Fairness via Representation Neutralization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing bias mitigation methods for DNN models primarily work on learning debiased encoders. This process not only requires a lot of instance-level annotations for sensitive attributes, it also does not guarantee that all fairness sensitive information has been removed from the encoder. To address these limitations, we explore the following research question: *Can we reduce the discrimination of DNN models by only debiasing the classification head, even with biased representations as inputs?* To this end, we propose a new mitigation technique, namely, Representation Neutralization for Fairness (RNF) that achieves fairness by debiasing only the task-specific classification head of DNN models. To this end, we leverage samples with the same ground-truth label but different sensitive attributes, and use their neutralized representations to train the classification head of the DNN model. The key idea of RNF is to discourage the classification head from capturing spurious correlation between fairness sensitive information in encoder representations with specific class labels. To address low-resource settings with no access to sensitive attribute annotations, we leverage a bias-amplified model to generate proxy annotations for sensitive attributes. Experimental results over several benchmark datasets demonstrate our RNF framework to effectively reduce discrimination of DNN models with minimal degradation in task-specific performance.

1 Introduction

Deep neural networks (DNNs) have made significant advances in recent times [1, 2, 3, 4], and have been deployed in many real-world applications. However, DNNs often suffer from biases and show discrimination towards certain demographics, especially in high-stake applications, such as criminal justice, employment, loan approval, credit scoring, etc [5, 6, 7]. For example, the recidivism algorithm is likely to associate African-American offenders with higher risk scores compared to Caucasians while having a similar profile [8]. This brings significant harm to both society and individuals leading to recent focus on mitigation techniques to alleviate the adverse effects of DNN biases.

Existing debiasing methods usually work on *learning debiased representations* at the encoder-level. One representative family of methods perform mitigation by explicitly learning debiased representations, either through adversarial learning [9, 10, 11] or invariant risk minimization [12, 13]. Another family of methods [14, 15, 16] implicitly learn debiased representations by incorporating explanation during model training to suppress it from paying high attention to biased features in the original input. Essentially, the above methods aim to remove the bias from deep representations.

Learning debiased representations is a technically challenging problem. Firstly, it is hard to remove all fairness sensitive information in the encoder. The suppression of fairness sensitive information might also remove useful information that is task relevant. Secondly, most existing debiasing methods assume access to additional meta-data such as fairness sensitive attributes and a lot of annotations corresponding to the protected groups to guide the learning of debiased representations. However, such resources are expensive to obtain, if not unavailable, for most real world applications.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

To address these limitations, we explore the following research question: *Can we reduce the discrimination of DNN models by only debiasing the task-specific classification head, even with a biased representation encoder?* Our work is motivated by the empirical observation that standard training can result in the classification head capturing spurious correlation between fairness sensitive information and specific class labels. Some recent works [17, 18, 19] have explored such spurious or shortcut learning behavior of DNNs in various applications. To this end, we propose the RNF (Representation Neutralization for Fairness) framework for mitigation, motivated by the Mixup work [20, 21]. We first train a biased teacher network via standard cross entropy loss. In the second stage, we freeze the representation encoder of the biased teacher, and only update the classification head via representation neutralization. This discourages the model from associating biased features with specific class labels, and enforces the model to focus more on task relevant information. To address low-resource settings, our RNF framework does not require any access to the protected attributes during training. To this end, we train a bias-amplified model using generalized cross entropy loss that is used to generate proxy annotations for sensitive attributes. Experimental results over several benchmark tabular and image datasets demonstrate our RNF framework to significantly reduce discrimination of DNN models with minimal degradation of the task performance. The major contributions of our work can be summarized as follows:

- We analyze bias propagation from the encoder representations to the final task-specific layer demonstrating that DNN models heavily rely on spurious correlations for prediction.
- We introduce RNF, a bias mitigation framework for DNN models via representation neutralization. Our RNF framework achieves mitigation without any access to instance-level sensitive attribute annotations, and instead relies on self-generated proxy annotations.
- Experimental results on several benchmark datasets demonstrate the effectiveness of our RNF framework via debiasing only the classification head while using biased representations as input. Additionally, we show RNF to be complementary to existing methods that learn debiased encoders and can be further improved within our framework.

2 Representation Neutralization for Fairness

In this section, we first analyze the task-specific classification head of a DNN to examine how bias is propagated from the encoder representation layer to the task-specific output layer. We empirically demonstrate the spurious correlation between fairness sensitive information in representations with specific class labels (Section 2.1). Based on the observation, we introduce the Representation Neutralization for Fairness (RNF) framework to debias DNN models (Section 2.2). Finally, we propose an approach to generate proxy annotations for sensitive attributes, enabling the RNF framework to be applicable to low-resource settings with no access to sensitive attribute annotations (Section 2.3).

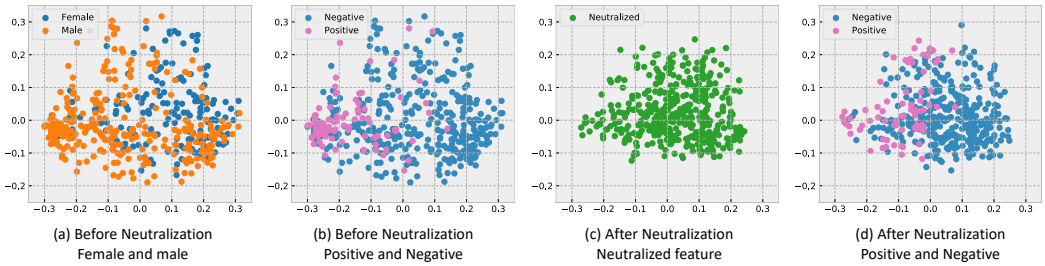
We first introduce the notations used in this work. Let $\mathcal{X} = \{x_i, y_i, a_i\}, i \in 1, ..., N$ be the training set, where x_i is the input feature, y_i denotes the ground truth label, and a_i represents the sensitive attribute (e.g., gender, race, age). For ease of notation, in the following sections, we consider binary sensitive attributes. Nevertheless, *our proposed mitigation framework can also be applied to non-binary sensitive attributes (e.g., age).*

Consider the classification model $f(x, \theta) = c(g(x))$, parameterized with θ as the model parameters. Here $g(x) : \mathcal{X} \rightarrow \mathcal{Z}$ represents the feature encoder, and $g(x) = z$ is the representation for x obtained from a DNN model. The predictor $c(z) : \mathcal{Z} \rightarrow \mathcal{Y}$ is the *multi-layer* classification head. It takes the encoded representation z as input and maps it to softmax probability. The final class prediction is denoted by $\tilde{y} = \arg\max c(z)$. In this work, we aim to reduce the discrimination of DNN models by *only debiasing the classification head* $c(z)$, with the biased representation encoder $g(x)$ as input.

2.1 Analysis of the Classification Head

In this section, we examine how bias manifests in the representation space $\mathcal{Z}$ as well as how the classification head $c(z)$ obtained with standard training scheme propagates bias from the representation layer to the model output layer. To this end, we train a biased network $f_T(x)$ via standard cross entropy loss, where the following experiment is performed using the Adult dataset [22].

Representation Probing Analysis. For the Adult training set, we generate representation vectors for 500 training samples using the biased network $f_T(x)$ and project them in 2D for visualization. To



(a) Before Neutralization
Female and male

(b) Before Neutralization
Positive and Negative

(c) After Neutralization
Neutralized feature

(d) After Neutralization
Positive and Negative

Figure 1: Representation analysis of z using Kernel PCA. (a) Protected attribute a (i.e., gender) is a discriminative feature. In this plot, the male group primarily lies in the lower left interval, whereas the female group is located primarily on the upper right interval. (b) Predicted positive label and negative task-label distribution. (c) Representation neutralization (please refer to Sec. 2.2) to reduce the discriminative power of a for dimensions relevant to the sensitive information, comparing to the distribution in (a). (d) Neutralization still preserves the useful task relevant information.

this end, we utilize the Kernel Principal Component Analysis (KPCA) [23] with a sigmoid kernel, which is a tool to visualize high-dimensional data. Since the classification head $c(z)$ contains multiple non-linear layers, we choose kernel PCA instead of a linear dimensionality reduction method such as linear PCA. The visualization is shown in Figure 1 (a). The plot depicts that the low dimensional projection separates the two protected groups in two areas, where the male group is primarily located in the lower left area, whereas the female group primarily occupies the upper right area. Comparing the task-label y distribution in Figure 1 (b) with the protected group distribution in Figure 1 (a), we observe that the protected attribute information is a discriminative feature that could be exploited by the task classification head for prediction.

Role of the Biased Classification Head. The above demonstrative analysis indicates that the model representation captures both useful task relevant classification information as well as bias information from protected attributes. Specifically, the model captures strong correlation between the fairness sensitive information and the class labels. On analyzing the data distribution, we observe this to be an artifact of the conditional label distribution with respect to the sensitive attributes being skewed. The model relies on this shortcut for prediction, resulting in bias amplification. We observe the male neurons to positively correlate to the desirable label (also refer to the experimental analysis in Sec. 3.3), whereas the female neurons positively correlate to undesirable label. This depicts a *spurious correlation* between sensitive information with certain class labels in the model.

Our Motivation. Based on the above empirical observations, we propose to neutralize the training samples (Fig 1 (c)) so as to reduce the discriminative power of the fairness sensitive information, while at the same time preserving task relevant information (Fig 1 (d)). With the neutralized training data, we propose to re-train the classification head. Our goal is to adjust the decision boundary to implicitly de-correlate the fairness sensitive information in the representation space with class labels.

2.2 Representation Neutralization for Debiasing Classification Head

Based on the aforementioned empirical observations, in this section we propose a simple yet effective bias mitigation framework via Representation Neutralization for Fairness (RNF). RNF does not require any prior knowledge about existing bias in the representation space; nor does it require any knowledge about specific dimension(s) encoding the sensitive attributes – making it widely useful for arbitrary applications. Our goal is to encourage the model to ignore the sensitive attributes and instead focus more on task relevant information.

RNF is implemented in two steps (see Figure 2). In the first step, we train the model using cross entropy loss, and obtain a biased teacher network $f_T(x)$ (see Figure 2 (a)). During the second step, we freeze the encoder $g(x)$ for $f_T(x)$, and use it as our backbone encoder for learning representations. We then finetune only the classification head $c(z)$ using feature neutralization (see Figure 2 (b)).

Representation Neutralization. To this end, while training the classification head, for an input sample $\{x_1, y, a_1\}$, we *randomly select another sample* $\{x_2, y, a_2\}$, with the same class label y but a different sensitive attribute a_2 compared to a_1 in the input sample. Now we compute the corresponding

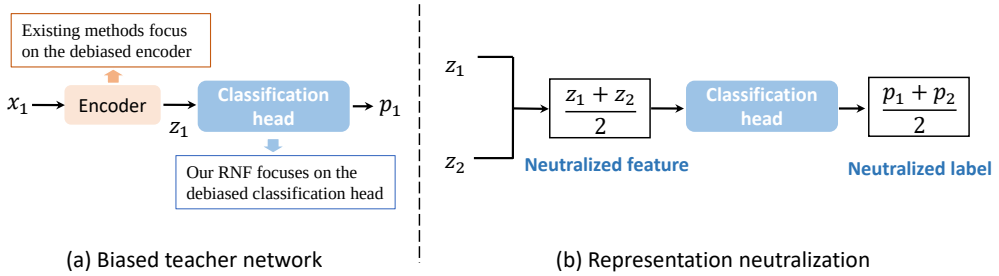


Figure 2: Debiasing with representation neutralization. (a) We first train a biased teacher network using only cross entropy loss. For two inputs x_1 and x_2 that with the same class label y and different sensitive attribute a, we obtain the representations z_1 and z_2, and softened probabilities p_1 and p_2. (b) We freeze parameters of the biased encoder and only finetune the classification head using the neutralized representation $\frac{z_1+z_2}{2}$ as input, and softened probability $\frac{p_1+p_2}{2}$ as supervision signal.

representations $z_1 = g(x_1)$ and $z_2 = g(x_2)$ and finetune the classification head using the neutralized representation $z = \frac{z_1+z_2}{2}$ as input. For the supervision label y for the classification head, we utilize the neutralized soft probability $y = \frac{p_1+p_2}{2}$ after temperature scaling obtained as follows. Given the logit vector z_1 for input x_1, the probability for class i is computed as $p_{1,i} = \frac{\exp(z_{1,i}/T)}{\sum_j \exp(z_{1,j}/T)}$, where $T \geq 1$. This can be regarded as a form of knowledge distillation [24], where $T > 1$ softens the softmax score. A larger temperature prevents the model from assigning over-confident prediction probabilities. A special case for RNF is when $T = 1$, where p_1 and p_2 are the standard softmax probabilities obtained from the biased teacher network $f_T(x)$.

We use the knowledge distillation loss. In particular, the mean squared error (MSE) loss is used as a distance-based metric to measure the similarity between model prediction and the supervision signal.

$$\mathcal{L}_{\text{MSE}} = (\hat{y}_i - y)^2 = \{c(\frac{1}{2}z_1 + \frac{1}{2}z_2) - (\frac{1}{2}p_1 + \frac{1}{2}p_2)\}^2. \tag{1}$$

where, c is the classification head to project representations to softmax prediction probability.

There are two main benefits of the aforementioned training scheme. From the input perspective, the neutralization of representations suppresses the model from capturing the spurious correlation between fairness sensitive information in the representation with the class labels. From the output perspective, the softened label encourages the model to assign similar predictions to different sensitive groups. Optimizing Eq. (1) can lead to reduced generalization gap between the two groups.

Theorem 1 *Given a well-trained representation encoder $g(x) = z$ satisfying $||z_1 - z_2||_2 \leq \lambda_z |p_1 - p_2|$ for $x_1, x_2 \in \mathcal{X}$ and a bounded loss function $L(c(z_i), p_i) = (1 - p_i) l(c(z_i), y = 0) + p_i l(c(z_i), y = 1)$, where $l(c(z_i), y = j) \leq \epsilon_L$ for $x_i \in \mathcal{X}$ and $j = 0, 1$; if the classification head c minimizes the loss between neutralized representation and soft probabilities, i.e. $\left|\left|\nabla_z L(c(z), p)\right|_{z=\frac{z_1+z_2}{2}, p=\frac{p_1+p_2}{2}}\right|\right|_2 \leq \epsilon_c$, where $z = g(x)$, $x_1 \sim P(x_1 \mid a_i = 0)$, $x_2 \sim P(x_2 \mid a_i = 1)$, $|p_1 - p_2| \leq \epsilon_p$, the gap of generalization loss between groups $a = 0$ and $a = 1$ is bounded by:*

$$\left|\mathbb{E}_{x_i \sim P(x_i|a_i=0)} L(c(z_i), p_i) - \mathbb{E}_{x_j \sim P(x_j|a_j=1)} L(c(z_j), p_j)\right| \leq \epsilon_p(\lambda_z \epsilon_c + \epsilon_L) \tag{2}$$

For detailed proof, please refer to Section A in Appendix.

Smoothing Neutralization. To further enforce the model to ignore sensitive attributes, we construct augmented training samples using a hyper-parameter λ to control the degree of neutralization of the samples $\{z_1, p_1, y\}$ and $\{z_2, p_2, y\}$. The augmented neutralized sample is given by $z = \lambda z_1 + (1 - \lambda) z_2$, $\lambda \in [\frac{1}{2}, 1)$. We encourage the classification head to give similar prediction scores for the augmented and the neutralized sample (with $\lambda = \frac{1}{2}$). The regularization loss is given by:

$$\mathcal{L}_{\text{Smooth}} = \sum_{\lambda \in [\frac{1}{2}, 1)} |c(\lambda z_1 + (1 - \lambda) z_2) - c(\frac{1}{2}z_1 + \frac{1}{2}z_2)|. \tag{3}$$

By varying λ we control the degree of sensitive information for the augmented samples. We linearly combine the MSE loss in Eq. (1) with the regularization term as follows:

$$\mathcal{L}_{\text{MSE}} = \mathcal{L}_{\text{MSE}} + \alpha \mathcal{L}_{\text{Smooth}}. \tag{4}$$

We finetune the classification head using the loss function in Eq. (4). Eventually we combine the original encoder of $f_T(x)$ and finetuned classification head as the debiased student network $f_S(x)$. The teacher $f_T(x)$ is later discarded and the debiased student network $f_S(x)$ is used for prediction.

2.3 Generating Proxy Annotations for Sensitive Attributes

The aforementioned feature neutralization is limited in that it requires instance-level sensitive attribute annotations $\{a_i\}_{i=1}^N$ for all training samples. Such resource-extensive annotations are difficult to obtain for many practical applications particularly due to the nature of the sensitive attributes. To address this limitation, we propose a method to generate proxy annotations $\{\hat{a}_i\}_{i=1}^N$ for the sensitive attributes based on the model uncertainty. The key idea is that a biased model generates over-confident predictions for one demographic group, while giving much lower scores for the alternative group.

To better facilitate the model to generate uncertainty scores, we train another biased model by intentionally amplifying the bias via generalized cross entropy loss (GCE) [25]. The bias-amplified model is denoted as $f_B(x)$ and the loss function is given as follows:

$$\text{GCE}(f(x;\theta), y) = \frac{1 - f_y(x;\theta)^q}{q}, \tag{5}$$

where $f_y(x;\theta)$ denotes the output probability for ground truth label y. The hyper-parameter $q \in (0, 1]$ controls the degree of bias amplification. When $\lim_{q \to 0}$, the GCE loss in Equation (2) approaches $-\log p$ which is equivalent to standard cross entropy loss. The core idea is that for more biased samples, i.e., samples with larger $f_y(x;\theta)$ value, the model assigns higher weights f_y^q while updating the gradient.

$$\frac{\partial\, \text{GCE}(p, y)}{\partial \theta} = f_y^q \frac{\partial\, \text{CE}(p, y)}{\partial \theta}. \tag{6}$$

In this setup, the model $f_B(x)$ learns from bias-amplified samples compared to the model $f_T(x)$ trained with standard cross entropy loss.

The confidence score of $f_B(x)$ is used to indicate whether a sample belongs to a privileged or unprivileged group. Specifically, for a desired ground truth label, samples with over-confident scores are grouped into the privileged group, whereas subsets of samples with low prediction scores are grouped into the unprivileged group. In contrast, for the undesired ground truth label, samples with over-confident scores are grouped into the unprivileged group, and vice versa. Based on this criterion, we generate proxy sensitive attribute annotation $\hat{a}$ for each training sample x to split samples into two groups, which are subsequently used for feature neutralization.

The overall process of our RNF mitigation framework is given in Algorithm 1, which contains two stages. In the first stage, we train the biased teacher network $f_T(x)$ and the bias-amplified network $f_B(x)$. In the second stage, we first use $f_B(x)$ to generate proxy sensitive attribute annotations for all training samples. We use the ratio γ to partition the training set to generate proxy annotations for protected attributes that are subsequently used for feature neutralization. Note that the ratio γ is determined by the fairness-accuracy trade-off on the validation set. Then we use representation neutralization and the loss function in Eq. (4) to retrain the classification head of $f_T(x)$. Eventually, we combine the original encoder $g(x)$ of $f_T(x)$ and the refined classification head $c(z)$ to give us the debiased student network $f_S(x)$.

Algorithm 1: RNF mitigation framework.

Input: Training data $D = \{(x_i, y_i)\}_{i=1}^N$.
Set hyperparameter q, α.
while *first stage* **do**
  Train teacher network $f_T(x)$ and bias-amplified network $f_B(x)$.
while *second stage* **do**
  Determine the splitting threshold γ for $f_B(x)$;
  Calculate proxy annotation $\hat{a}_i$ for each training sample $\{(x_i)\}_{i=1}^N$ based on γ and $f_B(x)$;
  Use loss function in Eq. (4) to retrain classification head $c(z)$ of $f_T(x)$
Output: The debiased student network $f_S(x)$.

3 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our RNF framework.

3.1 Experimental Setup

3.1.1 Fairness Metrics, Benchmark Datasets and Baselines

We use two group fairness metrics: Demographic Parity [26] and Equalized Odds [27]. Demographic Parity (DP) measures the ratio of the probability of favorable outcomes between unprivileged and privileged groups: DP $= \frac{p(\hat{y}=1|a=0)}{p(\hat{y}=1|a=1)}$. Equalized Odds ($\Delta$EO) require favorable outcomes to be independent of the protected class attribute a, conditioned on the ground truth label y. Specifically, it calculates the summation of the True Positive Rate difference and False Positive Rate difference:

$$\Delta\text{EO} = \{P(\hat{y}=1 \mid a=0, y=1) - P(\hat{y}=1 \mid a=1, y=1)\} + \{P(\hat{y}=1 \mid a=0, y=0) - P(\hat{y}=1 \mid a=1, y=0)\}. \quad (7)$$

Under the above metrics, it is desirable to have a DP value closer to 1 and ΔEO value closer to 0.

We use two benchmark tabular datasets and one image dataset to evaluate the effectiveness of RNF. For the Adult income dataset (**Adult**), the goal is to predict whether a person's income exceeds \$50K/yr [22]. We consider *gender* as the protected attribute where vanilla trained models depict bias towards the female group. For the Medical Expenditure dataset (**MEPS**), we consider two groups *white* and *non-white* [28]. Here the task is to predict whether a person would have a 'high' utilization, where vanilla DNN shows discrimination towards the non-white group. The CelebFaces Attributes (**CelebA**) dataset is used to predict whether the hair in an image is wavy or not [29]. We consider two groups *male* and *female*, where vanilla trained models show discrimination towards the male group. We split all datasets into three subsets with statistics reported in Table 1. More details of the datasets are included in Sec. B in the Appendix.

Table 1: Dataset statistics.

	Adult	**MEPS**	**CelebA**
# **Training**	33120	11362	194599
# **Validation**	3000	1200	4000
# **Test**	9102	3168	8000

We compare two variants of our framework **RNF** (using *proxy* sensitive attribute annotations) and **RNF$_{\mathbf{GT}}$** (using *ground truth* sensitive attribute annotations) against baselines such as DNNs trained using only cross entropy loss (referred as **Vanilla**) and two regularization based mitigation methods, namely, adversarial training (**Adversarial**) [30] and Equalized Odds Regularization (**EOR**) [31]. Among them, the Adversarial method achieves fairness via learning debiased representations, whereas EOR directly optimizes the EO metric in Eq. (7). All three baselines control the fairness-accuracy trade-off via hyper-parameters. More details on the baselines are included in Sec. C in the Appendix.

3.1.2 Implementation Details

For the image classification task, we use ResNet-18 [1] (we add one more fully connected layer). We set the representation encoder $g(x)$ as the convolutional layers and use the remaining two fully connected layers as the classification head $c(z)$. For tabular datasets, we use a three-layer MLP (multilayer perceptron) as the classification model, where the first layer is set as the encoder and the remaining two layers are used as the classification head. Dropout is used for the first two layers with the dropout probability fixed at 0.2. We use the same batch size of 64 for tabular datasets and 390 for the image dataset. The hyper-parameters (e.g., learning rate) are determined based on the model performance on the validation set, and early-stopping based on validation loss is used to avoid overfitting. As the random seeds lead to variance in the accuracy and fairness metrics, we train the model three times with different seeds and report the average result. The hyper-parameter q in Eq. (5) and the threshold γ are set as $(0.2, 0.993)$, $(0.6, 0.993)$, $(0.3, 0.990)$ for Adult, MEPS and CelebA datasets respectively. The optimal temperature T is set as 2.0, 5.0, 2.0 for Adult, MEPS and CelebA datasets respectively. For Eq. (3), we sample λ from the list $[0.6, 0.7, 0.8, 0.9]$.

3.2 Mitigation Performance Analysis

We compare the mitigation performance of RNF (with proxy attribute annotations) and RNF$_{\text{GT}}$ (with ground-truth attribute annotations) with other competing methods and illustrate their fairness-accuracy curves for the three datasets in Figure 3. The hyper-parameter α in Eq. (4) controls the trade-off between accuracy and fairness for RNF. For Adversarial and EOR, we vary their regularization weights to obtain the corresponding performance curves. Overall, we make the following observations.

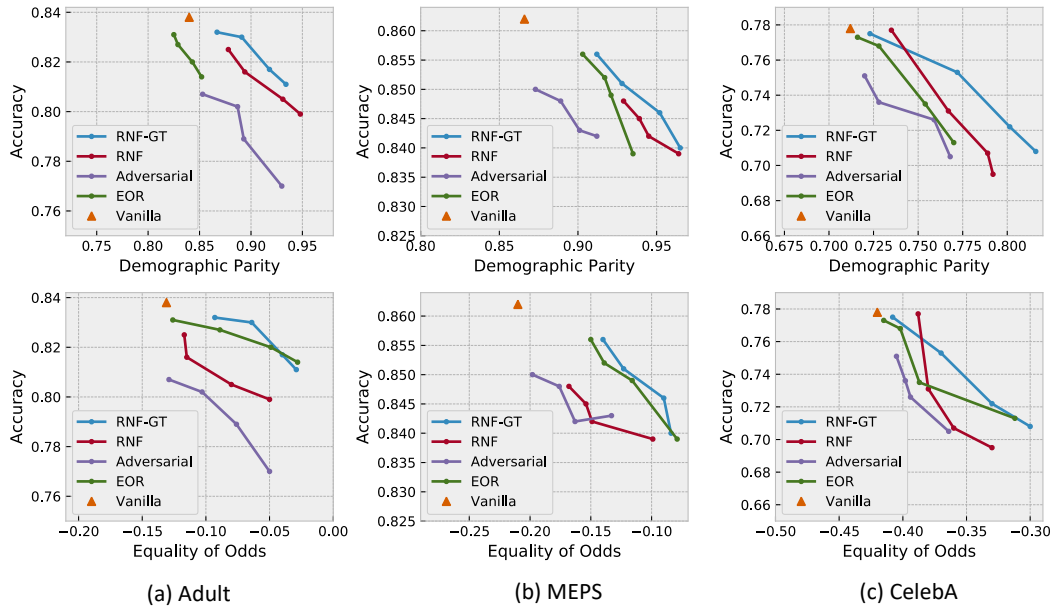


Figure 3: The fairness-accuracy curve comparison of RNF and other baselines. The first and second row depict the DP accuracy and ΔEO accuracy trade-off curves, respectively.

- Even though RNF does not rely on annotations for the sensitive attributes, it outperforms baseline methods with access to such information, e.g, Adversarial training. This makes RNF readily usable for real-world applications where protected attributes are not available in the training set.
- RNF_{GT} improves mitigation performance over RNF by 10% on an average across all datasets and metrics, thereby, demonstrating the benefit of using ground truth sensitive attribute annotations. In some instances like the CelebA dataset, RNF marginally outperforms RNF_{GT}.
- The soft labels of RNF and RNF_{GT} (obtained using a higher temperature T) discourage the model to assign overconfident predictions, thereby, suppressing it from capturing spurious correlation between fairness sensitive information and class labels.
- Direct optimization of the equality of odds metric (i.e., EOR) achieves comparable performance to RNF_{GT} for all the datasets. However, it has limited improvement in terms of the demographic parity metric. Note that EOR requires instance-level annotations for the protected attributes.
- We observe that Adversarial training also performs effective mitigation by learning debiased representations. However, this happens at the expense of a higher accuracy drop for the task performance. This likely results from the loss of task relevant information while suppressing sensitive information from the representations. Additionally, we observe adversarial training to be unstable, especially for relatively complex task like image classification.

3.3 Classification Head Analysis

In this section, we use explainability and an auxiliary prediction task to analyze the classification head $c(z)$ for the Adult dataset. Particularly, we leverage explainability as a debugging tool to analyze the attention difference between Vanilla and RNF model with respect to the representations.

Auxiliary Sensitive Attribute Prediction Task. We perform a representation analysis using an auxiliary prediction task. To this end, we train another linear classifier to predict sensitive attributes using the biased representation $g(x) = z$ as input and the sensitive attribute annotations $\{a_i\}_{i=1}^{N}$ as the supervision signal. The linear classifier is denoted by $L_{\text{SENS}}(z) = Wz + b$, where W and b represent the weight matrix and bias for the linear classifier respectively. The weight matrix W can be used to measure the degree of bias in each dimension of the representation z.

Explanation Analysis. We use post-hoc explainability to analyze the contribution of the classification head $c(z)$. Our goal is to figure out the contribution of each dimension within the biased representation $g(x) = z$ towards the model prediction $f(x, \theta) = c(g(x))$. We train a linear classifier $L_{\text{explan}}(z) = Wz + b$ to mimic the decision boundary of the multi-layer classification head $c(z)$.

We compare the weight matrix of the two linear classifiers $L_{\text{SENS}}(z)$ and $L_{\text{explan}}(z)$ using cosine similarity. For models trained using Adult dataset, we extract the weight matrix corresponding to the protected attribute *Male* and task label *Positive*, and then we calculate the cosine similarity between the Male vector and the Positive vector. This follows from our observation in Figure 1 that the vanilla model makes use of male relevant information to make positive predictions. For the Vanilla model and RNF models listed in Figure 3 (a), we calculate the cosine similarity and report the DP-Similarity performance in Figure 4. We observe that RNF dramatically reduces the cosine similarity between positive predictions and male relevant information compared to Vanilla (from 0.272 to 0.075), by adjusting the decision boundary. This eventually helps the head $c(z)$ to shift its attention from fairness sensitive information to task relevant information.

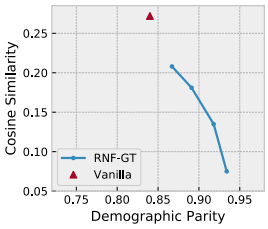


Figure 4: Analysis for the classification head.

3.4 Hyper-parameter and Ablation Analysis

Effectiveness of GCE Loss. We conduct ablation studies via MEPS dataset to examine the GCE loss in terms of generating proxy annotations for the protected attributes. The results are shown in Table 2. Firstly, we compare the fairness performance of GCE loss with standard CE loss (Vanilla). As expected, we observe that GCE is more biased than Vanilla in terms of the fairness metrics with 0.2% accuracy difference. Secondly, we compare RNF with several of its variants: 1) replacing GCE loss with cross entropy (CE) loss to generate proxy labels, and 2) using random annotations for the protected attributes. After switching to the CE loss, the partition ratio γ to separate the two protected groups is 0.9 (see Algorithm 1). For a fair comparison, except for the protected attribute annotations, we use the same set of hyper-parameters for different variants. Even though the same level of regularization is performed for the three RNF variants, RNF-GCE has a higher mitigation performance than RNF-CE with 3.0% DP improvement and only 0.3% accuracy difference, thereby, demonstrating the effectiveness of GCE in terms of separating the protected groups.

Table 2: Ablation analysis.

	MEPS		
Models	Accuracy	DP	ΔEO
Vanilla	0.862	0.866	-0.210
GCE	0.860	0.839	-0.249
RNF-GCE	0.839	0.964	-0.099
RNF-CE	0.842	0.936	-0.128
RNF-Random	0.856	0.902	-0.163

Varying Layers for the Classification Head. We also study the effect of an important hyper-parameter of our framework in terms of which layers to use for the encoder. Our classification model $f(x)$ contains the encoder $g(x)$ and task predictor $c(z)$. In this experiment, we vary the depth of the representation layer to examine the effect of layer selection. In particular, we investigate the question: *can we debias only the very last layer, with the remaining layers as the biased encoder?* We report the results for MEPS and CelebA datasets in Table 3, where the second row depicts the result of debiasing only the last layer. On MEPS dataset, RNF-Last has better performance in terms of the two fairness metrics with only 0.2% accuracy difference. Similarly, for CelebA dataset, with 2.7% accuracy difference, RNF-Last has better fairness performance (with 13% DP improvement). This demonstrates that debiasing only the last layer can achieve similar performance compared to debiasing the last several layers.

Table 3: Varying the layers for classification head.

	MEPS			CelebA		
Models	Accuracy	DP	ΔEO	Accuracy	DP	ΔEO
RNF	0.839	0.964	-0.099	0.668	0.836	-0.290
RNF-Last	0.837	0.971	-0.076	0.641	0.966	-0.052

3.5 Representation Neutralization with Debiased Encoder

In the discussions so far, we reported the performance of RNF while debiasing only the classification head. In this section, we analyze the impact of RNF built on top of a debiased encoder. We first use Adversarial or EOR training to learn a debiased encoder – which is subsequently used as the backbone encoder for updating the classification head using RNF. This experiment is performed on the MEPS dataset where both Adversarial and EOR methods achieve competitive performance. We use the same hyper-parameters for different RNF variants (with α fixed at 0.2) and *report a single point in the fairness-accuracy curve*. The results are shown in Table 4. We

Table 4: RNF with debiased encoder.

	MEPS		
Models	Accuracy	DP	ΔEO
Vanilla	0.862	0.866	-0.210
RNF	0.839	0.964	-0.099
RNF_EOR	0.834	0.980	-0.049
RNF_Adversarial	0.826	0.971	-0.085

observe RNF_EOR to achieve better fairness performance over RNF, with DP metric improvement of 1.6% and ΔEO metric moves closer to 0. However, such improvement in the fairness metrics incur some loss in the task performance – where the accuracy reduces by 0.5%. We observe a similar trend with the combination of RNF and Adversarial, where the joint combination improves the fairness metrics DP and ΔEO. Similar to the previous case, this fairness improvement is achieved at the expense of some task performance degradation, where the accuracy drops by 1.3%. This indicates that our RNF is complementary to using a debiased encoder where the joint combination performs better than either of them in terms of the fairness metrics with some loss in task performance.

4 Related Work

We briefly review bias mitigation and broader robustness literature which are most relevant to ours.

Bias Mitigation. Recent studies have indicated that DNN models exhibit social bias towards certain demographic groups. This has led to increased attention to bias mitigation techniques in recent times [32, 33, 34, 35, 36]. Existing mitigation methods can be generally grouped into three broad categories. The first one is based on adversarial training [9, 10, 11]. It leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit. However, this method assumes that the sensitive attribute annotations are known, and uses those annotations to learn debiased representations. The second representative family of mitigation methods is based on explainability [14, 15, 16, 37]. These methods require fine-grained feature-level annotations to specify which subset of features are fairness sensitive. The third category falls under the umbrella of causal fairness [38, 12, 13]. The main idea is to enforce the model to concentrate more on task relevant causal features, and getting rid of superficial correlations [39]. This results in the model capturing debiased representations. For instance, [40] minimizes the correlation between sentence representations and bias words using a contrastive learning framework. Different from the above-mentioned methods that aim to train debiased representations, our work aims to *suppress the classification head* from capturing spurious correlations between sensitive attributes and class labels.

Spurious Learning in DNNs. Our work is motivated by the observation that DNNs with standard training are prone to exploit spurious correlations (or shortcuts in the dataset) for prediction [17]. Beyond algorithmic discrimination, recent studies show that *spurious learning* [18] can result in other undesirable consequences like poor generalization and adversarial vulnerability. Specifically, this leads to a high performance degradation for previously unseen inputs, especially for those data beyond held-out test set. Representative tasks depicting such behavior include reading comprehension [41], natural language inference [19], visual question answering[42] and deepfake detection [43].

The most similar work to ours also focuses on refining the classification head [40]. Similar to the aforementioned solutions, this work also heavily relies on sensitive attribute annotations in the training set. This limits the application scenarios of the mitigation algorithms, especially for many real-world applications without readily available annotations dealing with sensitive attributes. Our representation neutralization framework achieves comparable or better performance to such techniques without relying on such sensitive annotations by leveraging proxy annotations obtained from a bias-intensified version of our framework, thereby, making it broadly applicable to arbitrary real-world applications.

5 Conclusions and Future Work

In this work, we demonstrate that even when input representations are biased, we can still improve fairness by debiasing only the classification head of the DNN models. We introduce the RNF framework for debiasing the classification head by neutralizing training samples that have the same ground truth label but with different sensitive attribute annotations. To reduce the reliance on sensitive attribute annotations (as used in existing works), we generate proxy annotations by training a bias-intensified model and then annotating samples based on its confidence level. Experimental results indicate our RNF framework to dramatically reduce the discrimination of DNN models, without requiring access to annotations for the sensitive attributes for all the training samples. Experimental analysis further demonstrates our RNF framework to further improve in conjunction with other debiasing methods. Specifically, our RNF framework built on top of a debiased backbone encoder leads to better mitigation performance with negligible accuracy drop in the task performance. In future, we plan to explore more choices of combination of RNF with other debiasing methods.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, 2012.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[6] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 2020.

[7] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[8] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *Fairness, Accountability, and Transparency in Machine Learning (FAT/ML)*, 2018.

[9] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.

[10] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. *International Conference on Computer Vision (ICCV)*, 2019.

[11] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[12] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[13] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning (ICML)*, 2020.

[14] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Andrea Zunino, Sarah Adel Bargal, Riccardo Volpi, Mehrnoosh Sameki, Jianming Zhang, Stan Sclaroff, Vittorio Murino, and Kate Saenko. Explainable deep classification models for domain generalization. *arXiv preprint arXiv:2003.06498*, 2020.

[16] Laura Rieger, Chandan Singh, W James Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. *International Conference on Machine Learning (ICML)*, 2020.

[17] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint arXiv:2011.09468*, 2020.

[18] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[19] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[20] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*, 2018.

[21] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, 2019.

[22] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.

[23] Gökhan H Bakır, Jason Weston, and Bernhard Schölkopf. Learning to find pre-images. *Advances in neural information processing systems (NIPS)*, 2004.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS 2014 Deep Learning Workshop*, 2015.

[25] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[26] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[27] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems (NIPS)*, 2016.

[28] Steven B Cohen. Design strategies and innovations in the medical expenditure panel survey. *Medical care*, pages III5–III12, 2003.

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[30] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)*, 2018.

[31] Yahav Bechavod and Katrina Ligett. Penalizing unfairness in binary classification. *Fairness, Accountability and Transparency in Machine Learning (FAT/ML)*, 2017.

[32] Peizhao Li, Han Zhao, and Hongfu Liu. Deep fair clustering for visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] Xiaoqian Wang and Heng Huang. Approaching machine learning fairness through adversarial network. *arXiv preprint arXiv:1909.03013*, 2019.

[34] Yongkai Wu, Lu Zhang, Xintao Wu, and Hanghang Tong. Pc-fairness: A unified framework for measuring causality-based fairness. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[35] Hao Wang, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning (ICML)*, 2019.

[36] Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffry Svacha, and Madeleine Udell. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the conference on fairness, accountability, and transparency*, 2019.

[37] Frederick Liu and Besim Avci. Incorporating priors with feature attribution on text classification. *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[38] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in neural information processing systems (NIPS)*, 2017.

[39] Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[40] Ching-Yao Chuang and Youssef Mroueh. Fair mixup: Fairness via interpolation. *International Conference on Learning Representations (ICLR)*, 2021.

[41] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[42] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. Fake face detection methods: Can they be generalized? In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section E in the Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work addresses the fairness problem. The goal is to enable DNNs to better serve our society.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Please refer to Section A in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] Both our codes and data will be released to GitHub upon acceptance of this submission.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Random seeds could lead to variance in model training. We did not use random bars, and instead reported experimental results using average values over three runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Our model is trained using a single GPU: GeForce RTX 2080 Ti.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We provide the URLs for the datasets that we have used in Sec.B in the Appendix.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]