A Appendix

A.1 Datasets & Model Architectures

Datasets. We explain additional details for benchmark datasets used in our preliminary and main experiments.
1) Colored MNIST. Colored MNIST (CMNIST) has a distinctive color bias assigned to each class, as shown in Fig. 5. We construct CMNIST dataset by injecting pre-defined RGB values into each MNIST image by following a generation protocol [23]. Moreover, we control the injected color distribution by randomly selecting the standard deviation among 0.05, 0.02, 0.01, and 0.005. Accordingly, we obtained 55,000 images as a training set and varied the ratio of bias-aligned samples with the percentages of 99.5%, 99.0%, 98.0%, and 95.0%, respectively. The number of bias-align and bias-conflict samples for training and test sets is described in Table 4. In addition, we use 5,000 images for all cases in the validation set with the same bias ratio as the training set.
2) Corrupted CIFAR10. Hendrycks and Dietterich proposed Corrupted CIFAR10 (CCIFAR10) with a well-defined generation protocol [12]. The CCIFAR10 has a unique corruption bias applied to each class as follows: "Plane and Snow," "Car and Frost," "Bird and Defocus Blur," "Cat and Brightness," "Deer and Contrast," "Dog and Spatter," "Frog and Frosted Glass Blur," "Horse and JPEG Compression," "Ship and Pixelate," and "Truck and Saturate." For bias-conflict samples, each class is applied with other classes' corruptions, excluding those used in bias-align samples. For example, "Plane" class will contain bias-conflict samples containing "Frost," "Defocus Blur" and other different corruptions, not "Snow". We use official dataset, containing the ratio of bias-aligned samples with 99.5%, 99.0%, 98.0%, and 95.0% percentages. The number of bias-align and bias-conflict samples are described in Table 4. Similar to CMNIST, the validation set is 5,000 images with the same bias ratio as the training set.
3) Biased FFHQ. Biased FFHQ (BFFHQ) is a subset of facial images focusing on two features (age and gender) extracted from the FFHQ [14] dataset proposed by Lee et al. [19]. The BFFHQ dataset has two classes, "Old" and "Young" with a strong correlation between age and gender since young women are selected in an age range from 10 to 29 years old, and old men are selected between 40 and 59 years old in the training set of the FFHQ dataset. The BFFHQ dataset is consisted of 19,200 samples (19,104 for bias-align and 96 for bias-conflict) in the training set with bias ratio of 99.5%. The number of the training and test data is specified in Table 4. For the validation set, we use 1,000 images with the same bias ratio of 99.5%.
4) Biased Action Recognition. Biased Action Recognition (BAR) is a real-world dataset with texture (background) bias proposed by Nam et al. [23]. The BAR dataset has six action classes with distinct places as follows: "Climbing and Rock Wall," "Diving and Underwater," "Fishing and Water Surface," "Racing and Racing Track," "Throwing and Playing Ground," and "Vaulting and Sky." The images are collected from internet sources, imSitu, Stanford 40 Actions, and Google Image Search. In addition, the authors have conducted a user study

Table 4: The number of benchmark datasets. The bias ratio indicates a proportion of bias-align and bias-conflict samples in each dataset. The bias ratio of 'One-Shot' indicates that the dataset has only one bias-conflict sample for each class.

Dataset	**Bias Ratio**	**Training**		**Test**	
		Bias-align	**Bias-conflict**	**Bias-align**	**Bias-conflict**
CMNIST	One-Shot	54,729	10	1,006	8,994
	99.5%	54,729	271		
	99%	54,454	546		
	98%	53,904	1,096		
	95%	52,254	2,746		
CCIFAR10	One-Shot	44,832	10	1,000	9,000
	99.5%	44,832	228		
	99%	44,527	442		
	98%	44,145	887		
	95%	42,820	2,242		
BFFHQ	One-Shot	19,104	2	500	500
	99.5%	19,104	96		
BAR	100%	1,941	N/A	N/A	654

Table 5: The model architecture used in our experiments. We provide component details used in our model architecture depending on the datasets.

Dataset	Encoder	Feature vector	Classifier
CMNIST	3 layer MLP	16	16×10
CCIFAR10	ResNet-18	512	512×10
BFFHQ			512×2
BAR			512×6

to divide the collected images into bias-align and bias-conflict samples. Then, they assigned bias-align samples as 'training set' and bias-conflict samples as 'test set.' Therefore, 2,595 samples (1,941 for training and 654 for testing) are available to train and test the classifiers. For the validation set, we randomly split the training set into training and validation sets with an 8:2 ratio.

5) One-shot Scenario. We extend the biased scenario into more extreme situations, where only one bias-conflict sample per class exists in each dataset. We use ten bias-conflict samples for CMNIST and CCIFAR10 and two bias-conflict samples for the BFFHQ dataset. The number of samples for the one-shot scenario is presented in Table 4.

Model Architectures. We provide a detailed explanation for the model architectures used in our experiments in Table 5. For CMNIST, we use a multi-layer perceptron (MLP) consisting of three hidden layers that have 100 hidden units, with 16 final feature vectors. For other datasets, we utilize ResNet-18 [11] with 512 final feature vectors. Furthermore, we adopt a classifier as a simple linear layer that maps the final feature vectors with the number of classes for each dataset.

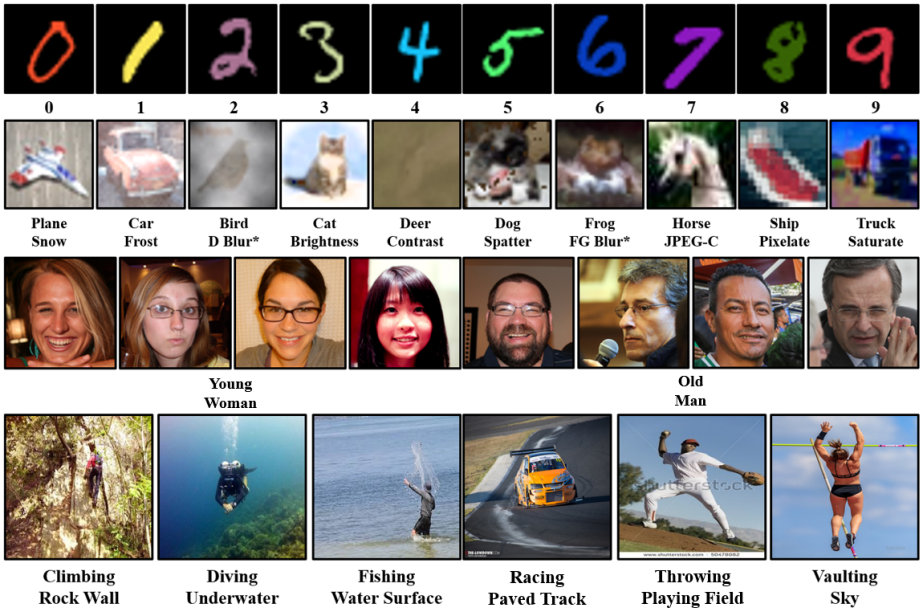

Fig. 5: Description for bias-align samples in benchmark datasets. Each row represents CMNIST, CCIFAR10, BFFHQ, and BAR datasets from top to bottom. The biases (i.e., color, corruption, maturity with gender, and place) are illustrated for each dataset. D Blur and FG Blur denote the defocus blur and frosted glass blur, respectively.

A.2 Training Details for Adaptive Augmentation (A^2)

Biased Classifier. We describe a training scheme for a biased classifier to extract bias-conflict samples as explained in Section 4. We train each classifier with GCE loss for 100 epochs, described in Table 5. The classifier is optimized via Adam optimizer [5] with learning rates of 0.01, 0.001, 0.0001, and 0.0001 for CMNIST, CCIFAR10, BFFHQ, and BAR datasets, respectively.

Projection Scheme. To translate a bias-align sample x_b into a bias-conflict sample x_d, we first need to find a biased latent vector z_b representing x_b. It can be performed by projecting x_b into the learned latent space [14] of the biased generator G_b. To this end, we initialize a random vector z_b and minimize the distance between a real bias-align sample x_b and a generated bias-align sample $G_b(z_b)$. For minimizing both perceptual and pixel-level distance, we exploit perceptual and reconstruction loss as follows:

$$L_{recon}(G_b(z_b), x_b) = \|(G_b(z_b) - x_b)\|^2, \tag{10}$$

$$L_{perc}(G_b(z_b), x_b) = \|VGG_{16}(G_b(z_b)) - VGG_{16}(x_b)\|^2, \tag{11}$$

[5] https://pytorch.org/docs/stable/optim.html

Table 6: Detailed configurations implemented for our models. Depending on the given dataset, we vary image size, batch size, decay rate, augmentation, and usage of pretrained weights.

Dataset	Image size	Batch size	Scheduler (Decay Rate)	Augmentation	ImageNet Pretrained
CMNIST	28 × 28	256	0.5 decay (every 20 epochs)	X	X
CCIFAR10	32 × 32				X
BFFHQ	224 × 224	64	0.1 decay (every 20 epochs)	Random Crop Horizontal Flip	X
BAR	256 × 256				O

where, the output of VGG_{16} indicates intermediate features maps from VGG networks [14]. Thus, our projection scheme is given by

$$z_b^* = \arg\min_{z_b} L_{recon}(G_b(z_b), x_b) + L_{perc}(G_b(z_b), x_b), \tag{12}$$

where we update z_b via Adam optimizer with a learning rate of 0.1 for n iterations. We use different hyperparameter n , considering the complexity of the datasets: 500, 1,000, 1,000, 1,000, and 3,000 iterations for CMNIST, CCIFAR10, BFFHQ, and BAR datasets. After projection, we generate the bias-conflict sample x_d by forwarding the obtained vector z_b into the adapted generator G_d. Note that this projection scheme can also be performed in parallel.

A.3 Baseline Implementation Details

Datasets & Baselines. We tried our best to implement the official repository for benchmarks datasets and baselines. The datasets, CMNIST, CCIFAR10, and BFFHQ, can be found in the official repository [6] provided by Lee et al. [19]. The BAR dataset can be found in the official repository [7] provided by Nam et al. [23]. Furthermore, we use the official implementations for the baseline models, LfF and DisEnt, to compare with our method.

Training Configuration. We provide the training configuration details in Table 6. For training and evaluating the classifiers, we use image sizes of 28 × 28, 32 × 32, 224 × 224, and 256 × 256 for CMNIST, CCIFAR10, BFFHQ, and BAR datasets. Since the complexity of each dataset is different, we apply individual preprocessing and augmentation techniques to each dataset. First, we use the batch size of 256 for CMNIST and CCIFAR10 and 64 for BFFHQ and BAR respectively. Second, the learning rate is decayed every 20 epochs by 0.5 for CMNIST and CCIFAR10 and 0.1 for BFFHQ and BAR. Third, augmentation is adopted for BFFHQ and BAR datasets, consisting of two consecutive approaches: Random Cropping and Random Horizontal Flip provided by the Torchvision library. Lastly, we use ImageNet pretrained weights.

Software & Hardware Configuration. We attempted to implement and reproduce the baseline methods precisely described in their original documents

[6] https://github.com/kakaoenterprise/Learning-Debiased-Disentangled
[7] https://github.com/alinlab/BAR

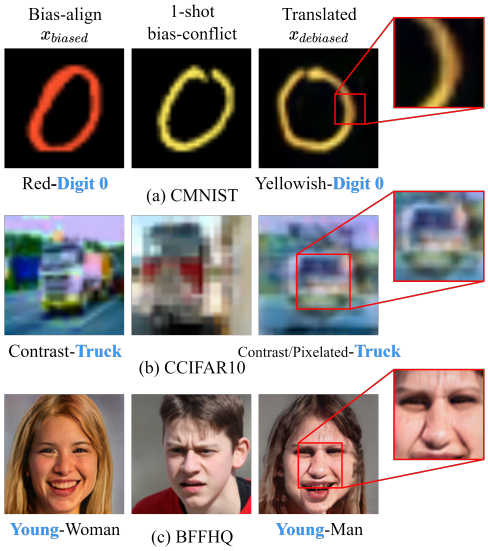

Fig. 6: The translated images via A^2 in one-shot testing scenarios. The blue text indicates label information for each image. We observe that the translated images have bias-conflict features, mixed with bias-align features.

papers. For software configuration, we implemented our models using PyTorch v1.8.1 with Torchvision v0.9.1 on Python v3.7.0 and Numpy v1.20.1. We used Intel XEON Gold 6230 2.1GHz CPU and four NVIDIA RTX 3090 24GB GPUs for hardware configuration based on CUDA v11.2.

A.4 Further Analysis

Translated images in one-shot environment. Figure 6 shows the translation results of one-shot experiments. For CMNIST, we can observe similar results in Section 6.2, such as the mixture of yellowish colors; however, it has a distinct feature that reflects both colors between bias-align and bias-conflict. Furthermore, we can also observe this phenomenon in CCIFAR10, where the translated image has contrast (align) and pixelated (conflict) objects. Regarding the BFFHQ dataset, we can identify slightly overfitted features in a translated image by the landmarks of the face denoted with a red box. Nevertheless, this demonstrates that translated images retain bias-conflict characteristics. Thus, our method is flexible enough to reflect bias-conflict features even in one-shot environment.

A.5 Summary of findings

To the best of our knowledge, we are the first to apply a generative model for debiasing through augmentation. We observed that adapting the learned gener-

ative model to another distribution can be performed efficiently by minimizing the distance between generated samples. The augmented images have a mixture of task-irrelevant features while retaining label information, which implies our method successfully reflects bias-conflict distribution.

Furthermore, we demonstrate that A^2 can be applied to other general datasets that include non-face and non-digit data (i.e., CCIFAR10 and BAR) as well as face and digit data (i.e., BFFHQ and CMNIST) that GAN can easily generate [15]. Thus, we demonstrate that A^2 could push the limits of classifiers to learn debiased representation via augmentation.