification using corpus-level constraints. arXiv preprint arXiv:1707.09457 (2017)