BiasAdv: Bias-Adversarial Augmentation for Model Debiasing

Jongin Lim[1*] Youngdong Kim[1] Byungjai Kim[1] Chanho Ahn[1]
Jinwoo Shin[2] Eunho Yang[2] Seungju Han[1]
[1]Samsung Advanced Institute of Technology (SAIT)
[2]Korea Advanced Institute of Science and Technology (KAIST)

2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.00373

Abstract

Neural networks are often prone to bias toward spurious correlations inherent in a dataset, thus failing to generalize unbiased test criteria. A key challenge to resolving the issue is the significant lack of bias-conflicting training data (i.e., samples without spurious correlations). In this paper, we propose a novel data augmentation approach termed Bias-Adversarial augmentation (BiasAdv) that supplements bias-conflicting samples with adversarial images. Our key idea is that an adversarial attack on a biased model that makes decisions based on spurious correlations may generate synthetic bias-conflicting samples, which can then be used as augmented training data for learning a debiased model. Specifically, we formulate an optimization problem for generating adversarial images that attack the predictions of an auxiliary biased model without ruining the predictions of the desired debiased model. Despite its simplicity, we find that BiasAdv can generate surprisingly useful synthetic bias-conflicting samples, allowing the debiased model to learn generalizable representations. Furthermore, BiasAdv does not require any bias annotations or prior knowledge of the bias type, which enables its broad applicability to existing debiasing methods to improve their performances. Our extensive experimental results demonstrate the superiority of BiasAdv, achieving state-of-the-art performance on four popular benchmark datasets across various bias domains.

1. Introduction

Real-world datasets are often inherently biased [2, 34], where certain visual attributes are spuriously correlated with class labels. For example, let us consider a binary classification task between cats and dogs. Unbeknownst to us, our dataset could consist of most cats indoors and most dogs outdoors, as illustrated in Figure 1. When trained on such a biased dataset, neural networks often learn unintended shortcuts [2, 8, 34, 39] (*e.g.*, making predictions based on the background) and fail to generalize in a new unbiased test environment. To tackle the problem, conventional methods have utilized explicit bias annotations [1, 19, 39, 43] or prior knowledge of the bias type [2, 3, 5, 9, 46]. However, bias annotations are expensive and laborious to obtain, and presuming certain bias types in advance limits the capability to be universally applicable to various bias types.

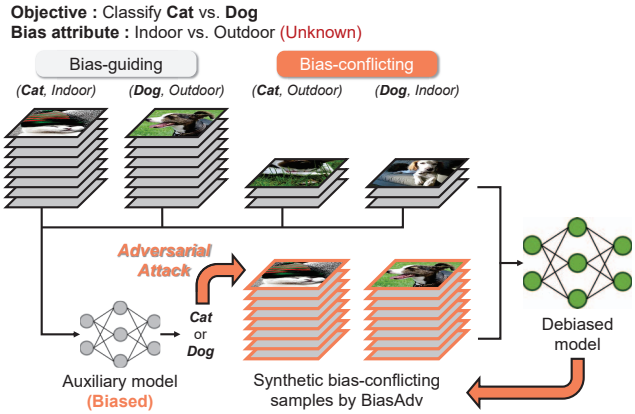


Figure 1. **An overview of BiasAdv.** In MetaShift [26], the bias attribute {Indoor, Outdoor} is spuriously correlated to the class label {Cat, Dog}. In this work, we refer to data with such spurious correlations as *bias-guiding* samples and without such correlations as *bias-conflicting* samples, respectively. Using the biased dataset, we train an auxiliary model to be biased, and BiasAdv supplements bias-conflicting samples using adversarial images which attack the biased predictions of the auxiliary model while preserving the predictions of the debiased model. By leveraging the diversified bias-conflicting data, BiasAdv allows the debiased model to learn generalizable representations for unbiased classification.

To train a debiased model without bias annotations, the main line of recent research [6, 22, 30, 34, 42] has commonly utilized an intentionally biased model as an auxiliary model under the idea that bias attributes are *easy-to-learn*. In essence, these methods identify bias-conflicting samples based on the auxiliary model and train the debiased model in a way that focuses more on the identified samples (*i.e.*, re-

*Corresponding author: jonny.lim@samsung.com

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.00373

weighting based on the auxiliary model). Although recent re-weighting methods have achieved remarkable success in debiasing without bias annotations, they have inherent limitation; since the number of bias-conflicting samples is often too small for a model to learn generalizable representations, the model is prone to over-fitting [25]. Consequently, re-weighting methods suffer from the degraded performance on bias-guiding samples [20, 44], which raises the question of whether these methods truly make models debiased or simply deflect models in unintended directions.

To resolve the aforementioned issues, data augmentation methods have recently been proposed to supplement bias-conflicting samples. For example, BiaSwap [20] conducts image-to-image translation to synthesize bias-conflicting samples. However, it requires delicate training of complex and expensive image translation models [36], limiting its applicability. On the other hand, DFA [25] utilizes feature-level swapping based on disentangled representations between bias-guiding and bias-conflicting features. Learning disentangled representations, however, is often challenging on real-world datasets [27, 28, 31].

In this paper, we devise a much simpler yet more effective approach to generate bias-conflicting samples, coined **Bias**-**Adv**ersarial augmentation (BiasAdv). Figure 1 shows an overview of BiasAdv. We utilize an auxiliary model that intentionally learns biased shortcuts, likewise [30, 34]. The key idea of BiasAdv is that an adversarial attack on the biased auxiliary model may generate adversarial images that alter the bias cue from the input images (*i.e.*, bias-conflicting samples). Concretely, we formulate an optimization problem to generate adversarial images that attack the predictions of the biased auxiliary model without ruining the predictions of the desired debiased model. Then, the generated adversarial images are used as additional training data to train the debiased model. It is noteworthy that, unlike previous data augmentation methods [20, 25], BiasAdv does not require complex image translation models or disentangled representations, so it can be seamlessly applied to any debiasing method based on the biased model. Furthermore, we show that BiasAdv, despite its simplicity, can generate surprisingly useful synthetic bias-conflicting samples, which significantly improves debiasing quality.

The main contributions of our work are three-fold:

- We propose BiasAdv, a simple and effective data augmentation method for model debiasing, which utilizes adversarially attacked images as additional training data. Our method does not require any bias annotations or prior knowledge of the bias type during training.
- BiasAdv can be easily applied to existing re-weighting methods without architectural or algorithmic changes. We confirm that BiasAdv significantly improves the performance, achieving up to 22.8%, 13.4%, 7.9%, and 8.0% better performance than the state-of-the-art results on CIFAR-10C [25], BFFHQ [25], BAR [34], and MetaShift [26], respectively.
- We demonstrate the effectiveness of BiasAdv through extensive ablation studies and analyses. Our key finding is that BiasAdv helps to learn generalizable representations and prevents over-fitting; it does not degrade the performance of bias-guiding samples and improves model robustness against input corruptions.

2. Related Work

Debiasing with bias supervision. To alleviate dataset bias, a majority of previous methods have exploited bias annotations [4, 12, 18, 33, 39, 40], balancing the data distribution through re-weighting. However, these methods are impractical since bias supervisions are costly, demanding extensive labor. Recently, to reduce annotation costs, several methods have utilized only a small amount of bias-labeled data [16, 35]. Yet, obtaining a small set of bias-labeled data could be still expensive since identifying which attributes exhibit spurious correlations requires thorough analysis of dataset [42]. Instead of using bias annotations directly, several methods have designed bias-tailored debiasing models by leveraging the prior knowledge of the bias type [2, 3, 5, 9, 46]. However, presuming certain bias types in advance limits the applicability to various bias types.

Debiasing without bias supervision. Recent debiasing methods without bias supervision [6, 22, 30, 34, 42] have focused on identifying bias-conflicting samples and re-weighting them. LfF [34] identifies bias-conflicting samples by an intentionally biased model trained by Generalized Cross Entropy (GCE) loss [51], while JTT [30] considers misclassified samples from standard ERM model as bias-conflicting samples. EIIL [6] infers a partition of bias-guiding and bias-conflicting by the invariance principle. BPA [42] conducts clustering in feature space to identify bias-conflicting samples. LWBC [22] employs committee of auxiliary classifiers to identify bias-conflicting samples more reliably. Unlike these methods, we focus on an orthogonal direction (*i.e.*, augmenting bias-conflicting samples), and BiasAdv can be easily applied to them to improve performance. Recently, several data augmentation methods [20, 25] have been proposed; BiaSwap [20] learns an image translation model [36], while DFA [25] presents feature-level augmentation based on disentangled representations. In contrast, our BiasAdv augments bias-conflicting samples by using adversarial images without generative models or disentangled representations.

Adversarial data augmentation. Utilizing adversarial images as additional training data has been extensively studied, particularly for improving the model robustness against adversarial attacks [11, 24, 32, 48, 49]. Related to our work,

there have been several attempts to debias the model using adversarial images [38,47,50]. However, these methods attack the *explicit* prediction models that directly classify the bias attribute, and hence, require full bias annotations. In contrast, BiasAdv leverages *implicit* information by employing the auxiliary model and *does not* require any bias annotations or prior knowledge of the bias type. In addition, M2m [21] for long-tailed classification, which translates the majority samples to the minority samples, shares a similar motive to our method. However, M2m requires information on whether the sample belongs to the majority class or the minority class, which is not given in our case. To the best of our knowledge, our work is the first attempt to utilize adversarial images without bias annotations, showing another good use case of adversarial attacks for debiasing.

3. Proposed Method

In this section, we describe BiasAdv in detail. We first present our problem setup in Section 3.1. Then, we describe how BiasAdv generates synthetic bias-conflicting samples and present the overall training procedure in Section 3.2. In Section 3.3, we discuss the underlying effects of BiasAdv.

3.1. Problem Setup

We consider a task of learning a classifier that classifies an input image $x \in \mathcal{X}$ as one of C classes $y \in \mathcal{Y}$ in the presence of dataset bias. Specifically, we consider a biased training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where a certain visual attribute $a \in \mathcal{A}$ of the image x is spuriously correlated to the class label y while in fact there is no causal relationship between them. In this work, we assume that we do not have annotations on the bias attribute a in the training dataset since they are expensive and laborious to obtain.

Let $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ be a classification model parameterized by $\theta \in \Theta$, which we want to optimize. A standard setting of Empirical Risk Minimization (ERM) with a proper loss function $\mathcal{L}(x, y; \theta) : \mathcal{X} \times \mathcal{Y} \times \Theta \rightarrow \mathbb{R}_+$ (*e.g.,* cross entropy loss) minimizes $\mathcal{R}(\theta)$ defined as

$$\mathcal{R}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathcal{L}(x, y; \theta)\big]. \tag{1}$$

However, since most training data are bias-guiding samples, f_θ trained by ERM exhibits high test errors for bias-conflicting samples when evaluated on unbiased test set.

In recent years, re-weighting methods [6, 22, 30, 34, 42] have been widely studied. Based on the assumption that the bias attribute a is learned more preferentially than other intrinsic attributes [30,34], these methods employ an auxiliary classification model $g_\phi : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by $\phi \in \Phi$, which is intentionally trained to make biased decisions (*i.e.,* predicting y based on a). Based on the auxiliary model g_ϕ, re-weighting methods first identify bias-conflicting samples and then train the model f_θ to be debiased in a way that emphasizes the identified bias-conflicting samples. Formally, the existing re-weighting methods can be formulated in a unified manner that minimizes the weighted empirical risk $\mathcal{R}_w(\theta)$ defined as follows,

$$\mathcal{R}_w(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\mathcal{W}(x, y; \theta, \phi) \cdot \mathcal{L}(x, y; \theta)\big], \tag{2}$$

where $\mathcal{W}(x, y; \theta, \phi)$ denotes the sample weight of (x, y). In essence, the re-weighting scheme prevents learning from being dominated by bias-guiding samples, improving the performance of bias-conflicting samples. However, due to the significant scarcity of bias-conflicting samples in the given dataset $\mathcal{D}$, re-weighting methods suffer from overfitting problems [25] and fail to learn generalizable representations, resulting in degrading performance of bias-guiding samples [20, 44]. In this work, to resolve the aforementioned issues, we propose BiasAdv, a novel data augmentation method that generates diversified bias-conflicting samples using adversarial images.

3.2. Bias-Adversarial Augmentation

Given a training pair $(x, y) \in \mathcal{D}$, the goal of BiasAdv is to generate an adversarial image x_{adv} that can act as a synthetic bias-conflicting sample for training the debiased model f_θ. We utilize a biased model g_ϕ as an auxiliary model. Note that we do not assume a specific auxiliary model, and BiasAdv can be combined with any existing re-weighting methods that can be formulated as Eq. (2). Given the biased auxiliary model g_ϕ, our insight is that an adversarial attack [10, 24, 32] on g_ϕ may alter the bias cue from the input image x, generating a synthetic bias-conflicting sample. However, since we do not use bias annotations during training, g_ϕ is not an ideal biased predictor, and the naive attack on g_ϕ risks ruining intrinsic attributes for class prediction. Therefore, to ensure that only the bias attribute is attacked, we constrain x_{adv} not to affect the class prediction of the debiased model f_θ. To this end, BiasAdv generates x_{adv} by solving the following optimization problem,

$$x_{\text{adv}} = \underset{\tilde{x}:=x+\epsilon}{\text{argmax}} \Big[\mathcal{L}(\tilde{x}, y; \phi) - \lambda \cdot \mathcal{L}(\tilde{x}, y; \theta)\Big], \tag{3}$$

where $\mathcal{L}$ denotes the cross entropy loss, $\lambda > 0$ denotes a tunable hyperparameter, and ϵ denotes an adversarial perturbation. Note that we can use any attacker to obtain ϵ, and Projected Gradient Descent (PGD) [32] is employed in this work. The first term attacks the prediction of g_ϕ, while the second term preserves the prediction of f_θ, and thus preventing intrinsic attributes from being compromised by adversarial perturbations. In a nutshell, BiasAdv translates the original image x to go across the decision boundary of g_ϕ while preserving the prediction of f_θ.

Then, the generated adversarial example x_{adv} is used as additional training data for learning the debiased model f_θ. Concretely, we train f_θ with a mixture of adversarial data

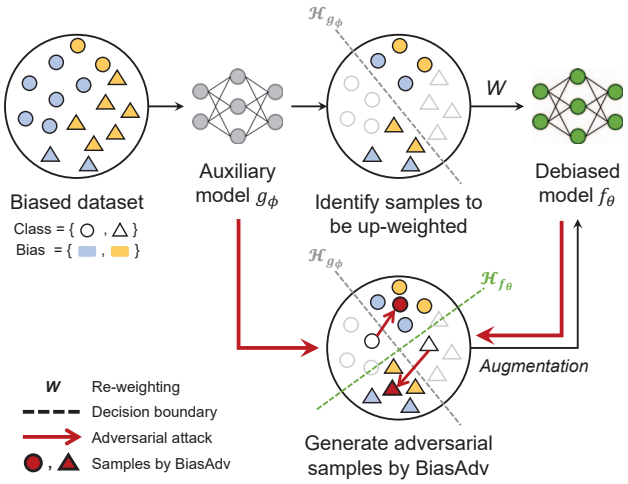


Figure 2. **The overall training procedure with BiasAdv.** During training, BiasAdv generates adversarial samples on-the-fly that go across the decision boundary of g_ϕ while preserving the prediction of f_θ. Note that BiasAdv can be easily applied to any re-weighting methods based on the auxiliary model, illustrated in the upper.

and original data, minimizing $\mathcal{R}_a(\theta)$ defined as

$$\mathcal{R}_a(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\omega_x \mathcal{L}(x,y;\theta) + \omega_{\text{adv}} \mathcal{L}(x_{\text{adv}},y;\theta)\right], \quad (4)$$

where ω_x and ω_{adv} denote the sample weights of x and x_{adv}, respectively. For ω_x, we can take advantage of existing re-weighting formula $\mathcal{W}(x,y;\theta,\phi)$ in Eq. (2) by defining $\omega_x = \mathcal{W}(x,y;\theta,\phi)$. That is, BiasAdv can be combined with any existing re-weighting methods that utilize the auxiliary model. In this case, we design ω_{adv} to trade off the sample weight ω_x as follows: $\omega_{\text{adv}} = \beta \cdot (1-\omega_x)$ where $\beta > 0$ denotes a hyperparameter that controls the importance of adversarial data. Intuitively, our method can be interpreted as complementing the insufficient learning of bias-guiding samples due to the re-weighting by translating them into synthetic bias-conflicting samples through the adversarial attack. Figure 2 illustrates the overall training procedure with our BiasAdv.

3.3. Discussion

Here, we discuss the underlying effects of BiasAdv in two aspects: (1) BiasAdv extends the decision boundary to include bias-conflicting samples. By solving Eq. (3), BiasAdv generates a set of synthetic data points $\{x_{\text{adv}}\}$ near the decision boundary of g_ϕ. At this time, as the bias attribute a is a shortcut in the learning process, we can expect that attacking the bias attribute is again the easiest shortcut to achieve Eq. (3). As a result, $\{x_{\text{adv}}\}$ can act as synthetic bias-conflicting samples. By incorporating $\{x_{\text{adv}}\}$, f_θ learns an extended decision boundary, improving the generalization of bias-conflicting samples. In Section 4.3, we will demonstrate that BiasAdv actually generates synthetic samples near bias-conflicting samples in the embedding space. (2) BiasAdv utilizes diverse and affluent information from bias-guiding samples. Since most of the samples in the dataset are bias-guiding samples, x_{adv} is mostly translated from the bias-guiding sample. As an adversarial example [15, 48], x_{adv} still has enough information about the original image. Hence, BiasAdv can be regarded as one of natural ways to leverage diverse intrinsic attributes of bias-guiding samples, allowing f_θ to learn generalizable representations. By leveraging the sample diversity, BiasAdv prevents over-fitting and improves performance not only for bias-conflicting samples but also for bias-guiding samples, which will be validated in Section 4.

4. Experiments

4.1. Experimental Setup

Datasets. To evaluate the generalization of the proposed method across various bias domains, we used one synthetic dataset and three real-world datasets: (1) Corrupted CIFAR-10 (CIFAR-10C) [25] is a synthetic dataset built upon CIFAR-10 [23] and contains spurious correlations between object classes and injected textures designed in [14]. The ratio of bias-conflicting samples in the training set was set to $p \in \{0.5\%, 2\%, 5\%\}$. For the test set, we considered unbiased test criteria where the texture biases were distributed uniformly at random. (2) Biased FFHQ (BFFHQ) [25] is a real-world facial dataset curated from FFHQ [17] where the gender attribute {Male, Female} is spuriously correlated to the class label {Young, Old}. The ratio of bias-conflicting samples in the training set was set to 0.5% following [25], and we evaluated the performance on the unbiased test set. (3) Biased Action Recognition (BAR) [34] is a real-world dataset that contains spurious correlations between six human action classes and six place attributes. Following [37], the ratio of bias-conflicting samples in the training set was set to $p \in \{1\%, 5\%\}$, and the test set consisted of only bias-conflicting samples. (4) MetaShift [26] is a recently introduced real-world dataset for evaluating contextual distribution shifts. We used "Cat vs. Dog", a subset of MetaShift, where the background context {Indoor, Outdoor} is spuriously correlated to the class label {Cat, Dog}. The ratio of bias-conflicting samples in the training set was set to $p \in \{1\%, 6\%, 12\%\}$ following the original setting [26], and we evaluated the performance on the unbiased test set.

Evaluation metrics. For quantitative evaluation, we adopted three metrics; AVERAGE (*i.e.*, accuracy (%) of all samples), CONFLICTING (*i.e.*, accuracy (%) of bias-conflicting samples), and WORST-GROUP (*i.e.*, minimum accuracy (%) among groups where each group is defined by the class label and the bias attribute). To ensure statistical robustness, we ran three independent trials and reported the mean and the standard deviation.

Table 1. **Comparison with state-of-the-art methods on CIFAR-10C.** BS denotes the model explicitly leverages bias annotations or prior knowledge of the bias type. † and ∗ denote the numbers reported from [25] and the original paper, respectively. Underline indicates performance improvement when applying BiasAdv. Best results are marked in bold.

Method	BS	AVERAGE		
		p=0.5%	p=2%	p=5%
HEX [46]†	✓	13.87	15.20	16.04
EnD [43]†	✓	22.89	31.31	40.26
ReBias [2]†	✓	22.27	31.66	43.43
BiaSwap [20]*	✗	29.11	35.25	41.62
DFA [25]†	✗	29.95	41.78	51.13
ERM	✗	21.29±0.31	29.66±0.27	37.05±0.37
ERM + BiasAdv	✗	28.43±0.45	36.80±0.23	48.00±0.11
JTT [30]	✗	23.66±0.78	31.44±0.47	41.20±0.19
JTT + BiasAdv	✗	28.83±0.65	40.10±0.37	48.44±0.18
LfF [34]	✗	28.81±0.44	40.66±0.70	50.72±1.31
LfF + BiasAdv	✗	**36.78**±0.20	**48.36**±0.59	**57.78**±0.33

Implementation details. For all experiments, we used the same ResNet-18 [13] architecture for both auxiliary and debiased models for fair comparisons. For BAR and MetaShift, we started training from the pre-trained weights on ImageNet [7], following prior works [26, 34]. Except for the experiments on BAR and MetaShift, we trained the models from scratch. To generate adversarial examples on-the-fly, we used PGD [32] attackers for all experiments with different perturbation sizes and attack steps. Specifically, we set λ in Eq. (3) to $\{1, 0.5, 0.5, 0.5\}$, the perturbation size ϵ to $\{0.7, 0.3, 0.3, 0.5\}$, the number of attack steps S to $\{5, 5, 7, 3\}$, and the weights of adversarial images β to $\{1.5, 0.5, 0.5, 1\}$ for {CIFAR-10C, BFFHQ, BAR, MetaShift}, respectively. Following [48], we applied the auxiliary batch normalization for adversarial images, since adversarial and clean images have different underlying distributions. We applied BiasAdv to three different methods to verify its effectiveness: a vanilla ERM (ERM + BiasAdv), LfF [34] (LfF + BiasAdv), and JTT [30] (JTT + BiasAdv). For ERM + BiasAdv, we set $\omega_x = 1$ and $\omega_{\text{adv}} = \beta$ in Eq. (4), and trained the auxiliary model with the GCE [51] loss, as in [25, 34]. For LfF + BiasAdv and JTT + BiasAdv, ω_x, ω_{adv}, and the design choices of the auxiliary model were defined by the re-weighting formula proposed in LfF and JTT, respectively. To implement LfF and JTT, we used the codes provided by the authors and followed the same settings.

4.2. Main Results

We compared BiasAdv with a standard ERM and recent state-of-the-art debiasing methods including HEX [46], EnD [43], ReBias [2], LfF [34], JTT [30], BiaSwap [20], DFA [25], IRMCon-IPW [37], and LWBC [22]. Note that EnD [43] leverages explicit bias annotations, and HEX [46] and ReBias [2] explicitly leverage prior knowledge of the bias type during the training phase.

Table 2. **Comparison with state-of-the-art methods on BFFHQ.** BS denotes the model explicitly leverages bias annotations or prior knowledge of the bias type. † and ∗ denote the numbers reported from [25] and the original paper, respectively. Underline indicates performance improvement when applying BiasAdv. Best results are marked in bold.

Method	BS	AVERAGE	CONFLICTING
HEX [46]†	✓	-	52.83
EnD [43]†	✓	-	56.87
ReBias [2]†	✓	-	59.46
BiaSwap [20]*	✗	79.00	58.87
DFA [25]†	✗	-	63.87
ERM	✗	76.67±0.12	54.07±0.34
ERM + BiasAdv	✗	78.67±0.12	57.73±0.19
JTT [30]	✗	80.93±0.69	62.20±1.34
JTT + BiasAdv	✗	**82.20**±0.65	64.87±1.20
LfF [34]	✗	75.23±1.60	62.97±3.22
LfF + BiasAdv	✗	81.97±1.02	**72.40**±1.34

Table 3. **Comparison with state-of-the-art methods on BAR.** BS denotes the model explicitly leverages bias annotations or prior knowledge of the bias type. † and ∗ denote the numbers reported from [37] and the original paper, respectively. Underline indicates performance improvement when applying BiasAdv. Best results are marked in bold.

Method	BS	CONFLICTING	
		p=1%	p=5%
ReBias [2]†	✓	52.10	65.00
DFA [25]†	✗	52.30	63.50
IRMCon-IPW [37]†	✗	55.30	67.90
LWBC [22]*	✗	62.03	-
ERM	✗	57.65±2.36	68.60±2.25
ERM + BiasAdv	✗	60.78±2.33	72.25±1.07
JTT [30]	✗	58.17±3.30	68.53±3.29
JTT + BiasAdv	✗	62.22±3.29	**73.29**±1.37
LfF [34]	✗	57.71±3.12	67.48±0.46
LfF + BiasAdv	✗	**63.20**±2.64	72.62±0.11

CIFAR-10C. Table 1 shows the overall results on CIFAR-10C with different bias ratios for the training set. To ensure fair comparisons, we conducted experiments under the same evaluation settings [25]. For all combined methods, BiasAdv consistently and significantly improved their performances. Notably, BiasAdv with the standard ERM already outperformed HEX and ReBias, which utilize bias-tailored modules for the specific bias type, and EnD, which

Table 4. **Comparison with state-of-the-art methods on MetaShift.** BS denotes the model explicitly leverages bias annotations or prior knowledge of the bias type. Underline indicates performance improvement when applying BiasAdv. Best results are marked in bold.

Method	BS	AVERAGE			WORST-GROUP		
		p=1%	p=6%	p=12%	p=1%	p=6%	p=12%
ERM	✗	76.91±0.93	79.69±0.86	80.56±0.14	51.85±3.42	57.41±2.29	61.34±2.29
ERM + BiasAdv	✗	77.26±1.12	78.70±1.71	80.96±1.99	53.94±2.40	58.33±3.46	63.89±1.59
JTT [30]	✗	76.97±0.71	78.65±0.49	80.38±0.99	53.47±1.13	56.48±1.82	60.65±1.43
JTT + BiasAdv	✗	**78.01**±0.72	79.34±1.07	80.79±0.43	**55.09**±0.87	**61.34**±2.80	**65.51**±2.34
LfF [34]	✗	75.06±0.79	79.28±0.45	80.85±0.14	52.78±2.04	57.17±1.99	62.73±2.29
LfF + BiasAdv	✗	76.91±1.48	**79.92**±0.74	**81.42**±1.33	53.47±2.91	58.10±1.43	64.35±1.45

uses bias annotations. Moreover, applying BiasAdv to the recent re-weighting methods, LfF and JTT, made further performance improvements. In particular, LfF + BiasAdv improved AVERAGE accuracy by 22.8%, 15.7%, and 13.0% compared to previous state-of-the-art results at $p = 0.5\%$, 2%, and 5%, respectively, significantly outperforming the complex data augmentation methods (BiaSwap and DFA).

BFFHQ. In Table 2, we report AVERAGE and CONFLICTING accuracies (%) on BFFHQ under the same evaluation settings as in [25]. Again, applying BiasAdv significantly improved performances of all baselines. In particular, LfF + BiasAdv dramatically improved AVERAGE (75.23% → 81.97%) and CONFLICTING (62.97% → 72.40%) accuracies of LfF, suggesting that BiasAdv makes the debiased model learn more generalizable representations by leveraging sample diversity, as discussed in Section 3.3. We also achieved new state-of-the-art results: 4.1% and 13.4% higher AVERAGE and CONFLICTING accuracies, respectively, compared to previous state-of-the-art methods.

BAR. Table 3 summarizes the overall results on BAR under the same evaluation settings as in [37]. The results clearly demonstrated consistent performance improvements by applying BiasAdv, which achieved very clear margins over the state-of-the-art results. In particular, when combined with JTT and LfF, BiasAdv further improved their performances remarkably and outperformed LWBC, the most recent method that utilizes multiple auxiliary models.

MetaShift. We evaluated our method on a very recently introduced real-world dataset MetaShift [26]. The results are presented in Table 4. As in other datasets, BiasAdv achieved promising improvements in both AVERAGE and WORST-GROUP accuracies for all baselines. In particular, BiasAdv markedly improved the WORST-GROUP accuracy, clearly demonstrating its debiasing capability.

Overall, applying BiasAdv to existing methods significantly improved their performances. Our successful results on four benchmark datasets across different bias domains prove the effectiveness of BiasAdv and its general applicability to recent debiasing methods.

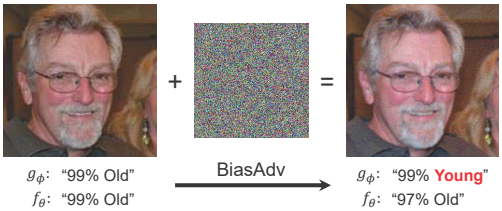


Figure 3. **An example of generated image by BiasAdv.** BiasAdv attacks the prediction of g_ϕ while preserving the prediction of f_θ. The noise image is normalized to [0, 1] for better visibility.

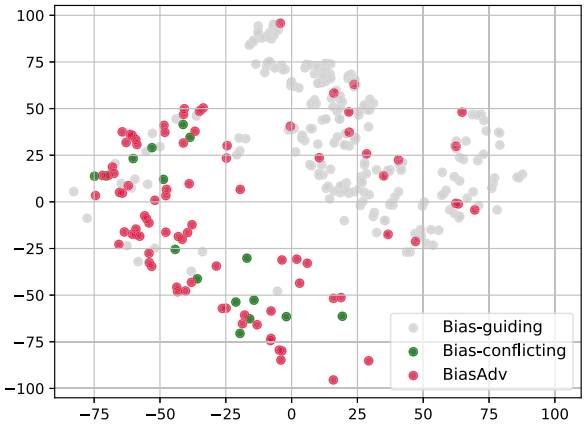
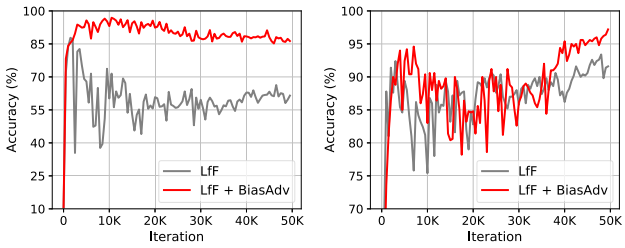


Figure 4. **Visualization of feature embeddings via t-SNE [45].** We compare the original bias-guiding and bias-conflicting samples from the BFFHQ dataset with the samples generated by BiasAdv. In this plot, we visualize the distribution of samples that have the same class label (*i.e.*, Young). The generated samples by BiasAdv can act as synthetic bias-conflicting samples.

4.3. Analysis

Does BiasAdv generate bias-conflicting samples? Figure 3 shows an example of the generated adversarial image by BiasAdv. Although BiasAdv clearly changed the prediction of g_ϕ (*i.e.*, Old → Young) while preserving the prediction of f_θ, the resulting image raises the question whether it can really act as a synthetic bias-conflicting sam-



(a) CIFAR-10C (b) BFFHQ

Figure 5. **Performance of bias-guiding samples.** Average accuracies (%) of bias-guiding samples in the test set during training on CIFAR-10C (a) and BFFHQ (b), respectively. BiasAdv achieves significantly higher bias-guiding accuracy than the baseline at the end of training.

Table 5. **Performance comparison of ablation models.** We evaluate AVERAGE and CONFLICTING accuracies (%) of variants of BiasAdv with ERM and LfF [34] on the BFFHQ dataset.

Method	AVERAGE	CONFLICTING
ERM	76.67±0.12	54.07±0.34
+ Random	77.22±0.27	55.00±0.55
+ AdvProp [48]	75.10±0.46	50.68±1.03
+ BiasAdv ($\lambda = 0$)	77.80±0.30	56.16±0.74
+ BiasAdv	**78.67**±0.12	**57.73**±0.19
LfF [34]	75.23±1.60	62.97±3.22
+ Random	80.37±0.63	64.27±1.27
+ AdvProp [48]	79.38±0.43	60.80±1.05
+ BiasAdv ($\lambda = 0$)	81.07±0.74	66.13±2.17
+ BiasAdv	**81.97**±1.02	**72.40**±1.34

ples (*i.e.*, old woman), since the adversarial perturbation is almost negligible from the human perspective. To verify that BiasAdv generates meaningful bias-conflicting samples from a network perspective, we visualized and compared penultimate features of original bias-guiding samples, original bias-conflicting samples, and generated samples by BiasAdv, using t-SNE [45]. Specifically, we first trained an auxiliary model and a debiased model using LfF [34] on the BFFHQ dataset [25]. We then applied BiasAdv to these trained models to generate adversarial images. Finally, we used t-SNE to visualize feature embeddings of the original samples from the test set and the generated samples where each embedding was obtained from the output of the penultimate layer of the trained debiased model. Figure 4 illustrates the results. As expected, bias-guiding samples and bias-conflicting samples were distributed separately from each other. Notably, the samples generated by BiasAdv were overlaid with bias-conflicting samples. This observation suggests that these synthetic adversarial images by BiasAdv can genuinely act as bias-conflicting samples for training the debiased model, even if adversarial perturbations are rarely recognized at the human level as shown in Figure 3. It is also aligned to a recent claim that adversarial perturbation is not a *bug* in neural networks, but a *generalizable* feature [15, 21, 48].

Performance of bias-guiding samples. Comparing ERM and LfF in Table 2, LfF improved CONFLICTING accuracy but rather degraded AVERAGE accuracy (- 1.44%p) due to the poor performance of bias-guiding samples, as reported in [20]. A well-generalized model should work well for both bias-guiding and bias-conflicting samples. Hence, we demonstrate the effectiveness of BiasAdv on maintaining the performance of bias-guiding samples. In Figure 5, we display the average accuracies (%) of bias-guiding samples in the test set during the training of LfF and LfF + BiasAdv, on CIFAR-10C with p=0.5% and BFFHQ, respectively. In the case of LfF, the performance of bias-guiding samples degraded as training progresses, which implies that LfF was over-fitted to an insufficient number of bias-conflicting samples. Applying BiasAdv, on the other hand, maintained good bias-guiding performance and achieved significantly higher bias-guiding accuracy at the end of training. These results support that BiasAdv leads to learning generalizable representations and reducing over-fitting.

Ablation studies. To validate the effectiveness of BiasAdv, we compared BiasAdv to three ablation models. First, to analyze whether the performance improvements brought about by BiasAdv were simply the result of the regularization power of adversarial images, we considered two ablation models: Random and AdvProp [48]. The Random model augments data by adding random noise instead of BiasAdv. The AdvProp model uses adversarial images that attack the debiased model instead of the auxiliary model (*i.e.*, adversarial training as in [48]). Lastly, to verify the effect of the regularization term $\lambda \cdot \mathcal{L}(\tilde{x}, y; \theta)$ of BiasAdv in Eq. (3), we considered an ablation model in which λ is set to 0; BiasAdv ($\lambda = 0$). For all ablation models and BiasAdv, ERM and LfF [34] were considered as the baselines. Table 5 summarizes the overall results of the ablation models and BiasAdv on the BFFHQ dataset. Adding random noise yielded slight performance improvements but was not promising. However, AdvProp, which adds adversarial noise that attacks the debiased model seriously degraded performance. In contrast, BiasAdv ($\lambda = 0$), which only attacks the auxiliary model yielded significant performance improvements. This observation reveals that attacking the biased auxiliary model plays a pivotal role in making our method work. That is, the performance improvement resulting from BiasAdv is attributed to the generation of synthetic bias-conflicting samples, as discussed in Figure 4, rather than the regularization power of adversarial images. Lastly, the use of the regularization term in Eq. (3) contributed to promising performance improvements. The regularization term prevents

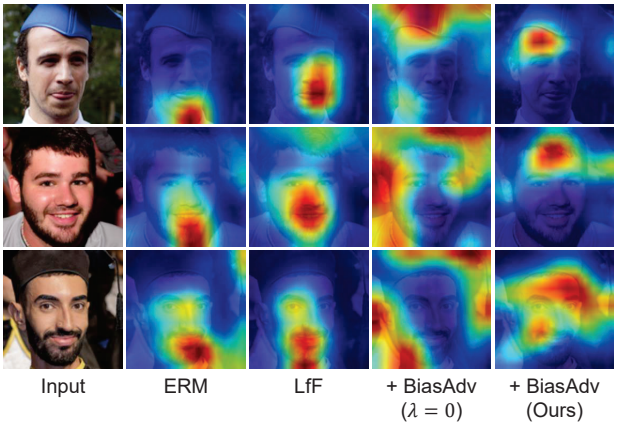


Figure 6. **Grad-CAM [41] comparison of ablation models.** We analyze the class activation maps of the test set images on the BFFHQ dataset.

intrinsic attributes from being compromised by adversarial perturbations and improves the quality of the generated samples, providing additional performance gains.

Visualization of Grad-CAM. To understand the qualitative effects of BiasAdv, we investigated the Grad-CAM [41] of the test set images of BFFHQ. Specifically, we compared our LfF + BiasAdv model to three baselines; ERM, LfF, and LfF + BiasAdv ($\lambda = 0$). In Figure 6, we show the class activation maps for predicting the age. It is noteworthy that ERM and LfF highlighted beard and mustache, which are strongly related to the gender (*i.e.*, Male), implying that these models made decisions based on the bias attribute. Only attacking the biased auxiliary model g_ϕ (*i.e.*, $\lambda = 0$) drived the model to focus on areas other than the bias attribute, but it was often, a completely wrong area such as background. With the proposed regularization constraint to maintain the prediction score of the debiased model f_θ, BiasAdv contributed to better semantic focus, attending on discriminative regions for the age prediction, yet neutral from the gender, such as the forehead. These observations imply that our BiasAdv guides the debiased model to capture intrinsic attributes for the target class, supporting the superior generalization performances for unbiased test criteria presented in Section 4.2.

Model robustness. To further demonstrate the effectiveness of BiasAdv, we evaluated the model robustness to various input corruptions following the protocol [29]. Specifically, we considered eight corruption types that were not used for training; additive noises (Gaussian and Salt & Pepper), dropping pixels (cutout and dropout), affine transformation (rotation and perspective), and image quality deterioration (JPEG-compression and Gaussian blur). After training on the original BFFHQ dataset, we evaluated the CONFLICT accuracy with corrupted test images. In Table 6, we compare the results of ERM, LfF, and LfF + BiasAdv. ERM and LFF yielded severely degrading performance despite small changes in the input image, achieving near 50% accuracy. In contrast, applying BiasAdv significantly improved the model robustness, achieving robust and superior performance regardless of corruption type. In particular, the superiority of BiasAdv was more obvious as the corruption worsens such as Cutout or JPEG-compression. The results clearly demonstrate that BiasAdv allows the model to learn more generalizable representations that are less affected by distracting noises.

Table 6. **Model robustness to unseen input corruptions.** We report the accuracy (%) of conflicting samples (*i.e.*, CONFLICTING) with eight unseen input corruptions (*i.e.*, not used in the training) on the BFFHQ dataset. + BiasAdv denotes that BiasAdv is applied to LfF [34]. Best results are marked in bold.

Input corruption	ERM	LfF [34]	+ BiasAdv
Additive noise:			
Gaussian	53.00±0.28	56.86±0.52	**69.60**±0.44
Salt & Pepper	52.86±0.18	56.67±0.19	**68.73**±0.99
Dropping pixels:			
Cutout	51.80±0.81	53.79±0.71	**68.00**±1.00
Dropout	51.20±1.45	52.72±0.94	**60.73**±1.08
Affine transformation:			
Rotation	52.06±0.66	53.40±1.56	**61.03**±1.97
Perspective	49.39±0.98	48.33±0.98	**58.27**±0.25
Image quality deterioration:			
JPEG-compression	48.72±1.51	53.66±0.19	**65.80**±0.59
Gaussian blur	49.72±0.96	49.93±0.74	**58.13**±0.66

5. Conclusion

In this paper, we propose BiasAdv, a novel data augmentation method for debiasing that supplements bias-conflicting samples using adversarial attacks. We find that BiasAdv can generate meaningful synthetic bias-conflicting samples, even with small adversarial perturbations. By leveraging the diversified synthetic bias-conflicting data, BiasAdv enables the model to learn more generalizable representations. The implementation of BiasAdv is simple and can be easily integrated into any debiasing methods based on re-weighting without architectural or algorithmic changes. Our extensive experimental results on four benchmark datasets across various bias domains demonstrate the effectiveness and general applicability of BiasAdv, and we achieve state-of-the-art performance by large margins for all benchmarks. We believe our work can provide a universal and promising data augmentation method for future work on learning debiased representations.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1

[2] 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*, pages 528–539. PMLR, 2020. 1, 2, 5

[3] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019. 1, 2

[4] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019. 2

[5] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in neural information processing systems*, 32, 2019. 1, 2

[6] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021. 1, 2, 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[8] 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. 1

[9] 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

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 3

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2

[12] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

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

[15] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 4, 7

[16] Sangwon Jung, Sanghyuk Chun, and Taesup Moon. Learning fair classifiers with partially annotated group labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10348–10357, 2022. 2

[17] 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

[18] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017. 2

[19] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020, 2019. 1

[20] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14992–15001, 2021. 2, 3, 5, 7

[21] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020. 3, 7

[22] Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. *arXiv preprint arXiv:2206.10843*, 2022. 1, 2, 3, 5

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[24] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. 2, 3

[25] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25123–25133. Curran Associates, Inc., 2021. 2, 3, 4, 5, 6, 7

[26] Weixin Liang and James Zou. Metashift: A dataset of datasets for evaluating contextual distribution shifts and training conflicts. *arXiv preprint arXiv:2202.06523*, 2022. 1, 2, 4, 5, 6

[27] Jongin Lim, Hyung Jin Chang, and Jin Young Choi. Pmnet: Learning of disentangled pose and movement for unsupervised motion retargeting. In *British Machine Vision Conference (BMVC)*, 2019. 2

[28] Jongin Lim, Youngjoon Yoo, Byeongho Heo, and Jin Young Choi. Pose transforming network: Learning to disentangle human posture in variational auto-encoded latent space. *Pattern Recognition Letters*, 112:91–97, 2018. 2

[29] Jongin Lim, Sangdoo Yun, Seulki Park, and Jin Young Choi. Hypergraph-induced semantic tuplet loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 212–222, 2022. 8

[30] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. 1, 2, 3, 5, 6

[31] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019. 2

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2, 3, 5

[33] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019. 2

[34] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[35] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *International Conference on Learning Representations*, 2021. 2

[36] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020. 2

[37] Jiaxin Qi, Kaihua Tang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Class is invariant to context and vice versa: On learning invariance for out-of-distribution generalization. In *ECCV*, 2022. 4, 5, 6

[38] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *European Conference on Computer Vision*, pages 19–37. Springer, 2020. 3

[39] 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. 1, 2

[40] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020. 2

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8

[42] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Unsupervised learning of debiased representations with pseudo-attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16742–16751, 2022. 1, 2, 3

[43] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13508–13517, 2021. 1, 5

[44] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. *arXiv preprint arXiv:2005.00315*, 2020. 2, 3

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6, 7

[46] 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, 5

[47] Zhibo Wang, Xiaowei Dong, Henry Xue, Zhifei Zhang, Weifeng Chiu, Tao Wei, and Kui Ren. Fairness-aware adversarial perturbation towards bias mitigation for deployed deep models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10379–10388, 2022. 3

[48] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020. 2, 4, 5, 7

[49] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 501–509, 2019. 2

[50] Yi Zhang and Jitao Sang. Towards accuracy-fairness paradox: Adversarial example-based data augmentation for visual debiasing. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4346–4354, 2020. 3

[51] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 2, 5