Learning Debiased Classifier with Biased Committee

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural networks are prone to be biased towards spurious correlations between classes and latent attributes exhibited in a major portion of training data, which ruins their generalization capability. This paper proposes a new method for training debiased classifiers with no spurious attribute label. The key idea of the method is to employ a committee of classifiers as an auxiliary module that identifies bias-conflicting data, *i.e.*, data without spurious correlations, and assigns large weights to them when training the main classifier. The committee is learned as a bootstrapped ensemble so that a majority of its classifiers are biased as well as being diverse, and intentionally fail to predict classes of bias-conflicting data accordingly. The consensus within the committee on prediction difficulty thus provides a reliable cue for identifying and weighting bias-conflicting data. Moreover, the committee is also trained with knowledge transferred from the main classifier so that it gradually becomes debiased along with the target and emphasizes more difficult data as training progresses. On five real-world datasets, our method outperforms existing methods using no spurious attribute label like ours and even surpasses those relying on bias label occasionally.

1 Introduction

Most supervised learning algorithms for classification rely on the empirical risk minimization (ERM) principle [37]. However, ERM has been known to cause a learned classifier to be biased toward spurious correlations between predefined classes and latent attributes that appear in a majority of training data [11]. In the case of hair color classification, for example, when most people with `blond-hair` (*i.e.*, target class) are `female` (*i.e.*, latent attribute) in a dataset, a classifier learned by ERM exploits `female` as a shortcut for the classification due to its spurious correlation with `blond-hair`, and often mis-classifies non-blonde-haired women as `blond-hair` in consequence. We call data with such spurious correlations and holding a majority of training data *bias-guiding samples*, and the other *bias-conflicting samples*, respectively. The issue of model bias has often been addressed by exploiting explicit spurious attribute labels [21, 26, 34, 1, 36, 35, 42] or knowledge about bias types given a priori [2]. However, these methods are impractical because such supervision and prior knowledge are costly, and the methods demand extensive post hoc analysis.

Hence, a body of research has been conducted for learning debiased classifiers with no additional label for spurious attributes [39, 25, 27, 31, 22, 24]. A common approach in this line of work is to employ an intentionally biased classifier as an auxiliary module [27, 31, 22, 24]. In this approach, samples that the biased classifier has trouble handling are considered bias-conflicting ones and assigned large weights when used for training the main classifier to reduce the effect of bias-guiding counterparts. Although it has driven remarkable success, this approach has drawbacks due to the use of a single biased classifier. First, the quality of the biased classifier could vary by hyper-parameters [28] and its initial parameter values [10]. Further, data that the biased classifier fails to handle could include not only bias-conflicting samples but also bias-guiding ones, which differs by the quality of the classifier.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

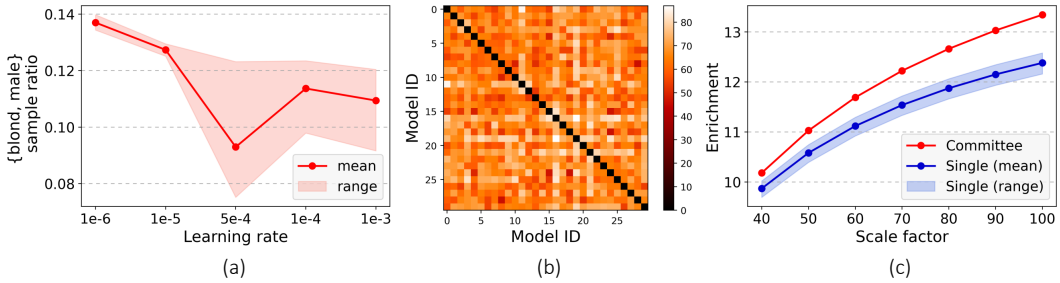


Figure 1: Analysis on the instability of a single biased classifier in mining and weighting bias-conflicting samples. The experiments are conducted on the CelebA dataset, in which samples with `blond` and `male` attributes are bias-conflicting. (a) The ratio of bias-conflicting samples to all incorrectly predicted by a single biased classifier. The ratio highly fluctuates by the learning rate of the classifier and varies up to 4%p due to different initialization even with a fixed learning rate, meaning that using a single biased classifier is sensitive to hyper-parameters. (b) Disagreement on predictions of biased classifiers. For pairs of biased classifiers initialized differently, we measure the number of bias-conflicting samples for which the classifiers predict differently. The results suggest that individual biased classifiers are sensitive to initialization. (c) Comparisons between a single biased classifier and a committee of biased classifiers. Higher enrichment implies more precise mining and weighting of bias-conflicting samples [28]. The committee clearly outperforms the single biased classifier in terms of enrichment.

These drawbacks limit the reliability and performance of debiasing methods depending on a single biased classifier, as demonstrated in Figure 1.

To overcome these limitations, we propose a new method using a committee of biased classifiers as the auxiliary module, coined *learning with biased committee* (LBC). LBC identifies bias-conflicting samples and determines their weights through consensus on their prediction difficulty within the committee. To this end, the committee is built as a bootstrapped ensemble, *i.e.*, each of its classifiers is trained from a randomly sampled subset of the entire training dataset. This strategy not only guarantees the diversity among the classifiers, but also lets a majority of the classifiers be biased since random subsets of training data are highly likely to be dominated by bias-guiding samples. Accordingly, a majority of the committee tends to classify bias-guiding samples correctly and fail to deal with bias-conflicting ones. The consensus on prediction difficulty within the committee thus gives a strong cue for identifying and weighting bias-conflicting samples. Also, using the consensus of multiple classifiers enables LCB to be robust to the varying quality of individual classifiers and consequently to focus more precisely on bias-conflicting samples, as shown in Figure 1.

Moreover, unlike the biased classifier trained independently of the main classifier in the previous work, the committee in LCB is trained with knowledge of the main classifier as well as the random subsets of training data to serve the main classifier better. Specifically, the knowledge is distilled in the form of classification logits of the main classifier [17], and each classifier of the committee utilizes the knowledge as pseudo labels of training data other than its own training set. We expect that this strategy allows the committee to become debiased gradually so that it does not give large weights to easy bias-conflicting samples, *i.e.*, those already well handled by the main classifier, and focuses more on difficult ones. Note that, even with this strategy, the classifiers of the committee are still biased differently due to their different training sets with ground-truth labels.

Finally, we further improve the proposed method by adopting a self-supervised representation as the frozen backbone of the committee and the main classifier. Since self-supervised learning is not dependent on class labels, it is less affected by the spurious correlations between classes and latent attributes, leading to a robust and less-biased representation. Also, by installing the committee and the main classifier on top of the representation, the classifiers can be implemented efficiently in both space and time while enjoying the rich and bias-free features given by the backbone.

LCB is validated extensively on five real-world datasets. It substantially outperforms existing methods using no bias label and even occasionally surpasses previous arts demanding bias labels. We also demonstrate that all of the main components, *i.e.*, the use of the committee, its training with knowledge transfer, and the self-supervised learning, contribute to the outstanding performance. The main contribution of this paper is four-fold:

- We present LBC, a new approach to learning a debiased classifier with no spurious attribute label. The use of consensus within the committee allows LBC to address limitations of previous work relying on a single biased classifier.
- We propose to learn the committee using knowledge of the main classifier, unlike the previous work whose auxiliary modules do not consider the main classifier.
- We investigate the potential of self-supervised learning for debiasing, and find that it is a solid yet unexplored baseline for the task.
- LCB demonstrates superior performance on five real-world datasets. It outperforms existing methods using no additional supervision like ours and even surpasses those relying on spurious attribute labels occasionally.

2 Related work

2.1 Debiasing with prior knowledge on bias

A majority of previous work on learning debiased classifiers mitigates the bias by leveraging explicit labels on bias types [21, 26, 34, 1, 36, 35, 42]. For instance, Kim et al. [21] build a debiased model to classify digits by leveraging RGB color labels explicitly in the Colored MNIST dataset. Other methods are designed to handle predefined domain-specific bias types (*e.g.*, texture bias in ImageNet) [38, 4, 2].

Recent methods try to reduce annotation costs for bias supervision by only utilizing a small set of bias-labeled data. For instance, Nam et al. [32] and Jung et al. [20] train auxiliary bias predictor with a small set of bias-labeled data and assign pseudo bias labels to the entire training set using its prediction. These methods have two inherent limitations. First, they cannot handle biases whose types are not predefined in training. Second, manually annotating all bias types is often expensive and laborious. Even obtaining a small set of bias labels could be expensive since bias labels existing in a real-world dataset is often long-tailed [29] or multi-labeled [15].

2.2 Debiasing without spurious attribute labels

Recent debiasing methods without any bias supervision [18, 7, 31, 24, 22, 27, 6] are based on the assumption that attributes of malignant bias are easier to learn than those of target classes. Through this assumption, previous methods identify the amount of biases within the samples and train a robust models by emphasizing bias-conflicting samples. To identify biases of samples, previous work leverage a high gradient of the latent vector [18, 7] or employ an intentionally biased classifier as an auxiliary module [31, 22, 24] by using generalized cross-entropy (GCE) loss [41]. Liu et al. [27] consider misclassified samples as bias-conflicting samples, and Creager et al. [6] derive bias-guiding and bias-conflicting partition that maximally violates the invariance principle. On the other hand, Bahng et al. [2] propose a method tailored to the texture bias, building a debiased model to learn independent features from a biased model in which receptive field size is limited to capture the texture bias in image classification.

While LBC is also based on the aforementioned assumption, we firstly propose to identify bias-conflicting samples and up-weight them through the consensus of a committee of the biased classifiers. By exploiting the consensus of multiple classifiers, LBC is robust against the varying quality of individual classifiers and allows to learn debiased classifiers more reliably and effectively.

3 Proposed method

We propose a new method that learns a debiased classifier with a committee of biased classifiers, dubbed LBC. It first learns a feature representation with self supervision, which is used as the frozen backbone providing rich and bias-free features to downstream modules (Section 3.1). Next, it trains a committee of m auxiliary classifiers $f_1, f_2, \ldots, f_m$ and the main classifier g on top of the self-supervised representation (Section 3.2); thanks to the self-supervised representation, the classifiers are designed concisely, using only two fully-connected layers for each.

In a nutshell, the committee identifies bias-conflicting samples and assigns them large weights to reduce the effect of bias-guiding samples during the training of the main classifier. To this end, the

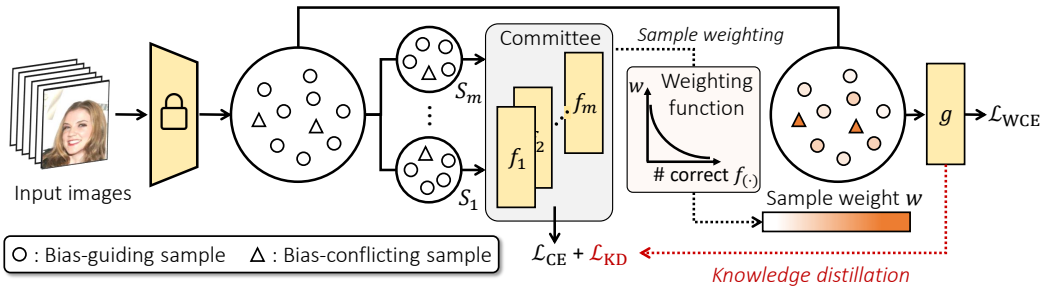


Figure 2: The overall pipeline of LBC. It adopts a frozen backbone trained by self-supervised learning. A committee of auxiliary classifiers f_i is trained on top of the backbone; a random subset of training data is assigned to each classifier of the committee as labeled data for supervised learning (Eq. (1)). The committee determines the weight of each training sample based on consensus of its members, *i.e.*, the number of members that correctly predict the class of the sample (Eq. (2)). The main classifier g is trained using the weights by minimizing the weighted cross entropy loss (Eq. (3)). In turn, knowledge of the main classifier is transferred to the committee through knowledge distillation (Eq. (4)).

committee is trained as a bootstrapped ensemble of classifiers so that a majority of its classifiers are biased as well as diverse, and intentionally fail to predict classes of bias-conflicting samples accordingly. Hence, the consensus within the committee on prediction difficulty of a sample (*e.g.*, the number of classifiers that fail to predict its class label) indicates how much likely the sample is bias-conflicting, and is used to compute weights for training samples. Moreover, the committee is trained also with knowledge of the main classifier so that it gradually becomes debiased along with the main classifier and emphasizes more difficult samples as training progresses. Note that the committee is an auxiliary module used only in training and thus does not impose additional computation or memory footprint in testing.

The overall process of LBC is illustrated conceptually in Figure 2 and given formally in Algorithm 1. The following sections elaborate on each step of LBC.

3.1 Self-supervised representation learning

As the feature extractor, we train a backbone network by self-supervised learning with BYOL [13] on the target dataset. During the self-supervised learning, a random patch of input image is cropped, resized to 224×224 pixels, flipped horizontally at random, and distorted by a random sequence of brightness, contrast, saturation, hue adjustments, and grayscale conversion.

A self-supervised model can capture diverse patterns shared by data without being biased towards a particular class even the training set is biased. We empirically demonstrate that adopting a self-supervised representation leads to a model less biased compared with a supervised representation.

Although the representation offers rich and less-biased features, the main classifier can be still biased when it is trained by ERM. In other words, the self-supervised representation alone is not enough to learn a debiased model, and the need to explore a debiasing method for a classifier arises. Hence, we propose LBC, a new debiasing method illustrated in the next section.

3.2 Learning a debiased classifier with a biased committee

First, we randomly sample m subsets of the same size, denoted by $\mathcal{S}_1, ..., \mathcal{S}_m$, from the entire training dataset $\mathcal{D}$ with replacement. Then m auxiliary classifiers $f_1, ..., f_m$ of the committee are initialized randomly, and each of the subsets $\mathcal{S}_l$ is assigned to each auxiliary classifier f_l as its training data.

The first step of LBC is warm-up training of the committee; this is required to ensure that the committee is capable of identifying and weighting bias-conflicting samples at the beginning of the main training process. Given a mini-batch $\mathcal{B}$ at each warm-up iteration, the committee is trained by minimizing the cross-entropy loss below:

$$\mathcal{L}_{\text{CE}} = \sum_{l=1}^{m} \sum_{(x,y)\in\mathcal{S}_l\cap\mathcal{B}} \text{CE}(f_l(x), y). \tag{1}$$

Algorithm 1 Learning a debiased classifier with a biased committee

input training set $\mathcal{D}$, batch size b, size of the committee m, learning rate η, scale hyper-parameter α, balancing hyper-parameter λ, total number of iterations t, number of warm-up iterations t_w

Draw m random subsets $\mathcal{S}_1, ..., \mathcal{S}_m$ of $\mathcal{D}$.
Initialize auxiliary classifiers of the committee $\{f_l(x;\theta_l)\}_{l=1}^m$.
Initialize the main classifier $g(x;\phi)$.
for $j = 1, ..., t_w$ **do**
  Draw a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^b$ from $\mathcal{D}$.
  $\theta_l \leftarrow \theta_l - \eta \nabla_{\theta_l} \mathcal{L}_{\text{CE}} \;\; \forall l = 1, ..., m$ Eq. (1)
end for
for $j = t_w + 1, ..., t$ **do**
  Draw a mini-batch $\mathcal{B} = \{(x_i, y_i))\}_{i=1}^b$ from $\mathcal{D}$.
  $w(x_i) = 1/(\sum_{l=1}^m \mathbb{1}(f_l(x_i) = y_i)/m + \alpha) \;\; \forall x_i \in \mathcal{B}$ Eq. (2)
  $\phi \leftarrow \phi - \eta \nabla_{\phi} \mathcal{L}_{\text{WCE}}$ Eq. (3)
  $\theta_l \leftarrow \theta_l - (1-\lambda)\eta \nabla_{\theta_l} \mathcal{L}_{\text{CE}} - \lambda \eta \nabla_{\theta_l} \mathcal{L}_{\text{KD}} \;\; \forall l = 1, ..., m$ Eq. (1), Eq. (4)
end for

Since each subset is sampled from the training set dominated by bias-guiding samples, a majority of auxiliary classifiers are also biased. At the same time, the classifiers are diverse due to their difference in initialization and training data.

After the warm-up stage, the main classifier and the committee are trained while interacting with each other. First, the main classifier is trained by the weighted cross entropy loss with the entire training set, where the sample weights are computed by considering consensus within the committee on prediction difficulty of the samples. Since a majority of auxiliary classifiers have trouble to handle bias-conflicting samples, we identity and weight bias-conflicting samples based on the number of auxiliary classifiers whose predictions are correct for the samples. The weight function w is given by

$$w(x) = \frac{1}{\sum_{l=1}^m \mathbb{1}(f_l(x) = y)/m + \alpha}, \tag{2}$$

where m is the size of the committee, f_l means the l-th classifier of the committee, and α is a scale hyper-parameter. The weight reflects how much the sample x is likely to be bias-conflicting and decreases rapidly when the number of correctly predicting classifiers increases. Then we train the main classifier g with emphasis on the bias-conflicting samples through the weight function w. The weighted cross entropy loss $\mathcal{L}_{\text{WCE}}$ is given by

$$\mathcal{L}_{\text{WCE}} = \sum_{(x,y)\in\mathcal{B}} w(x) \cdot \text{CE}(g(x), y), \tag{3}$$

where $\mathcal{B}$ is the mini-batch.

During training, as the main classifier is gradually debiased, samples useful for debiasing the main classifier change accordingly. To focus more on bias-conflicting samples difficult for the main classifier, we inform the quality of the main classifier to the committee by distilling the knowledge of the main classifier in the form of its classification logits [17] and transferring the knowledge by minimizing the following KD loss:

$$\mathcal{L}_{\text{KD}} = \sum_{l=1}^m \sum_{(x,y)\in\mathcal{B}-\mathcal{S}_l} \text{KL}\left(\text{softmax}\left(\frac{g(x)}{\tau}\right), \text{softmax}\left(\frac{f_l(x)}{\tau}\right)\right), \tag{4}$$

where τ is a temperature parameter. Note that we apply $\mathcal{L}_{\text{KD}}$ to the complement set of $\mathcal{S}_l$ to avoid auxiliary classifiers in the committee being identical to each other. By interacting with the main classifier, the committee gradually becomes debiased along with the main classifier. Since samples correctly predicted by the main classifier are less weighted and those with incorrect predictions are more weighted by the committee.

After the warm-up training, the main classifier and the auxiliary classifiers are alternately updated with a given mini-batch at each iteration. First, we forward every sample in a mini-batch to each auxiliary classifier and then compute weights of the samples using predictions of the auxiliary classifiers

Table 1: GUIDING, UNBIASED, and CONFLICTING metrics (%) for the CelebA dataset. For methods without spurious attribute labels, we mark the best and the second-best performance in **bold** and underline, respectively.

Method	Spurious attribute label	CelebA HairColor			CelebA HeavyMakeup		
		GUIDING	UNBIASED	CONFLICTING	GUIDING	UNBIASED	CONFLICTING
Group DRO [34]	✓	87.46	85.43$_{\pm 0.53}$	83.40$_{\pm 0.67}$	79.52	64.88$_{\pm 0.42}$	50.24$_{\pm 0.68}$
EnD [35]	✓	94.97	91.21$_{\pm 0.22}$	87.45$_{\pm 1.06}$	98.16	75.93$_{\pm 1.31}$	53.70$_{\pm 5.24}$
CSAD [42]	✓	91.19	89.36	87.53	82.32	67.88	53.44
ERM	✗	87.98	70.25$_{\pm 0.35}$	52.52$_{\pm 0.19}$	90.25	62.00$_{\pm 0.02}$	33.75$_{\pm 0.28}$
LfF [31]	✗	87.24	84.24$_{\pm 0.37}$	81.24$_{\pm 1.38}$	86.92	66.20$_{\pm 1.21}$	45.48$_{\pm 4.43}$
SSL+ERM	✗	**94.15**$_{\pm 0.57}$	80.48$_{\pm 0.91}$	66.79$_{\pm 2.20}$	**93.00**$_{\pm 0.49}$	66.30$_{\pm 1.15}$	39.50$_{\pm 2.47}$
LBC	✗	90.57$_{\pm 2.15}$	**88.90**$_{\pm 1.55}$	**87.22**$_{\pm 1.14}$	86.42$_{\pm 0.82}$	**70.29**$_{\pm 1.14}$	**51.28**$_{\pm 5.74}$

Table 2: UNBIASED and WORST-GROUP metrics (%) for the CelebA dataset. We also report the difference between UNBIASED and WORST-GROUP as GAP. For methods without spurious attribute labels, we mark the best and the second-best performance in **bold** and underline, respectively.

Method	Backbone network	Spurious attribute label	CelebA HairColor		
			UNBIASED	WORST-GROUP	GAP
Group DRO [34]	Resnet50	✓	93.1$_{\pm 0.21}$	88.5$_{\pm 1.16}$	4.6
SSA [32]	Resnet50	✓	92.8$_{\pm 0.11}$	89.8$_{\pm 1.28}$	3.0
ERM	Resnet50	✗	**95.6**	47.2	48.4
CVaR DRO [25]	Resnet50	✗	82.4	64.4	18.0
LfF [31]	Resnet50	✗	86.0	70.6	15.4
EIIL [6]	Resnet50	✗	91.9	83.3	8.6
JTT [27]	Resnet50	✗	88.0	81.1	6.9
SSL+ERM	Resnet18	✗	80.5$_{\pm 0.9}$	38.5$_{\pm 4.1}$	42.0
LBC	Resnet18	✗	88.9$_{\pm 1.6}$	**85.5**$_{\pm 1.4}$	**3.4**

(Eq. (2)). With the weights, the main classifier is updated by Eq. (3) and the knowledge of the updated main classifier is transferred to the auxiliary classifiers by Eq. (4). Thus, the auxiliary classifiers are updated by Eq. (1) and Eq. (4) as follows:

$$\mathcal{L}_{\text{committee}} = (1-\lambda)\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{KD}} \tag{5}$$

4 Experiments

4.1 Setup

Implementation details. For the training self-supervised model, we train ResNet-18 [14] following self-supervised learning with BYOL [13] on the target dataset. Since the color information is key feature to classify `HairColor` class on the celebA dataset, we adopt only brightness and contrast as color distortion when it trains celebA. Since NICO and BAR are very small dataset, we train self-supervised model from ImageNet [8] pretrained parameters as an initial point. Except for the experiments using BAR and NICO, we train self-supervised models from scratch. We use the self-supervised ResNet-18 as a backbone network except for the last fully connected layer. We set the batch size to {64, 64, 128, 256}, learning rate {1e-3, 1e-3, 1e-4, 6e-3}, the size of the committee m to {30, 30, 30, 40}, the size of subset $\mathcal{S}_l$ to {10, 10, 80, 300}, λ to {0.9, 0.6, 0.6, 0.6}, and τ to {1, 1, 1, 2.5}, respectively for {BAR, NICO, Imagenet-9, CelebA} and α to 0.02. Note that we average the accuracy of each training over three independent trials with random seeds.

Evaluation metrics. For evaluation, we adopt six metrics. VALIDATION / TEST: average accuracy on validation / test splits. GUIDING: average accuracy on bias guiding samples per class. CONFLICTING: average accuracy on bias conflicting samples per class. UNBIASED: average of 'Guiding' and 'Conflicting' per class. WORST-GROUP: minimum average accuracy of group.

Table 3: VALIDATION, UNBIASED, and TEST metrics (%) evaluated on the ImageNet-9 and ImageNet-A datasets. For methods without spurious attribute labels, we mark the best and the second-best performance in **bold** and underline, respectively.

Method	Spurious attribute label	ImageNet-9		ImageNet-A
		VALIDATION	UNBIASED	TEST
StylisedIN [12]	✓	$88.4_{\pm 0.5}$	$86.6_{\pm 0.6}$	$24.6_{\pm 1.4}$
LearnedMixin [5]	✓	$64.1_{\pm 4.0}$	$62.7_{\pm 3.1}$	$15.0_{\pm 1.6}$
RUBi [4]	✓	$90.5_{\pm 0.3}$	$88.6_{\pm 0.4}$	$27.7_{\pm 2.1}$
ERM	✗	$90.8_{\pm 0.6}$	$88.8_{\pm 0.6}$	$24.9_{\pm 1.1}$
Biased (BagNet18) [3]	✗	$67.7_{\pm 0.3}$	$65.9_{\pm 0.3}$	$18.8_{\pm 1.15}$
ReBias [2]	✗	$91.9_{\pm 1.7}$	$90.5_{\pm 1.7}$	$29.6_{\pm 1.6}$
LfF [31]	✗	86.0	85.0	24.6
CaaM [39]	✗	**95.7**	**95.2**	32.8
SSL+ERM	✗	$\underline{94.18}_{\pm 0.07}$	$\underline{93.18}_{\pm 0.04}$	$\underline{34.21}_{\pm 0.49}$
LBC	✗	$94.03_{\pm 0.23}$	$93.04_{\pm 0.32}$	$\mathbf{35.97}_{\pm 0.49}$

4.2 Datasets

CelebA. CelebA [30] is a dataset for face recognition where each sample is labeled with 40 attributes. Following the experiment configuration suggested by Nam et al. [31], we focus on `HairColor` and `HeavyMakeup` attributes that are spuriously correlated with `Gender` attributes, *i.e.*, most of the CelebA images with `blond hair` are women. As a result, the biased model suffers from performance degradation when predicting `HairColor` attribute on males. Therefore, we use `HairColor` as the target attribute and `Gender` as a spurious attribute, the same as `HeavyMakeup`.

ImageNet-9. ImageNet-9 [19] is a subset of ImageNet [33] containing nine super-classes. Following the setting adopted by [2], we conduct experiments with 54,600 training images and 2,100 validation images. ImageNet-9 tempts to have a correlated object class and image texture. We follow the evaluation scheme adopted by [2], and we report the unbiased accuracy of the validation set, which is computed as average accuracy on every object-texture combination.

ImageNet-A. ImageNet-A [16] contains real-world images misclassified by ImageNet-trained ResNet 50 [14]. Since such failure is caused when a model too heavily relies on the color, textures, and backgrounds. The ImageNet-A dataset could be a bias-conflicting set w.r.t. various ImageNet biases. This dataset is used only for evaluating a model trained on ImageNet-9.

BAR. The Biased Action Recognition (BAR) dataset [31] is a real-world image dataset designed to evaluate the spurious correlation between human action and place on real-world images. Originally the given training set of BAR consists of only 100% bias-guiding samples, and its test set consists of only bias-conflicting samples. In our setting, we use 10% of the original BAR training set as validation and set the bias-conflicting ratio of the training set to 1%.

NICO. NICO [15] is a real-world dataset for simulating the out-of-distribution image classification task. Following the setting used by Wang et al. [39], we use an animal subset of NICO, which is labeled with 10 object and 10 context classes for evaluating the debiasing methods. The training set consists 7 context classes per object class and they are long-tailed distributed (e.g., most dog images are appeared 'on grass' the others are appeared on 6 contexts). The validation and test set consists of 10 context classes with 3 unseen context classes per object class. We verify the ability of debiasing a model from object-context correlations by evaluating on NICO.

4.3 Quantitative results

LBC shows superior classification accuracy among the methods that do not use the spurious attribute label on the five real-world datasets. In Tables 1 & 2, we observe LBC outperforms existing debiasing methods using no spurious attribute label and shows comparable CONFLICTING performance with methods that exploit spurious attribute labels on CelebA, which reflects gender bias in the real world. Especially, the gap between the average accuracy of groups and worst-group accuracy of LBC is much smaller than the other methods, this means that our model fairly predicts a sample belongs

Table 4: CONFLICTING, VALIDATION, and TEST metrics (%) evaluated on the BAR dataset (a) and the NICO dataset (b). For methods without spurious attribute labels, we mark the best and the second-best performance in **bold** and underline, respectively.

Method	Spurious attribute label	BAR
		CONFLICTING
ReBias [2]	✗	$37.02_{\pm 0.26}$
ERM	✗	$35.32_{\pm 0.46}$
LfF [31]	✗	$48.15_{\pm 0.93}$
LDD [24]	✗	$52.31_{\pm 1.00}$
SSL+ERM	✗	$\underline{60.88}_{\pm 0.80}$
LBC	✗	$\mathbf{62.03}_{\pm 0.74}$

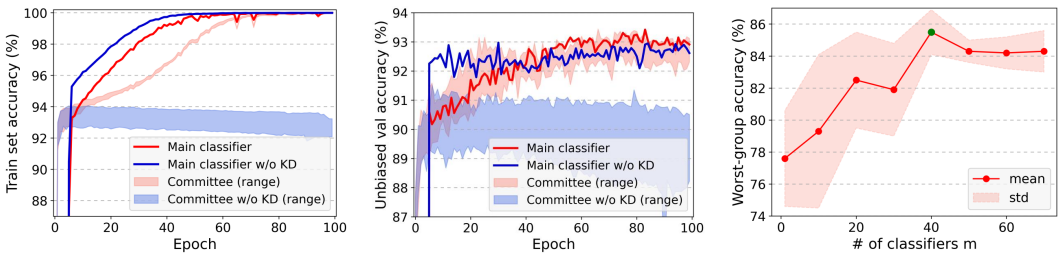
(a) BAR dataset

Method	Spurious attribute label	NICO	
		VALIDATION	TEST
Cutout [9]	✓	43.69	43.77
RUBi [4].	✓	43.86	44.37
IRM [1]	✓	40.62	41.46
Unshuffle [36]	✓	43.15	43.00
REx [23]	✓	41.00	41.15
ERM	✗	43.77	42.61
CBAM [40]	✗	42.15	42.46
ReBias [2]	✗	44.92	45.23
LfF [31]	✗	41.83	40.18
CaaM [39]	✗	46.38	46.62
SSL+ERM	✗	$\underline{55.63}_{\pm 0.54}$	$\underline{52.24}_{\pm 0.27}$
LBC	✗	$\mathbf{56.05}_{\pm 0.45}$	$\mathbf{52.84}_{\pm 0.31}$

(b) NICO dataset

Table 5: Ablation studies using WORST-GROUP metric (%) on the CelebA HairColor dataset. We study the impact of learning from a single biased classifier (row 2), learning by committee (row 3), and transferring the knowledge of the main classifier (row 4). We mark the best performance in **bold**.

Method					CelebA HairColor
Self-sup	ERM	Single	Committee	KD	WORST-GROUP
✓	✓	✗	✗	✗	$38.5_{\pm 4.1}$
✓	✗	✓	✗	✗	$64.1_{\pm 2.4}$
✓	✗	✗	✓	✗	$81.3_{\pm 2.3}$
✓	✗	✗	✓	✓	$\mathbf{85.5}_{\pm 1.4}$



(a) ImageNet-9 (b) CelebA HairColor

Figure 3: Effect of knowledge distillation (Eq. (4)) and the size of the committee m. (a) Accuracy of the main classifier and the committee with or without KD loss in ImageNet-9 train set (*left*) and unbiased validation set (*right*). (b) Worst-group accuracy of LCB versus the number of classifiers within the committee, where the green dot indicates the value used in the main paper.

to each group. Table 3 shows the results on the ImageNet-9, which is dominated by texture bias, and the ImageNet-A, which is regarded as a bias-conflicting set of ImageNet. LBC is 9.7% better than CaaM [40] on the ImageNet-A dataset, *i.e.*, LBC is robust to texture bias and various ImageNet biases. Table 4a shows results on the BAR dataset. LBC is 18.6% better than LDD, which is the previous state of the art. Compared to LfF [31] and LDD [24], which are debiasing methods with a single biased classifier, we demonstrate that learning a debiased classifier with the biased committee is more effective than a single classifier scheme. Table 4b shows the results on the NICO dataset. LBC is 13.3% better than CaaM, *i.e.*, LBC is well generalized to the unseen spurious attributes. Note that the validation and test set of NICO have unseen context classes and are unbiased.

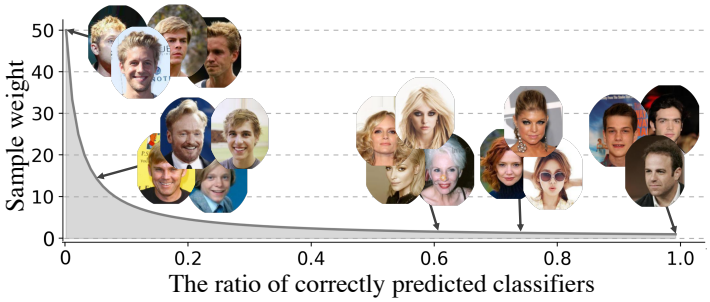


Figure 4: Qualitative examples on CelebA dataset. *Hair color* is target and *Gender* is the bias.

4.4 Ablation study

Self-supervised representation as a solid baseline. We empirically investigate the potential of self-supervised representation as a solid baseline for the debiasing task. We train a classifier on the top of the self-supervised representation by ERM. The results are denoted by 'SSL+ERM' and compared with 'ERM', which is a fully supervised classification model in Table 1, 2, 3, 4a, 4b. 'SSL+ERM' outperforms not only ERM but also the previous state of the art on all the datasets except for CelebA. 'SSL+ERM' is less biased than the model trained by fully supervised learning.

Importance of each module in LBC. Table 5 demonstrates through ablation studies [27, 31, 22, 24]: (1) learning from a single biased classifier, (2) learning with committee, and (3) transferring the knowledge of the main classifier. First, we train a classifier by ERM (row 1) then assign a weight value 50 to the wrongly predicted samples and 1 to other samples. Then re-training the classifier with the weights (row 2). Comparing these two results demonstrates that up-weighting scheme with a biased classifier is effective to debiasing a classifier. Then we increase the number of biased classifier and compute the sample weights using our weight function Eq. (2) (row 3). Learning with the committee shows a remarkable improvement on the worst-group accuracy. Moreover, knowledge distillation that enables the committee to interact with the main classifier further improves performance (row 4).

Effectiveness of transferring knowledge of the main classifier. Figure 3(a) shows the range of unbiased validation accuracy of classifiers in the committee and unbiased validation accuracy of main classifier during training. The mean accuracy of classifiers in the committee gradually increase following the accuracy of the main classifier. Also, the accuracy of the main classifier gradually increases as training progresses. On the other hand, the accuracy of classifiers in the committee trained without KD loss does not increase or even decrease.

Number of classifiers. We compare the results of the main classifier trained with the different number of auxiliary classifiers. Figure 3(b) shows the worst-group accuracy versus the number of auxiliary classifiers m on CelebA. With a single auxiliary classifier, the main classifier shows the lowest worst-group accuracy, but the accuracy increases as m increases. When larger than 40, the number of classifiers has little effect on learning the main classifier.

5 Conclusion

We have proposed a new method for learning a debiased classifier with a committee of auxiliary classifiers. The committee is learned in a way that consensus on predictions of its classifiers offers a strong and reliable cue to identify and weight bias-conflicting data. The main debiased classifier is then trained with emphasis on the bias-conflicting data to reduce the effect of bias-guiding counterparts. The committee is also trained to be debiased gradually along with the main classifier so that it highlights more challenging data as training progresses. Moreover, we demonstrated that self-supervised learning is a solid yet unexplored baseline for debiasing. Coupled with a self-supervised feature extractor, our method achieved the state of the art by large margins on most of existing real-world datasets.

Limitations. Self-supervised learning increases overall training complexity as it usually takes a longer time for convergence than supervised learning. Also, since the committee is designed as a bootstrapped ensemble, it involves randomness in training and thus causes noticeable performance fluctuation on CelebA, although it demonstrates stable performance on the other datasets.

References

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

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *Proc. International Conference on Machine Learning (ICML)*, pages 528–539. PMLR, 2020.

[3] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

[4] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 32, pages 841–852, 2019.

[5] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, 2019.

[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.

[7] Luke Darlow, Stanisław Jastrzębski, and Amos Storkey. Latent adversarial debiasing: Mitigating collider bias in deep neural networks. *arXiv preprint arXiv:2011.11486*, 2020.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[10] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[11] 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.

[12] 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 *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. In *Pattern Recognition*, volume 110, page 107383. Elsevier, 2021.

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2021.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Proc. European Conference on Computer Vision (ECCV)*, pages 124–140. Springer, 2020.

[19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 32, pages 125–136, 2019.

[20] Sangwon Jung, Sanghyuk Chun, and Taesup Moon. Learning fair classifiers with partially annotated group labels. *arXiv preprint arXiv:2111.14581*, 2021.

[21] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9012–9020, 2019.

[22] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 14992–15001, 2021.

[23] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *Proc. International Conference on Machine Learning (ICML)*, pages 5815–5826. PMLR, 2021.

[24] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

[25] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 33, pages 8847–8860, 2020.

[26] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9572–9581, 2019.

[27] 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 *Proc. International Conference on Machine Learning (ICML)*, pages 6781–6792. PMLR, 2021.

[28] 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 *Proc. International Conference on Machine Learning (ICML)*, pages 6781–6792. PMLR, 2021.

[29] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015.

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[31] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020.

[32] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. In *Proc. International Conference on Learning Representations (ICLR)*, 2022.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[34] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[35] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13508–13517, 2021.

[36] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization in visual question answering. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1417–1427, 2021.

[37] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[38] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[39] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. Causal attention for unbiased visual recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 3091–3100, 2021.

[40] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proc. European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[41] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2018.

[42] Wei Zhu, Haitian Zheng, Haofu Liao, Weijian Li, and Jiebo Luo. Learning bias-invariant representation by cross-sample mutual information minimization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 15002–15012, October 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] In fact, our method relieves societal discrimination problem of neural network.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1 and supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We average the accuracy of each training over three independent trials with random seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.1 and supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]