Discover and Mitigate Unknown Biases with Debiasing Alternate Networks - Appendix

Zhiheng Li[1], Anthony Hoogs[2], and Chenliang Xu[1]

[1]University of Rochester [2]Kitware, Inc.
{zhiheng.li,chenliang.xu}@rochester.edu anthony.hoogs@kitware.com

Appendix

A Pseudocode of DebiAN

We present the pseudocode of DebiAN for two tasks – 1) discover the unknown biases (Alg. 1); 2) mitigate the unknown biases (Alg. 2). To ensure that the sampled images have the same target attribute labels, we select images with the same target attribute label in a batch to compute the loss (line 3-7 in Alg. 1, 2, and line 10-14 in Alg. 2).

Algorithm 1: Discover unknown biases.

Input: C: trained *classifier*, T: number of iterations, K: number of target attribute classes
Output: D: *discoverer*
Data: $\mathcal{D}$: training set

1 **for** $t : 1 \dots T$ **do**
2 $\quad \mathcal{B} \coloneqq \{(\mathbf{I}_i, y_i)\}_{i=1}^{N} \sim \mathcal{D}$ // Sample a batch $\mathcal{B}$ with N pairs of images $\mathbf{I}_i$ and target attribute labels y_i
$\quad$ /* for each target attribute class t */
3 $\quad$ **for** $k : 1 \dots K$ **do**
4 $\quad\quad \mathcal{B}_k \coloneqq \{(\mathbf{I}_j, y_j) \mid y_j = k, (\mathbf{I}_j, y_j) \in \mathcal{B}\}_{j=1}^{M}$ // Select M pairs from $\mathcal{B}$ whose labels are k
5 $\quad\quad p(\hat{y} \mid \mathbf{I}_j) \coloneqq C(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // C predicts target attribute
6 $\quad\quad p(\hat{b} \mid \mathbf{I}_j) \coloneqq D(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // D predicts bias attribute groups
7 $\quad\quad \mathcal{L}_k = \mathcal{L}_{\text{EOV}} + \mathcal{L}_{\text{UA}}$ // Compute loss on $\mathcal{B}_t$.
8 $\quad$ update D with loss $1/K \sum_{k=1}^{K} \mathcal{L}_k$

B Implementation Details

In DebiAN, the *discoverer* and *classifier* use the same architecture but do not share the parameters.

Algorithm 2: Mitigate unknown biases.

Input: T: number of iterations, K: number of target attribute classes
Output: C: *classifier*, D: *discoverer*
Data: $\mathcal{D}$: training set

for $t : 1 \dots T$ **do**
  /* ======== Start: optimize C, freeze D ======== */
  $\mathcal{B} \coloneqq \{(\mathbf{I}_i, y_i)\}_{i=1}^{N} \sim \mathcal{D}$ // Sample a batch $\mathcal{B}$ with N pairs of images $\mathbf{I}_i$ and target attribute labels y_i
  /* for each target attribute class t */
  for $k : 1 \dots K$ **do**
    $\mathcal{B}_k \coloneqq \{(\mathbf{I}_j, y_j) \mid y_j = k, (\mathbf{I}_j, y_j) \in \mathcal{B}\}_{j=1}^{M}$ // Select M pairs from $\mathcal{B}$ whose labels are k
    $p(\hat{y} \mid \mathbf{I}_j) \coloneqq C(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // C predicts target attribute
    $p(\hat{b} \mid \mathbf{I}_j) \coloneqq D(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // D predicts bias attribute groups
    $\mathcal{L}_k^C = \mathcal{L}_{\text{RCE}}$ // Compute loss on $\mathcal{B}_k$
  update C with loss $1/K \sum_{k=1}^{K} \mathcal{L}_k^C$
  /* ======== End: optimize C, freeze D ======== */
  /* ======== Start: optimize D, freeze C ======== */
  $\mathcal{B} \coloneqq \{(\mathbf{I}_i, y_i)\}_{i=1}^{N} \sim \mathcal{D}$ // Sample a batch $\mathcal{B}$ with N pairs of images $\mathbf{I}_i$ and target attribute labels y_i
  /* for each target attribute class k */
  for $k : 1 \dots K$ **do**
    $\mathcal{B}_k \coloneqq \{(\mathbf{I}_j, y_j) \mid y_j = k, (\mathbf{I}_j, y_j) \in \mathcal{B}\}_{j=1}^{M}$ // Select M pairs from $\mathcal{B}$ whose labels are k
    $p(\hat{y} \mid \mathbf{I}_j) \coloneqq C(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // C predicts target attribute
    $p(\hat{b} \mid \mathbf{I}_j) \coloneqq D(\mathbf{I}_j), \mathbf{I}_j \in \mathcal{B}_k$ // D predicts bias attribute groups
    $\mathcal{L}_k^D = \mathcal{L}_{\text{EOV}} + \mathcal{L}_{\text{UA}}$ // Compute loss on $\mathcal{B}_k$
  update D with loss $1/K \sum_{k=1}^{K} \mathcal{L}_k^D$
  /* ======== End: optimize D, freeze C ======== */

On Multi-Color MNIST dataset (Sec. 4.1), we follow the same setting used in LfF [23]'s experiment on Colored MNIST. We use Adam [13] optimizer with 10^{-3} learning rate and 256 batch size. We use an MLP with three hidden layers (obtained from the LfF's official code[1]). All models are trained for 100 epochs.

In the experiments for gender bias mitigation on CelebA [21] dataset (Sec. 4.2), we follow most of the settings used in LfF. We use ResNet-18 [9] as the network architecture. We use horizontal flip for data augmentation during training. We use Adam optimizer with 10^{-4} learning rate and 256 batch size. All models are trained for 50 epochs. The only difference is that we use CelebA's validation set to choose the epoch where models achieve the best validation set accuracy and report the results on the testing set. Note that validation set accuracy does not use any bias attribute labels because unsupervised debiasing should not rely on any labels of bias attributes. LfF directly reports the results at the 50 epoch on

[1] https://github.com/alinlab/LfF

the validation set, which is hard to be replicated as reported by other users in their official code GitHub repository[2].

In the experiments for gender bias mitigation on bFFHQ dataset [12], we use the same setting in [12]. We use Adam as the optimizer with 256 batch size. All models are trained for 200 epochs. We use ResNet-18 as the backbone. We notice that Lee *et al.* [16] use a different setting on bFFHQ dataset with StepLR for the learning rate scheduling, which is more complicated than the one in the original paper [12]. Thus, we choose the former one as the setting on bFFHQ dataset.

In the experiment of mitigating multiple biases in gender classifier on CelebA dataset (Sec. 4.2), we choose 64 as the batch size and ResNet-50 as the backbone of classifiers. All models are trained with 50 epochs. We use CelebA's validation set to choose the epoch that has the best validation set accuracy for each method. We report the results on the testing set. We use Adam as the optimizer with 10^{-4} learning rate.

On Biased Action Recognition (BAR) [23] dataset, we use the setting in [23]. We use Adam as the optimizer with 10^{-4} learning rate. The batch size is 256. We use 224×224 random cropping for data augmentation. All models are trained with 90 epochs.

In the scene classification task, we choose ResNet-18 as the backbone of classifiers and 128 as the batch size. We use Adam as the optimizer with 10^{-4} learning rate. All models are trained only on the Places [31] dataset for 50 epochs. We choose the epoch where the model achieves the best accuracy on Places's validation set and report the results on LSUN's [29] validation set.

The code is based on PyTorch [24]. We modify LfF's code[3] that generates Colored MNIST to create Multi-Color MNIST dataset.

In implementation, we add $\epsilon = 10^{-6}$ to the denominators of $\bar{P}_{b^+}(\hat{y})$ and $\bar{P}_{b^-}(\hat{y})$ (Eq. 3) to avoid zero division.

For the *discoverer* D, we choose two different implementations for different numbers of classes of the target attribute. When the target attribute is binary (*e.g.*, experiments in Sec. 4.2), *i.e.* number of classes is two, D predicts one value for each image, which is the predicted bias attribute group. We denote this implementation as "global" since two classes globally share the predicted bias attribute groups. When the target attribute has $c > 2$ classes, *e.g.*, ten classes in the digit classification, action recognition, and scene classification in Sec. 4.1 and Sec. 4.3, D predicts c values, where each value is the predicted bias attribute group of the corresponding target attribute class. We denote this implementation as "per class" since D predicts bias attribute groups for each target attribute class. We provide an ablation study on this in Appendix C.5.

Table 7: Ablation study on Unbalanced Assignment (UA) penalty (*i.e.*, $\mathcal{L}_{\text{UA}}$) in mitigating gender bias of `Blond Hair` classifier on CelebA [21] dataset

	w/o $\mathcal{L}_{\text{UA}}$	DebiAN
Avg group Acc.	$79.6_{\pm 1.7}$	$\mathbf{84.0}_{\pm 1.4}$
Worst group Acc.	$38.5_{\pm 4.7}$	$\mathbf{52.9}_{\pm 4.7}$

Table 8: Ablation study on Unbalanced Assignment (UA) penalty (*i.e.*, $\mathcal{L}_{\text{UA}}$) in mitigating multiple biases of gender classifier on CelebA [21] dataset

bias attribute	metric	w/o $\mathcal{L}_{\text{UA}}$	DebiAN
`Wearing Lipstick`	Avg. Group Acc.	$87.7_{\pm 0.4}$	$\mathbf{88.5}_{\pm 1.1}$
	Worst Group Acc.	$58.1_{\pm 1.2}$	$\mathbf{61.7}_{\pm 4.2}$
`Heavy Makeup`	Avg. Group Acc.	$85.6_{\pm 1.2}$	$\mathbf{87.8}_{\pm 1.3}$
	Worst Group Acc.	$46.9_{\pm 5.2}$	$\mathbf{56.0}_{\pm 5.2}$

C Ablation Study

C.1 Unbalanced Assignment (UA) penalty

Here we show more ablation study results on the Unbalanced Assignment (UA) penalty on CelebA dataset. The results are shown in Tabs. 7 and 8, which further proves that $\mathcal{L}_{\text{UA}}$ can improve fairness results by avoiding the trivial solution—assigning all images into a single bias group (see Unbalanced Assignment (UA) penalty in Sec. 3.1).

C.2 Batch Size

In practice, $\{\mathbf{I}_i\}_{i=1}^n$ (defined in Sec. 3) is a mini-batch of images sampled from the dataset for optimizing the networks. One may have the concern that the sampled batch may not have enough images from different bias groups for the *discoverer* to assign. Therefore, we conduct an ablation study on different batch sizes on Multi-Color MNIST dataset with the same setting introduced in Sec 4.1, where the ratio of the `left color` is 0.99 and `right color` is 0.95. We report the accuracy results for images that are both bias-conflicting w.r.t. `left color` and `right color` bias attributes (see "both bias-conflicting" in Tab. 9). We also report the unbiased accuracy results. The results in Tab. 9 show that DebiAN can achieve better debiasing results under different batch sizes compared with the vanilla model.

[2] https://github.com/alinlab/LfF/issues/2
[3] https://github.com/alinlab/LfF/blob/master/make_dataset.py

Table 9: Ablation study on different batch sizes on Multi-Color MNIST dataset. We report the accuracy results for images that are both bias-conflicting w.r.t. `left color` and `right color` bias attributes. We also report the unbiased accuracy results. DebiAN achieves better debiasing results under all batch sizes

batch size		vanilla	**DebiAN (Ours)**
32	both bias-conflicting	$8.0_{\pm 0.5}$	$\mathbf{18.9}_{\pm 1.2}$
	unbiased accuracy	$61.7_{\pm 1.0}$	$\mathbf{75.0}_{\pm 0.8}$
64	both bias-conflicting	$8.2_{\pm 1.9}$	$\mathbf{18.1}_{\pm 0.8}$
	unbiased accuracy	$61.7_{\pm 1.4}$	$\mathbf{74.2}_{\pm 0.4}$
128	both bias-conflicting	$5.6_{\pm 1.8}$	$\mathbf{17.2}_{\pm 1.2}$
	unbiased accuracy	$58.7_{\pm 2.5}$	$\mathbf{72.1}_{\pm 0.7}$
256	both bias-conflicting	$5.2_{\pm 0.4}$	$\mathbf{16.0}_{\pm 1.8}$
	unbiased accuracy	$57.4_{\pm 0.7}$	$\mathbf{72.0}_{\pm 0.8}$
512	both bias-conflicting	$4.8_{\pm 1.0}$	$\mathbf{12.5}_{\pm 1.9}$
	unbiased accuracy	$56.1_{\pm 1.3}$	$\mathbf{70.1}_{\pm 1.1}$

C.3 Ablation Study on Different Ratios

We conduct an ablation study on different ratios of bias-aligned samples in Multi-Color MNIST's training set. We keep the ratio for the `right color` bias attribute to 0.95 and use different ratios for `left color` bias attribute, ranging from 0.995 to 0.95. The results are shown in Tab. 10. DebiAN achieves better unbiased accuracy results and accuracy results on samples that are bias-conflicting w.r.t. both bias attributes. The only exception is the accuracy results of the samples that are bias-conflicting w.r.t. both bias attributes when both ratios are 0.95 (last section in Tab. 10). Both LfF and DebiAN achieve 39.6 accuracy results. However, our method achieves a lower standard deviation (0.2) than LfF (6.9) and achieves much better final unbiased results (81.8 vs. 68.5). We also notice that LfF's debiasing results have a large standard deviation when the ratios of both bias attributes are 0.95. We provide an explanation in Appendix D.3.

C.4 Alternate Training

We conduct an ablation study on alternate training on the Multi-Color MNIST dataset with the same setting used in Sec. 4.1. To remove the alternate training from DebiAN, we follow EIIL [5] and PGI [1] to train the *discoverer* to identify the unknown biases in a *classifier* trained with one epoch. After training *discoverer*, we fix the parameters of *discoverer* and only train the *classifier* to perform debiasing. The results in Tab. 11 show that alternate training can improve the debiasing results, *e.g.*, higher unbiased accuracy and higher accuracy for samples that are bias-conflicting w.r.t. both bias attributes (4th row), which demonstrates the necessity of alternate training.

C.5 Bias Attribute Groups: Global vs. Per Class

As mentioned in Appendix B, the predicted bias attribute groups from the *discoverer* (D) are shared by both classes in the binary classification task. In the multi-class classification setting (*e.g.*, digit classification, scene classification task, *etc.*), D predicts binary bias group assignments for each class. We justify our implementation choices with the results in Tab. 12.

For the binary age classification on bFFHQ dataset, there is no significant difference between the two implementation choices (*i.e.*, differences are within error bars). Therefore, we choose "global" *discoverer* for the binary classification task due to its simplicity.

However, in the multi-class digit classification on Multi-Color MNIST dataset, we do observe the better result produced by the *discoverer* that predicts bias group assignment for each class (*i.e.*, improvement is greater than the error bar). We suspect that predicting bias attribute groups per class in the binary classification task is redundant because the binary target attribute is spuriously correlated with the binary bias attribute. For example, if the target attribute `age` is spuriously correlated with the bias attribute `gender`, *i.e.*, more young females and old males than old females and young males, then it is not necessary to predict bias attribute group for both genders since both genders share the same bias attribute groups. However, this may not be the case for multi-class settings. For example, in Multi-Color MNIST dataset, each digit class is spuriously correlated with a unique left color, *e.g.*, for bias-aligned samples, digit class 0's left color is red but digit class 1's left color is yellow (Fig. 4). In other words, the bias attribute values may not be shared globally across different target attribute classes. Therefore, we choose different numbers of outputs for the *discoverer* under different tasks.

C.6 Alternative Design for Debiasing: $\max_C \mathcal{L}_{\text{EOV}}$

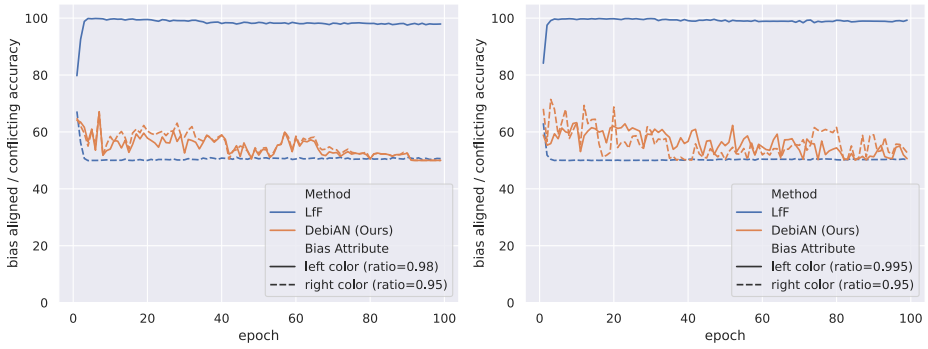
One may consider an alternative design for debiasing—train the *classifier* C to maximize the Equal Opportunity Violation (EOV) loss, or formally $\max_C \mathcal{L}_{\text{EOV}}$. This alternative design, to some degree similar to GAN [6]'s training strategy, may look more "unified" since it lets the *discoverer* D and *classifier* C play the minmax game:

$$\min_C \max_D |\bar{P}_{b^+}(\hat{y}) - \bar{P}_{b^-}(\hat{y})|, \tag{9}$$

where $\bar{P}_{b^+}(\hat{y})$ and $\bar{P}_{b^-}(\hat{y})$ are defined in Eq. 3. This alternative design enables C to *directly* meet the Equal Opportunity [8,25] fairness criterion. More concretely, we implement this alternative design of C's objectives by the following loss function:

$$\min_C -\log\left(1 - \left|\bar{P}_{b^+}(\hat{y}) - \bar{P}_{b^-}(\hat{y})\right|\right) + \text{CE}(p_t(\mathbf{I}_i), y_i), \tag{10}$$

where the first $-\log$ term implements C's objective in the minmax game (Eq. (9)) and the second term CE is the standard cross-entropy loss. We conduct an



(a) `left color`'s ratio = 0.98 **(b)** `left color`'s ratio = 0.995

Fig. 8: More bias discovery results w.r.t. `left color` and `right color` biases throughout the training epochs on Multi-Color MNIST dataset under different ratios of bias-aligned samples w.r.t. `left color`. The results are consistent with Fig. 5

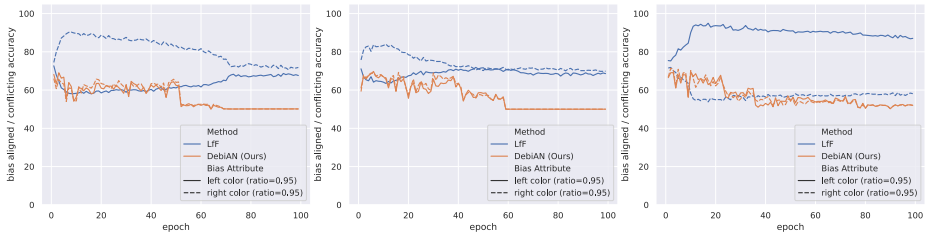


Fig. 9: Bias discovery results when `left color` and `right color` are equally salient (both ratios are 0.95). The three plots are results under three different random seeds. In the first two plots, LfF mainly discovers `right color` at the early training stage and gradually discovers both biases. In the third seed, it mainly discovers `left color` bias. The results show that LfF are unstable in bias discovery when two biases are equally salient. In contrast, our DebiAN consistently discovers both biases at the early stage under different random seeds and gradually converges 50% bias discovery accuracy as debiasing is performed in the *classifier*

ablation study on the design for debiasing (*i.e.*, playing minmax game vs. RCE loss (Eq. 6)) and the results on Multi-Color MNIST dataset are shown in Tab. 13. The results demonstrate that our RCE loss performs much better than the alternative design. We suspect the reason is that C in DebiAN has two goals – 1) fooling the *discoverer* to achieve fairer results; 2) achieving higher accuracy by optimizing the standard cross-entropy loss, which is different from GAN [6] where the generator only has one goal – fooling the discriminator to achieve better image quality of the synthesized images. Therefore, it is hard to control the balance between the two goals of C in this alternative design. In contrast, our RCE loss can better incorporate the two goals within a single objective function $\mathcal{L}_{\text{RCE}}$, leading to better debiasing results.

D Bias Discovery on Multi-Color MNIST

D.1 Implementation Details

To evaluate the bias discovery results, we transform the outputs from LfF's "biased model" and DebiAN's *discoverer* into bias-aligned / bias-conflicting prediction by the following approaches.

For LfF, since the biased model is trained to amplify the biases, the biased model's outputs predict ten colors aligned with digits. Thus, when its predicted color class is the same as the ground-truth digit class, *e.g.*, c-th color for the c-th class, we regard its prediction as bias-aligned. Otherwise, its prediction is bias-conflicting.

For DebiAN, we use *discoverer* and *classifier* to predict each image $\mathbf{I}_i$'s predicted bias group assignment $p(\hat{b} \mid \mathbf{I}_i)$ and predicted probability of the ground-truth class of the target attribute $p_t(\mathbf{I}_i)$ on the entire testing set, respectively. Then, we compute the weighted average predicted probabilities $\bar{P}_{b^+}(\hat{y})$ and $\bar{P}_{b^-}(\hat{y})$ (see Eq. 3) in two bias groups. If $\bar{P}_{b^+}(\hat{y}) \geq \bar{P}_{b^-}(\hat{y})$, we use $p(\hat{b} = 1 \mid \mathbf{I}_i)$ as the bias-aligned prediction and $p(\hat{b} = 0 \mid \mathbf{I}_i)$ as the bias-conflicting prediction since the positive bias group has higher weighted average predicted probability, *i.e.*, *classifier* performs better on the positive bias group. If $\bar{P}_{b^+}(\hat{y}) < \bar{P}_{b^-}(\hat{y})$, we use $p(\hat{b} = 1 \mid \mathbf{I}_i)$ as the bias-conflicting prediction and $p(\hat{b} = 0 \mid \mathbf{I}_i)$ as the bias-aligned prediction.

After obtaining bias-aligned and bias-conflicting predictions, we compute the bias-aligned and bias-conflicting accuracy for `left color` and `right color` biases as follows. Each testing image has two labels—(1) bias-aligned/bias-conflicting w.r.t. left color; (2) bias-aligned/bias-conflicting w.r.t. right color. Thus, the left color (or right color) bias discovery accuracy is computed based on the bias-aligned/bias-conflicting predictions against the left color (or right color) bias-aligned/bias-conflicting labels.

D.2 More Bias Discovery Results under Different Ratios

In the main paper, we evaluate LfF and DebiAN's bias discovery results when the ratio of `left color` is 0.99 in Fig. 5. Here, we show results under more ratios in Fig. 8, where the ratios of `left color` are 0.98 (Fig. 8 (a)) and 0.995 (Fig. 8 (b)). The results are consistent with Fig. 5—LfF can only discover the more salient `left color` and cannot identify the less salient `right color` bias, whereas DebiAN's *discoverer* can discover both biases at the early training stage and the bias discovery accuracy gradually converges to 50% when debiasing is performed in the *classifier*.

D.3 Bias Discovery Results under Equally Salient Biases

We further evaluate bias discovery results when `left color` and `right color` biases are equally salient, *i.e.*, ratios of both biases are 0.95. We found that LfF shows more unstable results under different random seeds than in previous

settings. Therefore, we show bias discovery results under three different random seeds in Fig. 9. Under the first two random seeds (left and middle plots in Fig. 9), LfF first discovers `right color` bias and gradually discovers both biases. However, under the third random seed (the right plot in Fig. 9), LfF mainly discovers the `left color` bias. Therefore, our Multi-Color MNIST dataset reveals another weakness of LfF—unstable bias discovery results when two biases are equally salient, which also explains LfF's unstable debiasing results under the equally salient biases (LfF has large error bars, *e.g.*, ± 33.7 and ± 25.9, of the accuracy results in Tab. 10). In contrast, DebiAN's bias discovery results are stable—consistent results across different random seeds and under different ratios.

D.4 Detailed Discussion on Bias Discovery

Why LfF's bias discovery accuracies do not converge to 50%? In Fig. 5 and Fig. 8, LfF's bias discovery accuracies maintain at about 100% or 50% throughout the training epochs. One may wonder why it does not converge to 50% as DebiAN does. The reason is that DebiAN discovers biases from the *classifier* ($\mathcal{L}_{\text{EOV}}$ is based on *classifier*'s output), whereas LfF identifies biases from the dataset. Concretely, LfF uses the assumption that the bias attribute is easier than the target attribute to define the bias and uses Generalized Cross-entropy (GCE) loss [30] to train a biased model. GCE loss is defined by $-p_t^q \log p_t$, where p_t is bias model's predicted probability of the ground-truth class and q is a hyperparameter. Intuitively, it up-weights easy examples (*i.e.*, high p_t) with high weight p_t^q and down-weights hard examples (*i.e.*, low p_t) with low weight p_t^q. Therefore, the "biased model" focuses more on the easy examples in the dataset. However, it does not know any biases in the *classifier* (no classifier's outputs used in GCE). Therefore, whether the classifier is performing debiasing will not affect LfF's bias discovery, making the bias attribute accuracy stays the same throughout the entire training stage. In contrast, DebiAN's *discoverer* actively identifies biases in the *classifier*. Therefore, *discoverer*'s bias discovery results will converge to 50% as debiasing is performing in the *classifier*, making the *discoverer* harder to find the biases.

Bias Discovery: EIIL and PGI Here, we discuss the connection and difference between DebiAN and two previous methods—EIIL and PGI. In contrast to LfF that finds biases from the dataset based on the assumption, all EIIL, PGI, and DebiAN actively find biases from the classifier. However, DebiAN differs from EIIL and PGI in terms of the objective function, network architecture, and training scheme. We mainly introduce EIIL because PGI is a follow-up work for EIIL with a difference in the network architecture.

In terms of objective function, EIIL (and PGI) inversely uses the debiasing objective function—IRMv1 [2]. In other words, while minimizing IRMv1 was designed for debiasing in previous works, EIIL maximizes the gradient norm penalty in IRMv1 to identify biases. However, this is suboptimal for two reasons. First, IRMv1 approximates IRM with gradient norm penalty to make it

computationally tractable. However, the zero gradient norm may only indicate a local minimum instead of the global minimum. In contrast, DebiAN's EOV loss uses the principled definition to define the bias—violation of equal opportunity fairness criterion. Second, since the gradient norm does not have an upper bound, maximizing it leads to an optimization problem. As a comparison, DebiAN's EOV loss minimizes a bounded negative log-likelihood (Eq. 2), which is easier to be optimized.

Regarding network architecture, EIIL does not train any networks but directly optimizes a vector $\mathbf{q} \in \mathbb{R}^N$ for N images in the training set to maximize IRMv1's gradient norm, which can be regarded as directly optimize the bias group assignments. Since the vector $\mathbf{q}$ is only fitted to the training set, we cannot evaluate EIIL's bias discovery results on the balanced testing set. PGI uses a slightly different approach by training a small MLP that takes the classifier's features as the input and predicts the bias group assignments. PGI's bias discovery objective function is identical with EIIL's—maximizing IRMv1. We use the trained MLP to infer the bias-aligned / bias-conflicting on the testing set on Multi-Color MNIST dataset under the setting used in Sec. 4.1 (ratio w.r.t. `left color` is 0.99 and ratio w.r.t. `right color` is 0.95) based on a classifier trained with one epoch (explain in the next paragraph). The bias discovery accuracy results w.r.t. `left color` and `right color` are 50.6±1.9 and 49.9±0.3. In contrast, DebiAN's *discoverer* has the same network architecture as the classifier to predict bias group assignments from raw images, which enables *discoverer* to learn its own feature of the bias attribute directly from the raw images. As a result, DebiAN achieves about 60% to 75% accuracy at the first epoch (see Fig. 5). Therefore, we empirically show that DebiAN can discover biases more accurately.

Finally, with respect to the training scheme, by hypothesizing that the classifier learns bias features in the early stage, EIIL and PGI use a fixed classifier trained with one epoch to discover biases. However, this might not be the case in the multi-bias setting where multiple biases may be learned at different training stages. In contrast, DebiAN trains *discoverer* and *classifier* in an alternate fashion, enabling *discoverer* to find biases in the *classifier* during the entire training stage. Our ablation study on the alternate training (Appendix C.4) further demonstrates its benefits.

E Results on Colored MNIST (single-bias setting)

We also compare with other methods on Colored MNIST in a single-bias setting. There are two variants of Colored MNIST datasets in previous works—(1) adding colors to the foreground (*i.e.*, digit) [1,11,17]; (2) adding colors to the background [3,26]. Therefore, we conduct experiment on both variants of Colored MNIST dataset. We denote the Colored MNIST with foreground color as "Colored MNIST (foreground)" and the Colored MNIST with background color as "Colored MNIST (background)." Same with the experiment setting on Multi-Color MNIST, we follow the setting used in LfF [23], including using MLP as the network architecture, training with 100 epochs, *etc*. Same with LfF,

we report the results under four ratios of bias-aligned samples in the training set—0.995, 0.99, 0.98, and 0.95. In terms of evaluation metrics, we follow LfF to report the accuracy results on bias-conflicting samples and unbiased accuracy. We additionally report the accuracy results on bias-aligned samples.

We report LfF's reported results on Colored MNIST (foreground). Besides, we also replicate LfF's results via their officially released code. We cannot replicate their reported results. This issue is also reported in the official code's GitHub repository by other people[4].

The results on Colored MNIST (foreground color) are in Tab. 14. Although LfF achieves better bias-conflicting accuracy results, it also achieves much lower bias-aligned accuracy, revealing that LfF's reweighing method overly focuses on the bias-conflicting samples than the bias-aligned samples. As a result, LfF achieves low unbiased accuracy. Overall, DebiAN achieves comparable or slightly lower unbiased accuracy results.

The results on Colored MNIST (background color) are in Tab. 15. Similar to the results on Colored MNIST (foreground color), LfF achieves better bias-conflicting accuracy but low bias-aligned accuracy and unbiased accuracy. Overall, DebiAN achieves comparable or slightly better unbiased accuracy results.

We also notice that PGI achieves inconsistent results on Colored MNIST (foreground color) and Colored MNIST (background color). While PGI achieves very good results on Colored MNIST (foreground color), *e.g.*, top-1 unbiased accuracy when ratio = 0.995 and ratio = 0.98 in Tab. 14, it achieves bad results on Colored MNIST (background color), *e.g.* the lowest unbiased accuracy results in Tab. 15. We suspect the reason is that PGI is overly sensitive to hyperparameters, *e.g.*, the coefficient of the KL-divergence for debiasing. In contrast, our method achieves good results across two Colored MNIST variants without tuning or changing any hyperparameters.

Finally, we restate that Colored MNIST is a single-bias setting, which may not be the case in real-world scenarios where multiple biases exist. Therefore, we regard that we should focus more on our new Multi-Color dataset to evaluate the debiasing results w.r.t. multiple biases (Tab. 10), where DebiAN achieves better debiasing results.

F More results on Mitigating Multiple Biases in Gender Classification

In Tab. 4, we show better DebiAN's better debiasing results w.r.t. `Wearing Lipstick` and `Heavy Makeup`. To demonstrate that DebiAN's better debiasing results w.r.t. more bias attributes, we evaluate on Transects [4] dataset (mentioned on L567 in the main paper). Transects dataset contains high-quality face images synthesized by StyleGAN2 [10], which are also balanced w.r.t. multiple biases such as `Hair Length` and `Skin Color`. The dataset does not contain training split because it is designed as a testing set to identify biases in gender

[4] https://github.com/alinlab/LfF/issues/1

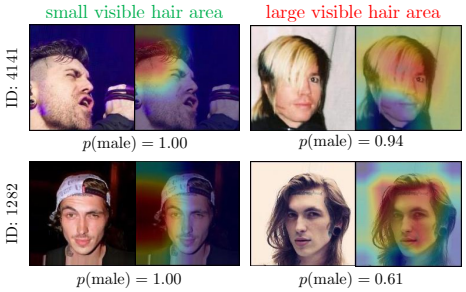


Fig. 10: Discovered bias of gender classifier: `visible hair area`. D's CAM saliency map is paired with each image. The vanilla gender classifier train on CelebA dataset performs worse for males with larger visible hair area

classifier. We use gender classifiers trained on CelebA dataset (same setting used in Sec. 4.2) to evaluate their debiasing performance w.r.t. `Hair Length` and `Skin Color` bias attributes on Transects dataset. The results are shown in Tab. 16, which demonstrates DebiAN's better capability in mitigating multiple biases simultaneously. Furthermore, the better results w.r.t. `Hair Length` can also reflect that DebiAN's *discoverer* identifies `visible hair area` bias attribute (Fig. 6 and Fig. 10).

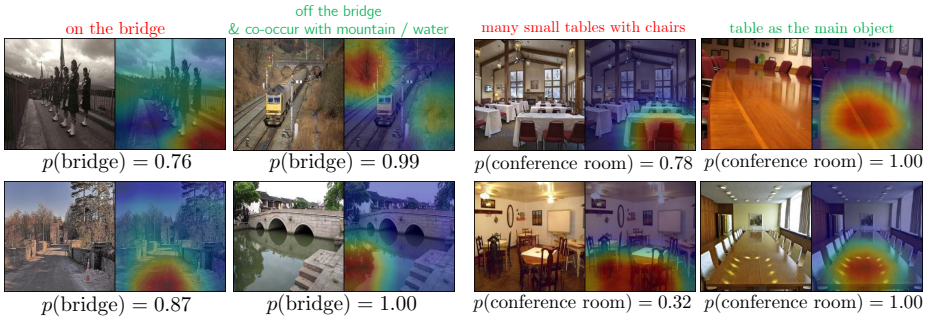
G More Qualitative Results

G.1 More Examples of Discovered Biases on Face Images

While Fig. 5 shows the discovered `visible hair area` bias of gender classifier on female images, we further show the male image examples in Fig. 10. D focuses on the `visible hair area` to separate images into "small visible hair area" and "large visible hair area" groups, which is consistent with the female examples shown in Fig. 5.

G.2 More Examples of Discovered Biases on Scene Images

We show more examples of discovered biases on the scene images in Fig. 11. For bridge images, D predicts two bias groups. When the photos are taken on the bridge, the vanilla classifier performs worse. In comparison, the vanilla classifier performs better when the photos are taken off the bridge with some correlated backgrounds, such as mountains or water. For conference room images, the vanilla classifier performs better when the table is the major object in the scene. However, it performs worse when the conference room images have many tables, and the tables do not occlude the chairs.



(a) Discovered bias in the bridge class. (b) Discovered bias in the conference room class.

Fig. 11: Discovered biases of vanilla scene classifiers. *D*'s CAM is paired with the image

H Discussion

H.1 Is DebiAN a hard negative method?

No. Hard negative methods focus on addressing the imbalanced problem by letting the classifier focus on hard misclassified examples and pay less attention to easy examples. A seminal hard negative method is focal loss [19], which reweighs the standard cross-entropy loss ($-\log(p_t)$, where p_t is the predicted probability of the ground-truth class) to $-\alpha_t(1-p_t)^\gamma \log(p_t)$, where α_t and γ are hyperparameters. Intuitively, it uses classifier's predicted probability to reweigh the cross-entropy loss, where hard examples (*i.e.*, low p_t) are up-weighted with high $\alpha_t(1-p_t)^\gamma$ weights and easy examples (*i.e.*, high p_t) are down-weighted with low $\alpha_t(1-p_t)^\gamma$ weights. Different from focal loss, our RCE loss (Eq. 6) uses *discoverer*'s predicted bias group assignments to reweigh the cross-entropy loss, and the *discoverer* is trained with $\mathcal{L}_{\text{EOV}}$ to differentiate *classifier*'s p_t on examples from the same target class where the Equal Opportunity is violated. Therefore, DebiAN is not a hard negative method because we do not use easy or hard samples (*i.e.*, low or high p_t) to perform reweighing, but rather use samples' estimated bias group assignments to perform debiasing.

We also compare with focal loss on Multi-Color MNIST dataset. We use $\alpha = 0.25$ and $\gamma = 2.0$ as they perform the best in [19]. The results are shown in Tab. 17, where focal loss's results are even worse than the vanilla model. The results prove that hard negative methods are not well-suited for debiasing. Since DebiAN is different from hard negative methods by using estimated bias group assignments to mitigate biases, our method achieves much better debiasing results.

H.2 Is EOV loss simply doing clustering?

No. EOV loss is used to train *discoverer* to classify different bias groups values. Therefore, instead of simply clustering *classifier*'s prediction, EOV loss guide the *discoverer* to do a classification for the bias group assignment. The results in

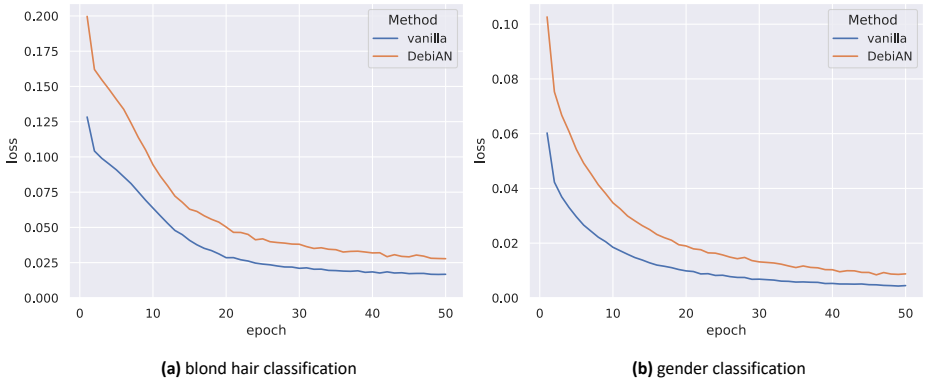


Fig. 12: Vanilla's cross-entropy loss and DebiAN's RCE loss in (a) blond hair classification and (b) gender classification

Fig. 5, Fig. 8, and Fig. 9 show that *discoverer* can accurately classify if the samples on the testing set (unseen during training) are bias-aligned or bias-conflicting, demonstrating that EOV loss guides the *discoverer* to do classification based on different bias groups values and it can generalize to testing set's images.

H.3 Will *classifier* achieve 100% accuracy such that *discoverer* cannot predict bias group assignments?

No. First, note that *discoverer*'s EOV loss is based on *classifier*'s predicted probabilities instead of thresholded hard predictions. Therefore, 100% accuracy does not indicate that *discoverer* cannot predict the bias group assignments. Second, we show the vanilla model's cross-entropy loss and DebiAN's RCE loss on blond hair classification and gender classification tasks in Fig. 12. The results show that the losses do not completely converge to zero, which proves that there always exist samples in the training set that *classifier* does not achieve 1.0 predicted probabilities. Therefore, it still leaves the room for *discoverer* to predict bias group assignments based on *classifier*'s different predictions on different samples.

H.4 What if the mini-batch only contains the samples from a single bias group?

It mainly happens under two conditions—(1) very strong spurious correlation; (2) small mini-batch size. For the first case, our ablation study on the ratios on Multi-Color MNIST dataset (Appendix D.2) shows that DebiAN achieves better debiasing results even when the ratio of `left color` is 0.995 (*i.e.*, very strong spurious correlation). For the second case, our ablation study on different batch sizes (Appendix C.2) shows that our method still achieves strong debiasing results when the batch size is small.

H.5 RCE loss compared with previous reweighing-based methods

LfF and focal loss are two previous reweighing-based methods for unsupervised debiasing. At a high level, LfF, focal loss, and DebiAN's RCE loss all target at up-weighting worse performed samples and down-weighting better-performed samples. The difference is how to compute the weight. LfF uses the ratio of cross-entropy loss between the biased model and the classifier to compute weights. Focal loss, as a hard negative method (see Appendix H.1), directly uses classifier's predicted probabilities to compute the weights. Different from previous methods, RCE loss uses *discoverer*'s predicted bias group assignment to compute the weights. Compared with LfF and focal, DebiAN achieves better debiasing results (Tab. 1-6 and Tab. 17).

H.6 Evaluation on Discovered Unknown Biases

In Fig. 6, Fig. 7, Fig. 10, and Fig. 11, we show some interesting unknown biases that human may not preconceive via saliency map. One may wonder if there exist other approaches to evaluate the results. First, it is hard to directly quantify the findings due to lack of annotations of the discovered bias attributes, *e.g.*, CelebA does not have attribute annotations or segmentation annotations of `visible hair area`. Using other datasets (*e.g.*, COCO [20]) with more attribute or segmentation ground-truth may not help since the discovered unknown biases may still be out of the annotations. Second, UDIS [14], a recent bias discovery method, also uses saliency maps to interpret the bias. We believe that using saliency maps is an established evaluation protocol in this task. Third, although it is hard to evaluate bias discovery in real-world dataset, our evaluation of bias discovery on Multi-Color MNIST (Fig. 5, Fig. 8, and Fig. 9) has shown that DebiAN achieves strong bias discovery results. Finally, we believe that our better debiasing results w.r.t. `Hair Length` bias attribute on the Transects dataset can also indirectly prove that *discoverer* identifies `visible hair area` bias (see Appendix F).

H.7 Why not add two colors to the foreground in Multi-Color MNIST?

The reason is that foreground digits are not always well aligned to the center of the images. If we assign two colors to the foreground digit based on whether the foreground is on the left or right, we may encounter cases where the digit is mainly on the right and only has a tiny area on the left, *e.g.* an italic digit "1." Thus, we choose to add colors to the background.

H.8 Difference between Multi-Color MNIST and Biased MNIST

Shrestha *et al.* [27] recently proposed the Biased MNIST dataset, which contains seven biases. However, all seven biases in Biased MNIST share the same bias-aligned ratio (*i.e.*, 0.7). In contrast, our Multi-color MNIST contains two biases

that are in different bias-aligned ratios, which we believe is more common in real-world scenarios and can better reveal the failure modes of existing debiasing methods. For example, while LfF performs the best in the Biased MNIST benchmark, our Multi-Color MNIST dataset reveals that LfF can only discover the more salient bias—the bias with a larger bias-align ratio.

H.9 Why evaluate on scene classification task?

First, we regard that scene classification as a core vision task on par with object classification. Second, while many previous debiasing works create datasets [12,23] that contain a single bias (*e.g.*, artificially introducing the spurious correlation w.r.t. a single bias), we believe that the classical cross-dataset generalization evaluation approach [28] does not have the single-bias assumption. The subgroup distribution w.r.t. multiple biases may vary across different datasets, which is closer to the real-world setting.

H.10 Limitations and Future Directions

We list some limitations that DebiAN has not fully resolved. First, we only assume that the bias attribute is binary or continuously valued from 0 to 1 (*i.e.*, two bias attribute groups). Future works can focus on extending DebiAN to discover and mitigate unknown biases with more than two groups. Second, DebiAN can only discover the biases caused by spurious correlation rather than lack of coverage. For example, suppose a face image dataset only contains long-hair female images and does not contain any short-hair female images. In that case, DebiAN cannot discover the `hair length` bias attribute because the *discoverer* does not have samples to categorize female images into two groups in terms of the `hair length` bias attribute. Finally, in terms of interpreting the discovered biases, DebiAN's approach, using the saliency maps on real-world images, is not as easy as interpreting biases from synthesized counterfactual images [15,18]. Future works can further explore better interpreting the discovered unknown biases on real-world images.

H.11 Potential Negative Social Impact

One potential negative social impact is that DebiAN's discovered biases could be used as a way to choose real-world images as the adversarial images to attack visual models in some safety-critical domains, *e.g.*, self-driving cars. Therefore, we encourage the defender to use DebiAN to mitigate the biases as the defense strategy.

Since our bias discovery approach relies on the fairness criterion based on equations, *e.g.*, equal true positive rates among two groups, our method cannot identify the biases that a fairness criterion cannot capture, *e.g.*, discrimination against the historically disadvantaged group. To mitigate this issue, we include a model card [22] in the released code to clarify that our method's intended use case

is discovering and mitigating biases that violate the equal opportunity fairness criterion [8], and the model's out-of-scope use case is identifying or mitigating other biases that cannot be captured by the equation of a fairness criterion, *e.g.*, discrimination against the historically disadvantaged group [7].

References

1. Ahmed, F., Bengio, Y., van Seijen, H., Courville, A.: Systematic generalisation with group invariant predictions. In: International Conference on Learning Representations (2021)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant Risk Minimization. arXiv:1907.02893 [cs, stat] (2020)
3. Bahng, H., Chun, S., Yun, S., Choo, J., Oh, S.J.: Learning De-biased Representations with Biased Representations. In: International Conference on Machine Learning (2020)
4. Balakrishnan, G., Xiong, Y., Xia, W., Perona, P.: Towards causal benchmarking of bias in face analysis algorithms. In: The European Conference on Computer Vision (ECCV) (2020)
5. Creager, E., Jacobsen, J.H., Zemel, R.: Environment Inference for Invariant Learning. In: International Conference on Machine Learning (2021)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
7. Hanna, A., Denton, E., Smart, A., Smith-Loud, J.: Towards a critical race methodology in algorithmic fairness. In: Conference on Fairness, Accountability, and Transparency (2020)
8. Hardt, M., Price, E., Srebro, N.: Equality of Opportunity in Supervised Learning. In: Advances in Neural Information Processing Systems (2016)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
11. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning Not to Learn: Training Deep Neural Networks With Biased Data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Kim, E., Lee, J., Choo, J.: BiaSwap: Removing dataset bias with bias-tailored swapping augmentation. In: The IEEE International Conference on Computer Vision (ICCV) (2021)
13. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations (2015)
14. Krishnakumar, A., Prabhu, V., Sudhakar, S., Hoffman, J.: UDIS: Unsupervised Discovery of Bias in Deep Visual Recognition Models. In: British Machine Vision Conference, BMVC (2021)
15. Lang, O., Gandelsman, Y., Yarom, M., Wald, Y., Elidan, G., Hassidim, A., Freeman, W.T., Isola, P., Globerson, A., Irani, M., Mosseri, I.: Explaining in Style: Training a GAN to explain a classifier in StyleSpace. In: The IEEE International Conference on Computer Vision (ICCV) (2021)

16. Lee, J., Kim, E., Lee, J., Lee, J., Choo, J.: Learning Debiased Representation via Disentangled Feature Augmentation. In: Advances in Neural Information Processing Systems (2021)
17. Li, Y., Vasconcelos, N.: REPAIR: Removing Representation Bias by Dataset Resampling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
18. Li, Z., Xu, C.: Discover the Unknown Biased Attribute of an Image Classifier. In: The IEEE International Conference on Computer Vision (ICCV) (2021)
19. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: The European Conference on Computer Vision (ECCV) (2014)
21. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep Learning Face Attributes in the Wild. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
22. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T.: Model Cards for Model Reporting. In: ACM Conference on Fairness, Accountability, and Transparency (2019)
23. Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J.: Learning from Failure: Training Debiased Classifier from Biased Classifier. In: Advances in Neural Information Processing Systems (2020)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems (2019)
25. Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., Weinberger, K.Q.: On Fairness and Calibration. In: Advances in Neural Information Processing Systems (2017)
26. Reddy, C., Sharma, D., Mehri, S., Romero-Soriano, A., Shabanian, S., Honari, S.: Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics. In: Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021)
27. Shrestha, R., Kafle, K., Kanan, C.: An Investigation of Critical Issues in Bias Mitigation Techniques. In: The IEEE Winter Conference on Applications of Computer Vision (WACV) (2022)
28. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
29. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. arXiv:1506.03365 [cs] (2016)
30. Zhang, Z., Sabuncu, M.: Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In: Advances in Neural Information Processing Systems (2018)
31. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)

Table 10: Ablation study on the ratios on Multi-Color MNIST dataset. Top-1 accuracy results are bolded, and the lowest accuracy results are underlined

left color ratio = 0.995	right color ratio = 0.95	vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	96.3$_{\pm 0.5}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	**98.7**$_{\pm 0.6}$	7.6$_{\pm 1.0}$	98.4$_{\pm 0.2}$	92.2$_{\pm 11.1}$	98.1$_{\pm 0.4}$
bias-conflicting	bias-aligned	6.5$_{\pm 1.0}$	**96.5**$_{\pm 1.6}$	46.8$_{\pm 0.5}$	27.7$_{\pm 14.2}$	55.4$_{\pm 2.1}$
bias-conflicting	bias-conflicting	2.0$_{\pm 0.4}$	5.9$_{\pm 1.3}$	7.4$_{\pm 0.1}$	6.3$_{\pm 2.4}$	**9.2**$_{\pm 0.8}$
unbiased		51.8$_{\pm 0.2}$	51.6$_{\pm 0.6}$	63.2$_{\pm 0.1}$	56.5$_{\pm 4.3}$	**65.7**$_{\pm 0.7}$
left color ratio = 0.99	right color ratio = 0.95	vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	99.6$_{\pm 0.5}$	**100.0**$_{\pm 0.0}$	98.6$_{\pm 2.3}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	97.1$_{\pm 0.5}$	4.7$_{\pm 0.5}$	**97.2**$_{\pm 1.5}$	82.6$_{\pm 19.6}$	95.6$_{\pm 0.8}$
bias-conflicting	bias-aligned	27.5$_{\pm 3.6}$	**98.6**$_{\pm 0.4}$	70.8$_{\pm 4.9}$	26.6$_{\pm 5.5}$	76.5$_{\pm 0.7}$
bias-conflicting	bias-conflicting	5.2$_{\pm 0.4}$	5.1$_{\pm 0.4}$	10.9$_{\pm 0.8}$	9.5$_{\pm 3.2}$	**16.0**$_{\pm 1.8}$
unbiased		57.4$_{\pm 0.7}$	52.0$_{\pm 0.1}$	69.7$_{\pm 1.0}$	54.3$_{\pm 4.0}$	**72.0**$_{\pm 0.8}$
left color ratio = 0.98	right color ratio = 0.95	vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	99.0$_{\pm 1.7}$	**100.0**$_{\pm 0.0}$	89.0$_{\pm 19.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	96.6$_{\pm 1.2}$	9.7$_{\pm 0.7}$	96.0$_{\pm 0.3}$	78.6$_{\pm 32.4}$	**97.1**$_{\pm 0.8}$
bias-conflicting	bias-aligned	64.4$_{\pm 2.3}$	**98.3**$_{\pm 0.9}$	84.0$_{\pm 0.6}$	69.5$_{\pm 27.7}$	85.1$_{\pm 3.4}$
bias-conflicting	bias-conflicting	12.4$_{\pm 1.1}$	11.5$_{\pm 1.1}$	16.0$_{\pm 1.7}$	16.4$_{\pm 1.1}$	**19.4**$_{\pm 1.3}$
unbiased		68.3$_{\pm 1.4}$	54.6$_{\pm 0.5}$	74.0$_{\pm 0.5}$	63.42$_{\pm 19.3}$	**75.4**$_{\pm 0.9}$
left color ratio = 0.95	right color ratio = 0.95	vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	93.4$_{\pm 5.8}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	91.1$_{\pm 2.3}$	71.1$_{\pm 33.7}$	92.7$_{\pm 0.5}$	76.5$_{\pm 17.8}$	**94.7**$_{\pm 0.9}$
bias-conflicting	bias-aligned	87.0$_{\pm 3.7}$	69.8$_{\pm 25.9}$	90.0$_{\pm 1.1}$	74.4$_{\pm 17.7}$	**92.7**$_{\pm 1.3}$
bias-conflicting	bias-conflicting	26.0$_{\pm 1.3}$	**39.6**$_{\pm 6.9}$	34.7$_{\pm 3.3}$	15.8$_{\pm 4.7}$	**39.6**$_{\pm 0.2}$
unbiased		76.0$_{\pm 1.6}$	68.5$_{\pm 3.2}$	79.3$_{\pm 0.7}$	66.7$_{\pm 10.0}$	**81.8**$_{\pm 0.6}$

Table 11: Ablation study on alternate training on Multi-Color MNIST dataset

left color ratio = 0.99	right color ratio = 0.95	w/o alternate training	DebiAN
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	**97.3**$_{\pm 0.1}$	95.6$_{\pm 0.8}$
bias-conflicting	bias-aligned	74.0$_{\pm 0.3}$	**76.5**$_{\pm 0.7}$
bias-conflicting	bias-conflicting	11.5$_{\pm 0.7}$	**16.0**$_{\pm 1.8}$
unbiased accuracy		70.7$_{\pm 0.1}$	**72.0**$_{\pm 0.8}$

Table 12: Results of ablation study on *discoverer*'s outputs. Bolded methods are used to report results in the main paper. For binary classification (*i.e.*, age classification on bFFHQ dataset) task, there is no significant difference whether or not the *discoverer* predicts bias attribute groups per class or globally. When it comes to the multi-class digit classification on Multi-Color MNIST dataset, predicting bias attribute groups per class has better accuracy results on the images that are bias-conflicting w.r.t. both `left color` and `right color` bias attributes

dataset	#classes	per class	**DebiAN (global)**
bFFHQ	2	$62.80_{\pm 0.60}$	$62.87_{\pm 0.61}$
dataset	#classes	global	**DebiAN (per class)**
Multi-Color MNIST	10	$13.5_{\pm 0.1}$	$\mathbf{16.0}_{\pm 1.8}$

Table 13: Results of an alternative design for debiasing in DebiAN – D and C play the minmax game (see DebiAN (minmax)) on Multi-Color MNIST dataset. Our RCE loss significantly outperforms the alternative design

left color ratio = 0.99	right color ratio = 0.95	DebiAN (minmax)	**DebiAN (RCE)**
bias-aligned	bias-aligned	$97.3_{\pm 4.6}$	$\mathbf{100.0}_{\pm 0.0}$
bias-aligned	bias-conflicting	$\mathbf{95.7}_{\pm \mathbf{0.8}}$	$95.6_{\pm 0.8}$
bias-conflicting	bias-aligned	$64.5_{\pm 2.0}$	$\mathbf{76.5}_{\pm 0.7}$
bias-conflicting	bias-conflicting	$7.7_{\pm 1.9}$	$\mathbf{16.0}_{\pm 1.8}$
unbiased accuracy		$66.3_{\pm 0.9}$	$\mathbf{72.0}_{\pm 0.8}$

Table 14: Results on Colored MNIST (foreground color) under different ratios of bias-aligned samples in the training set. We bold top-1 results (except LfF's results reported in the original paper since they cannot be replicated by their officially released code) and underline the lowest results based on the mean value. Although LfF achieves better bias-conflicting accuracy, it achieves lower bias-aligned accuracy, resulting in low unbiased accuracy. Overall, DebiAN achieves comparable or slightly lower unbiased accuracy results compared with other methods

ratio		vanilla	LfF (reported in the paper)	LfF (replicate via official code)	EIIL	PGI	**DebiAN (Ours)**
0.995	bias-aligned	$100.00_{\pm 0.00}$	-	$\underline{54.13_{\pm 6.33}}$	$99.90_{\pm 0.01}$	$99.86_{\pm 0.11}$	$\mathbf{100.00}_{\pm 0.00}$
	bias-conflicting	$7.92_{\pm 4.68}$	$63.49_{\pm 1.94}$	$\mathbf{57.11}_{\pm 6.22}$	$\underline{24.63_{\pm 0.37}}$	$27.01_{\pm 5.49}$	$24.83_{\pm 1.83}$
	unbiased accuracy	$\underline{53.96_{\pm 2.34}}$	$63.39_{\pm 1.97}$	$55.62_{\pm 6.26}$	$62.27_{\pm 1.85}$	$\mathbf{64.44}_{\pm 2.78}$	$62.41_{\pm 0.91}$
0.99	bias-aligned	$\mathbf{99.97}_{\pm 0.05}$	-	$\underline{61.96_{\pm 3.29}}$	$99.81_{\pm 0.19}$	$99.86_{\pm 0.07}$	$99.86_{\pm 0.05}$
	bias-conflicting	$\underline{18.73_{\pm 2.78}}$	$74.19_{\pm 0.94}$	$\mathbf{67.20}_{\pm 4.58}$	$40.71_{\pm 1.93}$	$41.88_{\pm 0.99}$	$43.33_{\pm 0.86}$
	unbiased accuracy	$\underline{59.35_{\pm 1.40}}$	$74.01_{\pm 2.21}$	$64.58_{\pm 2.22}$	$70.26_{\pm 1.06}$	$70.87_{\pm 0.53}$	$\mathbf{71.60}_{\pm 0.44}$
0.98	bias-aligned	$\mathbf{99.80}_{\pm 0.16}$	-	$\underline{73.11_{\pm 5.41}}$	$99.77_{\pm 0.14}$	$99.73_{\pm 0.11}$	$99.76_{\pm 0.05}$
	bias-conflicting	$\underline{39.23_{\pm 1.63}}$	$80.67_{\pm 0.56}$	$\mathbf{78.23}_{\pm 1.56}$	$54.44_{\pm 1.26}$	$57.46_{\pm 1.13}$	$55.46_{\pm 0.71}$
	unbiased accuracy	$\underline{69.52_{\pm 0.90}}$	$80.48_{\pm 0.45}$	$75.67_{\pm 2.95}$	$77.10_{\pm 0.55}$	$\mathbf{78.60}_{\pm 0.55}$	$77.61_{\pm 0.37}$
0.95	bias-aligned	$99.60_{\pm 0.17}$	-	$\underline{71.67_{\pm 0.68}}$	$99.61_{\pm 0.25}$	$99.37_{\pm 0.31}$	$\mathbf{99.70}_{\pm 0.17}$
	bias-conflicting	$70.99_{\pm 2.45}$	$85.77_{\pm 0.66}$	$\mathbf{82.37}_{\pm 1.49}$	$73.01_{\pm 1.03}$	$\underline{70.63_{\pm 2.24}}$	$73.04_{\pm 2.20}$
	unbiased accuracy	$85.30_{\pm 1.30}$	$85.39_{\pm 0.94}$	$\underline{77.02_{\pm 4.11}}$	$86.31_{\pm 0.46}$	$85.00_{\pm 1.05}$	$\mathbf{86.37}_{\pm 1.10}$

Table 15: Results on Colored MNIST (background color) under different ratios of bias-aligned samples in the training set. We bold top-1 results and underline the lowest results based on the mean value. Similar to the results in Colored MNIST (foreground color) (Tab. 14), although LfF achieves better bias-conflicting accuracy, it achieves lower bias-aligned accuracy, resulting in low unbiased accuracy. Overall, DebiAN achieves comparable or slightly higher unbiased accuracy results compared with other methods

ratio		vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
0.995	bias-aligned	99.97±0.05	42.98±1.67	**100.00**±0.00	**100.00**±0.00	**100.00**±0.00
	bias-conflicting	1.98±0.35	**61.37**±1.69	14.16±4.89	9.95±3.28	20.94±3.92
	unbiased accuracy	50.98±0.18	52.17±1.62	57.08±2.44	54.98±1.64	**60.47**±1.96
0.99	bias-aligned	**100.00**±0.00	52.24±1.84	99.97±0.05	99.26±1.10	**100.00**±0.00
	bias-conflicting	5.43±0.32	**67.94**±0.88	35.71±1.78	12.89±1.62	42.57±0.36
	unbiased accuracy	52.75±0.16	60.09±1.26	67.84±0.90	56.07±0.81	**71.29**±0.18
0.98	bias-aligned	**99.97**±0.05	71.90±4.86	99.90±0.09	97.83±3.54	99.90±0.00
	bias-conflicting	21.84±5.39	**80.28**±0.98	51.25±2.86	18.06±5.47	53.41±1.21
	unbiased accuracy	60.91±2.71	76.09±2.91	75.58±1.46	59.75±4.02	**76.66**±0.60
0.95	bias-aligned	99.87±0.15	69.71±7.07	99.90±0.16	96.78±4.90	**99.90**±0.01
	bias-conflicting	55.14±2.06	**81.67**±4.54	69.17±3.15	33.90±15.55	70.70±0.76
	unbiased accuracy	77.51±1.09	75.69±5.50	84.53±1.54	65.34±9.80	**85.30**±0.37

Table 16: Average group accuracy results of gender classification on Transects [4] dataset. All models are trained on CelebA dataset and evaluated on Transects w.r.t. two bias attributes—`Hair Length` and `Skin Color`. DebiAN achieves better results, which demonstrates that DebiAN better mitigate multiple biases simultaneously in the real-world multi-bias setting. Besides, it also reflects that DebiAN discovers `visible hair area` bias attribute to achieve better debiasing results w.r.t. `Hair Length` bias attribute

bias attribute	vanilla	LfF	EIIL	PGI	**DebiAN (Ours)**
`Hair Length`	55.1±5.8	54.7±2.9	54.0±0.4	56.2±1.3	**60.5**±1.7
`Skin Color`	53.5±5.3	53.3±2.9	53.1±0.08	57.4±0.3	**60.1**±1.2

Table 17: Comparing with focal loss [19], a hard negative method, on Multi-Color MNIST dataset

left color skew = 0.995	right color skew = 0.95	vanilla	focal	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	**98.7**$_{\pm 0.6}$	97.9$_{\pm 0.9}$	98.1$_{\pm 0.4}$
bias-conflicting	bias-aligned	6.5$_{\pm 1.0}$	0.4$_{\pm 0.2}$	**55.4**$_{\pm 2.1}$
bias-conflicting	bias-conflicting	2.0$_{\pm 0.4}$	1.2$_{\pm 0.3}$	**9.2**$_{\pm 0.8}$
unbiased		51.8$_{\pm 0.2}$	49.2$_{\pm 0.2}$	**65.7**$_{\pm 0.7}$
left color skew = 0.99	right color skew = 0.95	vanilla	focal	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	**97.1**$_{\pm 0.5}$	95.7$_{\pm 0.6}$	95.6$_{\pm 0.8}$
bias-conflicting	bias-aligned	27.5$_{\pm 3.6}$	3.3$_{\pm 2.0}$	**76.5**$_{\pm 0.7}$
bias-conflicting	bias-conflicting	5.2$_{\pm 0.4}$	2.4$_{\pm 0.3}$	**16.0**$_{\pm 1.8}$
unbiased		57.4$_{\pm 0.7}$	50.3$_{\pm 0.4}$	**72.0**$_{\pm 0.8}$
left color skew = 0.98	right color skew = 0.95	vanilla	focal	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	96.6$_{\pm 1.2}$	85.1$_{\pm 2.1}$	**97.1**$_{\pm 0.8}$
bias-conflicting	bias-aligned	64.4$_{\pm 2.3}$	15.9$_{\pm 4.4}$	**85.1**$_{\pm 3.4}$
bias-conflicting	bias-conflicting	12.4$_{\pm 1.1}$	6.0$_{\pm 0.1}$	**19.4**$_{\pm 1.3}$
unbiased		68.3$_{\pm 1.4}$	51.7$_{\pm 0.8}$	**75.4**$_{\pm 0.9}$
left color skew = 0.95	right color skew = 0.95	vanilla	focal	**DebiAN (Ours)**
bias-aligned	bias-aligned	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$	**100.0**$_{\pm 0.0}$
bias-aligned	bias-conflicting	91.1$_{\pm 2.3}$	63.7$_{\pm 3.8}$	**94.7**$_{\pm 0.9}$
bias-conflicting	bias-aligned	87.0$_{\pm 3.7}$	54.4$_{\pm 4.1}$	**92.7**$_{\pm 1.3}$
bias-conflicting	bias-conflicting	26.0$_{\pm 1.3}$	11.3$_{\pm 0.1}$	**39.6**$_{\pm 0.2}$
unbiased		76.0$_{\pm 1.6}$	57.3$_{\pm 1.2}$	**81.8**$_{\pm 0.6}$