DebiasBench: Benchmark for Fair Comparison of Debiasing in Image Classification

Jungsoo Lee$^{1,2}$, Juyoung Lee$^2$, Sanghun Jung$^1$, and Jaegul Choo$^1$

$^1$ KAIST AI, South Korea  
\{bebeto, shjung13, jchoo\}@kaist.ac.kr  
$^2$ Kakao Enterprise, South Korea  
\{bebeto.lee, michael.jy\}@kakaoenterprise.com

Abstract. Image classifiers often rely overly on peripheral attributes that have a strong correlation with the target class (i.e., dataset bias) when making predictions. Recently, a myriad of studies focus on mitigating such dataset bias, the task of which is referred to as debiasing. However, these debiasing methods often have inconsistent experimental settings (e.g., datasets and neural network architectures). Additionally, most of the previous studies in debiasing do not specify how they select their model parameters which involve early stopping and hyperparameter tuning. The goal of this paper is to standardize the inconsistent experimental settings and propose a consistent model parameter selection criterion for debiasing. Based on such unified experimental settings and model parameter selection criterion, we build a benchmark named DebiasBench which includes five datasets and seven debiasing methods. We carefully conduct extensive experiments in various aspects and show that different state-of-the-art methods work best in different datasets, respectively. Even, the vanilla method, the method with no debiasing module, also shows competitive results in datasets with low bias severity. We publicly release the implementation of existing debiasing methods in DebiasBench to encourage future researchers in debiasing to conduct fair comparisons and further push the state-of-the-art performances.

Keywords: Debiasing, Image classification, Benchmark

1 Introduction

Image classification models tend to heavily rely on the correlation between peripheral attributes and target class, which is termed as dataset bias \cite{29,23,18}. Such dataset bias is found in datasets of which the majority of images include visual attributes that frequently co-occur with the target class but do not inherently define it, referred to as bias attributes. To give an example with Fig. 1, a training set of bird images may include the majority of birds in the blue sky (the bias attribute) while birds can also be found in other places such as roads or trees. In such a case, the image classifier may use the blue sky as the visual cue for predicting the bird image while not learning the intrinsic attributes, the
Fig. 1: Description of the debiasing task. When trained with a biased dataset, the image classifier relies on the bias attribute (e.g., blue sky) as the visual cue to predict the target class (e.g., birds). This leads the model to misclassify test images without the bias attribute such as birds on roads or trees. The blue and green bordered images indicate the images with bias attributes and the ones without bias attributes, respectively.

visual attributes which innately define a certain class (e.g., beaks and wings of birds). This becomes problematic during the inference stage since the classification model may misclassify bird images without the blue sky. Instead, it may also wrongly predict images with other classes taken in the blue sky (e.g., airplanes in the blue sky) as birds since it heavily relies on the strong correlation between the blue sky (bias attribute) and birds (target class). The task of debiasing aims to mitigate such dataset bias.

Until recently, numerous debiasing methods have been proposed [19,32,10,27,39,21,28,23,18]. In the early stages, debiasing methods either define certain bias types to address in advance [3] or even explicitly use bias labels to train debiased models [12,28]. However, Nam et al. [23] propose a debiasing algorithm which does not predefine a bias type based on the intuition that the bias is easy to learn for image classification models. Lee et al. [18] further improve the work by increasing the diversity of data samples which again does not predefine bias types. While defining certain bias types in advance or explicitly using bias labels may improve the classification accuracy, we may not know the bias types in datasets used in real-world applications. Thus, building debiasing methods which do not require predefined bias types would be a realistic setting.

While such existing debiasing methods show remarkable performances, they have inconsistent experimental settings (e.g., datasets and neural network architectures) as shown in Table 1. Additionally, most of the debiasing methods do not specify how they choose their model parameters which involve the early-stopping iteration and hyper-parameter tuning (e.g., lambda values for balancing loss functions). Throughout the paper, the model parameters indicate the checkpoints saved every certain iteration during the training phase. A standardized model parameter selection is important since it prevents researchers from tuning the model on the test set. Gulrajani et al. [7] and Musgrave et al. [22] point
Table 1: Summary of the inconsistent experimental settings of existing debiasing studies. We include the usage of bias labels, bias types, different datasets, neural network architectures, and specification of the model parameter selection criteria. Cross mark and check mark indicate whether the algorithm requires such knowledge during the training phase (e.g., EnD [28] explicitly uses color labels for training Colored MNIST). Pretr. and init. indicates that the model is pretrained with ImageNet [5] and it is randomly initialized, respectively. Note that most of the previous studies do not specify how they select their model parameters.

| Algorithm | Bias labels | Bias types | Dataset | Intrinsic attr. | Bias attr. | Architecture | Model parameter selection |
|-----------|-------------|------------|---------|-----------------|------------|--------------|--------------------------|
| LNL [12] | ✓ | ✓ | Colored MNIST (Fore) | Digits | Color | Conv layers | ResNet18 (pretr.) | Not specified |
| EnD [28] | ✓ | ✓ | Colored MNIST (Back) | Digits | Color | Conv layers | ResNet18 (pretr.) | Specified |
| ReBias [1] | ✗ | ✓ | Colored MNIST (Back) | ImageNet classes | Color | Conv layers | ResNet18 (init.) | Not specified |
| LfF [23] | ✗ | ✗ | Colored MNIST (Fore) | Digits | Color | MLP | ResNet20 (init.) | Not specified |
| DisEnt [18] | ✗ | ✗ | Colored MNIST (Fore) | Digits | Color | MLP | ResNet18 (init.) | Not specified |

Our contributions are summarized as follows:

- We propose to select model parameters on the best accuracy of the validation set after a pre-determined minimum number of iterations.
- We unify the backbone model as the same neural network architectures for all debiasing methods, which previously differed among various existing approaches.
- We release the benchmark repository named DebiasBench which includes the downloaders for five benchmark datasets and implementations of existing seven debiasing methods to encourage fair comparisons among them for future researchers in debiasing.
- By conducting extensive experiments via DebiasBench, we reveal that 1) existing state-of-the-art models work best in different datasets, respectively, and 2) the vanilla module achieves the best classification accuracy in datasets with low bias severity.
2 Related Work

2.1 Debiasing

Early studies on debiasing define bias types in advance [19,27,3,30,2] or even use the bias labels for training [28,12]. Tartaglione et al. disentangle the intrinsic and bias features based on a regularization strategy which utilizes explicit bias labels for training [28]. Kim et al. minimize the mutual information between the debiased and biased representations while also explicitly using bias labels for the latter parameters [12]. Acquiring bias labels, however, may be challenging and labor intensive since it necessitates manual labeling by human annotators with sufficient understanding of the underlying bias of a given dataset [23]. While not using the bias labels during training, Bahng et al. propose ReBias which defines a bias type in advance and leverages neural network architectures tailored for the predefined bias type (e.g., using models of small receptive fields for addressing color bias) [3]. Still, predefining a bias type 1) limits the debiasing capacity in other bias types especially when they are unknown and 2) requires human labor to manually identify the bias types [18].

To address such an issue, Nam et al. propose LfF which does not 1) use bias labels during training or 2) define bias types in advance [23]. They emphasize the data samples helpful for mitigating the dataset bias by identifying them based on the intuition that bias is easy to learn. Another recent debiasing work which does not require predefined bias types or bias labels was proposed by Lee et al. [18]. They claim that the diversity of data samples is important for debiasing and propose a feature-level augmentation method for debiasing which further improves the debiasing performance. While these debiasing methods show remarkable achievements, we find that their experimental settings are inconsistent and most of them do not specify their model parameter selection criterion.

2.2 Standardizing inconsistent experimental settings

Unspecified model selection criterion was addressed in other computer vision tasks such as domain generalization [21], metric learning [24,33], and cross-domain few-shot learning [15]. A plethora of algorithms proposed in these fields have inconsistent experimental settings including using different datasets and neural network architectures. Due to this fact, researchers in each field build benchmark repositories and conduct fair comparisons via extensive experiments. For example, Gulrajani et al. build DomainBed which points out that domain generalization methods which do not specify model selection criterion may have used the test set to select parameters. [7]. Also, Musgrave et al. similarly propose an evaluation protocol in metric learning to fix the flaws and problems of unfair comparisons found in the existing metric learning studies. [22]. Both benchmark repositories include the standardized model parameter selection criterion and claim that work without specific model parameter selection criterion may have used the test set for selecting the model parameters [7,22]. Guo et al. reveal that the recent state-of-the-art cross-domain few-shot learning methods
show poor performance when evaluated with datasets containing large domain discrepancy by building a benchmark named Broader Study of Cross-Domain Few-Shot Learning [8]. Unlike Gulrajani et al. and Musgrave et al., Guo et al. claim that choosing the model parameters using the validation set in cross-domain few-shot learning may be controversial and leave such an issue as the future work.

As shown in these previous studies, choosing the best model parameters is a challenging yet important issue researchers should be aware of. Most of the existing debiasing studies do not specify how they choose their model parameters, so we propose a model parameter selection criterion specialized for debiasing.

3 Preliminaries

3.1 Debiasing

In a biased dataset, input $X_{train}$ with the target class $Y_{train}$ include the intrinsic attributes $I_{train}$ and the bias attributes $B_{train}$. To give the aforementioned bird image in the blue sky as an example, the wings and beaks of birds correspond to $I_{train}$ while the blue sky corresponds to $B_{train}$. The difference between $I$ and $B$ is that $I$ always appears in the target class while $B$ does not necessarily appear since it does not innately define the target class.

In supervised learning, classifier $f$ can output unintended results under different circumstances [6]. To be more specific, $f$ which is trained with the biased dataset learns the unwanted correlation between $B_{train}$ and $Y_{train}$ rather than learning $I_{train}$. $f$ may even rely on such visual cue to predict $Y_{train}$. However, $f$ trained in such a manner shows significant performance drop when $B_{test} \neq B_{train}$, meaning that $X_{test}$ do not include $B_{train}$. $f$ is prone to misclassifying $X_{test}$ especially when $B_{train}$ is included in other target classes (e.g., wrongly predicting airplanes in the blue sky as birds). Therefore, the debiasing task requires $f$ to learn $I$ of a target class regardless of $B_{train}$ since $I_{train} = I_{test}$ for a given target class.

We define key terminologies used in the debiasing task. Following the work of Nam et al. [23], we define bias-aligned samples as $X$ including $B_{train}$ (e.g., birds in the blue sky) and bias-conflicting samples as $X$ without $B_{train}$ (e.g., birds on roads or trees).

3.2 Difference between domain generalization

The debiasing task is different from domain generalization [25,13,17,18] in the following aspect. Domain generalization assumes that the distribution of $X_{train}$ is known while $X_{test}$ is unknown [7]. For example, $X_{train}$ are photo images of birds while $X_{test}$ are sketch images of birds which the model does not encounter during the training phase. On the other hand, the debiasing task assumes that the distribution of $X_{train}$ and $X_{test}$ are known and similar, however the distributions of $B_{train}$ and $B_{test}$ are different. For example, $X_{train}$ and $X_{test}$ are both photo images of birds but $X_{train}$ includes the blue sky as $B_{train}$ while $X_{test}$ includes the ground or trees as $B_{test}$.
Fig. 2: Learning curves of vanilla and LfF for Colored MNIST and Corrupted CIFAR10. The left and right plot the vanilla and DisEnt for each dataset, respectively. The dotted lines of ‘Best valid’ and ‘Best valid 25K’ indicate the iteration index with the best validation set accuracy 1) throughout the training and 2) after 25K iterations, respectively.

4 DebiasBench

This section describes how we select the model parameters specialized for the debiasing task. Then, we illustrate the datasets, debiasing methods, and the implementation details of DebiasBench.

4.1 Model Parameter Selection

When trained with a biased dataset, models are overfitted to the bias-aligned samples which are easy to learn [23], leading them to achieve the best validation accuracy in the early iterations. Using the model of the best validation accuracy to evaluate on the test set is a conventional method in machine learning. However, it may be unfair to evaluate debiasing methods achieving the best validation in the early iterations in such a manner since their proposed methods are not fully exploited at the step.

Such finding is demonstrated by our experiments. Fig. 2 shows the accuracy of the validation and test sets of vanilla [9] and LfF [23], with Colored MNIST and Corrupted CIFAR10, respectively. Since it achieves the best validation accuracy in the early stage of training, LfF shows significantly low performance in test sets when using the parameter of the best validation accuracy. For example, in the case of LfF trained with Colored MNIST, the model achieves the best validation accuracy of 92.78% at 1.5K iteration with the test set accuracy of 32.51% while it can achieve the best test set accuracy of 74.70%. Such a finding is also observed in DisEnt [18], another recently proposed seminal debiasing method (See the Supplementary). Based on this observation, we propose using the parameters with the best validation accuracy after certain iterations. We assume that all debiasing methods fully exploit their proposed methods after a predefined number of iterations, so it is fair to use the parameters of the best validation accuracy for evaluating on the test set. We also show the plot of the vanilla model to demonstrate that such an evaluation criterion does not penalize other models. We set the default value of the ‘certain iterations’ as the
Fig. 3: Datasets included in DebiasBench. DebiasBench includes 1) Colored MNIST and Corrupted CIFAR10 for the synthetic datasets, and 2) BFFHQ, Dogs & Cats and BAR for the real-world datasets. Each grid of Colored MNIST, Corrupted CIFAR10, and BAR indicates each class while classes are bisected in three columns for BFFHQ and Dogs & Cats.

Fig. 3 shows the five debiasing datasets we include in DebiasBench. The five datasets are Colored MNIST [18], Corrupted CIFAR10 [18], biased FFHQ (BFFHQ) [14], Dogs & Cats [12], and biased action recognition (BAR) [23]. Each dataset has different ratios of bias-conflicting samples in order to evaluate algorithms under varying levels of bias severity. For the test sets, we use unbiased test sets which do not have the correlation between $I_{test}$ and $B_{test}$ of a given target class $Y_{test}$. DebiasBench is built in Pytorch and provides data loaders and downloaders for easy accessibility to the five datasets. Further detailed descriptions of datasets are included in the Supplementary.

**Colored MNIST** has digits and colors for the intrinsic and bias attributes, respectively. While previous debiasing studies either inject the color on the foreground or background on MNIST [17] to construct the Colored MNIST, we use the foreground Colored MNIST in DebiasBench for a standardized experimental setting. For building the Colored MNIST, we follow the dataset used in Lee et al. [18]. The number of bias-aligned samples and bias-conflicting samples for different ratios of bias-conflicting samples are as follows: (54751, 249)-0.5%, (54509, 491)-1%, (54014, 986)-2%, and (52551, 2449)-5%.

**Corrupted CIFAR10** has classes of CIFAR10 [16] and corruption types for the intrinsic and bias attributes, respectively. Nam et al. [23] use two different types of Corrupted CIFAR10 which are termed as ‘Type 0’ and ‘Type 1’. We use Type 0 for a standardized experimental setting in DebiasBench. To be more specific, the corruption types are Snow, Frost, Fog, Brightness, Contrast, Spatter, Elastic, JPEG, Pixelate, Saturate. Each corruption type is highly correlated with one of the classes in CIFAR10 which are Plane, Car, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck. The number of bias-aligned samples and bias-conflicting samples

halves number of iterations (i.e., 25K) out of total iterations (i.e., 50K) used in training. In the Supplementary, we show that the order of the ranking between the debiasing methods is not highly sensitive to the steps of certain iterations (e.g., iterations of 35K and 45K).
for different ratios of bias-conflicting samples are as follows: (19104, 96)-0.5%, (19008, 192)-1%, (18816, 384)-2%, and (18240, 960)-5%.

BFFHQ has age (i.e., young and old) and gender for the intrinsic and bias attributes, respectively. Old male and young female correspond to the bias-aligned samples while young male and old female are the bias-conflicting samples. While Lee et al. only use 0.5% ratio for evaluating under BFFHQ, we also include 1%, 2%, and 5% in DebiasBench. BFFHQ is modified from FFHQ [11]. We follow the dataset used in Kim et al. [14] for biased FFHQ. The number of bias-aligned samples and bias-conflicting samples for different ratios of bias-conflicting samples are as follows: (44832, 228)-0.5%, (44527, 442)-1%, (44145, 887)-2%, and (42820, 2242)-5%.

Dogs & Cats has animal (i.e., dog and cat) and color for the intrinsic and bias attributes, respectively. The bias-aligned samples are dark cats and bright dogs while the bias-conflicting samples are bright cats and dark dogs. While Kim et al. used this dataset [1] previously, we find that the test sets they used do not have publicly available ground truths. Due to this fact, we split the original train set which includes ground truths into train, valid, and test sets. The number of bias-aligned samples and bias-conflicting samples for different ratios of bias-conflicting samples are as follows: (8037, 80)-1% and (8037, 452)-5%.

BAR has the action and background for the intrinsic and bias attributes, respectively. Nam et al. [23] did not evaluate debiasing methods according to different ratios of bias-conflicting samples in BAR. We intentionally build two sets of BAR dataset with different ratios of bias-conflicting samples for the unified experimental setting as done for other datasets. The number of bias-aligned samples and bias-conflicting samples for different ratios of bias-conflicting samples are as follows: (1761, 14)-1% and (1761, 85)-5%.

### 4.3 Debiasing Methods

DebiasBench includes seven debiasing methods: vanilla (four layers of convolutional layers for Colored MNIST and ResNet [9] for the rest of datasets), Hex [30], LNL [12], EnD [28], ReBias [3], LfF [23], and DisEnt [18]. Note that LfF and DisEnt do not define bias types in advance or use bias labels during training.

### 4.4 Implementation

Following the work of Bahng et al. [3], we use a classifier with four convolutional layers for Colored MNIST for all debiasing methods. Since the recent state-of-the-art models [23,18] use the MLP for Colored MNIST, we additionally include such experiments in Table 2. For the other datasets except for BAR, we use ResNet18 with random initialization. We use pretrained ResNet18 for BAR since BAR has an excessively small number of images compared to other datasets. We apply data augmentation for all datasets except for Colored MNIST. We set the batch size of 256 for Colored MNIST and Corrupted CIFAR10 and 64 for the rest of the datasets. All experiments are trained for 50K iterations and are
1) test sets after certain iterations, 2) test sets with the best accuracy, and 3) We report the image classification accuracy of debiasing methods evaluated on Colored MNIST and Corrupted CIFAR10. Using MLP for Colored MNIST or Type 1 for Corrupted CIFAR10 do not significantly change the order of rankings between the debiasing methods.

Table 2: Additional experiments on Colored MNIST and Corrupted CIFAR10.

Using MLP for Colored MNIST or Type 1 for Corrupted CIFAR10 do not significantly change the order of rankings between the debiasing methods.

Table 3: Image classification accuracy evaluated on test sets with model selection after certain iterations. The cross and check marks denote whether each algorithm requires 1) bias labels during training and 2) prior knowledge on bias types in that order. The best results are marked in bold.

reported over five independent trials with the mean and the standard deviation. We include the remaining implementation details in the Supplementary.

5 Experiments

We report the image classification accuracy of debiasing methods evaluated on 1) test sets after certain iterations, 2) test sets with the best accuracy, and 3)
validation sets. For the evaluation after certain iterations, we report the test set accuracy with the parameters achieving the best validation accuracy after 25K.

### 5.1 Test set: Unbiased set

Table 4 shows that DisEnt and LfF achieve comparable performance in both Colored MNIST and Corrupted CIFAR10. While DisEnt reports that ReBias outperforms their proposed method in Colored MNIST in their original paper, they use a classifier of four convolutional layers for ReBias while leveraging MLP for DisEnt. However, when evaluated with the same backbone, we find that DisEnt achieves better performance than ReBias. We also find that the vanilla model shows the best accuracy in Colored MNIST 5% and BFFHQ 5%. The recent debiasing methods show superior performance in synthetic datasets or datasets with an excessively small number of bias-conflicting samples. However, if there exists a non-trivial amount of bias-conflicting samples in a train set, the vanilla model may show better performance compared to the recent state-of-the-art methods.

Table 4 reports the best test set accuracy of the debiasing methods. We observe that such results achieve improved debiasing performance when compared to Table 3. We emphasize that future researchers should be restrained from using the model parameter achieving the best test set accuracy since it may lead to the overfitting of models on the test sets. In most cases except for the BAR dataset, DisEnt or LfF consistently achieve the best performance regardless of the model selection criterion. This indicates that how to...
Table 5: Image classification accuracies with the validation sets. The first and the second accuracy indicates the best validation set accuracy 1) throughout the training phase and 2) after 25K iterations. We only report the average accuracy of five trials due to the page limit. Results with the standard deviations are included in the Supplementary.

utilize easy-to-learn property [23] of the bias attribute is effective to improve the debiasing performance. Additionally, ReBias [3] outperforms other debiasing approaches on BAR dataset. Since the background (i.e., the bias attribute) includes a certain color or texture, ReBias [3] which predetermines the bias type as color or texture, shows superior performance compared to other methods. In other words, leveraging the prior knowledge on bias types brings debiasing performance gains.

5.2 Validation set: Biased set

The validation set is mostly composed of bias-aligned images similar to the train set. Although most of the previous debiasing studies do not report the image classification accuracy of the validation set, we claim that the future debiasing studies should also report the validation set accuracies. The first and the second accuracies in Table 5 indicate the best validation accuracy measured 1) throughout the training phase and 2) after 25K iterations, respectively. While LfF and DisEnt achieve the state-of-the-art performances in most of the test sets, they show a large performance drop on the validation sets. For example, LfF achieves the best test accuracy of 45.64% in Dogs & Cats 1% while the vanilla model achieves 32.64%. However, LfF shows 71.30% of the validation set accuracy while the vanilla model shows 97.14%. LfF requires 25.84% point decrease of validation set accuracy to obtain 13% point gain in the unbiased
test set. LfF and DisEnt even show large performance degradation in the best validation accuracy after 25K iterations, unlike other debiasing methods.

While the existing debiasing studies focus on obtaining high accuracy in the unbiased test sets, the performance on the validation sets has been relatively overlooked. We believe that debiasing is not the task of improving the test set accuracy with the sacrifice of the validation set accuracy. Debiasing methods should correctly classify not only bias-conflicting samples but also bias-aligned samples. In other words, researchers need to propose debiasing methods which achieve high test set accuracy while maintaining the validation set accuracy at a reasonable level. Therefore, the validation set accuracies should also be reported in future debiasing work.

6 Discussion

This section discusses the main findings from our extensive experiments via DebiasBench and the future directions of debiasing studies.

6.1 Towards In-the-Wild Datasets

DebiasBench includes two types of datasets: 1) synthetic datasets (i.e., Colored MNIST and Corrupted CIFAR-10) and 2) real-world datasets (i.e., BFFHQ, Dogs & Cats, and BAR). For the analysis of the debiasing method, the experimental settings in our DebiasBench assumes the real-world datasets include bias attributes. In other words, the intrinsic and bias attributes in our real-world datasets are already identified before evaluating the debiasing methods. In order to extend our evaluation framework to the in-the-wild dataset, identifying the bias attributes is necessitated. To be more specific, there are two steps: 1) identifying whether a dataset has a bias and 2) resolving the identified dataset bias (if there exists). To the current state, only few studies explore the first step [31] while the existing previous debiasing studies mainly focus on the second step. DebiasBench was proposed to evaluate the debiasing methods focusing on the second step. In this regard, the first step is beyond the scope of our work but we believe that it is a valuable future research direction. With a newly proposed bias-identification method, we believe DebiasBench supports evaluating such in-the-wild datasets with minor modifications.

6.2 Debiasing Methods based on Easy-to-Learn Intuition

Nam et al. [23] emphasize the bias-conflicting samples during training which are identified based on the intuition that the bias is easy to learn. Lee et al. [18] diversify the bias-conflicting samples via feature-level augmentation still relying on the easy-to-learn intuition. Such intuition does not require predefined bias types or bias labels during training. Despite their remarkable achievements, they show limitations when the ratio of bias-conflicting samples is sufficient enough (5% in Colored MNIST and BFFHQ). Additionally, they could not show superiority in
DebiasBench 13

Table 6: Analysis on various model architectures. Even trained with strong backbones (e.g., ResNet50 and ResNet101), the debiasing capacity remains at a similar level or even degrades in some cases.

| Dataset       | Model       | Vanilla | HEX | LNL | EnD | ReBias | LfF | DisEnt |
|---------------|-------------|---------|-----|-----|-----|--------|-----|--------|
|               | ResNet18    | 24.27±0.31 | 22.99±0.54 | 24.29±0.41 | 23.78±0.63 | 24.43±0.96 | **30.59±0.80** | **30.59±3.05** |
|               | ResNet50    | 24.95±0.56 | 20.79±2.86 | 24.41±0.58 | 24.66±0.46 | 24.00±0.44 | **31.61±0.86** | 28.53±4.60 |
|               | ResNet101   | 24.44±1.18 | 23.57±2.28 | 23.95±0.94 | 24.91±0.31 | 24.48±0.53 | **31.66±2.71** | 26.67±1.81 |
|               | Corrupted CIFAR-10c | 56.88±0.58 | 58.16±1.49 | 57.84±2.69 | 54.31±2.02 | 56.92±0.84 | **60.96±1.85** | 57.92±3.47 |
|               | ResNet18    | 56.32±1.79 | 56.12±2.02 | **58.00±0.90** | 54.36±2.72 | 54.96±0.95 | 57.24±2.62 | 57.44±2.27 |
|               | ResNet101   | 53.88±1.68 | 55.52±2.79 | 57.8±1.86 | 54.12±2.08 | **63.88±1.36** | 58.8±2.71 | 57.8±3.78 |
|               | BFFHQ       | 32.64±0.74 | 31.70±2.49 | 36.90±4.51 | 31.38±1.44 | 29.74±2.15 | **45.64±5.13** | 37.52±5.22 |
|               | ResNet18    | 26.73±0.78 | 30.40±1.29 | 30.08±1.76 | 23.76±1.43 | 24.56±1.83 | **52.34±2.30** | 47.36±2.67 |
|               | ResNet101   | 23.32±2.28 | 25.02±1.79 | 28.18±4.36 | 24.04±2.34 | 48.9±3.51 | **51.76±2.25** | 46.82±3.25 |
|               | Dogs & Cats | ResNet18   | 68.09±1.33 | 68.76±1.95 | 67.75±1.40 | 67.80±1.80 | **69.95±1.21** | 68.95±1.20 |
|               | ResNet50    | 73.18±1.89 | 72.16±3.28 | **73.62±1.67** | 72.68±0.88 | 71.75±1.97 | 71.95±3.32 | 72.30±1.69 |
|               | ResNet101   | 70.86±1.37 | 66.60±2.12 | 72.27±2.52 | **73.46±1.33** | 73.38±1.51 | 72.94±1.61 | 71.92±3.15 |

6.3 Using Strong Backbones for Debiasing

The main purpose of the debiasing technique is to prevent a model from being overfitted to a certain bias attribute. Ideally, even when given a backbone with a high model capacity, e.g., an over-parameterized model, the debiasing technique should prevent such a model from being overfitted to a certain bias.

Table 6 shows that training debiasing methods with strong backbones (e.g., ResNet50 and ResNet101) do not improve the debiasing capability generally. Unlike other image classification tasks (e.g., domain generalization), the current debiasing methods with increased model capacity fail to prevent the model from being over-fitted to the bias. For example, in domain generalization, training with ResNet50 increases the average accuracy of five benchmark datasets by 3.8% point compared to training with ResNet18. One of the main reasons for such a result is due to the strong correlation between bias attributes and the target class. This strong correlation prevents models from learning intrinsic attributes regardless of the backbone capacity. This result clearly demonstrates that debiasing is a non-trivial task which cannot be solved by simply using more parameters or layers, necessitating further research.
6.4 Inconsistent experimental settings of Colored MNIST and Corrupted CIFAR10

We also conduct experiments with Colored MNIST using MLP in Table 2. Using MLP for Colored MNIST may be unfair for ReBias [4]. To be more specific, ReBias leverages a debiased model and a biased model while the latter network requires convolutional layers with small receptive fields. When MLP and convolutional layers are used for the debiased model and biased model, respectively, there may exist a collapse of the joint training due to the significant difference of the model capacities. When trained with MLP, LfF and DisEnt again show comparable performances in Colored MNIST.

Table 7 shows the image classification accuracy of the vanilla model evaluated with both foreground and background of Colored MNIST 1%. We observe that using the foreground Colored MNIST achieves slightly higher accuracy compared to evaluating with the background Colored MNIST. Similarly, Table 2 shows that debiasing methods generally achieve higher accuracy when evaluated with Corrupted CIFAR10 Type 1 compared to ones evaluated with Corrupted CIFAR10 Type 0. While additionally including background Colored MNIST and Corrupted CIFAR10 Type 1 in DebiasBench would be another way to construct the benchmark, we find that the order of ranking between the debiasing methods does not change significantly. Therefore, we include the foreground Colored MNIST and Corrupted CIFAR10 Type 0 in DebiasBench.

7 Conclusion

In this work, we initially reveal that the existing debiasing studies have inconsistent experimental settings such as datasets and neural network architectures. Motivated by the finding, we achieve the following contributions. First, we propose a model parameter selection criterion which selects the parameters after a pre-determined minimum number of iterations which is specialized for debiasing. Second, we unify the backbone model with the same neural network architecture which differed among the previous studies. Third, we build DebiasBench which enables fair comparisons between the existing debiasing methods based on the standardized experimental settings. Last, through extensive experiments, we find that the state-of-the-art debiasing methods achieve the best accuracy in different datasets and bias-conflicting ratios, respectively. Additionally, the vanilla
method shows the best performance in datasets with less bias severity. DebiasBench is a living benchmark and we welcome adding new debiasing methods to improve the state-of-the-art performance. We hope the release of DebiasBench and our findings encourage future researchers in the field of debiasing to conduct research in a standardized setting.
References

1. Dogs vs. cats redux: Kernels edition, https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
2. Agrawal, A., Batra, D., Parikh, D.: Analyzing the behavior of visual question answering models. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1955–1960. Association for Computational Linguistics, Austin, Texas (Nov 2016). https://doi.org/10.18653/v1/D16-1203
3. Bahng, H., Chun, S., Yun, S., Choo, J., Oh, S.J.: Learning de-biased representations with biased representations. In: International Conference on Machine Learning (ICML) (2020)
4. Cha, J., Chun, S., Lee, K., Cho, H.C., Park, S., Lee, Y., Park, S.: Swad: Domain generalization by seeking flat minima. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A largescale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence 2, 665–673 (Nov 2020)
7. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=lQdXeXDoWtI
8. Guo, Y., Codella, N.C., Karlinsky, L., Codella, J.V., Smith, J.R., Saenko, K., Rosing, T., Feris, R.: A broader study of cross-domain few-shot learning. ECCV (2020)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
10. Hendricks, L.A., Burns, K., Saenko, K., Darrell, T., Rohrbach, A.: Women also snowboard: Overcoming bias in captioning models. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III (2018)
11. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
12. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
13. Kim, D., Yoo, Y., Park, S., Kim, J., Lee, J.: Sefreg: Self-supervised contrastive regularization for domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9619–9628 (2021)
14. Kim, E., Lee, J., Choo, J.: Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14992–15001 (October 2021)
15. Koniusz, P., Tas, Y., Zhang, H., Harandi, M.T., Porikli, F., Zhang, R.: Museum exhibit identification challenge for the supervised domain adaptation and beyond. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI. Lecture Notes in Computer Science, vol. 11220, pp.
16. Krizhevsky, A.: Learning multiple layers of features from tiny images pp. 32–33 (2009)
17. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010)
18. Lee, J., Kim, E., Lee, J., Lee, J., Choo, J.: Learning debiased representation via disentangled feature augmentation. In: Proc. the Advances in Neural Information Processing Systems (NeurIPS) (2021)
19. Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9572–9581 (2019)
20. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (2015)
21. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
22. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check (2020)
23. Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J.: Learning from failure: Training debiased classifier from biased classifier. In: Advances in Neural Information Processing Systems (2020)
24. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., Jin, R.: Softtriple loss: Deep metric learning without triplet sampling. In: IEEE International Conference on Computer Vision, ICCV 2019 (2019)
25. Qiao, F., Zhao, L., Peng, X.: Learning to learn single domain generalization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12556–12565 (2020)
26. Rothe, R., Timofte, R., Gool, L.V.: Deep expectation of real and apparent age from a single image without facial landmarks. International Journal of Computer Vision (2018)
27. Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. arXiv preprint arXiv:1911.08731 (2019)
28. Tartaglione, E., Barbano, C.A., Grangetto, M.: End: Entangling and disentangling deep representations for bias correction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13508–13517 (June 2021)
29. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR. pp. 1521–1528. IEEE Computer Society (2011), http://dblp.uni-trier.de/db/conf/cvpr/cvpr2011.htmlTorralbaE11
30. Wang, H., He, Z., Lipton, Z.L., Xing, E.P.: Learning robust representations by projecting superficial statistics out. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rJEjjo89K7
31. Wang, T., Zhao, J., Yatskar, M., Chang, K.W., Ordonez, V.: Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
32. Wang, Z., Qinami, K., Karakozis, I.C., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
33. Yuan, Y., Yang, K., Zhang, C.: Hard-aware deeply cascaded embedding. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 814–823. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.94

34. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: ICLR (2021)