Partition-and-Debias: Agnostic Biases Mitigation via A Mixture of Biases-Specific Experts

Jiaxuan Li
The University of Tokyo, Japan
li@nlab.ci.i.u-tokyo.ac.jp

Duc Minh Vo
The University of Tokyo, Japan
vmduc@nlab.ci.i.u-tokyo.ac.jp

Hideki Nakayama
The University of Tokyo, Japan
nakayama@ci.i.u-tokyo.ac.jp

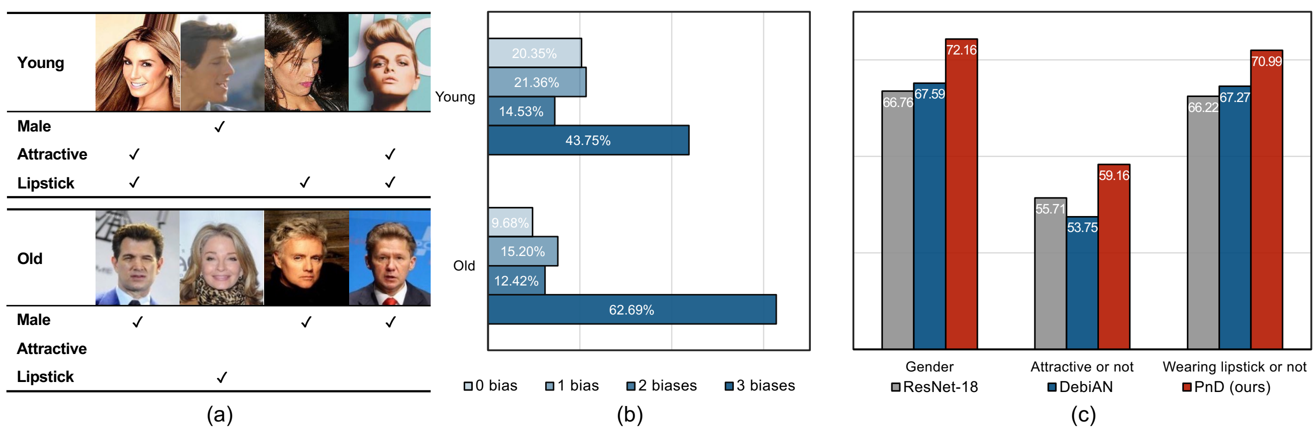

Figure 1: Taking the age attribute in the CelebA dataset as an example for analyzing the agnostic biases problem. (a) Representative samples in CelebA containing multiple biases. By analyzing the attribute distribution of all data within the *young/old* category, we found that three attributes can be biased: gender (*female/male*), attractiveness (*attractive/not attractive*), and wearing lipstick (*lipstick/no lipstick*). (b) Proportion of samples with 0 – 3 biases in a single image within the *young* and *old* groups. This indicates that the number of images with multiple biases dominated other cases in the dataset. (c) Age classification accuracy (%) of the existing methods for the worst groups of three bias attributes in CelebA degrades under this realistic bias scenario. For clarification, when discussing biases in this paper, we refer to abstract words like age as "attribute", and the *italic* words which describe the labels of age like *young/old* as "category".

Abstract

Bias mitigation in image classification has been widely researched, and existing methods have yielded notable results. However, most of these methods implicitly assume that a given image contains only one type of known or unknown bias, failing to consider the complexities of real-world biases. We introduce a more challenging scenario, ***agnostic biases mitigation****, aiming at bias removal regardless of whether the type of bias or the number of types is unknown in the datasets. To address this difficult task, we present the Partition-and-Debias (PnD) method that uses a mixture of biases-specific experts to implicitly divide the bias space into multiple subspaces and a gating module to find a consensus among experts to achieve debiased classification. Experiments on both public and constructed benchmarks demonstrated the efficacy of the PnD. Code is available at: https://github.com/Jiaxuan-Li/PnD.*

1. Introduction

One of the reasons for poor generalization in image classification is the presence of biased features in training data [19, 31, 13], which distracts the model from learning the target features associated with the classification objects.

Thus, accurately capturing the target features while reducing the influence of these biases[1] has become a critical issue, resulting in increased bias mitigation research [26].

Unlike most previous studies that implicitly assumed that only one type of known/unknown bias exists in a given image, we investigate the coexistence of multiple unknown biases in an image. For instance, most *young* samples in CelebA [17] are associated with the *female*, *attractive*, and *lipstick* categories (Fig. 1a), whereas the *old* samples have corresponding yet reversed ones. Consequently, for the age (*young*/*old*) classification in CelebA, these three biases, including gender, attractiveness, and wearing lipstick, degrade the prediction performance. Overall, we discovered that 43.75% of the *young* samples had three biases, which means they were all annotated with *female*, *attractive*, and *lipstick*, whereas 58.28% of the samples had at least two of them (Fig. 1b). The *old* samples show similar patterns. These observations imply that multiple biases are inevitable in a given image. At the same time, we cannot determine all types of bias that may appear in the image. Dealing with multiple unknown biases is thus emergent, and cannot be fully solved using prior methods (Fig. 1c) because (i) they fail to capture the biases of different types and (ii) removing a single bias does not always eliminate the effects of all biases. Therefore, we introduce a more challenging scenario, **agnostic biases**, in which the unknown biases include not only the type of bias, but also the number of types. Here, we use "agnostic biases" to bring attention to biases in real-world scenarios, where the bias type and number of types are unknown. We do not use "unknown biases" proposed in [9], because it ignores multiple unknown biases. Our scenario overcomes the existing bias constraints, boosting the performance of real-world applications.

We hypothesized and empirically found that the features of agnostic biases scatter at different depths of the network depending on the biases' nature. Even if multiple biases are entangled at the same depth, they can be be regarded as one type of bias. Thus, agnostic biases can be grouped by their feature levels and processed individually at different network depths. As a result, we propose a Partition-and-Debias (PnD) approach based on the divide-and-conquer strategy to capture and remove agnostic biases at different levels for debiased classification. Thus, the entire agnostic bias scenario space is divided into multiple subscenario spaces that can be handled by multiple biases-specific experts. The final prediction is obtained based on the consensus of all the experts using a gating module. Our contributions are:

- We point out the existence of multiple biases in the real world, proposing a new scenario with agnostic biases that fills in the gaps of previous works' bias assumptions.
- We present a Partition-and-Debias approach to solve the new scenario via a mixture of biases-specific experts.
- On both public and our constructed challenging bias datasets, experimental results show that the proposed method achieves cutting-edge performance.

[1]Similar to [25], *"bias"* refers to the attribute spuriously correlated with the target attribute.

2. Related Work

2.1. Bias mitigation

Bias mitigation learns the target features without influence by spurious correlations when training data is biased.
Known bias mitigation assumes the annotation of bias or the type of bias is accessible. The previous methods can be classified as supervised or unsupervised. The former includes reweighting samples with higher uncertainty [14], regularization [22, 25], data augmentation [20], and supervised bias estimation [2, 11, 1]. The latter often uses mixup [7], a two-branch network [18, 13], prioritizing simple target features while ignoring complex biased features [24], and MaskTune [3]. These methods make strong assumptions about the type of bias. For instance, bias can be easily learned [18, 13, 7], target features are simpler than bias features [24], and the bias is editable [3]. Furthermore, most studies considered only a single type of bias appearing in an image. Although [24] used a multiple-bias dataset in their experiments, they still adhered to the limitations of the assumptions on the type of bias. Our method belongs to the unsupervised approach, yet we relax the strong bias assumptions and use a partition-and-debias strategy.
Unknown bias mitigation does not require a pre-definition for bias in the dataset. Jeon *et al*. [9] proposed obtaining unbiased target features from the shallow layers of the classification network. However, their definition of unknown bias misses that there are multiple unknown biases in the data. For real-world datasets, spurious correlations are complex and cannot be defined simply as a result of a specific attribute. By contrast, our agnostic biases assumption emphasizes that both the type and number of bias types are unknown. Also, Li *et al*. [16] proposed an Equal Opportunity Violation loss to discover the most salient bias from unknown biases and then mitigate it by reweighting. Although they considered two biases in their experiments, their method theoretically could only eliminate one dominating bias, which was binary rather than multi-class bias. In contrast to them, our model overcomes these limitations.

2.2. Mixture of experts

The mixture of experts (MoE) technique was originally proposed by Jacobs *et al*. [8] to mitigate the effects of different types of samples on the training data. It divides data into different domains using a gating network and assigns multiple experts to handle each domain. Recently, Zuo *et*

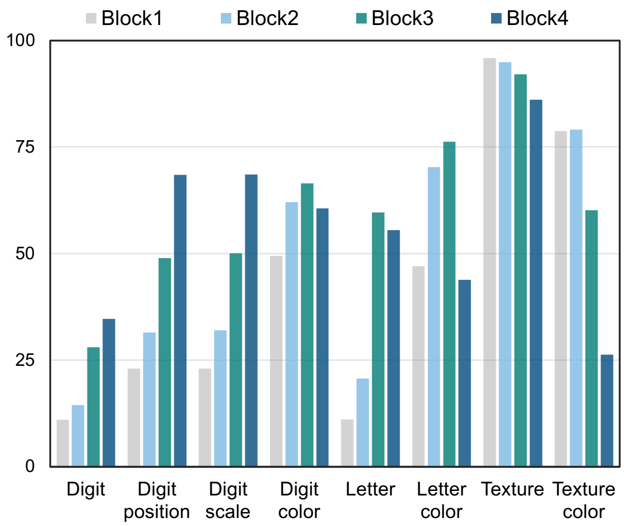

Figure 2: Classification accuracy scores (%) for the 8 attributes when retraining in features learned from target class classification at blocks of different depths in ResNet-18. We find that the classification performance for different attributes trends differently across the depths.

al. [32] used MoE in language models by breaking a pretrained model into multiple experts to speed up the inference process. Zhang *et al*. [29] combined MoE with fine-grained categorization by training each subsequent expert using prior information obtained from the previous expert. Unlike these methods, we employ the MoE strategy for debiasing and specifically design it to remove agnostic biases by inserting experts at different depths of the network.

3. Features of Different Levels Matters

Our hypothesis. When training a neural network with target categories, agnostic biases manifest as scattered features at different network depths.

Experimental setup. We use the Biased MNIST [24] dataset (see Sec. 5.1) in this exploratory experiment. The biases arise from the co-occurrence of each digit category with specific categories from all other attributes, such as digit color and digit position. Unlike an image with a single bias, one image in this dataset may have up to seven biases. The bias ratio, which denotes the probability of co-occurrence, is 0.95. We selected ResNet-18 [6] consisting of four residual blocks, as the classification network.

First, we trained a classification network from scratch using the target categories *0 – 9* in the digits and obtained an average classification accuracy of 33.73% for all categories. The learned features obtained from the trained model can then be visualized to investigate how bias features are distributed across the network when training the network with target categories. However, since many attributes such as digital color, digit position, and digit scale are interdependent, their features overlap in feature maps, making it difficult to distinguish their differences by simply looking at feature maps. We used the classification accuracy for each attribute separately in each block to determine their distributions. Specifically, we froze the trained network weights and added a binary classifier after each trained block. We trained the additional four classifiers to obtain the corresponding classification accuracies for all eight attributes.

Features of biases with different levels are distributed at different depths of the network. We obtained 4×8 accuracy results after retraining, as shown in Fig. 2. (i) From the perspective of different attributes, the classification accuracy of all attributes except texture color was notably higher than that of the digit in the last block (block 4), which is usually used to determine the final prediction. Furthermore, the other blocks followed the same pattern as the last block. This phenomenon implies that the previously learned features from the target attribute classification (here, digits) are more easily classifiable in the bias attribute classification than in the target attribute classification. We concluded that many spuriously correlated features exist at all depths of the network, degrading the target attribute predictions. (ii) From the perspective of different blocks, although most bias attributes can be classified in each layer, the classification performance for some bias attributes varies depending on the block. Texture-relevant attribute classifiers performed well in the first block, while those with position- and scale-relevant attributes performed better in the last block; the remaining attributes achieved the best results in the third block. These findings are consistent with our intuition regarding the distribution of image features, which holds that texture features are more abundant in the shallower parts of the network and that spatial and scale information are more prevalent in the deeper parts of the network. We conclude that each bias attribute feature exists at all network depths, yet these features are clustered at different network depths.

4. Proposed Partition-and-Debias

The above experiment suggests that an ideal strategy for resolving our problem should be able to remove as many biases from the network depths as possible. Thus, we adopt a partition-and-debias strategy in our method, namely PnD, which divides the entire agnostic bias scenario space into different subscenario spaces across the classification network depths. Multiple biases of the same level are allowed in each subscenario space because they can be viewed naturally as a type of bias. This simple concept overcomes the limitations of previous studies allow the model to simultaneously capture and remove multiple biases at one time.

Our PnD consists of a debiased encoder $\mathcal{D}$, bias encoder $\mathcal{B}$, biases-specific experts $\mathcal{E}$, and gating module (Fig. 3).

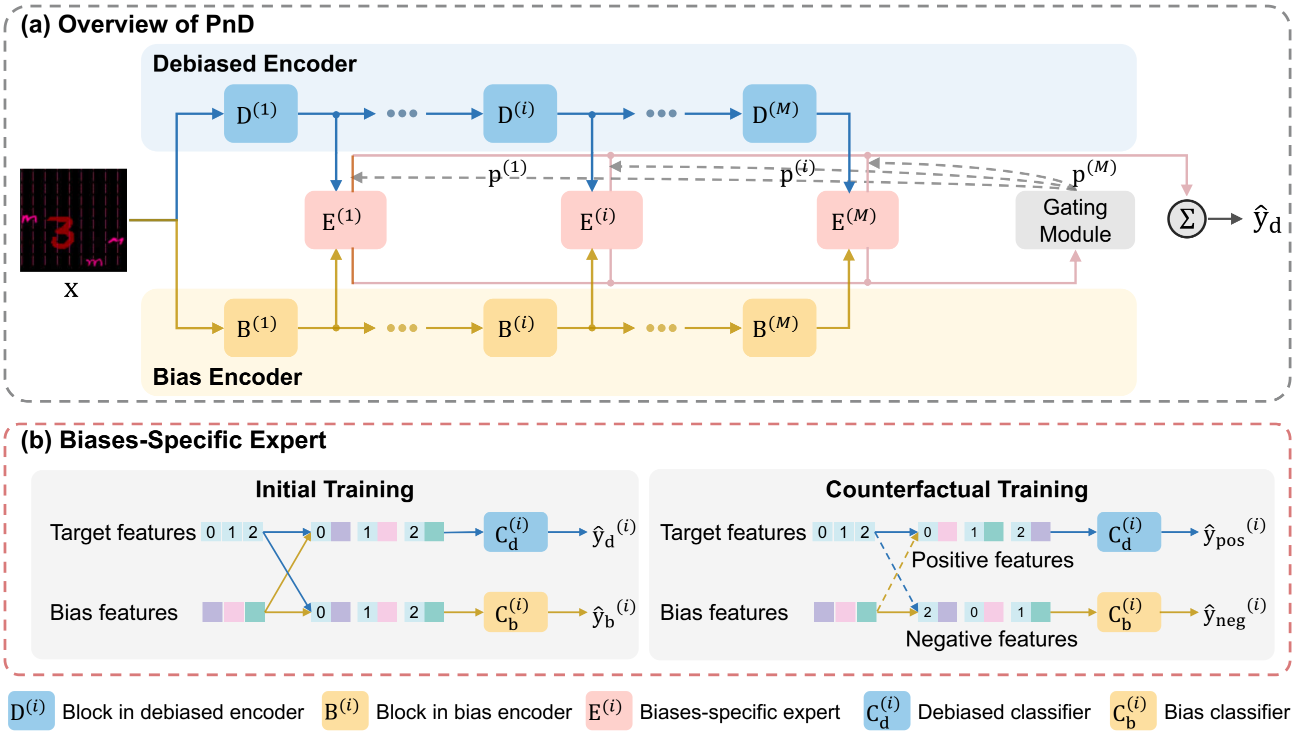

Figure 3: (a) Schematic of our proposed PnD, including a debiased encoder, a bias encoder, multiple biases-specific experts, and a gating module. The debiased encoder and bias encoder extract target and bias features from input images, which are fed into the biases-specific expert after each block for debiased classification and bias detection. The gating module adaptively mixes all the debiased classification results for the final output. (b) Biases-specific expert. It processes target features and bias features by combining them in the order of the original input batch for initial training, and recombining them to generate positive and negative samples during counterfactual training.

Both $\mathcal{D}$ and $\mathcal{B}$ contain several blocks of convolution layers to generate the target and bias features from an image (Sec. 4.1). $\mathcal{E}$ are responsible for purifying the target and bias features under agnostic bias scenarios (Sec. 4.2). Finally, a gating module adaptively gathers all the expert predictions before making a final decision (Sec. 4.3).

4.1. Target and bias features extraction

Given an image $\mathbf{x}$ with the target label $\mathbf{y}$ (vector-like is used to represent class y for simplicity), we use a debiased encoder $\mathcal{D} = \{D^{(i)}\}_{i=1}^{M}$ and bias encoder $\mathcal{B} = \{B^{(i)}\}_{i=1}^{M}$ to extract target and bias features separately. Note that $D^{(i)}$ and $B^{(i)}$ are residual blocks in ResNet [6] although the network architecture can be any and $M = 4$.

The image is fed into $\mathcal{D}$ to obtain the target features $\mathbf{z}_{\mathrm{d}}^{(i)}$ in i^{th} block. Simultaneously, we obtain the bias features $\mathbf{z}_{\mathrm{b}}^{(i)}$ for each block in $\mathcal{B}$. The size of $\mathbf{z}_{\mathrm{d}}^{(i)}$ is identical to that of $\mathbf{z}_{\mathrm{b}}^{(i)}$. We omitted the feature size when referencing the extracted features to simplify the process. Next, biases-specific experts processed these features.

4.2. Biases-specific experts

The biases-specific experts $\mathcal{E}$ consist of four experts $E^{(i)}$. Each of them contains two classifiers: a debiased classifier $C_{\mathrm{d}}^{(i)}$ and bias classifier $C_{\mathrm{b}}^{(i)}$. The inputs of each $E^{(i)}$ are created from the features $\mathbf{z}_{\mathrm{d}}^{(i)}$ and $\mathbf{z}_{\mathrm{b}}^{(i)}$ obtained from the corresponding $D^{(i)}$ and $B^{(i)}$. We combined $\mathbf{z}_{\mathrm{d}}^{(i)}$ and $\mathbf{z}_{\mathrm{b}}^{(i)}$ features in two ways, creating the original and counterfactual features used in our two-stage training (initial and counterfactual trainings). In both training stages, debiased classifier $C_{\mathrm{d}}^{(i)}$ and bias classifier $C_{\mathrm{b}}^{(i)}$ are used for debiased classification and bias detection, respectively.

4.2.1 Initial training

We combine the features $\mathbf{z}_{\mathrm{d}}^{(i)}$ and $\mathbf{z}_{\mathrm{b}}^{(i)}$ to create the original features $\mathbf{z}^{(i)} = [\mathbf{z}_{\mathrm{d}}^{(i)}; \mathbf{z}_{\mathrm{b}}^{(i)}]$ ($[\cdot;\cdot]$ denotes concatenation) (Fig. 3b, left). The i^{th} expert $E^{(i)}$ takes $\mathbf{z}^{(i)}$ as the input, and outputs a bias detection result $\hat{\mathbf{y}}_{\mathrm{b}}^{(i)}$ and a debiased classification result $\hat{\mathbf{y}}_{\mathrm{d}}^{(i)}$ made by $C_{\mathrm{b}}^{(i)}$ and $C_{\mathrm{d}}^{(i)}$, respectively.
Bias detection. This encourages the bias encoder to learn

the bias features. Because the bias features are easier to learn during training with target categories, we can concentrate our bias encoder on the more easily learned features by employing GCE loss [30] as discussed in [18, 13, 12], although the bias information in the dataset is unavailable:

$$\mathcal{L}_{\text{bias}} = \sum_{i=1}^{M} \text{GCE}\left(\hat{\mathbf{y}}_{\text{b}}^{(i)}, \mathbf{y}\right). \tag{1}$$

Debiased classification. To optimize the debiased and bias encoders separately, as opposed to bias detection, debiased classification should prioritize unbiased samples, which do not contain bias features and are difficult to fit using the bias encoder. Consequently, when bias detection is used, these samples are misclassified or classified with lower confidence, whereas debiased classification classifies them correctly or with higher confidence. Considering this, we follow [18] and add a weight $\mathrm{w}^{(i)}$ to each sample in debiased classification. $\mathrm{w}^{(i)}$ is defined as: $\mathrm{w}^{(i)} = \frac{\text{CE}\left(\hat{\mathbf{y}}_{\text{b}}^{(i)}, \mathbf{y}\right)}{\text{CE}\left(\hat{\mathbf{y}}_{\text{d}}^{(i)}, \mathbf{y}\right) + \text{CE}\left(\hat{\mathbf{y}}_{\text{b}}^{(i)}, \mathbf{y}\right)}$, where we use $\text{CE}(\hat{\mathbf{y}}_{\text{d}}^{(i)}, \mathbf{y})$ and $\text{CE}(\hat{\mathbf{y}}_{\text{b}}^{(i)}, \mathbf{y})$ to measure the relative difficulty between debiased classification and bias detection, $\text{CE}(\cdot, \cdot)$ denotes cross-entropy loss function. The loss for debiased classification is expressed as:

$$\mathcal{L}_{\text{debias}} = \sum_{i=1}^{M} \mathrm{w}^{(i)} \times \text{CE}\left(\hat{\mathbf{y}}_{\text{d}}^{(i)}, \mathbf{y}\right). \tag{2}$$

Combining Eq. 1 and Eq. 2, we obtain the total classification loss $\mathcal{L}_{\text{cls}}$ for debiased classification and bias detection: $\mathcal{L}_{\text{cls}} = \alpha \times \mathcal{L}_{\text{debias}} + \mathcal{L}_{\text{bias}}$, α is hyperparameter that balances $\mathcal{L}_{\text{debias}}$ and $\mathcal{L}_{\text{bias}}$; $\mathcal{L}_{\text{debias}}$ forces the debiased classification to focus more on unbiased samples with weight $\mathrm{w}^{(i)}$ added to CE loss, whereas $\mathcal{L}_{\text{bias}}$ focuses on bias features owing to the GCE loss to support easier-learned features.

Diversity penalty for biases-specific experts. To achieve diversified biases-specific experts, we introduc a Kullback-Leibler (KL) divergence-based loss function [28] to penalize the bias detection of each expert. The diversity loss for experts can be formulated as:

$$\mathcal{L}_{\text{div}} = \sum_{i=2}^{M} \exp\left(-\text{KL}\left(\hat{\mathbf{y}}_{\text{b}}^{(i)}, \hat{\mathbf{y}}_{\text{b}}^{(i-1)}\right)\right). \tag{3}$$

Thus, using Eq. 3, we can regularize the diversity of the bias detection by each expert, allowing them to capture as many biases as possible. In this way, each expert can focus on different level features, and thus, different biases.

4.2.2 Counterfactual training

We obtain relatively accurate bias and target features after warming the model during the initial training. Counterfactual training is used to further separate target features from bias features. This approach is based on two counterfactual procedures. (i) When we change the sample's target features while keeping its bias features unchanged, the model's decision should be changed; (ii) When we keep its target features unchanged while changing the sample's bias features, the model should make the same decision for the changed features as for the original features. To leverage these two procedures, we first synthesize counterfactual features before conducting counterfactual inference using contrastive loss.

Synthesizing counterfactual features. We randomly sample a mini-batch of K samples to construct the counterfactual features. For the j^{th} sample, in the mini-batch, we first randomly select one bias feature and P target features from the other samples as follows: $\tilde{\mathcal{Z}}_{\text{b}}^{(i)} = \{\mathbf{z}_{\text{b}_q}^{(i)}\}$, and $\tilde{\mathcal{Z}}_{\text{d}}^{(i)} = \{\mathbf{z}_{\text{d}_l}^{(i)}\}_{l=1}^{P}$, where $q \neq j$, $l \neq j$, and $0 < q \leq K$.

Subsequently, the target feature $\mathbf{z}_{\text{d}_j}^{(i)}$ is paired with the selected bias feature to construct positive features $\mathcal{Z}_{\text{pos}}{}^{(i)} = \{[\mathbf{z}_{\text{d}_j}^{(i)}; \mathbf{z}_{\text{b}_q}^{(i)}]\}$ (Fig. 3b, right). Similarly, the bias feature $\mathbf{z}_{\text{b}_j}^{(i)}$ is paired with the other P target features to construct its negative features $\mathcal{Z}_{\text{neg}}{}^{(i)} = \{[\mathbf{z}_{\text{d}_l}^{(i)}; \mathbf{z}_{\text{b}_j}^{(i)}]\}_{l=1}^{P}$.

Counterfactual inference. Positive and negative features were fed into the debiased classifier $C_{\text{d}}^{(i)}$ and bias classifier $C_{\text{b}}^{(i)}$, respectively. We obtain a positive prediction $\mathcal{Y}_{\text{pos}}^{(i)} = \{\hat{\mathbf{y}}_{\text{pos}}^{(i)}\}$ and a set of negative predictions $\mathcal{Y}_{\text{neg}}^{(i)} = \{\hat{\mathbf{y}}_{\text{neg}_l}^{(i)}\}_{l=1}^{P}$ for the original result $\hat{\mathbf{y}}_{\text{d}}^{(i)}$. We then use the contrastive loss $\mathcal{L}_{\text{con}}$ for this counterfactual inference:

$$\mathcal{L}_{\text{con}} = \sum_{i=1}^{M} -\log \frac{\exp\left(-\text{dist}\left(\hat{\mathbf{y}}_{\text{d}}^{(i)}, \hat{\mathbf{y}}_{\text{pos}}^{(i)}\right)\right)}{\sum_{\mathbf{y}' \in \mathcal{Y}_{\text{neg}}^{(i)} \cup \{\hat{\mathbf{y}}_{\text{pos}}^{(i)}\}} \exp\left(-\text{dist}\left(\hat{\mathbf{y}}_{\text{d}}^{(i)}, \mathbf{y}'\right)\right)},$$

where $\text{dist}(\cdot, \cdot)$ denotes Euclidean distance. This encourages the model to group samples with identical target features into the same category, regardless of their bias features. Conversely, even if samples have the same bias features, they can be classified into different categories if they have different target features.

4.3. Mixture of biases-specific experts using adaptively gating

The final output $\hat{\mathbf{y}}_{\text{d}}$ of the model is obtained by combining the debiased classification results $\hat{\mathbf{y}}_{\text{d}}^{(i)}$ from each biases-specific expert through an gating module. The gating loss for this operation can be presented as: $\mathcal{L}_{\text{gate}} = \text{CE}\left(\hat{\mathbf{y}}_{\text{d}}, \mathbf{y}\right)$. $\hat{\mathbf{y}}_{\text{d}} = \sum_{i=1}^{M} \mathrm{p}^{(\mathrm{i})} \times \hat{\mathbf{y}}_{\text{d}}^{(\mathrm{i})}$, where $\mathrm{p}^{(\mathrm{i})}$ denotes the probability value assigned to the biased classification result $\hat{\mathbf{y}}_{\text{d}}^{(i)}$ of $E^{(i)}$; it is the softmax result from the gating module by taking all experts' debiased classification results as the input.

Figure 4: Examples from Modified IMDB dataset, the left two are annotated with *young*, but also with *female* and *wearing glasses*. The right two are annotated with *old*, but also with *male* and *not wearing glasses*.

We call this module "gating" derived from the "gating" in MoE [8], where it refers to the weighted inputs from the gating network followed by a softmax function.

The complete loss for updating the entire model is:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{gate}} + \mathcal{L}_{\text{div}} & \text{initial training} \\ \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{gate}} + \mathcal{L}_{\text{div}} + \beta \times \mathcal{L}_{\text{con}} & \text{counterfactual training} \end{cases} \tag{4}$$

where β balances $\mathcal{L}_{\text{con}}$ with other terms.

5. Experiments

5.1. Datasets

Biased MNIST [24] contains ten digits (*0 – 9*) as its target categories and seven biases: digit color, digit scale, digit position, type of background texture, background texture color, co-occurring letter, and letter color. There are 50000, 10000, 10000 images for training, validation, and testing.

BAR [18] consists of typical action-place pairs, like *climbing* and *rockwall* in the training set; and unseen samples beyond the settled pairs in the test set. There are six target actions in 1941 training and 654 test images.

Modified IMDB is our constructed dataset using IMDB face images [21], containing 20000 training, 1617 validation, and 1617 test images. The targets are *young* and *old*, and the biases are gender and wearing glasses (Fig. 4).

MIMIC-CXR + NIH was constructed by simulating the biases brought about by different data sources when collecting the datasets. We mixed the MIMIC-CXR [10] and NIH [27] datasets into a MIMIC-CXR + NIH dataset. The target categories are *no finding* and *pneumonia*, and the biases come from two data sources where the correlation between the target and biases is not tangible. It contains 8500 training, 500 validation, and 500 test images.

5.2. Implementation details

Model architecture. We employed feature extraction layers of ResNet-18 [6] as the backbone of the debiased encoder and bias encoder. Two convolutional and two linear layers were used to design classifiers for biases-specific experts, and one linear layer was used to construct the gating module.

Training procedure. Our PnD was built using PyTorch, and all the experiments were conducted on an NVIDIA RTX A4000 GPU. For input to the PnD model, all images were resized to 160 × 160 × 3 except for the BAR where they were randomly cropped to $224 \times 224 \times 3$ and horizontally flipped following [18].

6. Results and Analysis

We compared our model to ResNet-18 [6], LfF [18], DFA [13], OccamNet [24], DebiAN [16], and UBNet [9]. ResNet-18 was pretrained using ImageNet [4] and simply used cross-entropy as its loss function without any debiasing strategy. In all the experiments, we calculated the means and standard deviations of accuracies of the test set across three runs for all datasets. Unless otherwise specified, the bias ratio is 0.95 for all cases in the following subsections.

6.1. Comparisons against state-of-the-art

Overall comparisons. We report the accuracy scores of all compared methods in Tab. 1. We used the results with two different bias ratios for each dataset except for BAR, because almost all of its training images were biased, and no bias labels were provided. We selected a relatively large bias ratio and a small bias ratio for this set of experiments. Nevertheless, due to the limitations of the original dataset, we could not set a smaller bias ratio for the Modified IMDB.

We can see that PnD outperforms all methods on Biased MNIST and MIMIC-CXR + NIH. Meanwhile, for the BAR, our method achieved the second-best performance, which is comparable to the results of DebiAN [16]. BAR has only one type of bias for each target category, and the images in the training set are purely biased. In addition, the proposed framework requires unbiased data. Therefore, our accuracy score is slightly lower than that of DebiAN [16]. For the Modified IMDB, PnD contains the best or second-best scores. Because there are only two biases in this dataset and this classification task is relatively simpler than that of the Biased MNIST, ResNet-18 also works well on it, whereas the SOTAs are inferior to PnD and ResNet-18 on this dataset. We conclude that PnD performs best in agnostic biases mitigation owing to the mixture of biases-specific experts, especially in the presence of multiple biases. Even when the number of biases was small, its performance was comparable to that of the others.

Robustness to different numbers of bias types. To evaluate the performance of all methods under different numbers of biases, we synthesized multiple biased MNISTs with varying numbers of biases (ranging from 1 to 7) by gradually adding digit color, digit scale, digit position, texture, texture color, letter, and letter color following the data synthesis operation in [24] as shown in the supplement.

Table 1: Accuracy scores (%) on Biased MNIST, Modified IMDB, MIMIC-CXR + NIH, and BAR datasets with different bias ratios. We compare our proposed method with ResNet-18 and other SOTA methods. Our method is clearly far superior or close to other methods. The best results are highlighted in **blue**, and the second-best results are in **red**.

Method	Biased MNIST (7 biases) 0.75	Biased MNIST (7 biases) 0.95	BAR (1 bias)	Modified IMDB (2 biases) 0.95	Modified IMDB (2 biases) 0.99	MIMIC-CXR + NIH (1 bias) 0.80	MIMIC-CXR + NIH (1 bias) 0.95
ResNet-18 [6]	94.49 ± 0.15	53.86 ± 1.90	51.85 ± 5.92	**74.31** ± 0.44	**67.04** ± 2.07	64.53 ± 2.07	56.83 ± 2.07
LfF [18]	84.58 ± 2.46	35.16 ± 9.80	62.98 ± 2.76	63.22 ± 1.53	62.01 ± 1.58	55.40 ± 0.00	56.23 ± 0.67
DFA [13]	90.79 ± 0.14	44.52 ± 2.51	58.97 ± 1.28	64.19 ± 2.92	62.46 ± 0.71	52.67 ± 2.70	50.93 ± 0.58
OccamNet [24]	**96.06** ± 0.33	**66.85** ± 0.55	52.60 ± 1.90	68.17 ± 0.99	61.60 ± 1.07	61.93 ± 0.40	52.15 ± 0.35
DebiAN [16]	90.90 ± 1.36	46.52 ± 2.65	**69.88** ± 2.92	72.42 ± 0.33	65.99 ± 0.80	**67.40** ± 0.96	**60.00** ± 1.40
UBNet [9]	90.40 ± 0.05	54.31 ± 1.13	61.93 ± 0.46	70.62 ± 0.25	63.02 ± 0.21	66.00 ± 0.46	55.00 ± 0.17
PnD	**96.60** ± 0.22	**70.43** ± 0.74	**69.83** ± 2.09	**74.34** ± 0.22	**66.58** ± 0.26	**67.87** ± 0.91	**60.73** ± 0.87

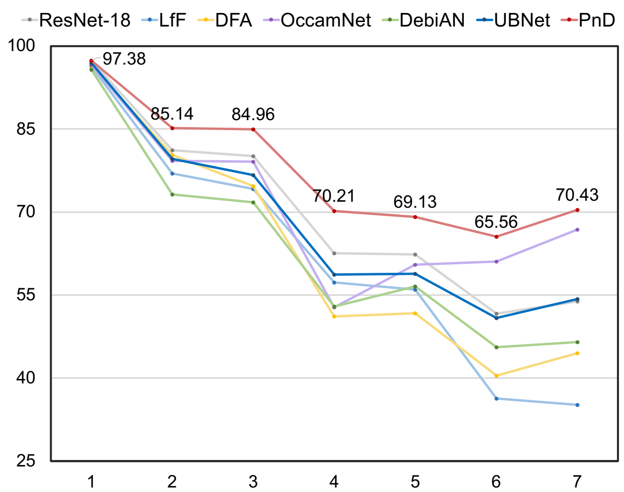

Figure 5: The accuracy scores (%) when changing the number of bias types from 1 to 7 in Biased MNIST. Regardless of the number of biases, PnD always achieves the best.

Table 2: Ablation study on Biased MNIST and MIMIC-CXR + NIH, including ablating the loss terms in PnD (3rd – 6th rows), the implementation strategies (7th – 8th rows), and training procedure (9th – 10th rows). The results reveal that each component in PnD is effective.

$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{gate}}$	$\mathcal{L}_{\text{div}}$	$\mathcal{L}_{\text{con}}$	Biased MNIST	MIMIC-CXR + NIH
✓				44.11 ± 0.76	54.30 ± 0.46
✓	✓			68.90 ± 0.38	57.87 ± 0.71
✓	✓	✓		69.88 ± 1.43	58.10 ± 0.99
✓	✓		✓	69.48 ± 1.78	59.20 ± 0.26
w/o concatenation				67.11 ± 0.90	58.57 ± 0.49
w/o adaptive gating				67.84 ± 0.38	58.13 ± 0.64
initial training only				69.03 ± 0.53	58.07 ± 1.27
counterfactual training only				69.61 ± 1.22	59.00 ± 0.56
PnD (full model)				70.43 ± 0.74	60.73 ± 0.87

Regardless of the number of biases, PnD always achieved the best performance (Fig. 5). When the number of biases was one (digit color), all methods achieved high scores. This is because, at this time, the digit only occupies a small area in the center of the images, resulting in difficulty in learning digit color features. Owing to the partition-and-debiasing strategy, our method does not eliminate its performance was as fast as those of the other methods after adding the second bias. Although our method suffers from a performance drop after the fourth bias, it still outperforms the other methods and remains nearly stable when additional biases are added.

6.2. Ablation study

Ablation study for different loss terms. The ablation study results on the biased MNIST and MIMIC-CXR + NIH datasets are shown in Tab. 2. We evaluated the impact of using multiple loss terms in Eq. 4 by dropping each loss term individually (3rd – 6th rows). Note that we drop each loss term in both training phases. For clarity, we only discuss the impact of the loss on the overall framework, not involving the analysis of the two training phases. The model with only $\mathcal{L}_{\text{cls}}$ (3rd row) (i.e., the model with only two encoders and two classifiers following the ends of the blocks) performed the worst. This is because it attempted to remove agnostic biases only once from the end of the network, as in previous studies, ignoring the fact that the number and type of agnostic biases are unknown. When other loss terms are gradually added, the performance improves. Particularly, the model with $\mathcal{L}_{\text{gate}}$ (4th row) boosts the performance significantly because we begin to process the agnostic biases according to the network depth. Moreover, the model with either $\mathcal{L}_{\text{div}}$ or $\mathcal{L}_{\text{con}}$ (5th and 6th rows) slightly improves the performance (less than 1%). When we used both $\mathcal{L}_{\text{div}}$ and $\mathcal{L}_{\text{con}}$ (11th row), the performance increased by 1.53% com-

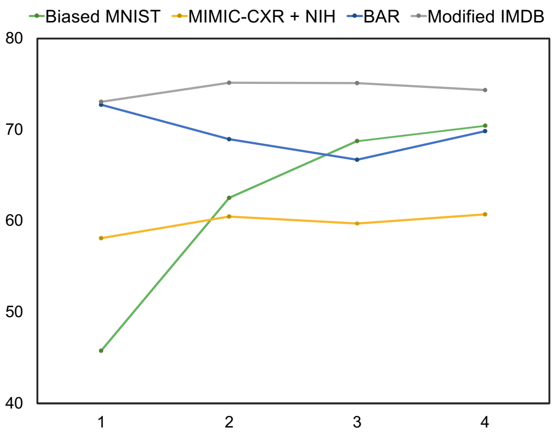

Figure 6: The accuracy scores (%) when changing the number of experts from 1 to 4 on Biased MNIST, Modified IMDB, MIMIC-CXR + NIH, and BAR datasets. In the case of multiple biases (Biased MNIST), the debiased classification accuracy is raised as the number of experts increases. But for two biases (Modified IMDB) or single bias (BAR and MIMIC-CXR + NIH), the number of experts does not affect the performance too much.

pared to the model with $\mathcal{L}_{\text{gate}}$. This is because $\mathcal{L}_{\text{div}}$ regularizes the diversity of each block in the bias encoder, which also increases the diversity of counterfactual features, thus improving the results. We conclude that the absence of any loss term reduces PnD's performance, indicating that all terms work properly and contribute to the final results.

Ablation study for different strategies used in our implementation. In PnD, we do not feed the bias and target features into the classifier separately for classification, but combine them and then classify them. The purpose, on the one hand, is to make the bias features a perturbation term on the target features to prevent overfitting, and on the other hand, to facilitate the synthesis of counterfactual features. We verified the results of a separate classification in our ablation experiments without concatenation (7th row), which showed significant decreases of 3.32% and 2.16%. For the adaptively gating operation, we demonstrated the effect of unweighted averaging (8th row), with a drop of 2.59% and 2.60% without adaptive gating.

Ablation study for two training phases. For a fair comparison, we still keep the same epochs. We can see that the two-stage training is better than only the first stage (9th row) or the second stage (10th row). The first training provides relatively purer target and bias features for the second stage, whereas the second stage further disentangles these two features via counterfactual inference. Consequently, performance improves when the two stages work together.

Ablation study for multiple biases-specific experts. We individually removed the expert modules inserted into the shallowest block of ResNet-18, to obtain the classification results when the number of experts ranged from 1 to 4 (Fig. 6). When multiple biases exist, we can see that the greater the number of blocks covered by the expert, the better the debiased classification effect. When the number of biases is two or one, the performance remains almost stable. This also confirmed the conclusion from the exploratory experiment and the sufficiency of our strategy.

Table 3: Further ablation study for multiple biases-specific experts on Biased MNIST and MIMIC-CXR + NIH. The accuracy (%) results reveal that inserting multiple biases-specific experts at different depths of PnD is effective.

Method	Biased MNIST	MIMIC-CXR + NIH
MoE on the last block	52.06 $\pm$1.10	58.83 $\pm$ 0.42
MoE on all blocks w/o other debiasing strategies	67.49 $\pm$1.41	58.47 $\pm$ 0.45
PnD	70.43 $\pm$ 0.74	60.73 $\pm$ 0.87

We further evaluate the performance of multiple experts ensemble in Tab. 3. We give the results of ensemble multiple biases-specific experts only in the last block (2nd row). When a single bias (MIMIC-CXR + NIH) exists, its performance drops by 1.9% compared to the PnD. However, when multiple biases (Biased MNIST), the performance decreases significantly by 18.37%. It fully illustrates the effectiveness of our idea that we should remove multiple biases from different depths in the network. In order to distinguish the effect of MoE and other debiasing strategies, we added an additional set of experiments, where we only keep the classification loss without weight for samples and the gating loss. The results show that the MoE strategy can also achieve relatively great results (3rd row). This is because we consider the features from the network in a depth-by-depth manner. The network focuses on more diverse regions, thus outputting a prediction result that is not limited to a single feature, which may be bias.

6.3. Detailed analysis

More results on real-world dataset. We additionally evaluate the performance on CelebA [17], a real-world dataset, in Tab. 4. For wearing lipstick or not classification, we show the accuracy scores of worst group and all groups in four bias attributes (2nd – 5th cols). From this table, we can see that the performance of PnD outperforms other methods in worst groups and all groups of almost all bias attributes. It demonstrates the advantage of our method in removing agnostic biases for real-world dataset.

Visualization of learned target and bias features. We visualized the region of interest of each expert using Grad-CAM [23] (Fig. 7) to qualitatively verify the debiased classification (upper) and bias detection (lower) performances of each block in PnD. In the debiased classification, all ex-

Table 4: Accuracy scores on worst groups and all groups for wearing lipstick or not classification on CelebA, where bias attributes are attractive or not, heavy makeup or not, high cheekbone or not, and gender. In the last column, we average the results in four bias attributes. The best results are highlighted in **blue**, and the second best results are in **red**.

Method	Attractive or not		Heavy makeup or not		High cheekbones or not		Gender		Average	
	Worst group	All groups	Worst group	All groups	Worst group	All groups	Worst group	All groups	Worst group	All groups
ResNet-18 [6]	**85.92** ± 0.19	**91.53** ± 0.29	**26.23** ± 1.73	**74.77** ± 0.07	**91.54** ± 0.48	**93.32** ± 0.05	25.96 ± 4.08	71.54 ± 0.99	57.41 ± 1.62	82.79 ± 0.35
DebiAN [16]	85.82 ± 1.56	91.49 ± 0.14	24.51 ± 6.24	74.37 ± 0.55	90.27 ± 2.14	93.03 ± 0.26	**32.05** ± 1.81	**73.41** ± 3.33	**58.16** ± 2.94	**83.07** ± 1.07
PAD	**87.33** ± 1.94	**91.77** ± 0.02	**27.94** ± 6.32	**75.08** ± 1.17	**92.00** ± 1.29	**93.37** ± 0.08	**32.69** ± 5.44	**73.39** ± 0.51	**60.00** ± 1.53	**83.40** ± 0.44

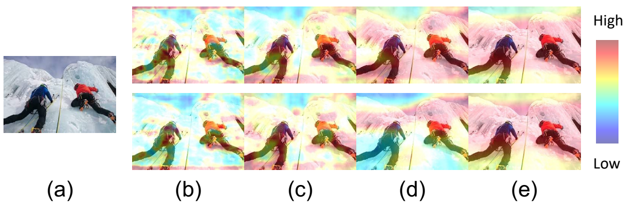

Figure 7: Regions of interest (ROIs) for biases-specific experts of our PnD in the debiased (upper) and bias (lower) encoders, when conducting action classification in the test set of BAR. (a) are original images, (b)-(e) are their saliency maps generated using Grad-CAM, from the first expert to the fourth expert. The ROIs for debiased classification and bias detection are changing as the network gets deeper, and there are also significant differences between the two tasks.

Table 5: Accuracy scores (%) for debiased classification results $\hat{\mathbf{y}}_{\mathrm{d}}^{(i)}$ of each expert (from $E^{(1)}$ to $E^{(4)}$), and the probability $\mathrm{p}^{(i)}$ (in parentheses) assigned by the gating module on Biased MNIST, Modified IMDB, MIMIC-CXR + NIH, and BAR datasets. We can see that each dataset has different trends in classification accuracy across experts.

	$E^{(1)}$	$E^{(2)}$	$E^{(3)}$	$E^{(4)}$	Final
Biased MNIST	48.47 (0.07)	68.75 (0.15)	70.66 (0.46)	68.19 (0.32)	70.43
BAR	33.44 (0.10)	49.54 (0.19)	61.57 (0.18)	69.83 (0.53)	69.83
Modified IMDB	68.13 (0.04)	72.89 (0.35)	74.71 (0.38)	74.25 (0.22)	74.34
MIMIC-CXR+NIH	51.77 (0.11)	57.90 (0.30)	60.70 (0.23)	61.00 (0.36)	60.73

perts were able to focus on the target region (climbing). Meanwhile, in bias detection, each expert could handle bias features differently; for instance, the first expert focuses on background texture, and the third expert concentrates on snowy slopes. We conclude that the two encoders can capture target and bias features properly and independently.

Analysis on the mixed debiased classification of experts. To analyze the effectiveness of mixing different expert results using adaptively gating, we checked the output from each expert and the probability assigned by the gating module. Tab. 5 shows that, across these four datasets, the highest debiased classification accuracy scores were located at 3rd or 4th expert due to the different complexities of biases. Additionally, the results of the expert modules slightly exceeded the final results coordinated by multiple experts. This is reasonable because shallow blocks may contain few target features, resulting in poor performance of the corresponding experts, and thus degrading the final results. This occurs particularly when the probability $\mathrm{p}^{(i)}$ assigned to the best expert is relatively low. However, we cannot remove shallow experts directly (see the results of reducing the number of experts in Fig. 6). We may be able to enhance the performance by selecting the output of the expert that has the highest $\mathrm{p}^{(i)}$ during testing.

Limitations. We employed multiple expert modules, which inevitably increased the number of network parameters. One potential solution is to increase the sparsity of expert networks [5]. Furthermore, as discussed in [15], removing multiple unknown biases without an inductive bias is difficult. Due to the requirement of unbiased training data, PnD does not perform as well on fully biased datasets such as BAR. We will examine these issues further in the future.

Societal impacts. This study introduces a more realistic bias scenario and provides a simple yet effective approach to ensure that deep learning-based decision processes are not biased toward agnostic attributes in the data. We believe that the proposed approach will encourage the development of more trustworthy AI applications. For example, it can increase racial and gender equity in face recognition systems by protecting minority populations from systemic biases.

7. Conclusion

Existing bias mitigation methods struggle to deal with multiple unknown biases in real-world scenarios. To address these limitations, we presented a novel bias scenario, namely, agnostic biases mitigation. First, we investigated our hypothesis that different bias features would cluster at different depths in a network. We then proposed a PnD method to address the new scenario by dividing the bias space into multiple subspaces across network depths and removing them using a mixture of biases-specific experts. Extensive experiments on both public and our constructed datasets demonstrated PnD's excellent performance.

Acknowledgement. This work was supported by JST SPRING Grant Number JPMJSP2108, Institute for AI and Beyond of the University of Tokyo, JSPS KAKENHI Grant Numbers JP23H03449, JP23KJ0404, and JP22K17947.

References

[1] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[2] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proc. European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[3] Saeid Asgari, Aliasghar Khani, Fereshte Khani, Ali Gholami, Linh Tran, Ali Mahdavi-Amiri, and Ghassan Hamarneh. Masktune: Mitigating spurious correlations by forcing to explore. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2022. 2

[4] 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. 6

[5] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23(120):1–39, 2022. 9

[6] 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)*, 2016. 3, 4, 6, 7, 9

[7] Inwoo Hwang, Sangjun Lee, Yunhyeok Kwak, Seong Joon Oh, Damien Teney, Jin-Hwa Kim, and Byoung-Tak Zhang. Selecmix: Debiased learning by contradicting-pair sampling. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2022. 2

[8] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 2, 6

[9] Myeongho Jeon, Daekyung Kim, Woochul Lee, Myungjoo Kang, and Joonseok Lee. A conservative approach for unbiased learning on unknown biases. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6, 7

[10] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):1–8, 2019. 6

[11] 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)*, 2019. 2

[12] 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)*, 2021. 5

[13] 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)*, 2021. 1, 2, 5, 6, 7

[14] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[15] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 9

[16] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and mitigate unknown biases with debiasing alternate networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 2, 6, 7, 9

[17] 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)*, 2015. 2, 8

[18] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 2, 5, 6, 7

[19] Jiaxin Qi, Kaihua Tang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Class is invariant to context and vice versa: on learning invariance for out-of-distribution generalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 1

[20] Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[21] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, 126(2-4):144–157, 2018. 6

[22] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2020. 2

[23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 8

[24] Robik Shrestha, Kushal Kafle, and Christopher Kanan. Occamnets: Mitigating dataset bias by favoring simpler hypotheses. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 6, 7

[25] 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)*, 2021. 2

[26] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[27] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[28] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: Regularized dropout for neural networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 5

[29] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[30] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. 2018. 5

[31] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017. 1

[32] Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. Moebert: from bert to mixture-of-experts via importance-guided adaptation. In *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2022. 3