Revisiting the Importance of Amplifying Bias for Debiasing

Jungsoo Lee[*1,2], **Jeonghoon Park**[*1,2], **Daeyoung Kim**[*1],
Juyoung Lee[2], **Edward Choi**[1], **Jaegul Choo**[1]

[1] KAIST
[2] Kakao Enterprise, South Korea
bebeto@kaist.ac.kr, jeonghoon_park@kaist.ac.kr, daeyoung.k@kaist.ac.kr,
michael.jy@kakaoenterprise.com, edwardchoi@kaist.ac.kr, jchoo@kaist.ac.kr

Abstract

In image classification, *debiasing* aims to train a classifier to be less susceptible to dataset bias, the strong correlation between peripheral attributes of data samples and a target class. For example, even if the frog class in the dataset mainly consists of frog images with a swamp background (*i.e.,* bias-aligned samples), a debiased classifier should be able to correctly classify a frog at a beach (*i.e.,* bias-conflicting samples). Recent debiasing approaches commonly use two components for debiasing, a biased model f_B and a debiased model f_D. f_B is trained to focus on bias-aligned samples (*i.e.,* overfitted to the bias) while f_D is mainly trained with bias-conflicting samples by concentrating on samples which f_B fails to learn, leading f_D to be less susceptible to the dataset bias. While the state-of-the-art debiasing techniques have aimed to better train f_D, we focus on training f_B, an overlooked component until now. Our empirical analysis reveals that removing the bias-conflicting samples from the training set for f_B is important for improving the debiasing performance of f_D. This is due to the fact that the bias-conflicting samples work as noisy samples for amplifying the bias for f_B since those samples do not include the bias attribute. To this end, we propose a *simple yet effective* data sample selection method which removes the bias-conflicting samples to construct a bias-amplified dataset for training f_B. Our data sample selection method can be directly applied to existing reweighting-based debiasing approaches, obtaining consistent performance boost and achieving the state-of-the-art performance on both synthetic and real-world datasets.

Introduction

When there exists a correlation between peripheral attributes and labels which is referred to as *dataset bias* (Torralba and Efros 2011) in the training dataset, image classification models often heavily rely on such a bias. Dataset bias is caused when the majority of data samples include bias attributes, the visual attributes that frequently co-occur with the target class but not innately defining it (Lee et al. 2021). For example, frogs are commonly observed in swamps (bias attribute), but frogs can also be found in other places such as grasses or beaches. In such a case, the image classification model trained with the biased dataset could use swamps as the visual cue for classifying frogs. In other words, it may fail to correctly classify frog images in other places. To mitigate such an issue, debiasing aims to train the image classification model to learn the intrinsic attributes, the visual attributes which inherently define a target class, such as the legs or eyes of frogs.

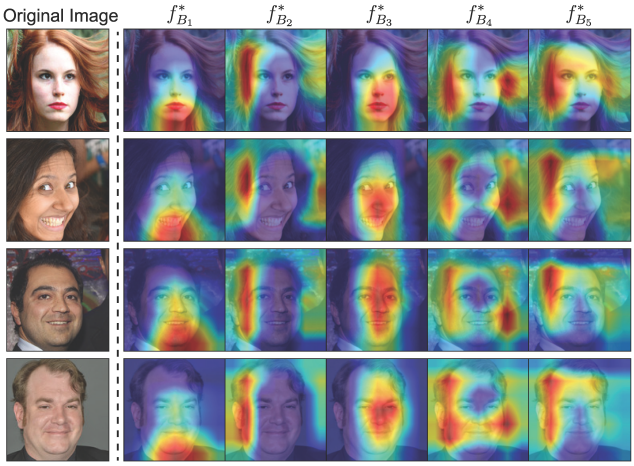


Figure 1: Visualization results of GradCAM applied to biased models with different initialization (*i.e.,* $f^*_{B_1}, \ldots, f^*_{B_5}$). The first column indicates the original images. Starting from the second column, we observe that each biased model with different initialization focuses on different visual attributes.

In a biased dataset, the data samples without the bias attribute (*i.e.,* bias-conflicting samples) such as frogs on grasses or beaches are excessively scarce compared to the samples including the bias attributes (*i.e.,* bias-aligned samples). Due to the scarcity, existing state-of-the-art debiasing studies (Nam et al. 2020; Lee et al. 2021) train a given model by *reweighting* the data samples which refers to imposing 1) high weight on losses of the bias-conflicting samples and 2) low weight on those of the bias-aligned ones. For example, Nam *et al.* (2020) reweight the data samples based on the finding that the bias attributes are *easy to learn* compared to the intrinsic attributes. To be more specific, they intentionally train a biased model f_B to be overfitted to the easily learned bias attribute. Then, they utilize f_B for computing a reweighting value w for each training sample, which the value is designed to be high for samples f_B fails to classify (*i.e.,* bias-conflicting samples). The data items are

[*]These authors contributed equally.

reweighted with w during training the model f_D to learn the debiased representation. In this regard, how well f_B is overfitted to the bias attribute influences the debiasing performance of f_D since it determines the reweighting value.

However, our careful analysis points out that f_B used in the existing reweighting-based approaches (Nam et al. 2020; Lee et al. 2021) fail to maximally exploit the bias attribute. They utilize a loss function which is designed to emphasize the bias-aligned samples in order to overfit f_B to bias attributes. Despite such a design, even the small number of bias-conflicting samples still interfere with f_B from being overfitted to the bias attribute since they work as noisy samples for learning the bias. In spite of the importance of such an issue, none of the previous studies shed light on removing the bias-conflicting samples from training sets for overfitting f_B to the bias attribute to improve the debiasing performance, especially challenging without explicit bias labels (*i.e.,* annotations of bias attributes) or prior knowledge on certain bias.

To this end, we propose a *simple yet effective* biased sample selection method that builds a refined dataset which discards bias-conflicting samples and mainly includes the bias-aligned ones in order to amplify bias when training f_B. While the bias attribute (*e.g.,* gender) is easy to learn (Nam et al. 2020), it is composed of multiple visual attributes (*e.g.,* make-up, hairstyle, beards). We found that it is challenging for a biased model to consider multiple visual attributes comprehensively for making biased predictions. To be more specific, as shown in Fig. 1, differently initialized biased models only utilize certain visual attributes for making biased predictions. Additionally, these attributes are different among the models (*e.g.,* one focuses on the lips mainly while the other concentrates on the hairstyle for predicting the gender). Such a finding aligns with the previous studies which found that deep neural networks learn in different ways with different random initialization (Bian and Chen 2021; Zaidi et al. 2021). This observation indicates that we can better understand the bias by capturing diverse visual attributes of the bias by utilizing the predictions of multiple biased models.

The procedure of our proposed method is as follows. First, in order to capture diverse visual attributes of a bias, we pretrain multiple biased models with different random initialization for a small number of iterations by using the *easy-to-learn* property of bias attributes (Nam et al. 2020). By utilizing the predictions of the differently initialized biased models, we refine the train dataset with the bias-conflicting samples discarded. The newly refined dataset which mainly includes bias-aligned samples is then used to train f_B. Training with the bias-amplified dataset encourages f_B to maximally exploit the bias attribute when making predictions, and improve the debiasing performance of f_D overall.

In summary, the main contributions of our paper are as follows:

- Based on our preliminary analysis, we reveal that how well f_B is overfitted to the bias influences the debiasing performance crucially, an important observation overlooked in the previous reweighting-based approaches.
- We propose a *simple yet effective* biased sample selection method which better captures a bias attribute by considering multiple visual attributes of a bias.
- Our method can be easily adopted to existing reweighting-based approaches, and we achieve the new state-of-the-art performances on both synthetic and real-world datasets.

Related Work

Existing early studies of debiasing explicitly use bias labels during training (Kim et al. 2019; Tartaglione, Barbano, and Grangetto 2021; Sagawa et al. 2020) or implicitly predefine the bias types (*e.g.,* focusing on mitigating the color bias) (Wang et al. 2019; Geirhos et al. 2019; Bahng et al. 2020). Bias labels or prior knowledge on the bias types are generally used to identify bias-conflicting samples. Although not utilizing explicit bias labels, ReBias (Bahng et al. 2020) predefines a certain bias type (*e.g.,* color and texture) and focuses on mitigating such bias by leveraging a color- and texture-oriented network with small receptive fields (Brendel and Bethge 2019) to capture the predefined color or texture bias. However, acquiring bias labels or predefining a bias type 1) necessitates humans to identify the bias type of a given dataset and 2) limits the debiasing performance on unknown bias types (Lee et al. 2021).

Recent debiasing studies proposed several methods to address such an issue (Darlow, Jastrzebski, and Storkey 2020; Huang et al. 2020; Nam et al. 2020; Lee et al. 2021). Nam *et al.* (2020) propose LfF which identifies the bias-conflicting samples based on the intuitive finding that the bias attributes are *easy to learn* compared to the intrinsic attributes. By using the fact that f_B outputs a relatively high loss value for bias-conflicting samples, they impose high weight on (*i.e.,* emphasize) bias-conflicting samples and low weight on the bias-aligned samples during training f_D. Lee *et al.* (2021) augment the bias-conflicting samples via disentangled feature-level augmentation, emphasizing them along with the bias-conflicting samples in the original training set by using the reweighting method. Although the previous studies utilize f_B for computing the reweighting value, we reveal that they overlooked the importance of amplifying bias for f_B, crucial for improving the debiasing performance of f_D.

Importance of Amplifying Bias

Background

Overfitting model to the bias. Since annotating bias labels or identifying the bias types in advance is challenging and labor intensive (Lee et al. 2021), recent studies leverage the Generalized Cross Entropy (GCE) loss (Zhang and Sabuncu 2018) that does not require such information for amplifying the bias (Nam et al. 2020; Lee et al. 2021). The GCE loss is defined as:

$$\mathcal{L}_{\text{GCE}}(p(x;\theta),y) = \frac{1-p_y(x;\theta)^q}{q}, \tag{1}$$

where q is a scalar value which controls the degree of amplification, and $p(x;\theta)$ and $p_y(x;\theta)$ are the softmax outputs of

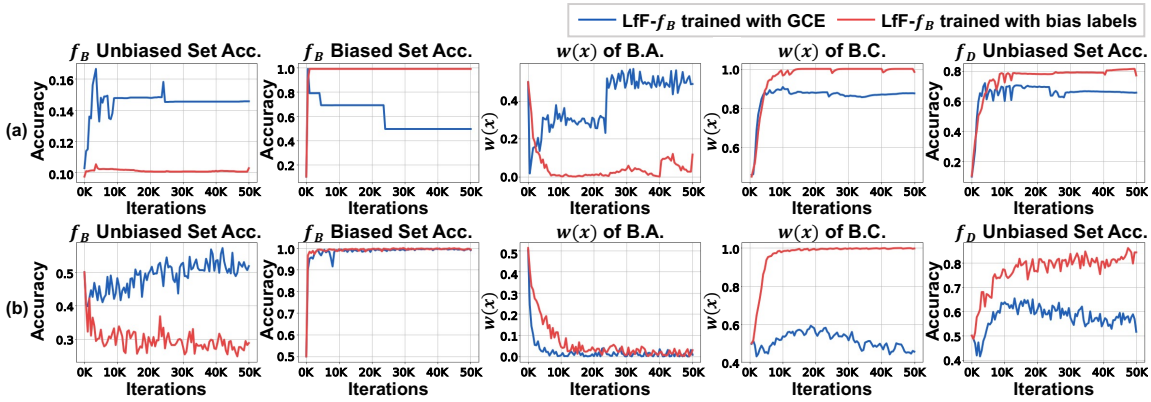


Figure 2: Comparison of LfF utilizing f_B trained with 1) GCE loss (blue) and 2) explicit bias labels (red). (a) and (b) indicate the results on Colored MNIST and BFFHQ, respectively. Starting from the first column, each graph represents the 1) unbiased test set accuracy of f_B, 2) biased test set accuracy of f_B, 3) averaged reweighting value $w(x)$ of bias-aligned samples (abbreviated as B.A.), 4) that of bias-conflicting samples (abbreviated as B.C.), and 5) unbiased test set accuracy of f_D.

the network parameterized by θ and the softmax probability of the target class y, respectively. The GCE loss assigns high weights on the gradients of the samples with the high prediction probability on the target class y, which can be formulated as:

$$\frac{\partial \mathcal{L}_{\text{GCE}}(p,y)}{\partial \theta} = p_y^q \frac{\partial \mathcal{L}_{\text{CE}}(p,y)}{\partial \theta}. \tag{2}$$

The GCE loss encourages the model to focus on the easy samples with high probability values. As revealed in the work of Nam *et al.* (2020), the bias attributes are easy to learn compared to the intrinsic attributes, so a model predicts bias-aligned samples with high probability values. Due to this fact, in a biased dataset, the GCE loss encourages the model to focus mainly on the bias-aligned samples, leading the model to be biased.

Reweighting-based approaches. Recent state-of-the-art debiasing methods (Nam et al. 2020; Lee et al. 2021) reweight data samples by utilizing two different models: 1) a biased model f_B and 2) a debiased model f_D. The former one is trained to be overfitted to the bias attribute while the latter one is mainly trained with the bias-conflicting samples, those which are identified by utilizing f_B. To be more specific, since f_B heavily relies on the bias attributes for making predictions, it fails to correctly classify the bias-conflicting samples, those without the bias attributes. Due to this fact, the Cross Entropy (CE) loss values of bias-conflicting samples are relatively high compared to those of bias-aligned ones. By utilizing such a characteristic, the loss of each data sample x is reweighted for training f_D with the reweighting value $w(x)$. Specifically, Nam *et al.* (2020) formulated $w(x)$ as

$$w(x) = \frac{\mathcal{L}_{\text{CE}}(f_B(x), y)}{\mathcal{L}_{\text{CE}}(f_B(x), y) + \mathcal{L}_{\text{CE}}(f_D(x), y)}, \tag{3}$$

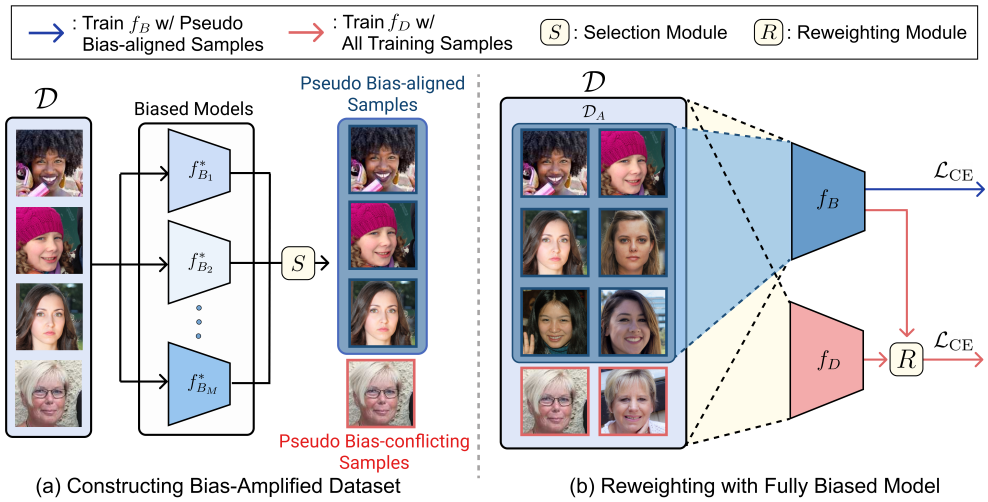
where $f_B(x)$ and $f_D(x)$ indicate the prediction outputs of f_B and f_D, respectively, and y is the target label of the sample x. Using the formula, the reweighting value $w(x)$ is designed to be imposed 1) high for the bias-conflicting samples and 2) low for the bias-aligned samples in order to improve the debiasing performance of f_D. In this regard, how well f_B is overfitted to the bias attribute determines the $w(x)$ which crucially influences the debiasing performance of f_D.

Revisiting f_B in Debiasing Methods

In this section, we show that the existing state-of-the-art reweighting methods fail to fully overfit f_B to the bias, resulting in an unsatisfactory reweighting of the data samples during training f_D overall. For the experiments, we use Colored MNIST (Lee et al. 2021) and biased FFHQ (BFFHQ) (Kim, Lee, and Choo 2021) to demonstrate that our analysis is applicable both on synthetic and real-world datasets. Bias-conflicting samples consist 1% of both training sets in this analysis. Detailed descriptions of the datasets are included in Supplementary. In Fig. 2, we compare 1) LfF (Nam et al. 2020) training f_B with GCE loss (blue) and 2) LfF training f_B with explicit bias labels using Cross Entropy (CE) loss (red). For the evaluation, we use 1) a biased test set, a dataset having a similar data distribution as the biased training set, and 2) the unbiased test set, a dataset which has no correlation found in the biased training set.

Imperfectly biased f_B. A fully biased f_B is likely to achieve 1) high accuracy on the biased test set and 2) low accuracy on the unbiased test set since it only uses the bias attribute as the visual cue for predictions. In other words, the gap between the biased test set accuracy and the unbiased test set accuracy increases as f_B focuses on the bias attribute. As shown in Fig. 2, however, f_B trained with GCE loss (blue) shows relatively higher unbiased test set accuracy compared to f_B trained with the explicit bias labels (red). Assuming that f_B trained with the explicit bias labels is perfectly overfitted to the bias attribute, such results demonstrate that f_B trained with GCE loss is less overfitted to the bias. In other words, even the small number of bias-conflicting samples work as noisy samples for learning bias.

Debiasing f_D via f_B. The reweighting value $w(x)$ determines the degree of how much f_D should focus on a given sample x during the training phase. It is crucial to satisfy two conditions simultaneously for training a debiased classifier f_D: imposing 1) high $w(x)$ on the bias-conflicting samples and 2) low $w(x)$ on the bias-aligned samples. In other words,



(a) Constructing Bias-Amplified Dataset (b) Reweighting with Fully Biased Model

Figure 3: Illustration of BiasEnsemble. (a) By filtering out pseudo bias-conflicting samples detected via utilizing M pretrained biased models $(f^*_{B_1}, \ldots, f^*_{B_M})$, we obtain the bias-amplified dataset $\mathcal{D}_A$. (b) Then, we train f_B with $\mathcal{D}_A$ while training f_D with the original training set $\mathcal{D}$. S and R indicate the sample selection module and reweighting module, respectively. Although not used for training f_B, the pseudo bias-conflicting samples are still fed to f_B for obtaining the reweighting value used for training f_D.

the difference between $w(x)$ of bias-conflicting samples and that of bias-aligned samples, $w(x)_{\text{diff}}$, should be large in order to improve the debiasing performance. We observe that LfF trained with GCE loss, however, outputs relatively small $w(x)_{\text{diff}}$ compared to LfF trained with explicit bias labels (third and fourth column in Fig. 2). This is due to utilizing a less overfitted f_B for computing $w(x)$. Low $w(x)$ on bias-conflicting samples indicates that they are less emphasized in training f_D, which should be emphasized for learning debiased features (fourth column in Fig. 2). Therefore, f_D of LfF trained with GCE loss shows lower test accuracy than the one trained with explicit bias labels (fifth column in Fig. 2). Based on the finding that how well f_B is overfitted to the bias significantly influences the debiasing performance of f_D, we propose an approach which further amplifies the bias for training f_B.

Debiasing with Bias-Amplified Dataset

Detecting Bias-Conflicting Samples

At a high level, since even a small number of bias-conflicting samples work as noisy samples for learning the bias, we discard them and build a bias-amplified dataset, mainly consisted of bias-aligned ones. We pretrain an additional biased model f^*_B with GCE loss for *a small number of iterations* by utilizing the property that the bias attribute is easy to learn in the early training phase (Nam et al. 2020). Note that f^*_B is a pretrained biased model while f_B is the biased model used for reweighting data samples during training f_D. Since we pretrain f^*_B only for a small number of iterations, our method requires a minimal amount of additional computational costs.

As f^*_B is overfitted to the bias at a certain degree, it mainly 1) correctly classifies the bias-aligned samples and 2) misclassifies the bias-conflicting samples. In other words, the model outputs 1) high confidence (*i.e.,* the softmax probability) on the target class for the bias-aligned samples and 2) low confidence for the bias-conflicting samples. By utilizing the probability of the target class p_y, we build the bias-conflicting detector *BCD* as follows:

$$BCD(x;\tau,f) = \begin{cases} 0, & \text{if } p_y(x;f) < \tau \\ 1, & \text{if } p_y(x;f) \geq \tau \end{cases}, \tag{4}$$

where τ is the confidence threshold. The detector regards the samples with confidence higher than the threshold as bias-aligned samples and vice versa.

Improving Detection via Multiple BCDs

While a single BCD may discard the bias-conflicting samples at a reasonable level, we empirically found that constructing $\mathcal{D}_A$ relying on only a single BCD shows large performance variations (Table 3 and Table 4). Although the bias attribute (*e.g.,* gender bias) is easy to learn compared to the intrinsic attribute in the early training phase, it may form as a combination of multiple visual attributes (*e.g.,* make-up, hairstyle, beards), especially in the real-world datasets. As shown in Fig. 1, differently initialized biased models only utilize certain visual attributes for learning the bias attribute. Also, the visual attributes utilized for the biased predictions are different among models. For example, one BCD captures the gender bias of a female image by mainly using the long hair as the visual cue while the other may recognize the bias mainly due to the lip makeups. Thus, each BCD may make different predictions on a same sample, leading to performance variation overall. One of the straight-forward solutions is considering both visual attributes to predict gender bias (*i.e.,* predicting an image as female if it includes both long hair and lip makeups). As demonstrated in the previous studies (Bian and Chen 2021; Fort, Hu, and Lakshminarayanan 2019; Zaidi et al. 2021), utilizing differently initialized models enables to induce diversity among models. Thus, we utilize multiple BCDs to better capture the bias via

considering diverse visual attributes consisting the bias attribute. Since we utilize multiple BCDs, we term our method as 'BiasEnsemble (BE)'.

To this end, we select data samples based on the predictions of multiple BCDs. To be more specific, we leverage multiple pretrained biased models ($f^*_{B_1}, f^*_{B_2}, ..., f^*_{B_M}$). We utilize the property that bias attribute is learned in the *early training phase* (Nam et al. 2020), so we only need a negligible training time for each f^*_B. Quantitative measurement on the marginal computational costs of our method is reported in our Supplementary. Then, M number of BCDs are built using each pretrained biased model f^*_B. Finally, we discard the sample that the majority of the detectors consider as the bias-conflicting sample. For example, setting M=5, a given sample is regarded as the bias-conflicting sample if more or equal to three BCDs considered it as the bias-conflicting one (*i.e.,* pseudo bias-conflicting sample) and vice versa. Note that all biased models have the same architecture, so we iteratively re-initialize biased models in order to save the memory space. As aforementioned, even such iterative re-initialization accompanies a marginal training time since we train each biased model for a small number of steps.

In summary, pseudo bias-aligned sample (PBA) can be formulated as

$$PBA = \begin{cases} 0, & \text{if } \sum_{i=1}^{M} BCD(x;\tau,f^*_{B_i}) < \lceil \frac{M}{2} \rceil \\ 1, & \text{if } \sum_{i=1}^{M} BCD(x;\tau,f^*_{B_i}) \geq \lceil \frac{M}{2} \rceil \end{cases}, \quad (5)$$

where M is the number of BCDs used. Finally, the data samples labeled as pseudo bias-aligned ones consist $\mathcal{D}_A$, used for training f_B.

In the Supplementary, given that bias labels are not provided, we show that BiasEnsemble is superior to simply ensembling multiple biased models (f_{B_1}, f_{B_2}, ..., f_{B_M}) in the main stage of debiasing. Without the bias-conflicting samples discarded, the ensembled predictions of the multiple biased models fail to emphasize the bias-conflicting ones for f_D. That is, each biased model (f_{B_i}) learns the intrinsic attribute from the bias-conflicting samples as training proceeds. While ensembling has been widely adopted in other fields to bring further performance gain, this experiment shows that ensembling without careful consideration does not guarantee performance gain in debiasing. Although ensembling itself may be regarded as a simple and naive approach, we believe that finding how to adjust ensembling to debiasing is important and needs careful consideration.

Training Debiased Model Using $\mathcal{D}_A$

After obtaining a bias-amplified dataset $\mathcal{D}_A$ which contains a significantly smaller number of bias-conflicting samples compared to the original training dataset $\mathcal{D}$, we train f_B using $\mathcal{D}_A$. When applying BiasEnsemble to existing reweighting-based approaches, LfF (Nam et al. 2020) and DisEnt (Lee et al. 2021), we do not modify the training procedure of f_D. Thus, BiasEnsemble can be easily applied to existing methods that leverage f_B for *reweighting* data samples. Note that f_B is utilized for reweighting all the training data samples during training f_D, although the pseudo bias-conflicting ones (*i.e.,* the samples not included in $\mathcal{D}_A$) are not used for *training* f_B. Both f_B and f_D are trained with the CE loss.

Experiment

Experimental Settings

Dataset. Following the previous studies, we conduct experiments under four datasets: Colored MNIST (Lee et al. 2021), biased FFHQ (BFFHQ) (Kim, Lee, and Choo 2021), Dogs & Cats (Kim et al. 2019), and biased action recognition (BAR) (Nam et al. 2020). Each dataset has an intrinsic attribute and a bias attribute: Colored MNIST - {digit, color}, BFFHQ - {age, gender}, Dogs & Cats - {animal, color}, and BAR - {action, background}. The former and the latter visual attribute in the bracket correspond to the intrinsic and bias attribute, respectively. We conduct experiments under various ratios of bias-conflicting samples (*i.e.,* the number of bias-conflicting samples out of the total number of training samples) in each dataset to evaluate the debiasing algorithms under different levels of bias severity, following the previous studies (Nam et al. 2020; Lee et al. 2021). For evaluating the debiasing performance, we use unbiased test sets which include images without the correlation found in the training set. We use datasets with 1% ratio of bias-conflicting samples for in-depth analyses.

Implementation details. Following Nam *et al.* (2020) and Lee *et al.* (2021), we use a multi-layer perceptron (MLP) which consists of three hidden layers for Colored MNIST. For the other datasets except for BAR, we train ResNet18 (He et al. 2015) with the random initialization. Since BAR has an extremely small number of images compared to other datasets, we utilize a pretrained ResNet18. We set M=5, meaning that we pretrain five biased models (*i.e.,* $f^*_{B_1}, f^*_{B_1}, \ldots, f^*_{B_5}$). While all experiments are trained for 50K iterations, each f^*_B is pretrained for 0.5K iterations on all datasets, requiring negligible amount of additional computational costs. We set the confidence threshold τ for the BCD as 0.99. Note that all the hyper-parameters are constant across all datasets and bias ratios. We report the mean of the best unbiased test set accuracy over five independent trials. We include the remaining details of datasets and implementation in the Supplementary.

Comparisons on Unbiased Test Sets

Table 1 compares the image classification accuracies of the debiasing approaches on the unbiased test sets. As aforementioned, we applied BiasEnsemble on the state-of-the-art reweighting-based approaches, LfF (Nam et al. 2020) and DisEnt (Lee et al. 2021). We found that using BiasEnsemble for the two methods significantly improves the debiasing performances in four datasets regardless of the bias severities. We also observe that applying BiasEnsemble brings larger performance gain when evaluated with real-world datasets compared to the synthetic dataset. For example, using BiasEnsemble on DisEnt shows 7.48% and 9.56% performance gain on BFFHQ with 1% and 2% ratio of bias-conflicting samples, respectively. Note that we could not evaluate the debiasing methods requiring bias labels (LNL and EnD) on BAR dataset since the dataset does not include explicit bias labels.

Utilizing our approach on DisEnt outperforms ReBias (Bahng et al. 2020) on BAR. BAR dataset is biased towards the background which mainly contains the color

Method		Colored MNIST				BFFHQ				Dogs & Cats		BAR	
		0.5%	1.0%	2.0%	5.0%	0.5%	1.0%	2.0%	5.0%	1.0%	5.0%	1.0%	5.0%
Vanilla (He et al. 2015)	✗✗	34.75	51.14	65.72	82.82	55.64	60.96	69.00	82.88	48.06	69.88	70.55	82.53
HEX (Wang et al. 2019)	✗✓	42.25	47.02	72.82	85.50	56.96	62.32	70.72	83.40	46.76	72.60	70.48	81.20
LNL (Kim et al. 2019)	✓✓	36.29	49.48	63.30	81.30	56.88	62.64	69.80	83.08	50.90	73.96	-	-
EnD (Tartaglione, Barbano, and Grangetto 2021)	✓✓	35.33	48.97	67.01	82.09	55.96	60.88	69.72	82.88	48.56	68.24	-	-
ReBias (Bahng et al. 2020)	✗✓	60.86	**82.78**	**92.00**	**96.45**	55.76	60.68	69.60	82.64	48.70	65.74	73.04	83.90
LfF (Nam et al. 2020)	✗✗	63.55	76.81	84.18	89.65	65.19	69.24	73.08	79.80	71.72	84.32	70.16	82.95
DisEnt (Lee et al. 2021)	✗✗	68.49	79.99	84.09	89.91	62.08	66.00	69.92	80.68	65.74	81.58	70.33	83.13
LfF + BE	✗✗	69.70	81.17	85.20	90.04	67.36	**75.08**	**80.32**	**85.48**	**81.52**	**88.60**	**73.36**	83.87
DisEnt + BE	✗✗	**71.34**	82.11	84.66	90.15	**67.56**	73.48	79.48	84.84	80.74	86.84	73.29	**84.96**

Table 1: Image classification accuracy on unbiased test sets with varying ratios of bias-conflicting samples. The *cross* and *check* represent whether each model 1) uses bias labels during training and 2) requires predefined bias type. Best performing results are marked in bold.

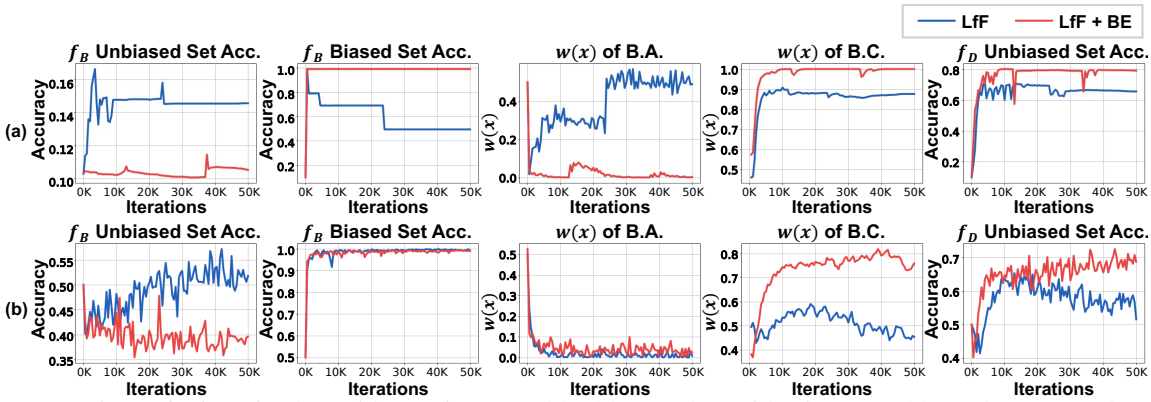


Figure 4: Comparison of LfF trained 1) without BiasEnsemble (blue) and 2) with BiasEnsemble (red) on (a) Colored MNIST and (b) BFFHQ. Each column corresponds to the ones in Fig. 2.

and texture bias. Since ReBias uses BagNet (Brendel and Bethge 2019) which is a color- and texture-oriented model to identify bias (*i.e.,* leveraging a prior knowledge on the bias type), it showed the state-of-the-art performance before using BiasEnsemble on existing reweighting-based approaches. However, even without such prior knowledge on the bias type, applying BiasEnsemble on DisEnt outperforms ReBias regardless of bias severities.

For the Colored MNIST, ReBias utilizes four layers of convolutional neural network while the other debiasing methods use three layers of multi-layer perceptron. We inevitably use the convolutional neural network for ReBias since it leverages a small receptive field of convolutional layers to capture the color bias. When comparing with the baselines using the same architecture, leveraging BiasEnsemble on LfF and DisEnt achieve the state-of-the-art debiasing performance on Colored MNIST.

Analysis

Amplified bias of f_B. Similar to Fig. 2 which describes the motivation of our work, we compare LfF trained 1) without BiasEnsemble and 2) with BiasEnsemble in Fig. 4. While achieving comparable or higher biased test set accuracy, f_B with BiasEnsemble shows lower unbiased test set accuracy compared to f_B without applying BiasEnsemble. This leads to increase the $w(x)_{\text{diff}}$, the difference between $w(x)$ of bias-conflicting samples and that of bias-aligned ones. Then, bias-conflicting samples are further emphasized for training f_D, improving the debiasing performance overall. Such improvement is valid in both synthetic (*i.e.,* Colored MNIST) and the real-world dataset (*i.e.,* BFFHQ). This visualization demonstrates that our proposed method indeed improves debiasing performance of f_D by further amplifying bias of f_B.

How to construct bias-amplified $\mathcal{D}_A$ for debiasing. We found two important factors when constructing $\mathcal{D}_A$: 1) discarding sufficient number of bias-conflicting samples and 2) maintaining a reasonable number of bias-aligned ones. To understand how the data samples composing $\mathcal{D}_A$ affects the debiasing performance, Table 2 compares the debiasing performances of LfF by adjusting the number of bias-aligned and bias-conflicting samples in $\mathcal{D}_A$. In Table 2, # of B.A. and # of B.C. indicate the remaining number of bias-aligned

# of B.A. # of B.C.	100% 100%	100% 60%	100% 20%	60% 20%	20% 20%
Colored MNIST	58.48	73.87	81.58	75.75	63.21
BFFHQ	62.10	73.96	79.36	75.88	69.12
Dogs&Cats	53.10	71.00	79.22	63.86	60.78

Table 2: Unbiased test set accuracies with adjusted number of bias-aligned samples (# of B.A.) and that of bias-conflicting ones (# of B.C.) in the bias-amplified dataset $\mathcal{D}_A$ utilized for training f_B of LfF with CE loss. First two rows represent the ratio of samples in $\mathcal{D}_A$ compared to $\mathcal{D}$.

	LfF	LfF + BE	
M	-	1	5
Colored MNIST	76.81±4.56	79.51±1.56	**81.17**±0.68
BFFHQ	69.24±2.07	71.52±2.68	**75.08**±2.29
Dogs & Cats	71.72±4.56	76.98±6.63	**81.52**±1.13
BAR	70.16±0.77	71.63±1.59	**73.36**±0.97

Table 3: Unbiased test set accuracies on 1) LfF, 2) applying our method on LfF with a single BCD and 3) multiple BCDs. M indicates the number of BCDs when using our method.

samples and that of bias-conflicting samples in $\mathcal{D}_A$, respectively, computed in ratio compared to the original training set. For example, in the case of adjusted ratios of (20%, 20%), 50000 bias-aligned samples and 100 bias-conflicting samples in $\mathcal{D}$ are adjusted to 10000 and 20 in $\mathcal{D}_A$, respectively. We trained f_B by using the adjusted dataset while using the original training set for training f_D.

When fixing the number of bias-aligned samples constant (100%), the debiasing performance improves as the number of bias-conflicting samples decreases (from 100% to 20%). The main reason is that the bias-conflicting samples, preventing f_B from learning the bias attribute, are discarded. This demonstrates that discarding sufficient number of bias-conflicting samples is important for improving the debiasing performance which is straight-forward. On the other hand, we also observe that debiasing performance deteriorates when the number of bias-aligned samples decreases (from 100% to 20%) with the constant number of bias-conflicting samples (20%). This indicates that f_B also requires a sufficient number of bias-aligned samples to learn the bias attributes. We want to emphasize that simply discarding numerous number of training samples for the purpose of eliminating entire bias-conflicting samples may fail to bring large performance gain since it also filters out bias-aligned ones, those important for learning a bias. This analysis demonstrates the importance of considering both factors when constructing $\mathcal{D}_A$. In the Supplementary, along with the standard deviation, we gradually change the adjusted number of samples (*e.g.,* 80%, 40%) to show the detailed tendency of change in debiasing performance with respect to the adjusted number of samples.

Superiority of multiple BCDs over single BCD. We compare the debiasing performance of using BiasEnsemble with a single BCD (M=1) and multiple BCDs (M=5) on LfF in Table 3. While using a single BCD brings performance gain compared to LfF, the standard deviation of the performance is larger when compared to using multiple BCDs. For example, using a single BCD to train LfF shows the standard deviation of 6.63% on Dogs & Cats dataset. This is due to the fact that we rely on a single BCD for constructing $\mathcal{D}_A$. When the single BCD fails to be overfitted to the bias, it fails to filter out the bias-conflicting samples for building $\mathcal{D}_A$. However, such an issue is mitigated when using multiple BCDs since they better capture the bias attribute by considering multiple visual attributes of the bias, compared to using a single BCD. We provide the further analysis on the performance variations of single BCD and multiple BCDs in Supplementary.

	# of B.A.(%)↑		# of B.C.(%)↓	
Dataset	M=1	M=5	M=1	M=5
Colored MNIST	84.10±11.01	**99.96**±0.03	4.64±0.79	**1.42**±0.78
BFFHQ	84.51±4.34	**92.22**±0.26	24.47±4.63	**24.37**±3.07
Dogs&Cats	85.89±2.61	**88.60**±1.02	12.00±6.25	**9.50**±3.75
BAR	97.24±0.27	**98.39**±0.19	60.00±13.24	**51.42**±5.34

Table 4: The remaining number of bias-aligned samples (# of B.A.) and bias-conflicting ones (# of B.C.) in the bias-amplified dataset $\mathcal{D}_A$ after applying our method. The remaining numbers are shown in ratios of samples compared to the original training dataset $\mathcal{D}$.

Such result is mainly due to the number of bias-aligned samples and bias-conflicting ones included in $\mathcal{D}_A$, as demonstrated in Table 2. Table 4 shows the remaining number of bias-aligned samples and bias-conflicting samples in $\mathcal{D}_A$ after applying BiasEnsemble, each computed in ratio compared to the original training dataset $\mathcal{D}$. We observe that utilizing multiple BCDs 1) maintains a significant number of bias-aligned samples and 2) further reduces the number of bias-conflicting samples compared to using a single BCD. Additionally, the standard deviations of the remaining number of samples are considerably larger when using the single BCD, demonstrating that a single BCD fails to fully capture the bias attribute at a stable level.

Conclusion

In this work, we propose a biased sample selection method, BiasEnsemble, in order to train f_B to maximally exploit the bias attribute. Our main finding is that how well f_B is overfitted to the bias influences the debiasing performance of f_D which was overlooked in the previous debiasing studies. While training f_B to overfit to the bias, the bias-conflicting samples interfere with learning bias for f_B, so we filter them out to construct a refined bias-amplified dataset $\mathcal{D}_A$. To do so, we utilize differently randomly initialized biased models to consider diverse visual attributes to better capture the bias attribute and discard the bias-conflicting samples for constructing $\mathcal{D}_A$. Such a simple approach improves the recent state-of-the-art reweighting-based debiasing approaches. We believe that we shed light on an important debiasing component f_B which has been relatively overlooked compared to f_D, and provide insightful findings for future researchers in debiasing.

Acknowledgments

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-00075 and No.2022-0-009840101003, Artificial Intelligence Graduate School Program (KAIST)), Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [22ZS1200, Fundamental Technology Research for Human-Centric Autonomous Intelligent Systems], and Kakao Enterprise.

References

Bahng, H.; Chun, S.; Yun, S.; Choo, J.; and Oh, S. J. 2020. Learning De-biased Representations with Biased Representations. In *International Conference on Machine Learning (ICML)*.

Bian, Y.; and Chen, H. 2021. When Does Diversity Help Generalization in Classification Ensembles? *IEEE Transactions on Cybernetics*, 1–17.

Brendel, W.; and Bethge, M. 2019. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *International Conference on Learning Representations*.

Darlow, L.; Jastrzebski, S.; and Storkey, A. 2020. Latent Adversarial Debiasing: Mitigating Collider Bias in Deep Neural Networks. *arXiv preprint arXiv:2011.11486*.

Fort, S.; Hu, H.; and Lakshminarayanan, B. 2019. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*.

Huang, Z.; Wang, H.; Xing, E. P.; and Huang, D. 2020. Self-Challenging Improves Cross-Domain Generalization. In *ECCV*.

Kim, B.; Kim, H.; Kim, K.; Kim, S.; and Kim, J. 2019. Learning Not to Learn: Training Deep Neural Networks With Biased Data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, E.; Lee, J.; and Choo, J. 2021. BiaSwap: Removing Dataset Bias With Bias-Tailored Swapping Augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14992–15001.

Lee, J.; Kim, E.; Lee, J.; Lee, J.; and Choo, J. 2021. Learning Debiased Representation via Disentangled Feature Augmentation. In *Advances in Neural Information Processing Systems*.

Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from Failure: Training Debiased Classifier from Biased Classifier. In *Advances in Neural Information Processing Systems*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*.

Tartaglione, E.; Barbano, C. A.; and Grangetto, M. 2021. EnD: Entangling and Disentangling Deep Representations for Bias Correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13508–13517.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. 1521–1528. IEEE Computer Society.

Wang, H.; He, Z.; Lipton, Z. L.; and Xing, E. P. 2019. Learning Robust Representations by Projecting Superficial Statistics Out. In *International Conference on Learning Representations*.

Zaidi, S.; Zela, A.; Elsken, T.; Holmes, C. C.; Hutter, F.; and Teh, Y. 2021. Neural Ensemble Search for Uncertainty Estimation and Dataset Shift. In *Advances in Neural Information Processing Systems*, volume 34.

Zhang, Z.; and Sabuncu, M. R. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*.