BiaSwap: Removing Dataset Bias with Bias-Tailored Swapping Augmentation

Eungyeup Kim* KAIST eykim94@kaist.ac.kr
Jihyeon Lee* KAIST jihyeonlee@kaist.ac.kr
Jaegul Choo KAIST jchoo@kaist.ac.kr

Abstract

Deep neural networks often make decisions based on the spurious correlations inherent in the dataset, failing to generalize in an unbiased data distribution. Although previous approaches pre-define the type of dataset bias to prevent the network from learning it, recognizing the bias type in the real dataset is often prohibitive. This paper proposes a novel bias-tailored augmentation-based approach, BiaSwap, for learning debiased representation without requiring supervision on the bias type. Assuming that the bias corresponds to the easy-to-learn attributes, we sort the training images based on how much a biased classifier can exploits them as shortcut and divide them into bias-guiding and bias-contrary samples in an unsupervised manner. Afterwards, we integrate the style-transferring module of the image translation model with the class activation maps of such biased classifier, which enables to primarily transfer the bias attributes learned by the classifier. Therefore, given the pair of bias-guiding and bias-contrary, BiaSwap generates the bias-swapped image which contains the bias attributes from the bias-contrary images, while preserving bias-irrelevant ones in the bias-guiding images. Given such augmented images, BiaSwap demonstrates the superiority in debiasing against the existing baselines over both synthetic and real-world datasets. Even without careful supervision on the bias, BiaSwap achieves a remarkable performance on both unbiased and bias-guiding samples, implying the improved generalization capability of the model.

1. Introduction

Recent deep neural networks have shown remarkable performances in computer vision tasks including classification and object detection. However, these models often achieve their goals by erroneously relying on the peripheral features that have spurious correlations with their labels, so-called *dataset bias* [1]. For instance, imagine a classifier for recognizing a *camel*, when most of the camels in the training images appear in the desert. This unintended correlation causes the classifier to overly rely on the attributes of the desert, failing to recognize the camel standing on the road. In other words, the classifier trained on the biased dataset often shows drastic failures for the images without such bias, which raises a question on its generalization capability in unbiased image classification.

Existing methods attempt to address this issue by using an explicit definition of the bias type in their debiasing strategies. Some approaches [2, 3, 4] assume the texture bias in the image classification task and propose a hand-crafted module for such bias type. Similarly, textual modality, *i.e.,* question and answer, are pre-defined as the bias [5, 6] in visual question answering task and resolved by leveraging the question-only network.

However, assuming an already-known bias is quite unrealistic in that bias attributes can vary according to the composition of the training dataset. Moreover, unlike synthetic datasets where bias attributes are manually designated, *e.g.,* color is set as bias in Colored MNIST as shown in Figure 2-(a), it is highly demanding to acquire the prior knowledge on the bias that inherently exists in the real-world dataset. Therefore, an unsupervised debiasing with no definition in advance is an appropriate approach for learning generalized representations over various datasets. Moreover, maintaining the classification ability for biased samples as well as unbiased samples needs to be considered crucial for a desirable representation, which has been overlooked by the previous studies [6, 7].

This paper mainly focuses on 1) removing the dataset bias without explicit supervision by leveraging the bias-tailored swapping augmentation, and 2) achieving superior performance on bias-contrary as well as bias-guiding samples against other baselines. We propose an image translation-based augmentation framework, *BiaSwap*, which transfers the attributes appearing on the regions of the image where the classifier often exploits as a shortcut for prediction. To this end, we first exploit the reasonable observation that the biased classifier often learns to exploit *easy-to-learn* attributes in the early learning phase, proposed in Nam *et al*. [7] This enables us to obtain the class activation map (CAM) [8] which indicates the bias-relevant regions for each image without requiring an explicit definition of the bias type in advance. By integrating the CAM

* indicates equal contribution

into the image translation framework, we augment the images with their bias attributes being translated by those of another exemplar image. At the same time, we present a simple and intuitive criterion based on the same assumption (*i.e.,* bias is *easy-to-learn*) for discriminating between the bias-guiding and the bias-contrary samples among the training set. Therefore, given the pairs, BiaSwap mainly translates the bias-guiding image into the bias-contrary one by transferring the specific attributes corresponding to the bias. These augmented images, termed as *bias-swapped*, make the proportion of bias-guiding images to be less dominant in the training dataset, removing the dataset bias in the end. We provide extensive experiments representing that BiaSwap achieves the state-of-the-art debiasing results against the baselines across various datasets from synthetic (*i.e.,* Colored MNIST, Corrupted CIFAR10) to real-world (*i.e.,* BAR, bFFHQ) datasets, even without explicit supervision on the bias type.

2. Preliminaries

In this section, we first provide the formulation of the dataset bias (Section 2.1). Afterward, we categorize the various existing debiasing approaches in terms of prior assumption on bias type (Section 2.2).

2.1. Definition of unwanted correlation in dataset

Consider a training dataset $\mathcal{D}$ where each image $x \in \mathcal{X}$ has its corresponding class label $y \in \mathcal{Y}$. Each x can be explained by its various visual attributes, such as shape and color, and some of them are exploited by a classifier in the image classification task. Among these attributes, let z_g the one that is essential for predicting a target label y, meaning that every image for class y must contain z_g. Therefore, a classifier becomes generalized in the unbiased distribution when learning this attribute as a cue. In contrast, let z_b denote the attribute which is less essential, but have a strong correlation with target label y. In addition, z_b often acts as the bias attribute when it is easier for the classifier to learn compared to z_g. Eventually, the model becomes biased by overly exploiting the z_b instead of z_g when trained in the biased dataset, failing to predict the samples which do not contain the z_b. For example, in the Colored MNIST, most of the images in each class are highly correlated with the specific color, as illustrated in Figure 2-(a). On the other hand, an unbiased test set contains the samples whose colors are uniform at random, having no correlation with their target label. In this case, attributes z_g corresponds to the digit, while z_b indicates the color in each image. Throughout the paper, we term z_b *bias-guiding* attribute and the image containing the z_b as bias-guiding image. While most of the samples with the same class in the training distribution share the z_b, there might be a small portion of samples that have attributes that are conflicting to z_b, which we term z_{-b}. For example, in the Colored MNIST, while most of the samples in class 0 contain *red*, a few samples contain non-red color, such as blue or green. As this z_{-b} attribute is contradictory against z_b, the biased network cannot rely on it anymore. We term z_{-b} *bias-contrary* attribute, and the image with z_{-b} as bias-contrary image.

Since the bias-guiding samples with z_b are dominant in the training dataset, it leads the classifier to rely on z_b rather than the essential attribute z_g. Therefore, removing the dataset bias by increasing the proportion of the bias-contrary samples with z_{-b} can encourage the model to learn z_g by preventing it from relying solely on the z_b for classification. Our proposed image translation-based augmentation approach generates the images with their visual aspects of z_b being transferred into z_{-b}, while maintaining the essential features z_g. We term this augmented sample as a *bias-swapped* image. This, as a result, leads our classifier to achieve consistent performance in unbiased dataset distribution, where most of the samples are bias-contrary.

2.2. Existing debiasing approaches

Remove bias with prior knowledge Several approaches with an explicit label on the bias type have been proposed [9, 10, 11, 12, 13]. Li and Vasconcelos [10] and Kim *et al*. [9] set the particular RGB values to be a bias cue in the Colored MNIST dataset, where a specific color is correlated with each digit. Agarwal *et al*. [11] propose to synthesize the data with a generative algorithm by involving manually curated heuristics which selects the objects to remove. Besides, Sagawa *et al*. [12] and Goel *et al*. [13] utilize the clustering of the bias subgroups which require expensive supervision on the bias type.

Other approaches pre-define the bias type in advance and build a bias-tailored module for addressing the certain bias type [3, 2, 4, 5, 6, 14, 15, 16]. Wang *et al*. [2] assume the texture bias in the image classification task, and propose a projection method in the latent space to learn the independent features from the texture-biased ones. Geirhos *et al*. [4] propose a style transfer-based augmentation method with adaptive instance normalization [17], which enhances the robustness against the texture bias. Bahng *et al*. [3] introduce the model with limited capacity for capturing the texture bias in image classification or static bias in video action recognition, respectively, and propose the learning of the statistically independent representation against it.

However, these approaches have limitations in that assuming the certain type of bias does not guarantee the generalized debiasing in the dataset with other types of bias. As the bias-guiding attributes z_b is determined by the characteristics of dataset, such as the composition of images and the attribute complexity, learning debiased representation without prior assumption on the certain bias type is essential.

Remove bias without explicit supervision Still, learning

debiased representation in an unsupervised manner is an ideal but demanding problem. Darlow *et al*. [18] utilize the adversarial perturbation in the latent space for synthesizing the images against the bias that the classifier learns. Nam *et al*. [7] observe the general aspect of bias as easy-to-learn in the early training phase and adopt the generalized cross-entropy loss [19] to train a biased network. The samples in which such biased network fails to classify are then emphasized through weighted cross-entropy loss in the training of the debiased network.

A truly debiased classifier learns the generalized attribute z_g, which should correctly classify the samples in the unbiased as well as the biased dataset. However, existing baselines [6, 7] often suffer from the significant performance degradation in the biased dataset (*i.e.*, bias-guiding samples). This implies that they implicitly learn to *avoid* the bias-guiding attributes, not fully learning the z_g. Learning debiased representation without bias supervision remains challenging, and is thus fairly under-explored. Our proposed bias-tailored augmentation effectively removes the dataset bias, achieving the generalized debiasing capability in both biased and unbiased test set.

3. Proposed Approach

This section provides a detailed description of distinguishing a bias-guiding and bias-contrary samples (Section 3.1), training a bias-tailored swapping autoencoder (Section 3.2), and training a classifier with debiased representation (Section 3.3).

3.1. Separation of bias-contrary samples

We propose a simple, yet effective method that divides the training samples into bias-guiding and bias-contrary groups. Our method assigns the bias label for the images adaptively according to the training dataset, without the explicit supervision on the bias. As mentioned in Section 2.1, bias-guiding samples have the unwanted correlations that are *easy-to-learn* [7, 20], while bias-contrary samples are *hard-to-learn*. Therefore, a biased classifier becomes *certain* and *correct* for the bias-guiding samples. In contrast, as the bias-contrary samples do not include the attributes the classifier mainly relies on, the classifier can be either 1) *certain* and *incorrect* or 2) *uncertain* for the bias-contrary samples. Based on these characteristics, we introduce a *pseudo-bias label* which determines whether each image belong to bias-guiding or bias-contrary by observing both the classification correctness and the confidence of the model for the image.

To distinguish the binary category (*i.e.,* bias-contrary or bias-guiding) more certainly, we first train the biased classifier f_{bias} by exploiting the generalized cross-entropy (GCE) loss [19] in a similar manner to Nam *et al*. [7] GCE loss is originally proposed as a noise-robust alternative to the categorical cross-entropy (CE) loss. In our setting, it amplifies the biased representation since its gradient is written as $\frac{\partial GCE(p,y)}{\partial \theta} = \frac{p_y^q \partial CE(p,y)}{\partial \theta}$ where p_y is probability corresponding to the target label y, $q \in (0, 1]$ is a hyperparameter, and θ denotes the network parameters. GCE loss puts greater importance on the samples which are easy to learn compared to the CE loss. As these samples are bias-guiding in our training dataset, our classifier becomes biased.

Assume the biased classifier f_{bias} outputs the resulting logits $\mathbf{z} = (z_1, ..., z_K) \in \mathbb{R}^K$ after the last linear layer, where K denotes the number of the target class. We first define the bias score of each sample x by obtaining the absolute difference between the correctness and the max values of probability described as

$$\text{score}(x) = \left| \mathbb{1}_{\text{argmax}_k f_{\text{bias}}(x)=y} - \texttt{max}\left(\frac{e^{f_{\text{bias}}(x)}}{\sum_{j=1}^{K} e^{f_{\text{bias}}(x)_j}} \right) \right|, \tag{1}$$

where $\mathbb{1}$ is an indicator function which outputs one when satisfying the given condition and zero in vice versa, $\texttt{max}$ returns the maximum probability value after the softmax operation, and $f_{\text{bias}}(x)_j$ as the j-th logit values of the biased classifier. For the bias-guiding image which contains z_b, the model correctly predicts the target label y (*i.e.,* first term in Eq. 1 becomes 1) with high confidence (*i.e.,* second term in Eq. 1 becomes high), resulting the calculated score to be close to 0. Contrariwise, the score becomes close to 1 when the model makes the wrong prediction (*i.e.,* first term in Eq. 1 becomes 0) with high confidence (*i.e.,* second term in Eq. 1 becomes high), which might be mostly observed for the bias-contrary samples. In addition, for the occasional cases where the classifier correctly predicts the bias-contrary images with low confidence, the score would be placed between 0 and 1. Given such score, we determine the pseudo bias label $\tilde{y}_{\text{bias}}(x)$ for each data which is

$$\tilde{y}_{\text{bias}}(x) = \begin{cases} 1 & \text{if } \text{score}(x) > \frac{1}{N} \sum_{i=0}^{N} \text{score}(x_i) \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where y denotes a ground-truth target label and N as the total number of training images. We consider the image assigned with $\tilde{y}_{\text{bias}} = 0$ as bias-guiding and that with $\tilde{y}_{\text{bias}} = 1$ as bias-contrary. We adopt the arithmetic mean of the scores over the entire samples as the threshold for determining whether each sample is bias-guiding or bias-contrary. We empirically find that such mean values of the scores can act as a simple and effective threshold for discriminating the bias-guiding images and bias-contrary images. In Section 4.2, we validate this simple criterion reasonably performs over the various datasets utilized in the paper.

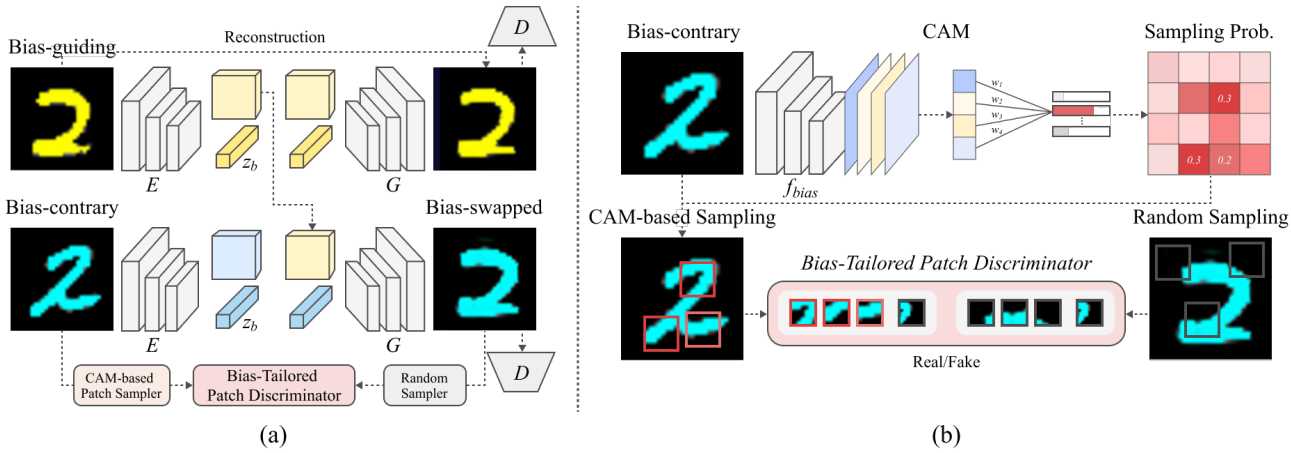


Figure 1: Illustration of the proposed method, BiaSwap. The figure (a) shows the overall pipeline of the swapping augmentation framework and the figure (b) describes the patch samplers and bias-tailored patch discriminator in detail. We generate the bias-swapped images from this framework to augment the training dataset for learning debiased representation.

3.2. Bias-tailored swapping autoencoder

Given the pair of bias-guiding and bias-contrary images using the $\tilde{y}_{\text{bias}}$ as shown in Figure 1-(a), we leverage the state-of-the-art image-to-image translation method called swapping autoencoder (SwapAE) [21] as our backbone network for translation. To enable the translation of the bias-aware attributes in the bias-aligned samples to be bias-contrary, we propose a novel variant of patch cooccurrence discriminator, which mainly focuses on bias attributes based on the class activation map (CAM) [8] of a biased classifier.

Swapping autoencoder SwapAE [21] consists of the encoder E which maps the image into the latent features z, and the generator G which reconstructs the images x from z. Specifically, E encodes the image into its content features z_c and style features z_s, and G takes them to synthesize the image which can be explained by them. SwapAE first utilizes the reconstruction loss and adversarial loss [22] for generating both realistic reconstruction of an input image x. Both losses are written as

$$\begin{aligned}\mathcal{L}_{\text{recon}}(E,G) &= \mathbb{E}_{x\sim\mathcal{X}}\left[\left\|x - G(E(x))\right\|_2^2\right], \\ \mathcal{L}_{\text{GAN,recon}}(E,G,D) &= \mathbb{E}_{x\sim\mathcal{X}}\left[-\log D(G(E(x)))\right],\end{aligned} \tag{3}$$

where $\mathcal{X}$ denotes a training dataset distribution and D a discriminator which classifies whether the image is real or fake. In addition, SwapAE learns to translate the style of an image, denoted as x^1, into that of another one, x^2, generating the translated image. This can be done by constructing the swapped pair of latent features from these images and decoding them into the image. In other words, each pair of (z_c^1, z_s^1) and (z_c^2, z_s^2) are encoded from x^1 and x^2, respectively, and the swapped pair, *i.e.,* (z_c^1, z_s^2), are decoded to generate the translated image, which contains the style of x^2 while maintaining the content of x^1. To ensure the translated images with swapped attributes to contain the same style with x^2, a patch cooccurrence discriminator D_{patch} is proposed. Such discriminator enforces the styles in the randomly sampled patches from the generated images to be identical to the ones in x^2. Therefore, the objective function can be written as

$$\begin{aligned}&\mathcal{L}_{\text{CooccurGAN}}(E,G,D_{\text{patch}}) = \\ &\mathbb{E}_{x^1,x^2\sim\mathcal{X}}\left[-\log(D_{\text{patch}}(\texttt{crop}_{\text{u}}(G(z_c^1,z_s^2)),\texttt{crops}_{\text{u}}(x^2)))\right],\end{aligned} \tag{4}$$

where $\texttt{crop}_{\text{u}}$ and $\texttt{crops}_{\text{u}}$ denote the operation of cropping *uniformly at random* in an image for a single patch and multiple patches, respectively. To make the generated images $G(z_c^1, z_s^2)$ realistic, the adversarial loss is added as

$$\begin{aligned}&\mathcal{L}_{\text{GAN,swap}}(E,G,D) = \\ &\mathbb{E}_{x^1,x^2\sim\mathcal{X},x^1\neq x^2}\left[-\log(D(G(z_c^1,z_s^2))\right].\end{aligned} \tag{5}$$

CAM-based patch sampling As D_{patch} randomly samples the patches from the entire spatial resolution, the style extracted from the patches does not reflect certain attributes. Instead, as we aim to transfer the styles corresponding to the attributes the classifier easily learns as shortcuts, sampling the patches related to these attributes are required. Therefore, we leverage the biased classifier f_{bias} and integrate its CAM [8], which identifies the discriminative regions used by such a classifier, into the patch sampling method in D_{patch}. Specifically, given an image, our classifier produces an activation map $f_{\text{bias,k}}(x,y)$, where k denotes a channel index and (x,y) the coordinate for the spatial location. Then, following Zhou *et al.* [8], we calculate a logit for each class c as $\sum_k w_k^c F_k$, where $F_{\text{bias,k}}$ denotes the result of global average pooling for channel k and w_k^c indicates a weight

which maps the $F_{\text{bias,k}}$ into each class probability. The logits for class c can be written as

$$\begin{aligned}\sum_k w_k^c F_{\text{bias,k}} &= \sum_k w_k^c \sum_{x,y} f_{\text{bias,k}}(x,y) \\ &= \sum_{x,y}\sum_k w_k^c f_{\text{bias,k}}(x,y).\end{aligned} \tag{6}$$

Therefore, the importance of the activation map at spatial location (x, y) for classifying the class c by the classifier f_{bias} can be presented as

$$I_c(x,y) = \sum_k w_k^c f_{\text{bias,k}}(x,y). \tag{7}$$

As the classifier is biased, the large value of $I_c(x, y)$ demonstrates the location where the bias attributes are highly obtained by the classifier. In this regard, we convert $I_c(x, y)$ into the sampling probability $P(x, y)$ for each spatial location of patch (x, y) and utilize such probability in the discriminator D_{patch} for style extraction, as shown in Figure 1-(b). In other words, instead of random `crop` operation in Eq. 4, we utilize the cropping of local patches according to the probability described as

$$P(x,y) = \frac{e^{I_c(x,y)}}{\sum_{x,y} e^{I_c(x,y)}}. \tag{8}$$

This encourages the more frequent sampling of patches corresponding to the bias attribute in the images compared to others, enabling the translation of bias attributes from an image to another. Therefore, the variant of the objective function of Eq. 4 via bias-tailored patch discriminator can be described as

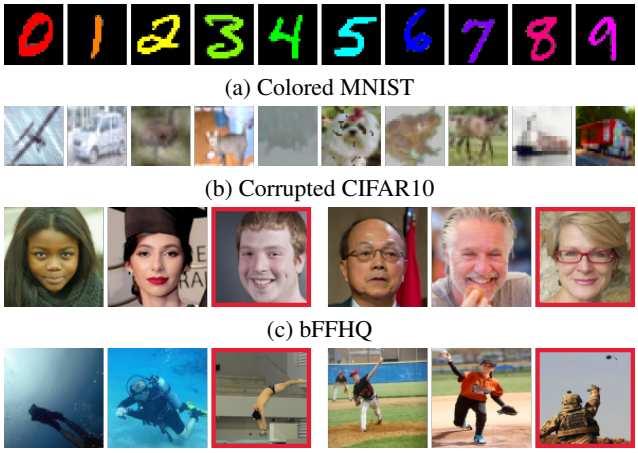
$$\begin{aligned}&\mathcal{L}_{\text{CooccurGAN}}(E,G,D_{\text{bias-tailored patch}}) = \\ &\mathbb{E}_{x^1,x^2\sim\mathcal{X}}\left[-\log(D_{\text{patch}}(\texttt{crop}_\text{b}(G(z_c^1,z_s^2)),\texttt{crops}_\text{b}(x^2)))\right],\end{aligned} \tag{9}$$

where $\texttt{crop}_\text{b}$ and $\texttt{crops}_\text{b}$ denote the operation of cropping under the probability of Eq. 8 in an image for a single patch and multiple patches, respectively. In consequence, BiaSwap generates the *bias-swapped* image, which contains the bias-relevant attributes from the bias-contrary image while preserving the bias-irrelevant features from the bias-guiding image, as shown in Figure 1-(a).

3.3. Training classifier with augmented dataset

By adding the generated bias-swapped images $\mathcal{X}_{\text{bias-swapped}}$, we can obtain our augmented training dataset $\mathcal{X}_{\text{aug}} = \mathcal{X} \cup \mathcal{X}_{\text{bias-swapped}}$. These reasonable amounts of bias-swapped samples in $\mathcal{X}$ alleviates the dataset bias caused by the dominant number of bias-guiding images in the dataset, thus preventing the model from learning biased representation. Finally, we train a classifier f_{debias} with these datasets with the classification loss as

$$\mathcal{L}_{\text{class}} = \mathbb{E}_{x\sim\mathcal{X}_{\text{aug}}}\left[-\sum_c y_c \log f_{\text{debias}}(x)\right]. \tag{10}$$



(a) Colored MNIST

(b) Corrupted CIFAR10

(c) bFFHQ

(d) BAR

Figure 2: Example images of each dataset we utilize in the paper. Rows (a) and (b) represent the bias-guiding samples which have a strong correlation between bias attribute and the target class. For rows (c) and (d), we additionally visualize the bias-contrary images with red boxes, which do not contain such correlation.

4. Experiments and Analysis

In Section 4.1, we first introduce the experimental setup, including the details of the biased datasets and implementation details. Afterward, Section 4.2 and Section 4.3 provide the quantitative and qualitative comparison between our method with existing baselines on synthetic and real-world datasets, respectively.

4.1. Experimental setup

We evaluate our method as well as the baselines across the synthetic dataset, *i.e.,* Colored MNIST, and Corrupted CIFAR10 [23], which are widely used in the previous literature. We also utilize the real-world datasets including BAR [7] and bFFHQ.

Datasets As shown in Fig. 2, Colored MNIST is an MNIST dataset [24] which has a correlation with certain colors. To inject the color bias, we select 10 distinct colors and inject each color into the MNIST images with a certain digit label (*e.g.,* red color for images of zero label). Bias-contrary samples have their colors sampled uniformly at random. Corrupted CIFAR10 is the CIFAR10 [25] dataset with texture corruptions, as proposed in Hendrycks and Dietterich [23]. Similar to Colored MNIST, each texture corruption has an injurious correlation with each object class. We newly construct the Gender-biased FFHQ dataset (bFFHQ) which has

Dataset	%	Bias-guiding				Unbiased			
		Vanilla	ReBias	LfF	BiaSwap	Vanilla	ReBias	LfF	BiaSwap
Colored MNIST	95.0	99.23	**100.0**	80.33	97.95	79.54	**96.28**	84.72	90.85
	98.0	99.80	**100.0**	69.14	98.33	62.62	**90.16**	75.88	85.29
	99.0	99.74	**100.0**	62.33	98.45	48.76	**84.19**	70.05	83.74
	99.5	99.44	**99.9**	72.85	98.49	32.67	62.82	61.61	**85.76**
Corrupted CIFAR10	95.0	**99.05**	98.23	62.74	95.53	35.68	**45.49**	42.32	41.62
	98.0	**98.97**	98.67	73.41	94.82	29.68	31.52	35.23	**35.25**
	99.0	98.79	**99.11**	74.73	96.98	24.51	25.04	29.27	**32.54**
	99.5	98.56	**99.29**	80.90	96.82	23.12	20.49	27.10	**29.11**

Table 1: Quantitative comparisons of bias-guiding and unbiased test accuracy on two synthetic datasets. Note that each method has a different supervision level. The methods with yellow background assume the bias type in advance, while those with blue do not require such type. We denote the best score with bold and the best score among unsupervised methods with under-lined scores.

age as a target label and gender as a correlated bias, and the images are from the FFHQ dataset [26]. The images include the dominant number of young women (*i.e.*, aged 10-29) and old men (*i.e.*, aged 40-59) in the training data. The biased action recognition (BAR) dataset is categorized by six human-action classes that are correlated with the distinct places [7]. Curated six typical action-place pairs are *(Climbing, RockWall), (Diving, Underwater), (Fishing, WaterSurface), (Racing, A PavedTrack), (Throwing, PlayingField)*, and *(Vaulting, Sky)*. For the experiments on synthetic datasets, we vary the ratio of bias-guiding samples which are 95.0%, 98.0%, 99.0%, and 99.5%. For bFFHQ, we utilize 99.0% of bias-guiding images. For BAR, we utilize typical action-place paired images for training, and bias-contrary ones only belong to the evaluation set. Although BAR only contains the bias-guiding training samples, undoubtedly there exist relatively easier samples, *i.e.,* more bias-guiding, than the others *i.e.,* less bias-guiding, in our proposed framework.

Evaluation sets To measure the generalization capability of the debiasing method, we consider the two types of evaluation sets, the unbiased and bias-guiding sets. The unbiased evaluation set is constructed in a way that the bias attributes are distributed uniformly at random among the data without any correlation with a certain target label, following the evaluation protocol of existing studies [3, 7, 18]. This set mainly evaluates how the debiasing method correctly classifies the bias-contrary test samples which do not include the strong correlation. Note that for the real-world datasets, we exclude the bias-guiding samples from the unbiased test set, and call the remaining ones as "bias-contrary" test set, as did in LfF [7]. In contrast, the bias-guiding set is composed of the bias-guiding images from the same distribution of the biased training dataset. Such an evaluation set enables us to evaluate how the debiasing method maintains the classification capability for the bias-aligned test images after learning the debiased representation. We believe that the truely debiased classifier should correctly predict the target labels of images in an unbiased as well as a biased test sets.

Implementation details For the biased classifier and debiased classifier, we use MLP with three hidden layers for Colored MNIST, and ResNet-18 [27] for Corrupted CIFAR10, bFFHQ, BAR datasets, respectively. For the SwapAE, we follow the same network architecture as proposed in Park *et al*. [21] To measure the bias score, hyperparameter $q = 0.7$ is used for the GCE loss, and thresholds are fairly chosen on 50 epochs. We provide a detailed description of experimental details in Section D in the supplementary material.

Comparison methods We compare BiaSwap with existing methods, ReBias [3] and LfF [7], which address the dataset bias problem in the image classification task. The vanilla classifier which is trained without any debiasing procedure is also included in the comparison. In addition, we add Stylised ImageNet (SIN) [4] as our baseline in validating the effectiveness of utilizing the realistic augmentation in debiasing. For the fair comparison with baseline models, we re-implement LfF [7], ReBias [3], and Stylised ImageNet (SIN) [4] with the dataset we evaluated.

4.2. Quantitative Evaluation

Synthetic datasets We verify our method on Colored MNIST and Corrupted CIFAR10. In Table 1, we report the classification accuracy on the two datasets with varying ratios of bias, evaluated on both bias-guiding and unbiased test set for each dataset. The more bias becomes severe, the more vanilla model failure to generalize on unbiased data, where the shortcut does not exist. In contrast, our pro-

Dataset		Vanilla	ReBias	LfF	BiaSwap
bFFHQ	Bias-guiding	98.60	98.09	59.85	**99.13**
	Bias-contrary	51.03	53.66	55.61	**58.87**
BAR	Bias-guiding	**95.00**	87.78	91.67	93.33
	Bias-contrary	49.59	39.29	52.13	**52.4**

Table 2: Quantitative comparisons of bias-guiding and bias-contrary test accuracy on two real-world datasets. We denote the best score with bold and the best score among unsupervised methods with under-lined scores.

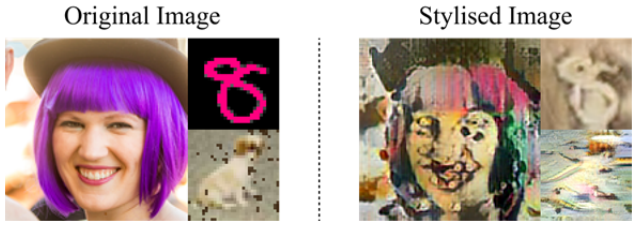
posed method maintains the robust debiasing capability on the unbiased test set, regardless of the bias ratio. In addition, the ReBias is observed to achieve the best score in Colored MNIST, which is mainly due to the network designed for capturing the texture bias. Note that BiaSwap obtains comparable accuracies compared to ReBias, even BiaSwap does not require any assumption on the type of bias in advance. Especially for the bias-guiding samples, our method achieves significant improvement, as success to generalize on the intended direction. Compared to LfF, where no prior knowledge on the bias type is required like BiaSwap, BiaSwap outperforms LfF on both bias-guiding and unbiased test accuracies for most of the dataset setup. As for the degraded bias-guiding accuracies of LfF, we assume that the oversampling of the limited number of bias-contrary images makes the network instead under-fitted to the downplayed bias-guiding images during training.

Real-world datasets To demonstrate the efficacy of our method on a realistic scenario, we provide the quantitative comparisons validated on bFFHQ and BAR datasets that contain complex types of bias in real-world images. Table 2 demonstrates that BiaSwap achieves superior debiasing performance against the existing baselines on these real-world datasets. ReBias reveals the significant drop of bias-contrary accuracy for those datasets where the texture no more causes the unwanted correlation. Likewise, LfF shows the degraded performance in the bias-guiding images. Therefore, we demonstrate that our method represents the debiasing approach with wide applicability on real-world datasets.

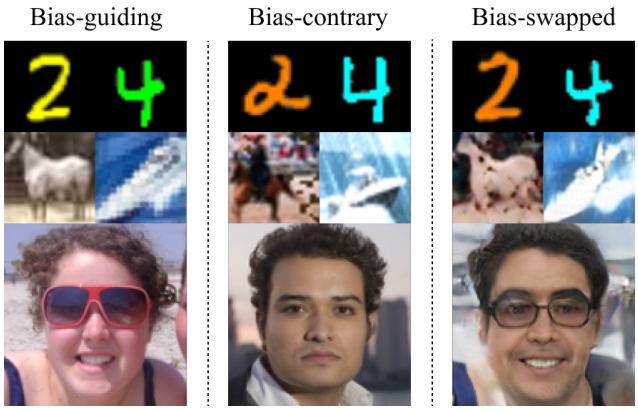
To validate the proposed methods, the ablation study is

Dataset	Colored MNIST	Corrupted CIFAR10	bFFHQ
Precision (%)	97.54	60.70	65.52
Recall (%)	92.12	87.28	70.62
F1 score (%)	94.74	66.13	67.70

Table 3: Quantitative evaluations on the $\tilde{y}_{\text{bias}}$ assignment via precision, recall, and F1 score metrics. We report the evaluation scores for 99.0% of each dataset, except BAR where no bias label is accessible.



(a) Generated images via Stylised ImageNet



(b) Generated images via BiaSwap

Figure 3: Qualitative comparisons between the augmented images from (a) SIN and (b) our bias-tailored swapping augmentation. BiaSwap generates a more realistic and bias-aware image compared to SIN.

provided in Section A of the supplementary material.

Evaluation on $\tilde{y}_{\text{bias}}$ assignment We provide the quantitative evaluation on the proposed method of assigning the pseudo-bias labels on the bias-guiding and bias-contrary images via Eqs. 1 and 2, as shown in Table 3. The table includes the precision, recall, and F1 Score of the binary classification on the four datasets with varying ratios of bias severity. Note that we consider identifying both the bias-guiding and bias-contrary images equally important in our framework, we first calculate each metric for both cases. Afterwards, we add them and divide them by two in order to obtain the overall scores for classifying both bias-guiding and bias-contrary images. As shown in Table 3, the proposed method mentioned in Section 3.1 achieves the reasonable performance on dividing the bias-guiding and bias-contrary images. Therefore, providing these paired sets of images enables the effective generation of bias-swapped images in the swapping autoencoder described in Section 3.2.

4.3. Qualitative analysis

Generation of bias-swapped image Figure 3-(b) depicts a set of bias-guiding, bias-contrary, and the generated bias-swapped images for Colored MNIST, Corrupted CIFAR10, and bFFHQ in each row. We observe that the bias-swapped images contain the bias attributes extracted from the bias-

	Colored MNIST	BAR	bFFHQ
Vanilla	48.76	49.59	74.86
SIN	40.79	50.51	69.86
BiaSwap	**83.74**	**52.44**	**78.98**

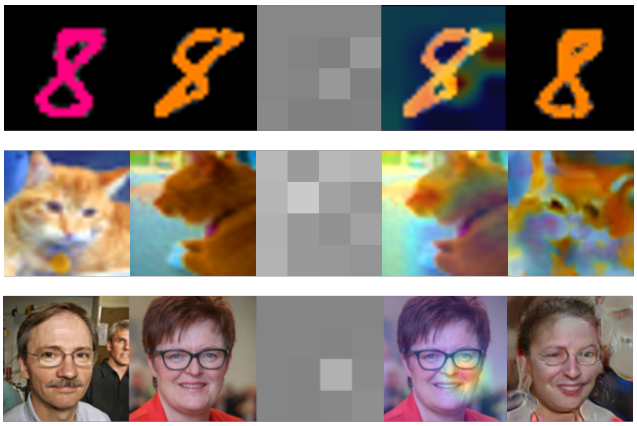
Table 4: Quantitative comparisons of unbiased test accuracy between BiaSwap and SIN. We utilize 99.0% of bias-guiding images for training.

contrary images while maintaining the bias-irrelevant attributes from the bias-guiding images. For example, our method translates a young female (first column) into a young male (third column) by reflecting the gender attribute from another young male (second column), while retaining the bias-irrelevant aspects, such as wearing sunglasses and smiling.

Comparison with stylised imagenet Although existing augmentation-based methods have achieved the improved classification performance [28, 29, 30], they may suffer from generating unrealistic images. Some of the methods often leverage the simple image-level augmentation techniques to combine the two different images, resulting in unrealistic images compared to natural ones. The recently proposed StylisedImageNet (SIN) [4] utilizes the AdaIN-based style transfer to augment the ImageNet images with different textures in order to solve the texture bias. However, as shown in Fig 3-(a), stylized results are much more unrealistic compared to the original images. In contrast, our approach synthesizes realistic images in a more natural way, and we believe that realistic augmented images help debiasing more than unrealistic ones.

To verify this, we compare the unbiased test accuracy of our method against that of SIN across the Colored MNIST, bFFHQ, and BAR datasets, in Table 4. We observe that SIN fails to learn the debiased representation on each dataset, and it may be caused by 1) the unrealistic augmented samples and 2) the augmentation without considering the bias attributes. To be specific, in Figure 3-(a), stylized pink eight loses the original shape of eight, and the texture of a facial image is changed while the gender attribute remains unchanged. On the other hand, BiaSwap only replaces the bias-relative attributes and generates visually plausible images, as shown in Figure 3-(b).

Visualization of CAM In order from left to right in Figure 4, bias-guiding sample, bias-contrary sample, CAM, CAM heatmap visualized on the image (b), and the bias-swapped image generated from BiaSwap are shown. Red regions in (d) correspond to the more discriminative region compared to the blue regions. As intended, the highlighted regions of CAM mainly appear in the regions where the bias attributes are exploited, *e.g.,* colors in Colored MNIST and face in bFFHQ. Note that by exploiting the attributes of those attended regions in the biased classifier, our bias-tailored patch discriminator generates the images considering bias-relevant attributes. For example, the cat in the (e) column contains the same corruption (*i.e.,* saturate) as the one in column (b) while maintaining the overall shape of cat in column (a).



(a) (b) (c) (d) (e)

Figure 4: Visualization of CAM utilized in our CAM-based patch sampler.

5. Discussion and Conclusion

In this paper, we propose a novel image translation-based debiasing approach which augments the realistic bias-contrary images for learning debiased representation. Based on the assumption that bias attribute is easy-to-learn, we leverage the patch cooccurence discriminator integrated with CAM and the GCE loss to generate the image with its bias attributes translated from the bias-contrary images while maintaining the other bias-irrelevant visual aspects. Extensive experiments demonstrate that our method successfully generates realistic bias-contrary images, achieving the state-of-the-art debiasing performance across diverse datasets. We acknowledge that the perfect translation of bias in images remains challenging, particularly when the dataset contains a complex combination of bias attributes or the number of training images is limited. However, we believe that our work can be viewed as a cornerstone of future debiasing works.

Acknowledgements This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2019R1A2C4070420). This work was also supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2021-0-01778, Development of human image synthesis and discrimination technology below the perceptual threshold)

References

[1] A. Torralba and A. A. Efros. Unbiased look at dataset bias. CVPR '11, 2011. 1

[2] Haohan Wang, Zexue He, Zachary L. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019. 1, 2

[3] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020. 1, 2, 6

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 1, 2, 6, 8

[5] Remi Cadene, Corentin Dancette, Hedi Ben younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 2

[6] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics. 1, 2, 3

[7] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020. 1, 3, 5, 6

[8] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016. 1, 4

[9] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[10] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9572–9581, 2019. 2

[11] V. Agarwal, Rakshith Shetty, and M. Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9687–9695, 2020. 2

[12] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 2

[13] Karan Goel, Albert Gu, Yixuan Li, and Christopher Re. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2021. 2

[14] Rakshith Shetty, B. Schiele, and M. Fritz. Not using the car to see the sidewalk — quantifying and controlling the effects of context in classification and segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8210–8218, 2019. 2

[15] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. pages 6642–6651, 06 2019. 2

[16] Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. In *EMNLP/IJCNLP*, 2019. 2

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[18] Luke Darlow, Stanisław Jastrzębski, and Amos Storkey. Latent adversarial debiasing: Mitigating collider bias in deep neural networks. *arXiv preprint arXiv:2011.11486*, 2020. 3, 6

[19] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018. 3

[20] 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. 3

[21] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, 2020. 4, 6

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014. 4

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 5

[24] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 5

[25] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 5

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 6

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6

[28] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 8

[29] Devesh Walawalkar, Zhiqiang Shen, Z. Liu, and M. Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3642–3646, 2020. 8

[30] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 8