Training Debiased Subnetworks with Contrastive Weight Pruning

Geon Yeong Park[1] Sangmin Lee[2] Sang Wan Lee[1*] Jong Chul Ye[1,2,3*]
[1]Bio and Brain Engineering, [2]Mathematical Sciences, [3]Kim Jaechul Graduate School of AI
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{pky3436, leeleesang, sangwan, jong.ye}@kaist.ac.kr

Abstract

Neural networks are often biased to spuriously correlated features that provide misleading statistical evidence that does not generalize. This raises an interesting question: "Does an optimal unbiased functional subnetwork exist in a severely biased network? If so, how to extract such subnetwork?" While empirical evidence has been accumulated about the existence of such unbiased subnetworks, these observations are mainly based on the guidance of ground-truth unbiased samples. Thus, it is unexplored how to discover the optimal subnetworks with biased training datasets in practice. To address this, here we first present our theoretical insight that alerts potential limitations of existing algorithms in exploring unbiased subnetworks in the presence of strong spurious correlations. We then further elucidate the importance of bias-conflicting samples on structure learning. Motivated by these observations, we propose a Debiased Contrastive Weight Pruning (DCWP) algorithm, which probes unbiased subnetworks without expensive group annotations. Experimental results demonstrate that our approach significantly outperforms state-of-the-art debiasing methods despite its considerable reduction in the number of parameters.

1. Introduction

While deep neural networks have made substantial progress in solving challenging tasks, they often undesirably rely on spuriously correlated features or dataset bias, if present, which is considered one of the major hurdles in deploying models in real-world applications. For example, consider recognizing desert foxes and cats from natural images. If the background scene (e.g., a desert) is spuriously correlated to the type of animal, the neural networks might use the background information as a shortcut to classification, resulting in performance degradation in different backgrounds (e.g., a desert fox in the house).

To investigate the origin of the spurious correlations, this paper considers shortcut learning as a fundamental architectural design issue of neural networks. Specifically, if any available information channels in deep networks' structure could transmit the information of spuriously correlated features (*spurious features* from now on), networks would exploit those features as long as they are sufficiently predictive. It naturally follows that pruning weights on spurious features can purify the biased latent representations, thereby improving performances on bias-conflicting samples[1]. We conjecture that this neural pruning may improve the generalization of the network in a way that reduces the effective dimension of spurious features, considering that the failure of Out-of-Distribution (OOD) generalization may arise due to high-dimensional spurious features [24, 31].

Recently, Zhang et al. [33] has empirically demonstrated the existence of subnetworks that are less susceptible to spurious features. Based on the modular property of neural networks [5], they prune out weights that are closely related to the spurious attributes. While [33] affords us valuable insights on the importance of neural architectures, the study has limitation in that such neural pruning requires sufficient number of ground-truth bias-conflicting samples. Thus, *how to discover the optimal subnetworks in practice when the dataset is highly biased?*

To address this, we first present a simple theoretical observation that reveals the limitations of existing substructure probing methods in searching unbiased subnetworks. Specifically, we reveal that there exists an unavoidable generalization gap in the subnetworks obtained by standard pruning algorithms in the presence of strong spurious correlations. Our analysis also shows that trained models may inevitably rely on the spuriously correlated features in a practical training setting with finite training time and a number of samples.

In addition, we show that sampling more bias-conflicting data makes it possible to identify spurious weights. Specifically, bias-conflicting samples require that the weights as-

[1]The *bias-aligned* samples refer to data with a strong correlation between (potentially latent) spurious features and target labels (e.g., cat in the house). The *bias-conflicting* samples refer to the opposite cases where spurious correlations do not exist (e.g., cat in the desert).

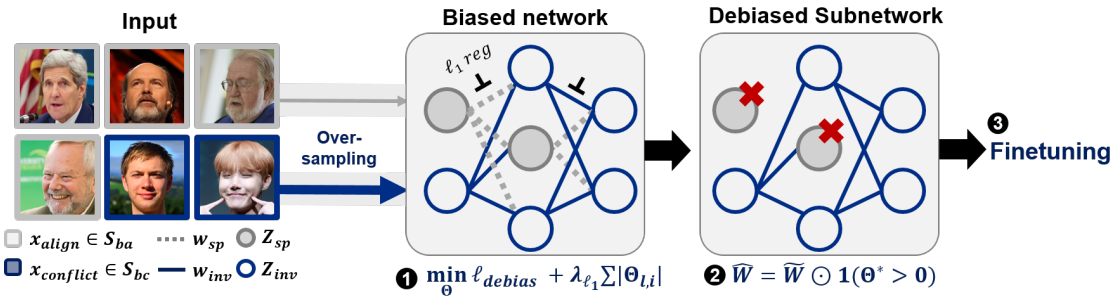


Figure 1. **Concept**: We demonstrate an inevitable generalization gap of subnetworks obtained by standard pruning methods including [33]. Based on these observations, we design a novel subnetwork probing framework by fully exploiting unbiased samples.

sociated with spurious features should be pruned out as the spurious features do not help predict bias-conflicting samples. Our theoretical observations suggest that balancing the ratio between the number of bias-aligned and bias-conflicting samples is crucial in finding the optimal unbiased subnetworks.

In practice, the dataset may severely lack diversity for bias-conflicting samples due to the potential pitfalls in data collection protocols or human prejudice. Since it is often highly laborious to supplement enough bias-conflicting samples, we propose a novel debiasing scheme called Debiased Contrastive Weight Pruning (DCWP) that uses the oversampled bias-conflicting data to search unbiased subnetworks.

As shown in Fig. 1, DCWP is comprised of two stages: (1) identifying the bias-conflicting samples without expensive annotations on spuriously correlated attributes, and (2) training the pruning parameters to obtain weight pruning masks with the sparsity constraint and *debiased* loss function. Here, the debiased loss includes a weighted cross-entropy loss for the identified bias-conflicting samples and an alignment loss to further reduce the geometrical alignment gap between bias-aligned and bias-conflicting samples within each class.

We demonstrate that DCWP consistently outperforms state-of-the-art debiasing methods across various biased datasets, including the Color-MNIST [21, 25], Corrupted CIFAR-10 [13], Biased FFHQ [19] and CelebA [23], even without direct supervision on the bias type. Our approach improves the accuracy on the unbiased evaluation dataset by $86.74\% \rightarrow 93.41\%$, $27.86\% \rightarrow 35.90\%$ on Colored-MNIST and Corrupted CIFAR-10 compared to the second best model, respectively, even when 99.5% of samples are bias-aligned.

2. Related works

Spurious correlations. A series of empirical works have shown that the deep networks often find shortcut solutions relying on spuriously correlated attributes, such as the texture of image [10], language biases [12], or sensitive variables such as ethnicity or gender [7, 26]. Such behavior is of practical concern because it deteriorates the reliability of deep networks in sensitive applications like healthcare, finance, and legal services [4].

Debiasing frameworks. Recent studies to train a debiased network robust to spurious correlations can be roughly categorized into approaches (1) leveraging annotations of spurious attributes, i.e., bias label [27, 32], (2) presuming specific type of bias, e.g., texture [1, 9] or (3) without using explicit kinds of supervisions on dataset bias [20, 25]. The authors in [15, 27] optimize the worst-group error by using training group information. For practical implementation, reweighting or subsampling protocols are often used with increased model regularization [28]. Liu et al.; Sohoni et al. [22, 29] extend these approaches to the settings without expensive group annotations. Goel et al.; Kim et al. [11, 19] provide bias-tailored augmentations to balance the majority and minority groups. In particular, these approaches have mainly focused on better approximation and regularization of worst-group error combined with advanced data sampling, augmentation, or retraining strategies.

Studying impacts of neural architectures. Recently, the effects of deep neural network architecture on generalization performance have been explored. Diffenderfer et al. [6] employ recently advanced lottery-ticket-style pruning algorithms [8] to design the compact and robust network architecture. Bai et al. [2] directly optimize the neural architecture in terms of accuracy on OOD samples. Zhang et al. [33] demonstrate the effectiveness of pruning weights on spurious attributes, but the solution for discriminating such spurious weights lacks robust theoretical justifications, resulting in marginal performance gains. To fully resolve the above issues, we carry out a theoretical case study, and build a novel pruning algorithm that distills the representations to be independent of the spurious attributes.

3. Theoretical insights

3.1. Problem setup

Consider a supervised setting of predicting labels $Y \in \mathcal{Y}$ from input samples $X \in \mathcal{X}$ by a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$. Following [33], let $(X^e, Y^e) \sim P^e$, where $X^e \in \mathcal{X}$ and $Y^e \in \mathcal{Y}$ refer to the input random variable and the corresponding label, respectively, and $e \in \mathcal{E} = \{1, 2, \dots E\}$ denotes the index of environment, P^e is the corresponding distribution, and the set $\mathcal{E}$ corresponds to every possible environments. We further assume that $\mathcal{E}$ is divided into training environmments $\mathcal{E}_{train}$ and unseen test environments $\mathcal{E}_{test}$, i.e. $\mathcal{E} = \mathcal{E}_{train} \cup \mathcal{E}_{test}$.

For a given a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathbb{R}^+$, the standard training protocol for the empirical risk minimization (ERM) is to minimize the expected loss with a training environment $e \in \mathcal{E}_{train}$:

$$\hat{\theta}_{ERM} = \arg\min_\theta \mathbb{E}_{(X^e, Y^e) \sim \hat{P}^e} \left[\ell(X^e, Y^e; \theta) \right], \quad (1)$$

where $\hat{P}^e$ is the empirical distribution over the training data. Our goal is to learn a model with good performance on OOD samples of $e \in \mathcal{E}_{test}$.

3.2. Motivating example

We conjecture that neural networks trained by ERM indiscriminately rely on predictive features, including those spuriously correlated ones [31].

To verify this conjecture, we present a simple binary-classification example $(\boldsymbol{X}^e, Y^e) \sim P^e$, where $Y^e \in \mathcal{Y} = \{-1, 1\}$ represents the corresponding target label, and a sample $\boldsymbol{X}^e \in \mathcal{X} = \{-1, 1\}^{D+1} \in \mathbb{R}^{D+1}$ is constituted with both the invariant feature $Z^e_{inv} \in \{-1, 1\}$ and spurious features $\boldsymbol{Z}^e_{sp} \in \{-1, 1\}^D$, i.e. $\boldsymbol{X}^e = (Z^e_{inv}, \boldsymbol{Z}^e_{sp})$. Suppose, furthermore, $Z^e_{sp,i}$ denote the i-th spurious feature component of $\boldsymbol{Z}^e_{sp}$. Note that we assume $D \gg 1$ to simulate the model heavily relies on spurious features $\boldsymbol{Z}^e_{sp}$ [24, 33].

We consider the setting where the training environment $e \in \mathcal{E}_{train}$ is highly biased. In other words, we suppose that $Z^e_{inv} = Y^e$, and each of the i-th spurious feature component $Z^e_{sp,i}$ is independent and identically distributed (i.i.d) Bernoulli variable: i.e. $Z^e_{sp,i}$ independently takes a value equal to Y^e with a probability p^e and $-Y^e$ with a probability $1 - p^e$, where $p^e \in (0.5, 1], \forall e \in \mathcal{E}_{train}$. Note that $p^e \to 1$ as the environment is severely biased. A test environment $e \in \mathcal{E}_{test}$ is assumed to have $p^e = 0.5$, which implies that the spurious feature is totally independent with Y^e. Then we introduce a linear classifier f parameterized by a weight vector $\boldsymbol{w} = (w_{inv}, \boldsymbol{w}_{sp}) \in \mathbb{R}^{D+1}$, where $w_{inv} \in \mathbb{R}$ and $\boldsymbol{w}_{sp} \in \mathbb{R}^D$. In this example, we consider a class of pretrained classifiers parameterized by $\tilde{\boldsymbol{w}}(t) = \big(\tilde{w}_{inv}(t), \tilde{w}_{sp,1}(t), \dots, \tilde{w}_{sp,D}(t)\big)$, where $t < T$ is a finite pretraining time for some sufficiently large T. Time t will be often omitted in notations for simplicity.

Our goal is to obtain the optimal sparse classifier with a highly biased training dataset. To achieve this, we introduce a binary weight pruning mask $\boldsymbol{m}$ as $\boldsymbol{m} = (m_{inv}, \boldsymbol{m}_{sp}) \in \{0, 1\}^{D+1}$ for the pretrained weights, which is a significant departure from the theoretical setting in [33]. Specifically, let $m_{inv} \sim Bern(\pi_{inv})$, where π_{inv} and $1 - \pi_{inv}$ represents the probability of preserving (i.e. $m_{inv} = 1$) and pruning out (i.e. $m_{inv} = 0$), respectively. Similarly, let $m_{sp,i} \sim Bern(\pi_{sp,i}), \forall i$. Then, our optimization goal is to estimate the pruning probability parameter $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{D+1}) = (\pi_{inv}, \pi_{sp,1}, \dots, \pi_{sp,D})$, where $\boldsymbol{m} \sim P(\boldsymbol{\pi})$ is a mask sampled with probability parameters $\boldsymbol{\pi}$. Accordingly, our main loss function for the pruning parameters given the environment e can be defined as follows:

$$\begin{aligned} \ell^e(\boldsymbol{\pi}) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, Y^e, \boldsymbol{m}}[1 - Y^e \hat{Y}^e] \\ &= \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, Y^e, \boldsymbol{m}} \left[1 - Y^e \cdot \operatorname{sgn}\left(\tilde{\boldsymbol{w}}^T (\boldsymbol{X}^e \odot \boldsymbol{m}) \right) \right], \end{aligned} \quad (2)$$

where $\hat{Y}^e$ is the prediction of binary classifier, $\tilde{\boldsymbol{w}}$ is the pretrained weight vector, $\operatorname{sgn}(\cdot)$ represents the sign function, and $\odot$ represents element-wise product.

We first derive the upper-bound of the training loss $\ell^e(\boldsymbol{\pi})$ to illustrate the difficulty of learning optimal pruning parameters in a biased data setting. The proof can be found in Supplementary Material.

Theorem 1. *(Training and test bound) Assume that $p^e > 1/2$ in the biased training environment $e \in \mathcal{E}_{train}$. Define $\tilde{\boldsymbol{w}}(t)$ as weights pretrained for a finite time $t < T$. Then the upper bound of the error of training environment w.r.t. pruning parameters $\boldsymbol{\pi}$ is given as:*

$$\ell^e(\boldsymbol{\pi}) \le 2 \exp\left(- \frac{2\big(\pi_{inv} + (2p^e - 1)\sum_{i=1}^D \alpha_i(t)\pi_{sp,i}\big)^2}{4\sum_{i=1}^D \alpha_i(t)^2 + 1} \right), \quad (3)$$

where the weight ratio $\alpha_i(t) = \tilde{w}_{sp,i}(t)/\tilde{w}_{inv}(t)$ is bounded below some positive constant. Given a test environment $e \in \mathcal{E}_{test}$ with $p^e = \frac{1}{2}$, the upper bound of the error of test environment w.r.t. $\boldsymbol{\pi}$ is given as:

$$\ell^e(\boldsymbol{\pi}) \le 2 \exp\Big(- \frac{2\pi_{inv}^2}{4\sum_{i=1}^D \alpha_i(t)^2 + 1} \Big), \quad (4)$$

which implies that there is an unavoidable gap between training bound and test bound.

The detailed proof of Theorem 1 is provided in the supplementary material. This mismatch of the bounds is attributed to the contribution of $\pi_{sp,i}$ on the training bound (3). Intuitively, the networks prefer to *preserve both* $\tilde{w}_{inv}$ and $\tilde{w}_{sp,i}$ in the presence of strong spurious correlations due

to the inherent sensitivity of ERM to all kinds of predictive features [16, 31]. This behavior is directly reflected in the training bound, where increasing either π_{inv} or $\pi_{sp,i}$, i.e., the probability of preserving weights, decreases the training bound. This inertia of spurious weights may prevent themselves from being primarily pruned against the sparsity constraint.

We note that the unintended reliance on spurious features is fundamentally rooted to the positivity of the weight ratio $\alpha_i(t)$. In the proof of Theorem 1 in Supplementary Material, we show some intriguing properties of $\alpha_i(t)$: (**1**) If infinitely many data and sufficient training time is provided, the gradient flow converges to the optimal solution which is invariant to $\boldsymbol{Z}^e_{sp}$, i.e., $\alpha_i(t) \to 0$. In this ideal situation, the gap between training and test bound is closed, thereby guaranteeing generalizations of obtained subnetworks. (**2**) However, given a finite time $t < T$ with a strongly biased dataset in practice, $\alpha_i(t)$ is bounded below by some positive constant, resulting in an inevitable generalization gap.

Theorem 1 implies that the classifier may preserve spurious weights due to the lack of bias-conflicting samples, which serve as counterexamples that spurious features themselves fail to explain. It motivates us to analyze the training bound in another environment η where we can systematically augment bias-conflicting samples. Specifically, consider $\boldsymbol{X}^\eta = (Z^\eta_{inv}, \boldsymbol{Z}^\eta_{sp})$, where $Z^\eta_{inv} = Y^\eta$ and *mixture distribution* of $\boldsymbol{Z}^\eta_{sp}$ given $Y^\eta = y$ is defined in an element wise as follows:

$$P^\eta_{mix}(Z^\eta_{sp,i} \mid Y^\eta = y) = \phi P^\eta_{debias}(Z^\eta_{sp,i} \mid Y^\eta = y) + (1-\phi) P^\eta_{bias}(Z^\eta_{sp,i} \mid Y^\eta = y), \tag{5}$$

where ϕ is a scalar mixture weight,

$$P^\eta_{debias}(Z^\eta_{sp,i} \mid Y^\eta = y) = \begin{cases} 1, & \text{if } Z^\eta_{sp,i} = -y \\ 0, & \text{if } Z^\eta_{sp,i} = y \end{cases} \tag{6}$$

is a debiasing distribution to weaken the correlation between Y^η and $Z^\eta_{sp,i}$ by setting the value of $Z^\eta_{sp,i}$ as $-Y^\eta$, and

$$P^\eta_{bias}(Z^\eta_{sp,i} \mid Y^\eta = y) = \begin{cases} p^\eta, & \text{if } Z^\eta_{sp,i} = y \\ 1-p^\eta, & \text{if } Z^\eta_{sp,i} = -y \end{cases} \tag{7}$$

is a biased distribution similarly defined in the previous environment $e \in \mathcal{E}_{train}$. Given this new environment η, the degree of spurious correlations can be controlled by ϕ. This leads to a training bound as follow:

Theorem 2. *(Training bound with the mixture distribution) Assume that the defined mixture distribution P^η_{mix} is biased, i.e., for all $i \in \{1, \dots, D\}$,*

$$P^\eta_{mix}(Z^\eta_{sp,i} = -y \mid Y^e = y) \le P^\eta_{mix}(Z^\eta_{sp,i} = y \mid Y^\eta = y). \tag{8}$$

Then, ϕ satisfies $0 \le \phi \le 1 - \frac{1}{2p^\eta}$. Then the upper bound of the error of training environment η w.r.t. the pruning parameters is given by

$$\ell^\eta(\boldsymbol{\pi}) \le 2\exp\left(-\frac{2(\pi_{inv} + (2p^\eta(1-\phi)-1)\sum_{i=1}^D \alpha_i(t)\pi_{sp,i})^2}{4\sum_{i=1}^D \alpha_i(t)^2 + 1}\right). \tag{9}$$

Furthermore, when $\phi = 1 - \frac{1}{2p^\eta}$, the mixture distribution is perfectly debiased, and we have

$$\ell^\eta(\boldsymbol{\pi}) \le 2\exp\Big(-\frac{2\pi^2_{inv}}{4\sum_{i=1}^D \alpha_i(t)^2 + 1}\Big), \tag{10}$$

which is equivalent to the test bound in (4).

The detailed proof is provided in the supplementary material. Our new training bound (9) suggests that the significance of $\pi_{sp,i}$ on training bound decreases as ϕ progressively increases, and at the extreme end with $\phi = 1 - \frac{1}{2p^\eta}$, it can be easily shown that $P^\eta_{mix}(Z^\eta_{sp,i} \mid Y^\eta = y) = \frac{1}{2}$ for both $y = 1$ and $y = -1$ so that $Z^\eta_{sp,i}$ turns out to be random. In other words, by plugging $\phi = 1 - \frac{1}{2p^\eta}$ into (9), we can minimize the gap between training and test error bound, which guarantees the improved OOD generalization.

4. Debiased Contrastive Weight Pruning

Our theoretical observations elucidate the importance of balancing between the bias-aligned and bias-conflicting samples in discovering the optimal unbiased subnetworks structure. While the true analytical form of the debiasing distribution is unknown in practice, we aim to approximate such unknown distribution with existing bias-conflicting samples and simulate the mixture distribution P^η_{mix} with modifying sampling strategy. To this end, we propose a Debiased Contrastive Weight Pruning (DCWP) algorithms that learn the unbiased subnetworks structure from the original full-size network.

Consider a L layer neural networks as a function $f_{\boldsymbol{W}} : \mathcal{X} \to \mathbb{R}^C$ parameterized by weights $\boldsymbol{W} = \{\boldsymbol{W}_1, \dots, \boldsymbol{W}_L\}$, where $C = |\mathcal{Y}|$ is the number of classes. Analogous to the earlier works on pruning, we introduce binary weight pruning masks $\boldsymbol{m} = \{\boldsymbol{m}_1, \dots, \boldsymbol{m}_L\}$ to model the subnetworks as $f(\cdot; \boldsymbol{m}_1 \odot \boldsymbol{W}_1, \dots, \boldsymbol{m}_L \odot \boldsymbol{W}_L)$. We denote such subnetworks as $f_{\boldsymbol{m} \odot \boldsymbol{W}}$ for the notational simplicity. We treat each entry of $\boldsymbol{m}_l$ as an independent Bernoulli variable, and model their logits as our new pruning parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \dots, \boldsymbol{\Theta}_L\}$ where $\boldsymbol{\Theta}_l \in \mathbb{R}^{n_l}$ and n_l represents the dimensionality of the l-th layer weights $\boldsymbol{W}_l$. Then $\pi_{l,i} = \sigma(\Theta_{l,i})$ denotes the probability of preserving the i-th weight of l-th layer $\boldsymbol{W}_{l,i}$ where σ refers to a sigmoid

function. To enable the end-to-end training, the Gumbel-softmax trick [17] for sampling masks together with ℓ_1 regularization term of $\mathbf{\Theta}$ is adopted as a sparsity constraint. With a slight abuse of notations, $\boldsymbol{m} \sim G(\mathbf{\Theta})$ denotes a set of masks sampled with logits $\mathbf{\Theta}$ by applying Gumbel-softmax trick.

Then our main optimization problem is defined as follows:

$$\min_{\mathbf{\Theta}} \ell_{debias}\Big(\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \mathbf{\Theta}\Big) + \lambda_{\ell_1} \sum_{l,i} |\Theta_{l,i}|, \quad (11)$$

where S denotes the index set of whole training samples, $\lambda_{\ell_1} > 0$ is a Lagrangian multiplier, $\tilde{\boldsymbol{W}}$ represents the pretrained weights and ℓ_{debias} is our main objective which will be illustrated later. Note that we freeze the pretrained weights $\tilde{\boldsymbol{W}}$ during training pruning parameters Θ. We interchangeably use $\ell_{debias}\Big(\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|S|}; \Theta\Big)$ and $\ell_{debias}\big(S; \Theta\big)$ in the rest of the paper. For comparison with our formulation, we recast the optimization problem of [33] with our notations as follows:

$$\min_{\mathbf{\Theta}} \ell\Big(\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \mathbf{\Theta}\Big) + \lambda_{\ell_1} \sum_{l,i} |\Theta_{l,i}|, \quad (12)$$

where [33] uses the cross entropy (CE) loss function for ℓ.

Bias-conflicting sample mining In the first stage, we identify bias-conflicting training samples which empower functional modular probing. Specifically, we train a bias-capturing model and treat an error set S_{bc} of the index of misclassified training samples as bias-conflicting sample proxies. Our framework is broadly compatible with various bias-capturing models, where we mainly leverage the ERM model trained with generalized cross entropy (GCE) loss [35]:

$$\ell_{GCE}(x_i, y_i; \boldsymbol{W}_B) = \frac{1 - p_{y_i}(x_i; \boldsymbol{W}_B)^q}{q}, \quad (13)$$

where $q \in (0, 1]$ is a hyperparameter controlling the degree of bias amplification, $\boldsymbol{W}_B$ is the parameters of the bias-capturing model, and $p_{y_i}(x_i; \boldsymbol{W}_B)$ is a softmax output value of the bias-capturing model assigned to the target label y_i. Compared to the CE loss, the gradient of the GCE loss up-weights the samples with a high probability of predicting the correct target, amplifying the network bias by putting more emphasis on easy-to-predict samples [25].

To preclude the possibility that the generalization performance of DCWP is highly dependent on the behavior of the bias-capturing model, we demonstrate in Section 5 that DCWP is reasonably robust to the degradation of accuracy on capturing bias-conflicting samples. Details about the bias-capturing model and simulation settings are presented in the supplementary material.

Upweighting Bias-conflicting samples After mining the index set of bias-conflicting sample proxies S_{bc}, we treat $S_{ba} = S \setminus S_{bc}$ as the index set of majority bias-aligned samples. Then we calculate the weighted cross entropy (WCE) loss $\ell_{WCE}\big(\{x_i, y_i\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \Theta\big)$ as follows:

$$\ell_{WCE}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big) := \mathbb{E}_{\boldsymbol{m}\sim G(\Theta)}\big[\lambda_{up}\ell_{bc}(S_{bc}; \boldsymbol{m}, \tilde{\boldsymbol{W}}) + \ell_{ba}(S_{ba}; \boldsymbol{m}, \tilde{\boldsymbol{W}})\big], \quad (14)$$

where $\lambda_{up} \geq 1$ is an upweighting hyperparameter, and

$$\ell_{bc}(S_{bc}; \boldsymbol{m}, \tilde{\boldsymbol{W}}) = \frac{1}{|S_{bc}|} \sum_{i \in S_{bc}} \ell_{CE}(x_i, y_i; \boldsymbol{m} \odot \tilde{\boldsymbol{W}}), \quad (15)$$

where ℓ_{CE} denotes the cross entropy loss. ℓ_{ba} is defined as similar to ℓ_{bc}.

The expectation is approximated with Monte Carlo estimates, where the number of mask $\boldsymbol{m}$ sampled per iteration is set to 1 in practice. To implement (14), we oversample the samples in S_{bc} for λ_{up} times more than the samples in S_{ba}. This sampling strategy is aimed at increasing the mixture weight ϕ of the proposed mixture distribution P_{mix}^{η} in (5), while we empirically approximate the unknown bias-conflicting group distribution with the sample set S_{bc}.

Note that although simple oversampling of bias-conflicting samples may not lead to the OOD generalization due to the inductive bias towards memorizing a few counterexamples in overparameterized neural networks [28], such failure is unlikely reproduced in learning *pruning* parameters under the strong sparsity constraint. We sample new weight masks $\boldsymbol{m}$ for each training iteration in a stochastic manner, effectively precluding the overparameterized networks from potentially memorizing the minority samples. As a result, DCWP exhibits reasonable performance even with few bias-conflicting samples.

Bridging the alignment gap by pruning To fully utilize the bias-conflicting samples, we consider the sample-wise relation between bias-conflicting samples and majority bias-aligned samples. Zhang et al. [34] demonstrates that the deteriorated OOD generalization is potentially attributed to the distance gap between same-class representations; bias-aligned representations are more closely aligned than bias-conflicting representations, although they are generated from the same-class samples. We hypothesized that well-designed pruning masks could alleviate such geometrical misalignment. Specifically, ideal weight sparsification may guide each latent dimension to be independent of spurious attributes, thereby preventing representations from being misaligned with spuriously correlated latent dimensions. This motivates us to explore pruning masks by contrastive learning. (Related illustrative example in appendix)

Following the conventional notations of contrastive learning, we denote $f_{\boldsymbol{W}}^{enc} : \mathcal{X} \to \mathbb{R}^{n_{L-1}}$ as an encoder

parameterized by $\boldsymbol{W} = (\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{L-1})$ which maps samples into the representations at penultimate layer. Let $f^{cls}_{\boldsymbol{W}_L} : \mathbb{R}^{n_L} \to \mathbb{R}^C$ be the classification layer parameterized by $\boldsymbol{W}_L$. Then $f_{\boldsymbol{W}}(\boldsymbol{x}) = f^{cls}_{\boldsymbol{W}_L}(f^{enc}_{\boldsymbol{W}}(\boldsymbol{x})), \forall \boldsymbol{x} \in \mathcal{X}$. We similarly define $f^{enc}_{\boldsymbol{m}\odot\boldsymbol{W}}$ and $f^{cls}_{\boldsymbol{m}_L\odot\boldsymbol{W}_L}$. For the i-th sample $\boldsymbol{x}_i$, let $\boldsymbol{z}_i(\boldsymbol{W}) = \text{norm}(f^{enc}_{\boldsymbol{W}}(\boldsymbol{x}_i))$ be the normalized representations lies on the unit hypersphere, and similarly define $\boldsymbol{z}_i(\boldsymbol{m} \odot \boldsymbol{W})$. We did not consider projection networks [3, 18] for architectural simplicity. Given index subsets of training samples $\mathcal{V}, \mathcal{V}^+$, the supervised contrastive loss [18] function is defined as follows:

$$\ell_{con}(\mathcal{V}, \mathcal{V}^+; \mathbf{W}) = \sum_{i\in\mathcal{V}} \frac{-1}{|\mathcal{V}^+(y_i)|} \sum_{j\in\mathcal{V}^+(y_i)} \log \frac{\exp\left(\boldsymbol{z}_i(\mathbf{W}) \cdot \boldsymbol{z}_j(\mathbf{W})/\tau\right)}{\sum_a \exp\left(\boldsymbol{z}_i(\mathbf{W}) \cdot \boldsymbol{z}_a(\mathbf{W})/\tau\right)}, \tag{16}$$

where $a \in \mathcal{V} \setminus \{i\}$, $\tau > 0$ is a temperature hyperparameter, and $\mathcal{V}^+(y_i) = \{k \in \mathcal{V}^+ : y_k = y_i, k \neq i\}$ indicates the index set of samples with target label y_i. Then, we define the debiased alignment loss as follows:

$$\ell_{align}\left(\{x_i, y_i\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \Theta\right) = \mathbb{E}_{\boldsymbol{m}\sim G(\Theta)}\Big[\ell_{con}(S_{bc}, S; \boldsymbol{m} \odot \tilde{\boldsymbol{W}}) + \ell_{con}(S_{ba}, S_{bc}; \boldsymbol{m} \odot \tilde{\boldsymbol{W}})\Big], \tag{17}$$

where the expectation is approximated with Monte Carlo estimates as in (14). Intuitively, (17) reduces the gap between bias-conflicting samples and others (first term), while preventing bias-aligned samples from being aligned too close each other (second term, more discussions in appendix).

Finally, our debiased loss in (11) is defined as follows:

$$\ell_{debias}\left(S; \tilde{\boldsymbol{W}}, \Theta\right) = \ell_{WCE}\left(S; \tilde{\boldsymbol{W}}, \Theta\right) + \lambda_{align}\ell_{align}\left(S; \tilde{\boldsymbol{W}}, \Theta\right), \tag{18}$$

where $\lambda_{align} > 0$ is a balancing hyperparameter.

Fine-tuning after pruning After solving (11) by gradient-descent optimization, we can obtain the pruning parameters $\boldsymbol{\Theta}^*$. This allows us to uncover the structure of unbiased subnetworks with binary weight masks $\boldsymbol{m}^* = \{\boldsymbol{m}_1^*, \ldots, \boldsymbol{m}_L^*\}$, where $\boldsymbol{m}_l^* = \{\mathbb{1}(\sigma(\Theta_{l,i}^*) > 1/2) | 1 \leq i \leq n_l\}, \forall l \in \{1, \ldots, L\}$, and n_l is a dimensionality of the l-th weight. After pruning, we finetune the survived weights $\hat{\boldsymbol{W}} = \boldsymbol{m}^* \odot \tilde{\boldsymbol{W}}$ using ℓ_{WCE} in (14) and $\lambda_{align}\ell_{align}$ in (17). Interestingly, we empirically found that the proposed approach works well without the reset [8] (Related experiments in Section 5). Accordingly, we resume the training while fixing the unpruned pretrained weights. The pseudo-code of DCWP is provided in Algorithm 1.

Algorithm 1 Debiased Contrastive Weight Pruning (DCWP)

Input: Dataset $D = \{(x_i, y_i)_{i=1}^{|S|}\}$, pruning parameters Θ, Training iterations T_1, T_2, T_3.
Output: Trained pruning parameters Θ^* and finetuned weights $\boldsymbol{W}^*$

Stage 1. *Mining debiased samples*
Update the weights of bias-capturing network $\boldsymbol{W}_b$ on D for T_1 iterations.
Identify S_{bc} and S_{ba}.

Stage 2. *Debiased Contrastive Weight Pruning*
Pretrain the main network on D. Denote the pretrained weights as $\tilde{\boldsymbol{W}}$.
for $t = 1$ **to** T_2 **do**
    Update Θ with $\ell_{debias}\left(S; \tilde{\boldsymbol{W}}, \Theta\right) + \lambda_{\ell_1} \sum_{l,i} |\Theta_{l,i}|$ as in (11).
end for
Prune out weight as $\hat{\boldsymbol{W}} = \tilde{\boldsymbol{W}} \odot \mathbb{1}(\Theta^* > 0)$.
Update $\hat{\boldsymbol{W}}$ with ℓ_{WCE} and $\lambda_{align}\ell_{align}$ on D for T_3 iterations.

Table 1. Unbiased test accuracy evaluated on CMNIST, CIFAR10-C and bias-conflict test accuracy evaluated on BFFHQ. Models requiring supervisions on dataset bias are denoted with ✓, while others are denoted with ✗. Results are averaged on 4 different random seeds.

Dataset	Ratio (%)	ERM	EnD	Rebias	MRM	LfF	DisEnt	DCWP
		✗	✓	✓	✗	✗	✗	✗
CMNIST	0.5	62.36	84.32	69.12	60.98	83.73	86.74	**93.41**
	1.0	81.73	94.98	84.65	80.42	88.44	93.15	**95.98**
	2.0	89.33	97.01	91.96	89.31	92.67	95.15	**97.16**
	5.0	95.22	98.00	96.74	95.23	94.90	96.76	**98.02**
CIFAR10-C	0.5	22.02	23.93	21.73	23.92	27.02	27.86	**35.90**
	1.0	28.00	27.61	28.09	27.77	31.44	34.62	**41.56**
	2.0	34.63	36.62	35.57	33.53	38.49	41.95	**49.01**
	5.0	45.66	43.67	48.22	47.00	46.16	49.15	**56.17**
BFFHQ	0.5	52.25	59.80	54.90	54.75	56.50	55.50	**60.35**

Table 2. Worst-group and average test accuracies on CelebA (Blonde). (✓, ✗) here represents `Idx` $= (6, 4)$ (w/ and w/o pruning) in Table 3, respectively, which shows the impacts of pruning.

Models	ERM	DisEnt	JTT [22]	DCWP (✗)	DCWP (✓)
Worst-group	47.02	65.26	76.80	67.85	**79.30**
Average	**97.80**	67.88	93.98	95.89	94.50

5. Experimental results

5.1. Methods

Datasets To show the effectiveness of the proposed pruning algorithms, we evaluate the generalization performance of several debiasing approaches on Colored MNIST (CM-

NIST), Corrupted CIFAR-10 (CIFAR10-C), Biased FFHQ (BFFHQ) with varying ratio of bias-conflicting samples, i.e., bias ratio. We report unbiased accuracy [20, 25] on the test set, which includes a balanced number of samples from each data group. We also report bias-conflict accuracy for some experiments, which is the average accuracy on bias-conflicting samples included in an unbiased test set. Specifically, we report the bias-conflict accuracy on BFFHQ in which half of the unbiased test samples are bias-aligned, while the model with the best-unbiased accuracy is selected (Unbiased accuracy in Table 4). For CelebA (blonde) [14, 27], we report worst-group and average accuracy following [27] considering that abundant samples are included in (`Blonde Hair=0`, `Male=0`) bias-conflicting group. We use the same data splits from [14].

Baselines We compare DCWP with vanilla network trained by ERM, and the following state-of-the-art debiasing approaches: EnD [30], Rebias [1], MRM [33], LfF [25], JTT [22] and DisEnt [20]. EnD relies on the annotations on the spurious attribute of training samples, i.e., bias labels. Rebias relies on prior knowledge about the type of dataset bias (e.g., texture). MRM, LfF, JTT and DisEnt do not presume such bias labels or prior knowledge about dataset bias. Notably, MRM is closely related to DCWP where it probes the unbiased functional subnetwork with standard cross entropy. Details about other simulation settings are provided in Supplementary Material.

5.2. Evaluation results

As shown in Table 1, we found that DCWP outperforms other state-of-the-art debiasing methods by a large margin. Moreover, the catastrophic pitfalls of the existing pruning method become evident, where MRM fails to search for unbiased subnetworks. It underlines that the proposed approach for utilizing bias-conflicting samples plays a pivotal role in discovering unbiased subnetworks.

5.3. Quantitative analyses

Ablation studies To quantify the extent of performance improvement achieved by each introduced module, we analyzed the dependency of model performance on: (a) pruning out spurious weights following the trained parameters, (b) using alignment loss or (c) oversampling identified bias-conflicting samples when training $\boldsymbol{\Theta}$ and $\hat{\boldsymbol{W}}$. To emphasize the contribution of each module, we intentionally use an SGD optimizer which results in lower baseline accuracy (and for other CMNIST experiments in this subsection as well). Table 3 shows that every module plays an important role in OOD generalization, while (a) pruning contributes significantly comparing (1→2, +7.19%), (3→5, +11.59%) or (4→6, +8.68%).

Dependency on bias-capturing models To evaluate the reliability of DCWP, we compare different version of DCWP which does not rely on the dataset-tailored mining algorithms. We posit that early stopping [22] is an easy plug-and-play method to train the bias-capturing model in general. Thus we newly train DCWP$_{ERM}$ which collects bias-conflicting samples by using the early-stopped ERM model. Table 4 shows that DCWP$_{ERM}$ outperforms other baselines even though the precision, the fraction of samples in S_{bc} that are indeed bias-conflicting, or recall, the fraction of the bias-conflicting samples that are included in S_{bc}, were significantly dropped. It implies that DCWP may perform reasonably well with the limited number and quality of bias-conflicting samples.

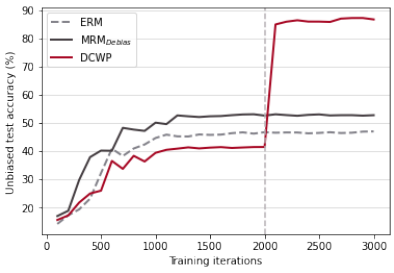
Table 3. Ablation study on CMNIST (Bias ratio=1%). Unbiased accuracy is reported. `Idx` $= 2$ uses ℓ_{WCE} only for training pruning parameters Θ while using ℓ_{CE} for retraining. `Idx` $= 3, 4$ does not conduct pruning and finetune the pretrained weights $\tilde{\boldsymbol{W}}$ by oversampling minorities or using alignment loss.

`Idx`	(a) Pruning	(b) ℓ_{align}	(c) ℓ_{WCE}	Accuracy (%)
1	-	-	-	43.10
2	✓	-	-	50.29
3	-	-	✓	73.20
4	-	✓	✓	79.28
5	✓	-	✓	84.79
6	✓	✓	✓	**87.96**

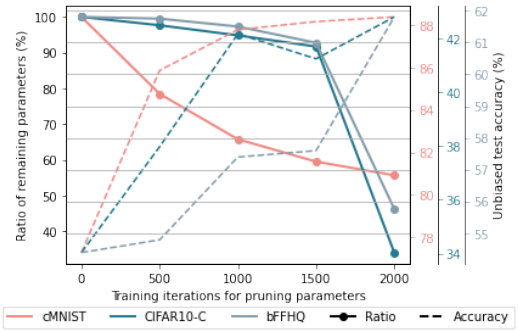
Table 4. Robustness dependency of DCWP on the performance of bias-capturing models. We set the bias ratio as 1% for CIFAR10-C. Results are averaged on 4 different random seeds.

Dataset	Model	Accuracy			Mining metrics	
		bias-align	bias-conflict	unbiased	precision	recall
CIFAR10-C	DisEnt	80.04	26.51	34.62	-	-
	DCWP$_{ERM}$	**94.33**	29.75	36.21	19.71	79.53
	DCWP	91.68	**35.99**	**41.56**	85.97	74.89
BFFHQ	DisEnt	89.80	55.55	72.68	-	-
	LfF	96.05	56.50	76.30	-	-
	DCWP$_{ERM}$	**99.45**	56.90	78.20	20.18	28.39
	DCWP	98.85	**60.35**	**79.60**	30.61	31.25

Do we need to reset weights? While it becomes widespread wisdom that remaining weights should be reset to their initial ones from the original network after pruning [8], we analyze whether such reset is also required for the proposed pruning framework. We compared the training dynamics of different models such as: (1) ERM model, (2) MRM$_{debias}$ which solves (11) instead of (12) to obtain the weight pruning masks, and (3) DCWP. Note that MRM$_{debias}$ reset the unpruned weights to its initialization after pruning. Figure 2a shows that although MRM$_{debias}$ makes a considerable advance, weight reset inevitably limits the performance gain. Moreover, finetuning the biased model significantly improves the generalization performance within only a few iterations, which implies that the proposed neural pruning can further boost the accuracy



(a) Weight reset



(b) Training iterations for Θ

Figure 2. (**a**) Comparison study on finetuning and weight resetting (CMNIST, bias ratio=1%). For DCWP, after pretraining weights for 2000 iterations, we pause and start training pruning parameters (vertical dotted line in the figure). After convergence, we mask out and finetune weights for another 1000 iterations. For MRM$_{debias}$, we reset the unpruned weight to its initialization and retrain for 3000 iterations. (**b**) Sensitivity analysis on the training iterations for pruning parameter Θ. Bias ratio=1% for both CMNIST and CIFAR10-C. Bias-conflict accuracy is reported for BFFHQ.

without weight reset. This finding allows us to debias large-scale pretrained models *without* retraining by simple pruning and finetuning.

Sensitivity analysis on training iterations We also analyzed the hyperparameter sensitivity on the training iterations of the pruning parameter $\boldsymbol{\Theta}$. The unbiased test accuracy is evaluated with weight pruning masks generated by $\boldsymbol{\Theta}$ trained for $\{500, 1000, 1500, 2000\}$ iterations on each dataset. Figure 2b shows that the accuracy increases as more (potentially biased) weights are pruned out. It implies that the proposed method can compress the networks to a substantial extent while significantly improving the OOD generalization performance.

Visualization of learned latent representations. We visualized latent representations of unbiased test samples in CMNIST after (a) pretraining, (b) pruning, and (c) finetuning. Note that we did not reset or finetune the weights in (b). As reported in Figure 3, biased representations in (a) are misaligned along with bias labels as discussed in section 4. However, after pruning, the representations were well-aligned with respect to the class of digits even without

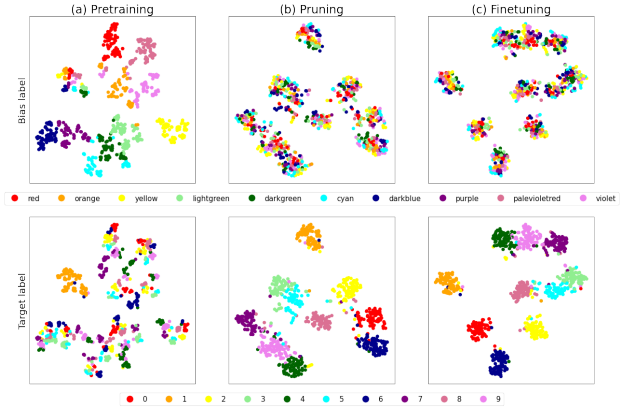


Figure 3. t-SNE visualization of representations encoded from unbiased test samples after (**a**) pretraining, (**b**) pruning and (**c**) finetuning (CMNIST, bias ratio=0.5%). Each point is painted following its label (i.e., bias label in first row, and target label in second row).

modifying the values of pretrained weights. It implies that the geometrical misalignment of representations can be addressed by pruning spurious weights while finetuning with ℓ_{debias} can further improve the generalizations.

6. Conclusion

This paper presented a novel functional subnetwork probing method for OOD generalization. Our goal was to find a winning functional lottery ticket [33], which can achieve better OOD performance compared to its counterpart full network, given a highly biased dataset in practice. We provided theoretical insights and empirical evidence to show that the minority samples provide an important clue for probing the optimal unbiased subnetworks. Simulations on various benchmark datasets demonstrated that our model significantly outperforms state-of-the-art debiasing methods. The proposed method is memory efficient and potentially compatible with many other debiasing methods.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019M3E5D2A01066267, NRF-2020R1A2B5B03001980), Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-00451), KAIST Key Research Institute (Interdisciplinary Research Group) Project and Field-oriented Technology Development Project for Customs Administration through National Research Foundation of Korea (NRF) funded by the Ministry of Science & ICT and Korea Customs Service(**NRF-2021M3I1A1097938**).

References

[1] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020. 2, 7

[2] Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 6

[4] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018. 2

[5] Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *arXiv preprint arXiv:2010.02066*, 2020. 1

[6] James Diffenderfer, Brian Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Advances in Neural Information Processing Systems*, 34:664–676, 2021. 2

[7] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015. 2

[8] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 2, 6, 7

[9] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34:27356–27368, 2021. 2

[10] 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. *arXiv preprint arXiv:1811.12231*, 2018. 2

[11] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775*, 2020. 2

[12] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018. 2

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 2

[14] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. *Advances in Neural Information Processing Systems*, 34:26449–26461, 2021. 7

[15] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018. 2

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

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 5

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 6

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

[20] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. 2, 7

[21] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019. 2

[22] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. 2, 6, 7

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 2

[24] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020. 1, 3

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

[26] Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, volume 1170, page 3, 2018. 2

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 2, 7

[28] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization

exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020. 2, 5

[29] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020. 2

[30] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13508–13517, 2021. 7

[31] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1, 3, 4

[32] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8919–8928, 2020. 2

[33] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021. 1, 2, 3, 5, 7, 8

[34] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022. 5

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