Supplementary material for
BiaSwap: Removing Dataset Bias with Bias-Tailored Swapping Augmentation

This material complements our paper with additional experimental results and their analysis. First of all, we present the ablation studies on the proposed modules of our framework, presented in Section A. Afterward, Section B provides the qualitative and quantitative analysis on the proposed pseudo-bias labels assignment. In Section C, we provide additional qualitative examples of augmented bias-swapped images generated by our proposed method, along with their class activation map (CAM) [9] visualization. Section D describes the implementation details, such as the settings for training and the construction of biased-FFHQ (bFFHQ) dataset. Lastly, we provide a detailed explanation of the datasets and baselines we utilized in Section E.

A. Ablation study

This section demonstrates the effectiveness of our two main contributions, 1) separation of bias-contrary and bias-guiding images and 2) CAM-based patch sampling in the bias-tailored swapping autoencoder (SwapAE). As explained in Sections 3.1 and 3.2 of the main paper, the separation of contrary and guiding images encourages the swapping autoencoder to translate the bias-guiding image into the bias-swapped one by reflecting the bias-contrary attributes. In addition, CAM obtained from the biased classifier enables the sampling of patches based on the highly discriminative (*i.e.,* highly bias-related) regions, enforcing the bias-tailored patch discriminator to translate the visual styles from them. To verify the effectiveness of such methods, we conduct the ablation studies on these two components denoted as **c1** and **c2** in Table 1 and compare the accuracy of the unbiased test set over Colored MNIST and bFFHQ datasets.

As the separation is ablated, a pair of two guiding images become more frequently provided in the SwapAE for augmenting the new image, compared to the bias-guiding and bias-contrary pairs. As a result, the translation between these guiding images generates another guiding image, which does not help to remove the dataset bias in the training distribution. It is observed in Table 1 that the model with **c1** ablated shows a critically degraded performance in an unbiased test set compared to the highest accuracies in the bias-guiding dataset on both Colored MNIST and bFFHQ. In contrast, our model trained with **c1** achieves superior performance both in guiding and unbiased test set, demonstrating that the classifier benefits from the augmented images generated from (*bias-guiding*, *bias-contrary*) pairs. When we ablate the second method **c2**, the patches are randomly sampled as exactly the same as the original patch co-occurrence discriminator proposed in the original paper [8] does. This simply transfers the overall style of a bias-contrary to a bias-guiding image without considering the regions of bias-attribute. However, utilizing the CAM-based patch sampling enables the further optimized image translation by focusing on transferring the bias-related attributes in the image. Table 1 indicates the proposed method equipped with **c2** achieves the best accuracy on the unbiased test set of both datasets.

Methods	Bias-guiding		Unbiased	
	Colored MNIST	bFFHQ	Colored MNIST	bFFHQ
BiaSwap w/o **c1**	99.92	99.2	43.04	45.0
BiaSwap w/o **c2**	98.98	99.0	84.16	51.2
BiaSwap (Full)	99.24	99.13	**86.03**	**58.87**

c1 : Separation of bias-contrary and bias-guiding pairs.
c2 : CAM-based patch sampling.

Table 1: Quantitative comparisons of our proposed method and its ablated versions on Colored MNIST and bFFHQ datasets. The separation of bias-contrary and bias-guiding pairs (**c1**), and CAM-based patch sampling (**c2**) are ablated.

B. Analysis on pseudo-bias label assignment

As introduced in Section 3.1 of the main paper, we utilize a bias score as well as the pseudo-bias label y_{pseudo} to divide the entire training dataset into bias-guiding and bias-contrary samples. To provide the qualitative and quantitative verification on the effectiveness of such division, this section consists of two parts, 1) qualitative examples classified as a bias-guiding and bias-contrary by the method and 2) quantitative evaluation of the robustness of y_{pseudo} assignment on the diverse dataset setups, as supplementary to the Table 3 in the main paper.

B.1. BAR images separated by pseudo-bias label

As mentioned in Section 4.1 in the main paper, the training dataset of BAR only contains bias-guiding samples categorized by Nam *et al.* [7]. However, even if we assume that

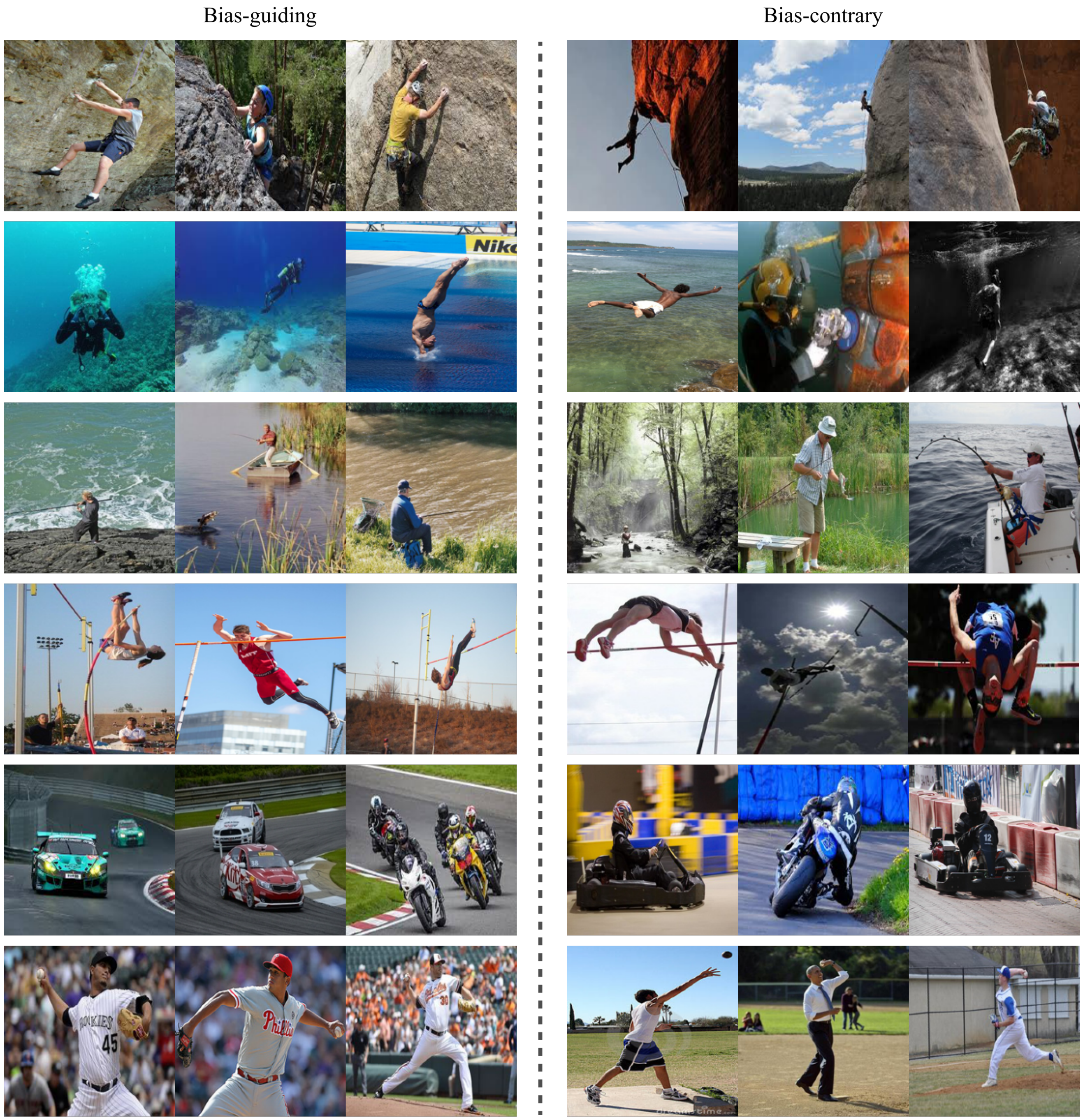

Figure 1: Qualitative examples of divided images of BAR. Images on the left side represent the images classified as a bias-guiding sample based on our threshold. Images on the right side correspond to the images classified as a bias-contrary sample.

all the samples have an unwanted correlation with the target label, there must exist a varying degree of bias between the training samples. In other words, some of the samples can be more bias-guiding compared to the other images. In this regard, the proposed dividing strategy can effectively capture this subtle difference and sorts the samples *from easy to hard* one. To be specific, the images with the same ground-truth target labels can be sorted by their bias score, which represents how much the image includes the bias attributes. Figure 1 shows the exemplar images which are assigned to the bias-guiding (left columns) or bias-contrary (right columns) via our bias score and the threshold de-

Dataset	%	Precision (%)	Recall (%)	F1 score (%)
Colored MNIST	95.0	97.09	98.43	93.55
	98.0	99.55	91.76	95.31
	99.0	97.54	92.12	94.74
	99.5	99.98	95.78	97.79
Corrupted CIFAR10	95.0	68.34	80.66	72.56
	98.0	64.38	82.14	69.49
	99.0	60.7	87.28	66.13
	99.5	58.61	87.09	63.64
bFFHQ	99.0	65.52	70.62	67.70

Table 2: Quantitative evaluations on the $\tilde{y}_{\text{bias}}$ assignment via precision, recall, and F1 score metrics. We report the evaluation scores for 99.5%, 99%, 98%, and 95% of both Colored MNIST and Corrupted CIFAR10, and 99% of bFFHQ, except BAR where no bias label is accessible.

scribed in Eqs 1 and 2 in the main paper, respectively. While unwanted correlations shown in the left columns are found in the most of training data, it turns out that some of the samples are relatively *hard-to-learn*, where there exist less severe correlations between the bias attributes and the target label. For instance, for the images labeled with climbing in the first row, most of the climbers are located on the rock which has an uneven texture and brown color. In this context, the examples with the sky in the most part of their backgrounds or with colors other than brown are classified as a bias-contrary sample. Similarly, for the images labeled with diving in the second row, divers are usually in the deep sea or taking a similar body motion. However, examples on the right side are conflicting with the bias in that they contain a unique diving pose. Some of them also are black-and-white pictures, which is uncommon in the training dataset. For the fishing images on the right side of the third row, the fisher in the lake surrounded by the dense trees or the fisher near the river represents that such places are not the usual cases in the training distribution. This implies that our method empirically well divides the samples based on the relative severity of the bias between the images.

B.2. Quantitative evaluation on pseudo-bias label

Table 2 demonstrates the quantitative evaluation scores of our proposed dividing method on the Colored MNIST, Corrupted CIFAR10, and bFFHQ datasets. As our method works as a binary classifier to discriminate whether the images are bias-guiding or bias-contrary, we measure the precision, recall, and F1 score for both bias-guiding and bias-contrary classes on the unbiased test set. Following the same evaluation protocol in Section 4.2 of the main paper, we add the scores of bias-guiding and bias-contrary ones and divide them by two in order to obtain the overall scores.

Table 2 indicates that the dividing method works well on classifying the bias-contrary as well as bias-guiding samples, achieving reasonable results in precision, recall, and F1 score. In consequence, the robustness of the method enables to guarantee of the effective augmentation of bias-swapped images in the bias-tailored swapping autoencoder.

C. Additional qualitative examples of bias-swapped images

To supplement Section 4.3 in the main paper, this section provides the additional qualitative results of the bias-swapped samples as well as their CAMs in Figure 2. In order from left to right, bias-guiding sample, bias-contrary sample, CAM, heatmap of CAM visualized on the bias-contrary image, and the bias-swapped image generated from BiaSwap are presented. The first and second rows include the examples of Colored MNIST and Corrupted CIFAR10, respectively. Similar to the ones in the main paper, CAM heatmaps on the Colored MNIST samples show that the biased classifier mainly focuses on the regions where the bias-correlated colors are located. For example, CAM described in the first row follows the region of blue colors in the digit. For the samples of the Corrupted CIFAR10, our model properly transfers the bias attributes of the second column images into the ones in the first column, maintaining the bias-irrelevant visual aspects unchanged. This results in the bias-swapped images shown in the last column. For example, the corruption applied on the car in the second column is transferred into another car in the first column, while the shape of the first column car is maintained in the generated bias-swapped car in the last column.

D. Implementation details

This section provides the specific values of the threshold for separation of bias-guiding and bias-contrary groups over each dataset. Afterward, we provide the detailed architecture of two main networks, bias-tailored swapping autoencoder and debiased classifier. In addition, we provide the training details, such as hyper-parameters for each objective function, over the dataset we utilized.

Threshold for division To separate the training samples into bias-contrary and bias-guiding sets in an unsupervised manner, we utilize the mean value of confusing scores of the images as our threshold in each dataset, as described in Eq. 2 in the main paper. Such threshold values correspond to 0.0358, 0.0903, 0.0232, and 0.008 for Colored MNIST, Corrupted CIFAR10, BAR, and bFFHQ, respectively.

Bias-tailored swapping autoencoder We follow the original network architecture of the encoder, decoder, and discriminator proposed in Park *et al.* [8] to maintain its image translation performance. However, to design a CAM-based patch sampling for the co-occurrence discriminator as proposed in Section 3.2 in the main paper, we sample

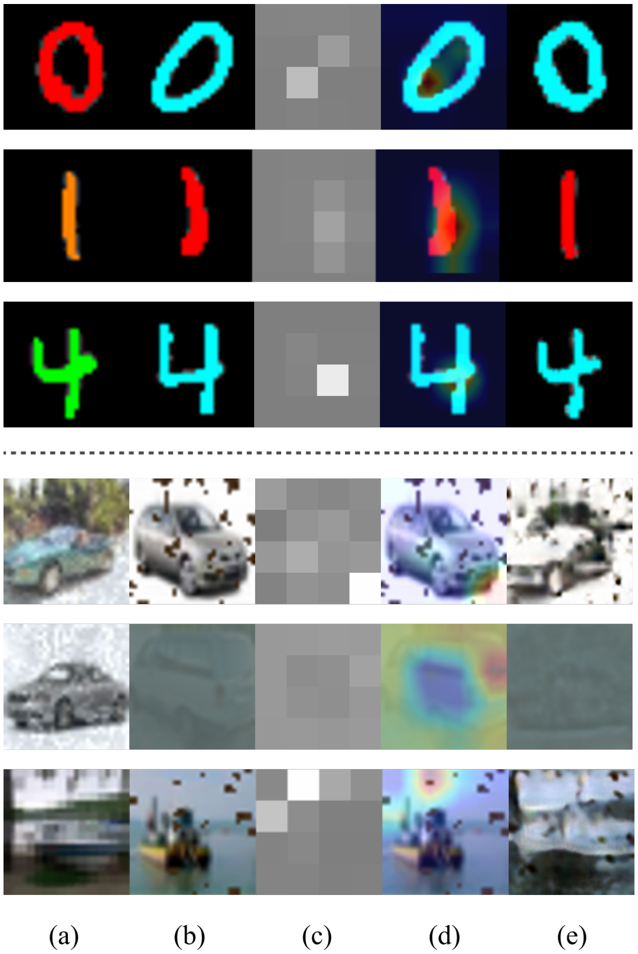

Figure 2: Qualitative examples of generated images via BiaSwap on Colored MNIST (top), Corrupted CIFAR10 (middle) and bFFHQ (bottom). Each sample is listed in order of bias-guiding, bias-contrary, CAM on the bias-contrary, heatmap of CAM on the bias-contrary, and bias-swapped image.

the patches using the patch-wise probability based on the CAM of the biased classifier, instead of random sampling. For the biased classifier, we use a multi-layer perceptron (MLP) with three hidden layers for Colored MNIST and ResNet-18 [2] for the rest of the datasets. The classifier is trained with the GCE loss with hyperparameter q of 0.7. As the parameters of the classifier are *not* jointly optimized with those of bias-tailored swapping autoencoder, the classifier is fully trained to be biased. We train and evaluate the biased classifier with the size of 28×28 and 32×32 images for Colored MNIST and Corrupted CIFAR10, and 128×128 for BAR and bFFHQ datasets. Each channel of the images are normalized with the mean of $(0.5, 0.5, 0.5)$ and the standard deviation of $(0.5, 0.5, 0.5)$. All other details are identical to the training of the debiased classifier. To train the autoencoder, we set the hyper-parameters for each loss functions as $\lambda_{\text{recon}} = \lambda_{\text{GAN,recon}} = \lambda_{\text{GAN,swap}} = \lambda_{\text{CooccurGAN}} = 1$. To prevent the patch discriminator from only sampling the same single patch due to the high probability close to one, we utilize the temperature scaling with $\tau = 10$, for smoothing the probability. Both the swapping autoencoder and the classifier are trained by using an Adam [5] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.99$.

Debiased classifier After we fully optimize the bias-tailored swapping autoencoder, we augment the dataset using the pairs of bias-guiding and contrary images given from the threshold. Given these additional images, which we call *bias-swapped* images in the main paper, we train an MLP with three hidden layers for Color MNIST and ResNet-18 for the rest of the datasets. We train and evaluate the classifier with the size of 28×28, 32×32, 128×128, and 224×224 images for Colored MNIST, Corrupted CIFAR10, bFFHQ, and BAR datasets, respectively. We use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and learning rate as 0.001. We use a batch size of 256 and train a classifier for 200 epochs for Color MNIST, Corrupted CIFAR10, BAR, and bFFHQ datasets.

E. Datasets and Baselines

E.1. Datasets

Corrupted CIFAR10 As proposed in Hendrycks and Dietterich [3], we apply the certain type of texture corruptions onto the CIFAR10 dataset [6]. Among 15 types of corruptions, we utilize the *Snow, Frost, Fog, Brightness, Contrast, Spatter, Elastic, JPEG, Pixelate*, and *Saturate* in our paper. Such corruptions are applied with the strong correlation with the original classes of CIFAR10 dataset, which are *Plane, Car, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck.* In addition, we utilize the corruptions with the highest degree of severity (*i.e.*, 4) in our dataset.

bFFHQ We newly construct the biased FFHQ dataset (bFFHQ) which has a strong correlation between the age (target) and gender (bias), based on the Flickr-Faces-HQ (FFHQ) dataset [4]. FFHQ consists of $70,000$ images at 1024×1024 resolution and contains the considerable variation of human faces in terms of age, ethnicity, and image background. Each face contains different attributes including head pose, gender, age, mustache, glasses, and emotion. Among these attributes, we utilize the *age* and *gender* attributes. To be specific, the attribute ‘young’ (*i.e.*, aged $10 - 29$) is highly correlated with ‘women’ and ‘old’ (*i.e.*, aged $40 - 59$) is connected with ‘men’. Among the total $70,000$ data, $19,200$ samples are utilized as the training dataset according to the criteria of unwanted correction, and 2,000 unbiased samples that each attribute is uniformly distributed are utilized as an evaluation set.

BAR This dataset includes the images which have a correlation between human actions and backgrounds, which is curated by Nam *et al*. [7]. It does not have the ground-truth bias label, unlike other datasets.

E.2. Baselines

ReBias ReBias [1] assumes the *texture* as the unwanted bias type and exploits the biased classifier with a limited kernel size in order to mainly capture the texture attributes from the images. By learning the representations statistically independent from such texture representations, ReBias achieves robust classification accuracies against the texture bias. We train ReBias with the same training protocol as suggested in the original paper including network architecture and the hyper-parameters. Note that for Colored MNIST, ReBias utilizes the convolutional network for capturing the texture cues, while other baselines including ours exploit the MLP with three hidden layers.

LfF As mentioned in Sections 1 and 2 of the main paper, LfF assumes the general characteristic of bias as “easy-to-learn" and proposes the re-weighting-based debiasing method based on the GCE loss. To the best of our knowledge, this work first learns the debiased representation without any prior assumption on the bias type. We follow the official implementation setups of LfF, except for the network architecture of ResNet-20 for the Corrupted CIFAR10 dataset. As a fair comparison, we utilize the ResNet-18 architecture for all the baselines including LfF.

SIN As mentioned in Section 4.3 of the main paper, we utilize SIN as another baseline for validating the importance of realistic image generation in dataset augmentation. For the augmentation of datasets we utilize, we follow the official implementation of SIN and only replace the original ImageNet dataset with each biased dataset. Style images are identical to official ones.

References

[1] 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. 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 4

[4] 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. 4

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. the International Conference on Learning Representations (ICLR)*, 2015. 4

[6] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 4

[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, 5

[8] 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. 1, 3

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