EFFICIENT DEBIASING WITH CONTRASTIVE WEIGHT PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks are often biased to spuriously correlated features that provide misleading statistical evidence that does not generalize. This raises a fundamental question: "Does an optimal unbiased functional subnetwork exist in a severely biased network? If so, how to extract such subnetwork?" While few studies have revealed the existence of such optimal subnetworks with the guidance of ground-truth unbiased samples, the way to discover the optimal subnetworks with biased training dataset is still unexplored 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).

We consider this shortcut as an inherent design issue of subnetworks. If any available information channels in deep networks' structure could transmit the information of spurious features, networks would exploit those features as long as they are sufficiently predictive. It naturally follows that pruning weights on spurious features or weights can purify the biased latent representations, leading to improved performances on bias-conflicting samples[1]. Accordingly, we hypothesize that such unbiased subnetworks may exist in the pretrained biased network.

Zhang et al. (2021) has empirically demonstrated the existence of subnetworks that are less susceptible to spurious features by using sufficient number of ground-truth bias-conflicting samples. Based on the modular property of neural networks (Csordás et al., 2020), they prune out weights that are irrelevant to the subtask, which is classification of the ground-truth bias-conflicting samples. Nonetheless, it is still unclear how to discover such optimal subnetworks when the dataset is highly biased.

To formulate this idea, we present a simple theoretical example in which, in the presence of strong spurious correlations, there exists an inevitable generalization gap of subnetworks obtained by standard pruning algorithms. Our example highlights the limitations of unbiased substructure probing combining the cross entropy loss and sparsity regularization (Zhang et al., 2021).

[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).

In addition, our example provides insight that sampling more bias-conflicting data makes it possible to identify incorrect weights. Specifically, bias-conflicting samples require that the weights associated with spurious correlations should be removed because the spurious features are not helpful in predicting the bias-conflicting samples. It leads to the exclusive preservation of non-spurious or *invariant* weights that are useful in deriving the upper bounds of the generalization error. Furthermore, 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. However, due to the potential pitfalls in data collection protocols or human prejudice, the dataset may severely lack diversity for bias-conflicting samples. Since it is often highly laborious to supplement enough bias-conflicting samples, it would be better to fully exploit a small set of given bias-conflicting samples in the training data.

To this end, we propose a novel debiasing scheme, called Debiased Contrastive Weight Pruning (DCWP), that uses the oversampled bias-conflicting data as a probe to search unbiased subnetworks. The proposed method comprises 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 *debiased* loss function and the sparsity constraint. More specifically, our debiased loss includes (1) a weighted cross-entropy loss that upweights the identified bias-conflicting samples and (2) an alignment loss that further reduces the geometrical alignment gap between bias-aligned samples 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 (Li & Vasconcelos, 2019; Nam et al., 2020), Corrupted CIFAR-10 (Hendrycks & Dietterich, 2019) and Biased FFHQ (Kim et al., 2021), 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 (Geirhos et al., 2018), language biases (Gururangan et al., 2018) or sensitive variables such as ethnicity or gender (Narayanan, 2018; Feldman et al., 2015). Such behavior is of practical concern because it deteriorates the reliability of deep networks in sensitive applications like healthcare, finance, and legal services (Corbett-Davies & Goel, 2018).

Debiasing frameworks. Recent studies have attempted to train a debiased network robust to spurious correlations, which can be roughly categorized into approaches (1) leveraging annotations of spurious attributes, i.e., bias label (Sagawa et al., 2019; Wang et al., 2020), (2) presuming certain type of bias, e.g., texture (Bahng et al., 2020; Ge et al., 2021) or (3) without using explicit kinds of supervisions on dataset bias (Nam et al., 2020; Lee et al., 2021). Sagawa et al. (2019); Hu et al. (2018) optimize the worst-group error by using training group information. For the practical implementation, reweighting or subsampling protocols are often used with increased model regularization (Sagawa et al., 2020). Liu et al. (2021); Sohoni et al. (2020) extend these approaches to the settings without expensive group annotations. Goel et al. (2020); Kim et al. (2021) provide bias-tailored augmentations to synthetically 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. In contrast to the approaches mentioned above, the effects of deep neural network architecture on generalization performance are relatively less explored. Diffenderfer et al. (2021) employ recently advanced lottery-ticket-style pruning algorithms (Frankle & Carbin, 2018) to design the compact and robust network architecture. Bai et al. (2021) directly optimize the neural architecture in terms of accuracy on OOD samples, but it cannot fundamentally eliminate the connections to the spurious input attributes. Zhang et al. (2021) 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, based on which we build a novel pruning algorithm that distills the representations to be independent of the spurious attributes.

3 Theoretical insights

3.1 Problem setup

We 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 Zhang et al. (2021), 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 environmnments $\mathcal{E}_{train}$ 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 ERM is to minimize the expected loss with a training environment $e \in \mathcal{E}_{train}$:

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

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 assume that neural networks trained by ERM indiscriminately rely on predictive features, including those spurious correlated ones (Tsipras et al., 2018). Specifically, ERM models may be sensitive to every strongly-correlated feature regardless of whether it is causally related.

To examine this issue, 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 $\boldsymbol{Z}^e_{sp}$ (Nagarajan et al., 2020; Zhang et al., 2021).

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 Zhang et al. (2021). 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:

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

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. The distribution function of each variable is omitted in expectations for notational simplicity. In practice, we apply strong sparsity constraint as ℓ_1 penalization in terms of $\boldsymbol{\pi}$ along with $\ell^e(\boldsymbol{\pi})$ to obtain a sparse solution.

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.

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 derived as:*

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

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 derived as:

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

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. The gap between (3) and (4) severely deteriorates the reliability of subnetworks obtained by training $\boldsymbol{\pi}$. 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 (Ilyas et al., 2019; Tsipras et al., 2018). 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.

Remarkably, we observe some intriguing properties of $\alpha_i(t)$: if infinitely many data and sufficient training time is provided, the gradient flow converges to the optimal solution which is invariant to $\boldsymbol{Z}_{sp}^e$, 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. 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. We provide details about the dynamics of $\alpha_i(t)$ in the appendix.

Then, how can we prioritize spurious weights to be pruned out? The above discussion illustrates the risk of reliance on spurious features. In this regard, Theorem 1 implies that the classifier may preserve pretrained 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 $\boldsymbol{X}^\eta = (Z_{inv}^\eta, \boldsymbol{Z}_{sp}^\eta)$, where $Z_{inv}^\eta = Y^\eta$ and *mixture distribution* of $\boldsymbol{Z}_{sp}^\eta$ given $Y^\eta = y$ is defined in an element wise as follows:

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

where ϕ is a scalar mixture weight,

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

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

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

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.,*

$$P^{\eta}_{mix}(Z^{\eta}_{sp,i} = -y \mid Y^{e} = y) \leq P^{\eta}_{mix}(Z^{\eta}_{sp,i} = y \mid Y^{\eta} = y), \quad \forall\, i \tag{8}$$

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

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

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

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

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 the gap between training and test error bound, which guarantees the improved OOD generalization.

4 Debiased Contrastive Weight Pruning (DCWP)

Our theoretical example elucidates 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 by identifying and exploiting a small set of existing bias-conflicting training samples.

Consider a L layer neural networks as a function $f_{\boldsymbol{W}} : \mathcal{X} \to \mathbb{R}^{C}$ parameterized by weights $\boldsymbol{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L\}$, where $C = |\mathcal{Y}|$ is the number of classes. Analogous to the earlier works on pruning, we introduce binary weight pruning masks $\boldsymbol{m} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_L\}$ to model the subnetworks as $f(\cdot; \boldsymbol{m}_1 \odot \boldsymbol{W}_1, \ldots, \boldsymbol{m}_L \odot \boldsymbol{W}_L)$. 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 $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_L\}$ 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 (Jang et al., 2016) for sampling masks together with ℓ_1 regularization term of $\boldsymbol{\Theta}$ is adopted as a sparsity constraint. With a slight abuse of notations, $\boldsymbol{m} \sim G(\boldsymbol{\Theta})$ denotes a set of masks sampled with logits $\boldsymbol{\Theta}$ by applying Gumbel-softmax trick.

Then our main optimization problem is defined as follows:

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

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 Zhang et al. (2021) with our notations as follows:

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

where Zhang et al. (2021) 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 (Zhang & Sabuncu, 2018):

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

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 upweights the samples with a high probability of predicting the correct target, amplifying the network bias by putting more emphasis on easy-to-predict samples (Nam et al., 2020).

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:

$$\ell_{WCE}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big) := \mathbb{E}_{\boldsymbol{m} \sim G(\Theta)}\left[\frac{\lambda_{up}}{|S_{bc}|} \sum_{i \in S_{bc}} \ell_{CE}(x_i, y_i; \boldsymbol{m} \odot \tilde{\boldsymbol{W}}) + \frac{1}{|S_{ba}|} \sum_{i \in S_{ba}} \ell_{CE}(x_i, y_i; \boldsymbol{m} \odot \tilde{\boldsymbol{W}})\right], \tag{14}$$

where $\lambda_{up} \geq 1$ is an upweighting hyperparameter, and ℓ_{CE} denotes the cross entropy loss. 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 Sagawa et al. (2020), 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. (2022) 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.

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 $f_{\boldsymbol{W}_L}^{cls} : \mathbb{R}^{n_L} \to \mathbb{R}^C$ be the classification layer parameterized by $\boldsymbol{W}_L$. Then $f_{\boldsymbol{W}}(\boldsymbol{x}) = f_{\boldsymbol{W}_L}^{cls}(f_{\boldsymbol{W}}^{enc}(\boldsymbol{x})), \forall \boldsymbol{x} \in \mathcal{X}$. We similarly define $f_{\boldsymbol{m} \odot \boldsymbol{W}}^{enc}$ and $f_{\boldsymbol{m}_L \odot \boldsymbol{W}_L}^{cls}$. For the i-th sample

$\boldsymbol{x}_i$, let $\boldsymbol{z}_i(\boldsymbol{W}) = \mathrm{norm}(f^{enc}_{\boldsymbol{W}}(\boldsymbol{x}_i))$ 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 (Chen et al., 2020; Khosla et al., 2020) for architectural simplicity. Given index subsets of training samples $\mathcal{V}, \mathcal{V}^+$, the supervised contrastive loss (Khosla et al., 2020) function is defined as follows:

$$\ell_{con}(\mathcal{V}, \mathcal{V}^+; \mathbf{W}) = \sum_{i \in \mathcal{V}} \frac{-1}{|\mathcal{V}^+(y_i)|} \sum_{j \in \mathcal{V}^+(y_i)} \log \frac{\exp\big(\boldsymbol{z}_i(\mathbf{W}) \cdot \boldsymbol{z}_j(\mathbf{W})/\tau\big)}{\sum_{a \in \mathcal{V} \setminus \{i\}} \exp\big(\boldsymbol{z}_i(\mathbf{W}) \cdot \boldsymbol{z}_a(\mathbf{W})/\tau\big)}, \tag{15}$$

where $\tau > 0$ 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:

$$\ell_{align}\Big(\{x_i, y_i\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \Theta\Big) = \mathbb{E}_{\boldsymbol{m} \sim G(\Theta)}\Big[\ell_{con}(S_{bc}, S; \boldsymbol{m} \odot \tilde{\boldsymbol{W}}) + \ell_{con}(S_{ba}, S_{bc}; \boldsymbol{m} \odot \tilde{\boldsymbol{W}})\Big], \tag{16}$$

where the expectation is approximated with Monte Carlo estimates as in (14). Intuitively, (16) 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:

$$\ell_{debias}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big) = \ell_{WCE}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big) + \lambda_{align}\ell_{align}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big), \tag{17}$$

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 $\boldsymbol{m}_l^* = \{\mathbb{1}(\sigma(\Theta_{l,i}^*) > 1/2) \mid 1 \leq i \leq n_l\}, \forall l \in \{1, \ldots, L\}$, 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 (16). Interestingly, we empirically found that the proposed approach works well without the reset (Frankle & Carbin, 2018) (Related experiments in Section 5). Accordingly, we resume the training while fixing the unpruned pretrained weights. The pseudo-code of DCWP is provided in the supplementary material.

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 (CMNIST), Corrupted CIFAR-10 (CIFAR10-C), and Biased FFHQ (BFFHQ) with varying ratio of bias-conflicting samples, i.e., bias ratio. We report unbiased accuracy (Nam et al., 2020; Lee et al., 2021) 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. We also compare the unbiased accuracy on BFFHQ in Table 3.

Baselines We compare DCWP with vanilla network trained by ERM, and the following state-of-the-art debiasing approaches: EnD (Tartaglione et al., 2021), Rebias (Bahng et al., 2020), MRM (Zhang et al., 2021), LfF (Nam et al., 2020) and DisEnt (Lee et al., 2021). EnD relies on the annotations on the spurious attribute of training samples, i.e., bias labels. Rebias rely on prior knowledge about the type of dataset bias (e.g., texture). MRM, LfF, 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, datasets, and baselines are provided in the 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.

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**

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) oversampling identified bias-conflicting samples when learning $\boldsymbol{\Theta}$ and $\boldsymbol{W}$, (b) pruning out spurious weights following the trained parameters, and using alignment loss for (c) training $\boldsymbol{\Theta}$, or for (d) finetuning $\boldsymbol{W}$, and (e) using GCE loss for training bias-capturing model. For those cases where GCE loss is not used, we replace it with a CE loss. To emphasize the contribution of each module, we intentionally use a SGD optimizer which results in lower baseline accuracy (and for other CMNIST experiments in this subsection as well). Table 2 shows that every module plays an important role in OOD generalization, while (b) pruning contributes significantly when comparing indices 3 and 8.

Table 2: Ablation study on CMNIST (Bias ratio=1%). Unbiased accuracy is reported.

Index	(a) Oversampling	(b) Pruning	(c) ℓ_{align}^{Θ}	(d) $\ell_{align}^{\boldsymbol{W}}$	(e) GCE	Accuracy (%)
1	-	-	-	-	-	43.10
2	✓	-	-	-	-	69.78
3	✓	-	-	-	✓	73.20
4	✓	✓	-	-	-	74.80
5	✓	✓	-	✓	-	75.15
6	✓	✓	✓	✓	-	76.49
7	✓	-	-	✓	✓	79.28
8	✓	✓	-	-	✓	84.79
9	✓	✓	✓	✓	✓	**87.96**

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 (Liu et al., 2021) 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 3 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.

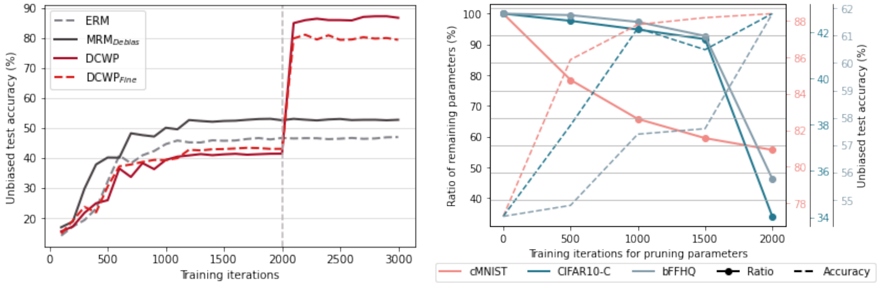
Do we need to reset weights? While it becomes widespread wisdom that remaining weights should be reset to their initial ones from the original network after pruning (Frankle & Carbin, 2018), we analyze whether such reset is also required for the proposed pruning framework. We compared the training dynamics of different models such as: (1) ERM model, (2) MRM$_{debias}$ which solves (11) instead of (12) to obtain the weight pruning masks, (3) DCWP$_{fine}$ which *skip* training $\boldsymbol{\Theta}$ and only conduct finetuning (index 7 in Table 2), and (4) DCWP. Note that MRM$_{debias}$ reset the

Table 3: Robustness dependency of DCWP on the performance of bias-capturing models. We set 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**	<u>29.75</u>	<u>36.21</u>	19.71	79.53
	DCWP	<u>91.68</u>	**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**	<u>56.90</u>	<u>78.20</u>	20.18	28.39
	DCWP	<u>98.85</u>	**60.35**	**79.60**	30.61	31.25

unpruned weights to its initialization after pruning. Figure 1a 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, while pruning further boosts the accuracy by about 9%. It implies that the proposed framework does not require parameter reset, which allows debiasing large-scale pretrained models without retraining by simple pruning and finetuning.

Sensitivity analysis on training iterations We also analyzed hyperparameter sensitivity to the amount of training iteration $\mathbf{\Theta}$. The unbiased test accuracy is evaluated with weight pruning masks generated by $\mathbf{\Theta}$ trained for $\{500, 1000, 1500, 2000\}$ iterations on every dataset. Figure 1b 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.



(a) Weight reset

(b) Training iterations for Θ

Figure 1: (**a**) Comparison study on finetuning and weight resetting (CMNIST, bias ratio=1%). For DCWP and $DCWP_{Fine}$, 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 Θ. Bias ratio=1% for both CMNIST and CIFAR10-C. Bias-conflict accuracy is reported for BFFHQ.

6 Conclusion

This paper presents a novel functional subnetwork probing method for OOD generalization. We provided theoretical insights and empirical evidence to show that the bias-conflicting samples provide an important clue for probing the optimal unbiased subnetworks. The proposed method is computationally efficient while fully compatible with many other debiasing methods.

REFERENCES

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*, pp. 528–539. PMLR, 2020.

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*, pp. 8320–8329, 2021.

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

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.

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.

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.

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*, pp. 259–268, 2015.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

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.

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.

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.

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.

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

Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.

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

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.

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

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

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.

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*, pp. 14992–15001, 2021.

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.

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*, pp. 9572–9581, 2019.

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*, pp. 6781–6792. PMLR, 2021.

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

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.

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

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.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

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.

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*, pp. 13508–13517, 2021.

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

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*, pp. 8919–8928, 2020.

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*, pp. 12356–12367. PMLR, 2021.

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.

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.

Bowen Zhao, Chen Chen, Qi Ju, and Shutao Xia. Learning debiased models with dynamic gradient alignment and bias-conflicting sample mining. *arXiv preprint arXiv:2111.13108*, 2021.

Appendix

The supplementary material is organized as follows. We begin with providing the algorithm of DCWP. Then we present the proof for Theorem 1 and 2. In section C, we extend the presented theoretical example in the main paper to illustrate the risks of geometrical misalignment of embeddings arising from strong spurious correlations. Section D presents additional experimental results and analyses. Optimization setting, hyperparameter configuration, and other experimental details are provided in section E.

A Pseudocode

Algorithm 1 Debiased Contrastive Weight Pruning (DCWP)

```
Input: Dataset $D = \{(x_i, y_i)_{i=1}^{|S|}\}$, pruning parameters $\Theta$, Training iterations $T_1, T_2, T_3$.
Output: Trained pruning parameters $\Theta^*$ 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 $\Theta$ with $\ell_{debias}\Big(S; \tilde{\boldsymbol{W}}, \Theta\Big) + \lambda_{\ell_1} \sum_{l,i} |\Theta_{l,i}|$ 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 $\ell_{WCE}$ and $\lambda_{align}\ell_{align}$ on D for $T_3$ iterations.
```

B Proofs

In this section, we present the detailed proofs for Theorems 1 and 2 explained in the main paper, followed by an illustration about the dynamics of weight ratio $\alpha_i(t) = \tilde{w}_{sp,i}(t)/\tilde{w}_{inv}(t)$.

B.1 Proof of Theorem 1

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 derived as:*

$$\ell^e(\boldsymbol{\pi}) \leq 2\exp\Bigg(-\frac{2\Big(\pi_{inv} + (2p^e-1)\sum_{i=1}^D \alpha_i(t)\pi_{sp,i}\Big)^2}{4\sum_{i=1}^D \alpha_i(t)^2+1}\Bigg), \tag{18}$$

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 derived as:

$$\ell^e(\boldsymbol{\pi}) \leq 2\exp\Big(-\frac{2\pi_{inv}^2}{4\sum_{i=1}^D \alpha_i(t)^2+1}\Big), \tag{19}$$

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

Proof. We omit time t in $\tilde{\boldsymbol{w}}(t)$ and $\alpha_i(t)$ for notational simplicity throughout the proof of Theorem 1 and 2.

The prediction from the classifier $\hat{Y}^e$ is defined in (2) as

$$\begin{aligned}\hat{Y}^e &= \operatorname{sgn}\big(\tilde{\boldsymbol{w}}^T(\boldsymbol{X}^e \odot \boldsymbol{m})\big)\\ &= \operatorname{sgn}\Big(\mathcal{O}^e\Big),\end{aligned} \tag{20}$$

where

$$\mathcal{O}^e := \tilde{w}_{inv} m_{inv} Z^e_{inv} + \sum_{i=1}^{D} \tilde{w}_{sp,i} m_{sp,i} Z^e_{sp,i}. \tag{21}$$

Assume that Y^e is uniformly distributed binary random variable. Then,

$$\mathbb{E}_{\boldsymbol{X}^e, Y^e, \boldsymbol{m}}[Y^e \hat{Y}^e] = \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = 1\Big] - \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = -1\Big], \tag{22}$$

where

$$\begin{aligned} \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = 1\Big] &= \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\operatorname{sgn}\Big(\mathcal{O}^e\Big) \,\Big|\, Y^e = 1\Big] \\ &= P\big(\mathcal{O}^e > 0 \,\big|\, Y^e = 1\big) - P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = 1\big) \\ &= 1 - 2P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = 1\big), \end{aligned} \tag{23}$$

and

$$\begin{aligned} \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = -1\Big] &= P\big(\mathcal{O}^e > 0 \,\big|\, Y^e = -1\big) - P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = -1\big) \\ &= -\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = 1\Big], \end{aligned} \tag{24}$$

where we use $P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = 1\big) = P\big(\mathcal{O}^e > 0 \,\big|\, Y^e = -1\big)$ and $P\big(\mathcal{O}^e > 0 \,\big|\, Y^e = 1\big) = P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = -1\big)$ thanks to the symmetry. Therefore, we have

$$\begin{aligned} \ell^e(\boldsymbol{\pi}) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, Y^e, \boldsymbol{m}}[1 - Y^e \hat{Y}^e] \\ &= \frac{1}{2} - \frac{1}{2}\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\hat{Y}^e | Y^e = 1\Big] \\ &= P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = 1\big). \end{aligned} \tag{25}$$

In order to derive a concentration inequality of $\ell^e(\boldsymbol{\pi})$, we compute a conditional expectation as follows:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\big[\mathcal{O}^e \,\big|\, Y^e = 1\big] &= \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\tilde{w}_{inv} m_{inv} Z^e_{inv} + \sum_{i=1}^{D} \tilde{w}_{sp,i} m_{sp,i} Z^e_{sp,i} \,\Big|\, Y^e = 1\Big] \\ &= \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\tilde{w}_{inv} m_{inv} + \sum_{i=1}^{D} \tilde{w}_{sp,i} m_{sp,i} Z^e_{sp,i} \,\Big|\, Y^e = 1\Big] \\ &= \tilde{w}_{inv}\pi_{inv} + \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\Big[\sum_{i=1}^{D} \tilde{w}_{sp,i} m_{sp,i} Z^e_{sp,i} \,\Big|\, Y^e = 1\Big] \\ &= \tilde{w}_{inv}\pi_{inv} + \sum_{i=1}^{D}(2p^e - 1)\tilde{w}_{sp,i}\pi_{sp,i}, \end{aligned} \tag{26}$$

where the last equality follows from the independence of $Z_{sp,\cdot}$ and $m_{sp,\cdot}$ as assumed in the main paper. Then,

$$\begin{aligned} P\big(\mathcal{O}^e < 0 \,\big|\, Y^e = 1\big) &= P\Big(\mathcal{O}^e - \mathbb{E}_{\boldsymbol{X}^e, \mathbf{m}}\big[\mathcal{O}^e\big] < -\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\big[\mathcal{O}^e\big] \,\big|\, Y^e = 1\Big) \\ &\leq P\Big(\,\Big|\,\mathcal{O}^e - \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\big[\mathcal{O}^e\big]\,\Big| > \mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\big[\mathcal{O}^e\big] \,\big|\, Y^e = 1\Big) \\ &\leq 2\exp\left(-\frac{2\mathbb{E}_{\boldsymbol{X}^e, \boldsymbol{m}}\big[\mathcal{O}^e \,\big|\, Y^e = 1\big]^2}{\tilde{w}^2_{inv} + \sum_{i=1}^{D} 4\tilde{w}^2_{sp,i}}\right) \\ &\leq 2\exp\left(-\frac{2\big(\tilde{w}_{inv}\pi_{inv} + \sum_{i=1}^{D}(2p^e - 1)\tilde{w}_{sp,i}\pi_{sp,i}\big)^2}{\tilde{w}^2_{inv} + \sum_{i=1}^{D} 4\tilde{w}^2_{sp,i}}\right) \\ &\leq 2\exp\left(-\frac{2\big(\pi_{inv} + \sum_{i=1}^{D}(2p^e - 1)\alpha_i\pi_{sp,i}\big)^2}{1 + \sum_{i=1}^{D} 4\alpha_i^2}\right), \end{aligned} \tag{27}$$

where the second inequality is obtained using Hoeffding's inequality, third inequality is from (26), and last inequality is obtained by dividing both denominator and numerator with $\tilde{w}_{inv}^2$. We use the definition of weight ratio $\alpha_i = \tilde{w}_{sp,i}/\tilde{w}_{inv}$. For the second inequality, we use that $\tilde{w}_{inv}m_{inv}Z_{inv}^e \in \{0, \tilde{w}_{inv}\}$ and $\tilde{w}_{sp,i}m_{sp,i}Z_{sp,i}^e \in \{-\tilde{w}_{sp,i}, 0, \tilde{w}_{sp,i}\}$ $\forall i$ in (21) to obtain the denominator.

Finally, the proof for the positivity of $\alpha_i(t)$ comes from Proposition 1 in B.3 in this appendix. This concludes the proof. □

B.2 Proof of Theorem 2

Theorem 2. *(Training bound with the mixture distribution) Assume that the defined mixture distribution P_{mix}^η is biased, i.e.,*

$$P_{mix}^\eta(Z_{sp,i}^\eta = -y \mid Y^e = y) \leq P_{mix}^\eta(Z_{sp,i}^\eta = y \mid Y^\eta = y), \quad \forall\, i \tag{28}$$

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

$$\ell^\eta(\boldsymbol{\pi}) \leq 2\exp\left(-\frac{2(\pi_{inv} + (2p^\eta(1-\phi)-1)\sum_{i=1}^D \alpha_i(t)\pi_{sp,i})^2}{4\sum_{i=1}^D \alpha_i(t)^2 + 1}\right). \tag{29}$$

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

$$\ell^\eta(\boldsymbol{\pi}) \leq 2\exp\Big(-\frac{2\pi_{inv}^2}{4\sum_{i=1}^D \alpha_i(t)^2 + 1}\Big), \tag{30}$$

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

Proof. Recall that $Z_{sp,i}^\eta$ follows the mixture distribution P_{mix}^η:

$$P_{mix}^\eta(Z_{sp,i}^\eta \mid Y^\eta = y) = \phi P_{debias}^\eta(Z_{sp,i}^\eta \mid Y^\eta = y) + (1-\phi)P_{bias}^\eta(Z_{sp,i}^\eta \mid Y^\eta = y). \tag{31}$$

Then, with definition in (6) and (7),

$$\begin{aligned} P_{mix}(Z_{sp,i}^\eta = -y|Y^\eta = y) &= \phi + (1-\phi)(1-p^\eta) \\ P_{mix}(Z_{sp,i}^\eta = y|Y^\eta = y) &= (1-\phi)p^\eta, \end{aligned} \tag{32}$$

for $y \in \{-1, 1\}$. Then, based on the assumption, $\phi + (1-\phi)(1-p^\eta) \leq (1-\phi)p^\eta$, which gives $\phi \leq 1 - \frac{1}{2p^\eta}$. Specifically, if $\phi = 1 - \frac{1}{2p^\eta}$, it turns out that $P_{mix}(Z_{sp,i}^\eta = -y|Y^\eta = y) = P_{mix}(Z_{sp,i}^\eta = y|Y^\eta = y) = \frac{1}{2}$, which implies that spurious features turns out to be random and the mixture distribution becomes perfectly debiased. If $\phi = 0$, the mixture distribution boils down into a biased distribution as similarly defined in the environment $e \in \mathcal{E}_{train}$.

The prediction from the classifier $\mathcal{O}^\eta$ is defined as similar to $\mathcal{O}^e$ in (21). Then in order to derive a concentration inequality of $\ell^\eta(\boldsymbol{\pi})$, we derive a conditional expectation of $\mathcal{O}^\eta$ as done in (26):

$$\begin{aligned} \mathbb{E}_{\boldsymbol{X}^\eta, \boldsymbol{m}}\big[\mathcal{O}^\eta \mid Y^\eta = 1\big] &= \mathbb{E}_{\boldsymbol{X}^\eta, \boldsymbol{m}}\Big[\tilde{w}_{inv}m_{inv}Z_{inv}^\eta + \sum_{i=1}^D \tilde{w}_{sp,i}m_{sp,i}Z_{sp,i}^\eta \,\Big|\, Y^\eta = 1\Big] \\ &= \mathbb{E}_{\boldsymbol{X}^\eta, \boldsymbol{m}}\Big[\tilde{w}_{inv}m_{inv} + \sum_{i=1}^D \tilde{w}_{sp,i}m_{sp,i}Z_{sp,i}^\eta \,\Big|\, Y^\eta = 1\Big]. \end{aligned} \tag{33}$$

Then, with the definition in (31), the second term in the above conditional expectation of (33) is defined as follows:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{X}^\eta,\boldsymbol{m}}\Big[\sum_{i=1}^{D}\tilde{w}_{sp,i}m_{sp,i}Z^\eta_{sp,i} \mid Y^\eta = 1\Big] \\
&= \sum_{i=1}^{D}\tilde{w}_{sp,i}\pi_{sp,i}\Big(\phi\mathbb{E}_{debias}[Z^\eta_{sp,i} \mid Y^\eta = 1] + (1-\phi)\mathbb{E}_{bias}[Z^\eta_{sp,i} \mid Y^\eta = 1]\Big) \\
&= \sum_{i=1}^{D}\tilde{w}_{sp,i}\pi_{sp,i}\Big(\phi\cdot(-1) + (1-\phi)(2p^\eta - 1)\Big) \\
&= \sum_{i=1}^{D}\tilde{w}_{sp,i}\pi_{sp,i}\big(2p^\eta(1-\phi) - 1\big),
\end{aligned}
\tag{34}
$$

where $\mathbb{E}_{debias}$ and $\mathbb{E}_{bias}$ in the first equality denote the conditional expectation with respect to distribution P^η_{debias} and P^η_{bias} in (6) and (7), respectively. Plugging (34) into (33), we get

$$
\mathbb{E}_{\boldsymbol{X}^\eta,\mathbf{m}}\big[\mathcal{O}^\eta \bigm| Y^\eta = 1\big] = \tilde{w}_{inv}\pi_{inv} + \sum_{i=1}^{D}\big(2p^\eta(1-\phi) - 1\big)\tilde{w}_{sp,i}\pi_{sp,i}. \tag{35}
$$

Then we can derive the upper bound of $\ell^\eta(\boldsymbol{\pi}) = P(\mathcal{O}^\eta < 0 \mid Y^\eta = 1)$ similarly to (27):

$$
\begin{aligned}
P\big(\mathcal{O}^\eta < 0 \bigm| Y^\eta = 1\big) &\le P\Big(\Big|\mathcal{O}^\eta - \mathbb{E}_{\boldsymbol{X}^\eta,\boldsymbol{m}}\big[\mathcal{O}^\eta\big]\Big| > \mathbb{E}_{\boldsymbol{X}^\eta,\boldsymbol{m}}\big[\mathcal{O}^\eta\big] \bigm| Y^\eta = 1\Big) \\
&\le 2\exp\Big(-\frac{2\mathbb{E}_{\boldsymbol{X}^\eta,\boldsymbol{m}}\big[\mathcal{O}^\eta \bigm| Y^\eta = 1\big]^2}{\tilde{w}_{inv}^2 + 4\sum_{i=1}^{D}\tilde{w}_{sp,i}^2}\Big) \\
&\le 2\exp\Big(-\frac{2\big(\tilde{w}_{inv}\pi_{inv} + \sum_{i=1}^{D}\big(2p^\eta(1-\phi) - 1\big)\tilde{w}_{sp,i}\pi_{sp,i}\big)^2}{\tilde{w}_{inv}^2 + 4\sum_{i=1}^{D}\tilde{w}_{sp,i}^2}\Big) \\
&\le 2\exp\Big(-\frac{2\big(\pi_{inv} + \sum_{i=1}^{D}(2p^\eta(1-\phi) - 1)\alpha_i\pi_{sp,i}\big)^2}{1 + \sum_{i=1}^{D}4\alpha_i^2}\Big),
\end{aligned}
\tag{36}
$$

where the first inequality is obtained by Hoeffding's inequality, and second inequality is from (35). The denominator is obtained as same as in (27), since $\tilde{w}_{inv}m_{inv}Z^\eta_{inv} \in \{0, \tilde{w}_{inv}\}$ and $\tilde{w}_{sp,i}m_{sp,i}Z^\eta_{sp,i} \in \{-\tilde{w}_{sp,i}, 0, \tilde{w}_{sp,i}\}\ \forall i$ as-is. If we plug-in the upper bound value of $\phi = 1 - \frac{1}{2p^\eta}$ obtained from (32) into (36), it boils down into the test bound in (4). □

B.3 Dynamics of the weight ratio

We omit an index of environment e in the proposition below for notational simplicity.

Proposition 1. *Consider a binary classification problem of linear classifier $f_{\boldsymbol{w}}$ under exponential loss. Let $(\boldsymbol{X}, Y) \sim P$, where each input random variable $\boldsymbol{X}$ and the corresponding label Y is generated by*

$$
\boldsymbol{X} = \begin{pmatrix} Z_{inv} \\ \boldsymbol{Z}_{sp} \end{pmatrix}, Y = Z_{inv},
$$

where $\boldsymbol{Z}_{sp} = (2\boldsymbol{z} - 1)Z_{inv}$ for a random variable $\boldsymbol{z} \in \{0,1\}^D$ which is chosen from multivariate Bernoulli distribution ($z_i \sim Bern(p)$) with $p > \frac{1}{2}$, i.e., p denotes p^e in the main paper. Let $\boldsymbol{w} = \begin{pmatrix} w_{inv} \\ \boldsymbol{w}_{sp} \end{pmatrix} \in \mathbb{R}^{D+1}$ be the weight of the linear classifier $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$. Assume that $0 < w_{inv}(0)$, i.e., w_{inv} is initialized with a positive value, and $0 < w_{sp,i}(0) < \frac{1}{2}\log\frac{p}{1-p}$. Then, after sufficient time of training, w_{inv} diverges to $+\infty$ and $w_{sp,i}$ converges to $\frac{1}{2}\log\frac{p}{1-p}$, which means $\alpha_i := \frac{w_{sp,i}}{w_{inv}}$ converges to 0 for all $i \in \{1, 2, \cdots, D\}$. More precisely,

$$
\log\Big(e^{w_{inv}(0)} + [4p(1-p)]^{\frac{D}{2}}t\Big) \le w_{inv}(t) \le \log\left(e^{w_{inv}(0)} + t\prod_{i=1}^{D}\Big(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\Big)\right).
$$

However, for a fixed $t < T$, each α_i is positive and its lower bound converges to some positive value.

Proof. In this proof, $w_{inv}(t)$ denotes the invariant weight at time t, while we often omit the time t and interchangeably use w_{inv} for notational simplicity, and likewise for $w_{sp,i}(t)$.

Note that the network output is given by

$$\begin{aligned} f_{\boldsymbol{w}}(\boldsymbol{x}) &= \boldsymbol{w}^T\boldsymbol{x} \\ &= Z_{inv}w_{inv} + \boldsymbol{Z}_{sp}^T\boldsymbol{w}_{sp} \\ &= Z_{inv}w_{inv} + \sum_{i=1}^{D} Z_{sp,i}w_{sp,i}. \end{aligned}$$

The exponential loss is defined by

$$\begin{aligned} L(\boldsymbol{w}) &= \mathbb{E}_{(\boldsymbol{X},Y)}[e^{-f_{\boldsymbol{w}}(\boldsymbol{X})Y}] \\ &= \mathbb{E}_{\boldsymbol{z}}\big[\exp\big(-(Z_{inv}w_{inv} + \sum_{i=1}^{D} Z_{sp,i}w_{sp,i})Z_{inv}\big)\big] \\ &= \mathbb{E}_{\boldsymbol{z}}\big[\exp(-w_{inv} - (2z_1-1)w_{sp,1} - \cdots - (2z_D-1)w_{sp,D})\big] \\ &= e^{-w_{inv}}\prod_{i=1}^{D}\mathbb{E}_{\boldsymbol{z}}[e^{-(2z_i-1)w_{sp,i}}] \\ &= e^{-w_{inv}}\prod_{i=1}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}). \end{aligned}$$

Then, thanks to symmetry of $\boldsymbol{w}_{sp}$, it is enough to consider $\alpha := \frac{w_{sp,1}}{w_{inv}}$. We first compute the gradient:

$$\begin{aligned} \frac{\partial L}{\partial w_{inv}} &= -e^{-w_{inv}}\prod_{i=1}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}) \\ \frac{\partial L}{\partial w_{sp,1}} &= -e^{-w_{inv}}(pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}})\prod_{i=2}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}). \end{aligned}$$

Since $\frac{d}{dt}w_{inv} = -\frac{\partial L}{\partial w_{inv}}$, the dynamics is given by the following differnetial equations.

$$\begin{aligned} \frac{d}{dt}w_{inv} &= e^{-w_{inv}}\prod_{i=1}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}) \\ \frac{d}{dt}w_{sp,1} &= e^{-w_{inv}}(pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}})\prod_{i=2}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}). \end{aligned}$$

First we show that $w_{inv}(t)$ diverges to $+\infty$ as t goes ∞. We show this by computing its lower bound.

$$\begin{aligned} \frac{d}{dt}w_{inv} &= e^{-w_{inv}}\prod_{i=1}^{D}(pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}) \\ &\geq e^{-w_{inv}}\prod_{i=1}^{D}(2\sqrt{p(1-p)}) \\ &= e^{-w_{inv}}[4p(1-p)]^{\frac{D}{2}}, \end{aligned}$$

where the inequality is obtained by AM-GM inequality. This implies $e^{w_{inv}}dw_{inv} \geq [4p(1-p)]^{\frac{D}{2}}dt$. Integrating both sides from 0 to t, we get

$$e^{w_{inv}(t)} - e^{w_{inv}(0)} \geq [4p(1-p)]^{\frac{D}{2}}t$$

or

$$w_{inv}(t) \geq \log\left(e^{w_{inv}(0)} + [4p(1-p)]^{\frac{D}{2}} t\right), \tag{37}$$

which shows that $w_{inv}(t)$ diverges to $+\infty$ as $t \to \infty$. Note also that w_{inv} strictly increases since $\frac{d}{dt} w_{inv} > 0$.

For $w_{sp,i}$, $\frac{d}{dt} w_{sp,i} = 0$ implies $w_{sp,i}$ converges to $w^*_{sp,i}$ such that

$$pe^{-w^*_{sp,i}} - (1-p)e^{w^*_{sp,i}} = 0,$$

namely, $w^*_{sp,i} = \frac{1}{2}\log\frac{p}{1-p}$.

As similar to w_{inv}, $w_{sp,1}$ strictly increases if and only if $w_{sp,1} < \frac{1}{2}\log\frac{p}{1-p}$. Based on the assumptions that $0 < w_{sp,i}(0) < \frac{1}{2}\log\frac{p}{1-p}$, we conclude that $w_{sp,1}$ monotonically converges to $\frac{1}{2}\log\frac{p}{1-p}$. As p goes to 1, $\frac{1}{2}\log\frac{p}{1-p}$ is sufficiently large and we can assume $w_{sp,i}(0) < \frac{1}{2}\log\frac{p}{1-p}$.

Now, we fix $0 < t < T$ for given T and compute an upper bound of w_{inv}. Using $w_{sp,i}(t) < \frac{1}{2}\log\frac{p}{1-p}$, we get

$$\begin{aligned}
\frac{d}{dt} w_{inv} &= e^{-w_{inv}} \prod_{i=1}^{D} (pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}) \\
&< e^{-w_{inv}} \prod_{i=1}^{D} \left(pe^{-w_{sp,i}(0)} + (1-p)\sqrt{\frac{p}{1-p}}\right) \\
&= e^{-w_{inv}} \prod_{i=1}^{D} \left(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\right)
\end{aligned}$$

which implies

$$e^{w_{inv}} dw_{inv} < \prod_{i=1}^{D} \left(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\right) dt.$$

Integrating both sides from 0 to t, we get

$$w_{inv}(t) < \log\left(e^{w_{inv}(0)} + \prod_{i=1}^{D} \left(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\right) t\right). \tag{38}$$

Similarly, we compute a lower bound of $w_{sp,1}$ on $0 < t < T$. Before we start, note that $w_{inv}(t) < w_{inv}(T) =: M$ from monotonicity.

$$\begin{aligned}
\frac{d}{dt} w_{sp,1} &= e^{-w_{inv}} (pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}}) \prod_{i=2}^{D} (pe^{-w_{sp,i}} + (1-p)e^{w_{sp,i}}) \\
&> e^{-M} (pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}}) \prod_{i=2}^{D} (2\sqrt{p(1-p)}) \\
&= e^{-M} [4p(1-p)]^{\frac{D-1}{2}} (pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}})
\end{aligned}$$

induces

$$\frac{1}{pe^{-w_{sp,1}} - (1-p)e^{w_{sp,1}}} dw_{sp,1} > e^{-M} [4p(1-p)]^{\frac{D-1}{2}} dt.$$

Integrating both sides from 0 to $t < T$, we get

$$\left[\frac{1}{\sqrt{p(1-p)}} \tanh^{-1}\left(\sqrt{\frac{1-p}{p}} e^{w_{sp,1}}\right)\right]_0^t > e^{-M} [4p(1-p)]^{\frac{D-1}{2}} t$$

or

$$w_{sp,1}(t) > \frac{1}{2}\log\frac{p}{1-p} + \log\ \tanh\left(\tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t\right). \tag{39}$$

Combining (38) and (39), we conclude that

$$\alpha_p(t) = \frac{w_{sp,1}(t)}{w_{inv}(t)} \tag{40}$$

$$> \frac{\frac{1}{2}\log\frac{p}{1-p} + \log\ \tanh\left(\tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t\right)}{\log\left(e^{w_{inv}(0)} + t\prod_{i=1}^{D}\left(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\right)\right)} \tag{41}$$

for $0 < t < T$. Note that $\alpha_p(t)$ is positive in $0 < t < T$, since both $w_{sp,1}(t)$ and $w_{inv}(t)$ is monotonically increasing in $0 < t < T$, and $0 < w_{sp,1}(0), w_{inv}(0)$ by assumptions.

The numerator becomes

$$\begin{aligned}&\frac{1}{2}\log\frac{p}{1-p} + \log\ \tanh\left(\tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t\right)\\ &= \log\left[\sqrt{\frac{p}{1-p}}\tanh\left(\tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t\right)\right]\\ &= \log\left[\sqrt{\frac{p}{1-p}}\left(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)} + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t\ \operatorname{sech}^2 c\right)\right]\end{aligned}$$

for some c such that

$$\tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) < c < \tanh^{-1}(\sqrt{\frac{1-p}{p}}e^{w_{sp,1}(0)}) + e^{-M}2^{D-1}[p(1-p)]^{\frac{D}{2}}t.$$

We use $f(x+y) = f(x) + yf'(c)$ by the Mean Value Theorem (MVT) at the last line.

Notably, if we take a limit $p \to 1$, the numerator becomes

$$\lim_{p\to 1}\log\left[e^{w_{sp,1}(0)} + e^{-M}2^{D-1}p^{\frac{D+1}{2}}(1-p)^{\frac{D-1}{2}}t\ \operatorname{sech}^2 c\right] = w_{sp,1}(0).$$

Similarly, the denominator becomes

$$\begin{aligned}&\lim_{p\to 1}\log\left(e^{w_{inv}(0)} + t\prod_{i=1}^{D}\left(pe^{-w_{sp,i}(0)} + \sqrt{p(1-p)}\right)\right)\\ &= \log\left(e^{w_{inv}(0)} + t\prod_{i=1}^{D}e^{-w_{sp,i}(0)}\right)\\ &= \log\left(e^{w_{inv}(0)} + t\exp\left(-\sum_{i=1}^{D}w_{sp,i}(0)\right)\right)\end{aligned}$$

Therefore, for a fixed $0 < t < T$, we conclude that

$$\begin{aligned}\lim_{p\to 1}\alpha_p(t) &= \lim_{p\to 1}\frac{w_{sp,1}(t)}{w_{inv}(t)}\\ &\geq \frac{w_{sp,1}(0)}{\log\left(e^{w_{inv}(0)} + t\exp\left(-\sum_{i=1}^{D}w_{sp,i}(0)\right)\right)}\\ &> \frac{w_{sp,1}(0)}{\log(e^{w_{inv}(0)} + T\exp\left(-\sum_{i=1}^{D}w_{sp,i}(0)\right))}\\ &\geq \frac{w_{sp,1}(0)}{\log T + \frac{1}{T}\exp\left(w_{inv}(0) + \sum_{i=1}^{D}w_{sp,i}(0)\right) - \sum_{i=1}^{D}w_{sp,i}(0)}\end{aligned} \tag{42}$$

where we use the inequality $\log(x+y) \leq \log x + \frac{y}{x}$ in the last line. □

The key insights from Proposition 1 can be summarized as follows:

(1) Weight ratio $\alpha_i(t)$ converges to 0 as $t \to \infty$.

(2) However, for a fixed $t < T$, $\alpha_i(t) > 0$.

(3) When $t < T$ and $p \to 1$, i.e., the environment is almost perfectly biased, the convergence rate of (1) is remarkably slow as in (42). In other words, there exists $c > 0$ such that $\frac{c}{\log t} < \alpha_p(t)$ over $0 < t < T$ if p is sufficiently close to 1.

This results afford us intriguing perspective on the fundamental factors behind the biased classifiers. If we situate the presented theoretical example in an ideal scenario in which infinitely many data and sufficient training time is provided, our result (1) shows that the pretrained classifier becomes fully invariant to the spurious correlations. However, in practical setting with finite training time and number of samples, our result (2) shows that the pretrained model inevitably rely on the spuriously correlated features.

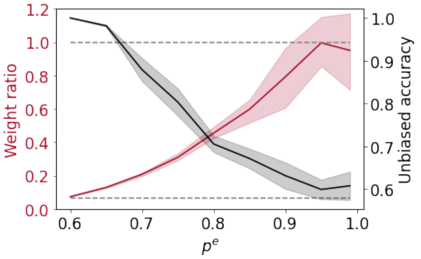


Figure 2: Implemented results of presented example.

Beyond theoretical results, we empirically observe that the weight ratio α_i of pretrained classifiers in Section 3 indeed increases as $p^e \to 1$. We simulate the example presented in section 3, where the dimensionality D is set to 15, and probability p^e varies from 0.6 (weakly biased) to 0.99 (severely biased). We train a linear classifier for 500 epochs with batch size of 1024, and measure the unbiased accuracy on test samples generated from environment $e \in \mathcal{E}_{test}$. We also measure weight ratio $\mathrm{mean}(\tilde{w}_{sp})/\tilde{w}_{inv}$, where $\mathrm{mean}(\bar{w}_{sp})$ denotes the average of pretrained spurious weights $\{w_{sp,i}\}_{i=1}^{D}$. To enable the end-to-end training, we use binary cross entropy loss instead of exponential loss, with setting $\mathcal{Y} = \{0, 1\}$ instead of $\mathcal{Y} = \{-1, 1\}$. We do not consider pruning process in this implementation. Figure 2 shows that the weight ratio increases to 1 in average as $p^e \to 1$. It implies that the spurious features $\boldsymbol{Z}_{sp}^e$ participate almost equally to the invariant feature Z_{inv}^e in the presence of strong spurious correlations. In this worst case, it is frustratingly difficult to discriminate weights necessary for OOD generalization in biased environment, resulting in the failure of learning optimal pruning parameters. Simulation results are averaged on 15 different random seeds.

C Example of geometrical misalignment

In this section, we present a simple example illustrating the potential adverse effect of spurious correlations on latent representations. Consider independent arbitrary samples within the same class $\boldsymbol{X}_i^b, \boldsymbol{X}_j^b \sim P^b_{\boldsymbol{X}^b|Y^b=y}$ and $\boldsymbol{X}^d \sim P^d_{\boldsymbol{X}^d|Y^d=y}$ for a common $y \in \{-1, 1\}$ and environments b, d where $b \in \mathcal{E}_{train}$ and $d \in \mathcal{E}_{test}$. Let $\boldsymbol{W} \in \mathbb{R}^{Q\times(D+1)}$ be a weight matrix representation of a linear mapping $T : \{-1, 1\}^{D+1} \to \mathbb{R}^Q$ which encodes the embedding vector of a given sample. We denote such embedding as $\boldsymbol{h}^e = \boldsymbol{W}\boldsymbol{X}^e$ for some $e \in \mathcal{E}$. We assume that $\boldsymbol{W}$ is initialized as to be semi-orthogonal (Saxe et al., 2013; Hu et al., 2020) for simplicity. Then the following lemma reveals the geometrical misalignment of embeddings in the presence of strong spurious correlations:

Lemma 1. *Given $y \in \{-1, 1\}$, let $\boldsymbol{h}_i^b, \boldsymbol{h}_j^b, \boldsymbol{h}^d$ be embeddings of $\boldsymbol{X}_i^b, \boldsymbol{X}_j^b, \boldsymbol{X}^d$ respectively. Then, the expected cosine similarity between $\boldsymbol{h}_i^b$ and $\boldsymbol{h}^d$ is derived as:*

$$\mathbb{E}\left[\frac{\langle \boldsymbol{h}_i^b, \boldsymbol{h}^d\rangle}{\|\boldsymbol{h}_i^b\| \cdot \|\boldsymbol{h}^d\|} \,\middle|\, Y^b = y, Y^d = y\right] = \frac{1}{D+1}, \tag{43}$$

while the expected cosine similarity between $\boldsymbol{h}_i^b$ and $\boldsymbol{h}_j^b$ is derived as:

$$\mathbb{E}\left[\frac{\langle \boldsymbol{h}_i^b, \boldsymbol{h}_j^b\rangle}{\|\boldsymbol{h}_i^b\| \cdot \|\boldsymbol{h}_j^b\|} \,\middle|\, Y^b = y\right] = \frac{1 + D(2p^b - 1)^2}{D+1}, \tag{44}$$

where p^b is a probability parameter of Bernoulli distribution of i.i.d variable $Z_{sp,i}^b$, similar to p^e in the main paper.

Proof. Let $\boldsymbol{X}^e = \boldsymbol{V}^e_{inv} + \boldsymbol{V}^e_{sp}$ for the sample from an arbitrary environment e in general, where $\boldsymbol{v}^e_{inv}, \boldsymbol{v}^e_{sp} \in \{-1, 1\}^{D+1}$ are invariant and spurious component vector, respectively:

$$V^e_{inv,j} = \begin{cases} Z^e_{inv}, & \text{if } j = 1 \\ 0, & \text{otherwise }, \end{cases} \tag{45}$$

$$V^e_{sp,j} = \begin{cases} Z^e_{sp,j}, & \text{if } j = 2, \dots, D+1 \\ 0, & \text{otherwise }. \end{cases} \tag{46}$$

Thus, $\boldsymbol{V}^e_{inv}$ and $\boldsymbol{V}^e_{sp}$ are orthogonal. Given $Y^b = y$ and $Y^d = y$ for some $y \in \{-1, 1\}$, the cosine similarity between $\boldsymbol{h}^b_i$ and $\boldsymbol{h}^d$ is expressed as follows:

$$\begin{aligned} \mathbb{E}\left[\frac{\langle \boldsymbol{h}^b_i, \boldsymbol{h}^d \rangle}{\|\boldsymbol{h}^b_i\| \|\boldsymbol{h}^d\|} \,\middle|\, Y^b = y, Y^d = y\right] &= \mathbb{E}\left[\frac{\langle \boldsymbol{X}^b_i, \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{X}^d \rangle}{\|\boldsymbol{h}^b_i\| \|\boldsymbol{h}^d\|} \,\middle|\, Y^b = y, Y^d = y\right] \\ &= \mathbb{E}\left[\frac{\langle \boldsymbol{X}^b_i, \boldsymbol{X}^d \rangle}{D+1} \,\middle|\, Y^b = y, Y^d = y\right] \\ &= \mathbb{E}\left[\frac{\langle \boldsymbol{V}^b_{i,inv} + \boldsymbol{V}^b_{i,sp}, \boldsymbol{V}^d_{inv} + \boldsymbol{V}^d_{sp} \rangle}{D+1} \,\middle|\, Y^b = y, Y^d = y\right] \\ &= \frac{1}{D+1}, \end{aligned} \tag{47}$$

where $\boldsymbol{V}^b_{i,inv}$ and $\boldsymbol{V}^b_{i,sp}$ represent the invariant and spurious component vector of $\boldsymbol{X}^b_i$, respectively, and the second equality comes from the semi-orthogonality of $\boldsymbol{W}$. The last equality comes from the orthogonality of spurious component vector from different environment $b \in \mathcal{E}_{train}$ and $d \in \mathcal{E}_{test}$.

On the other hand, the expected cosine similarity between two arbitrary embeddings $\boldsymbol{h}^b_i$ and $\boldsymbol{h}^b_j$ from the biased environment b is expressed as follows:

$$\begin{aligned} \mathbb{E}\left[\frac{\langle \boldsymbol{h}^b_i, \boldsymbol{h}^b_j \rangle}{\|\boldsymbol{h}^b_i\| \|\boldsymbol{h}^b_j\|} \,\middle|\, Y^b = y\right] &= \mathbb{E}\left[\frac{\langle \boldsymbol{V}^b_{i,inv} + \boldsymbol{V}^b_{i,sp}, \boldsymbol{V}^b_{j,inv} + \boldsymbol{V}^b_{j,sp} \rangle}{D+1} \,\middle|\, Y^e = y\right] \\ &= \frac{1 + D(2p^b - 1)^2}{D+1}, \end{aligned} \tag{48}$$

where the last equality comes from the expectation of product of independent Bernoulli variables. □

The gap between (43) and (44) unveils the imbalance of distance between same-class embeddings from different environments on the unit hypersphere; embeddings from the training environment are more closely aligned to other embeddings from the same environment than embeddings from test environment at initial even when all samples are generated within the same class. While the Lemma 1 is only applicable to the initialized $\boldsymbol{W}$ before training, such imbalance may be worsened if $\boldsymbol{W}$ learns to project the samples on the high-dimensional subspace where most of its basis are independent to the invariant features. This sparks interests in designing weight pruning masks to aggregate the representations from same-class samples all together. Indeed, in this simple example, we can address this misalignment by masking out every weight in $\boldsymbol{W}$ except the first column, which corresponds to the invariant feature.

From this point of view, we revisit the proposed alignment loss in main paper:

$$\ell_{align}\Big(\{x_i, y_i\}_{i=1}^{|S|}; \tilde{\boldsymbol{W}}, \Theta\Big) = \mathbb{E}_{\boldsymbol{m} \sim G(\Theta)}\Big[\ell_{con}(S_{bc}, S; \boldsymbol{m} \odot \tilde{\boldsymbol{W}}) + \ell_{con}(S_{ba}, S_{bc}; \boldsymbol{m} \odot \tilde{\boldsymbol{W}})\Big], \tag{49}$$

where the first term reduces the gap between bias-conflicting samples and others, while the second term prevents bias-aligned samples from being aligned too close each other. In other words, the first term is aimed at increasing the cosine similarity between representations of same-class samples with different spurious attributes, as $\boldsymbol{h}^b_i$ and $\boldsymbol{h}^d$ in this example. The second term serves as a regularizer that pulls apart same-class bias-aligned representations, as $\boldsymbol{h}^b_i$ and $\boldsymbol{h}^b_j$ in this example. Thus we can leverage abundant bias-aligned samples as negatives regardless of their class in second term, while Zhang et al. (2022) limits the negatives to samples with different target label but same bias label, which are often highly scarce in a biased dataset.

D Additional results

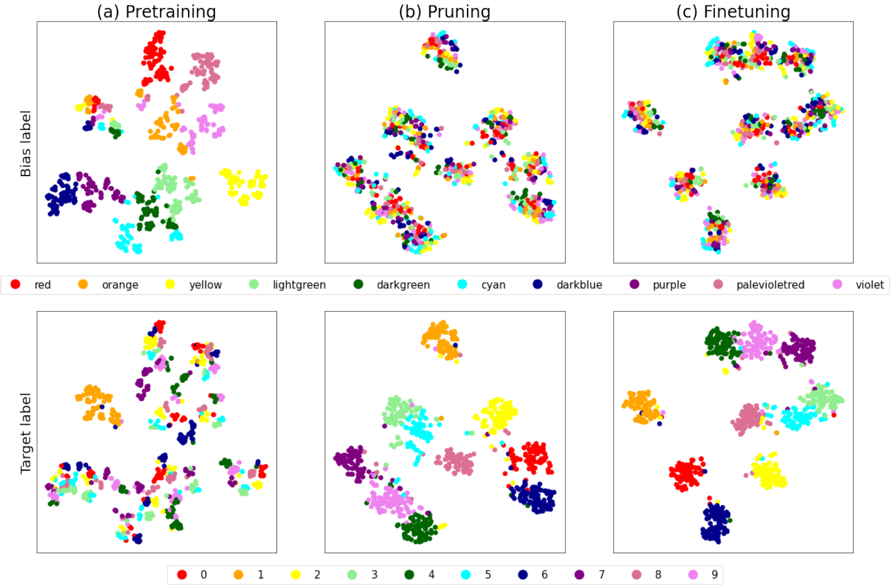


Figure 3: t-SNE visualization of representations encoded from unbiased test samples after (**a**) pretraining, (**b**) pruning and (**c**) finetuning (CMNIST, bias ratio=0.5%). Each point is painted following its label (i.e., bias label in first row, and target label in second row).

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 label as discussed in section 4 and C. However, after pruning, the representations were well-aligned with respect to the class of digits even without 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.

E Experimental setup

E.1 Datasets

We mainly follow Nam et al. (2020); Lee et al. (2021) to evaluate our framework on Color-MNIST (CMNIST), Corrupted CIFAR-10 (CIFAR10-C) and Biased FFHQ (BFFHQ) as presented in Figure 4.

CMNIST. We first consider the prediction task of digit class which is spuriously correlated to the pre-assigned color, following the existing works (Bahng et al., 2020; Nam et al., 2020; Lee et al., 2021; Tartaglione et al., 2021). Each digit is colored with certain type of color, following (Nam et al., 2020; Lee et al., 2021). The ratio of bias-conflicting samples, i.e., bias ratio, is varied in range of $\{0.5\%, 1.0\%, 2.0\%, 5.0\%\}$, where the exact number of (bias-aligned, bias-conflicting) samples is set to: (54,751, 249)-0.5%, (54,509, 491)-1%, (54,014, 986)-2%, and (52,551, 2,449)-5%.

CIFAR10-C. Each sample in this dataset is generated by corrupting original samples in CIFAR-10 with certain types of corruption. Among 15 different corruptions introduced in the original paper (Hendrycks & Dietterich, 2019), we select 10 types which are `Brightness`, `Contrast`, `Gaussian Noise`, `Frost`, `Elastic Transform`, `Gaussian Blur`, `Defocus Blur`, `Impulse Noise`, `Saturate`, and `Pixelate`, following Lee et al. (2021). Each of these corruption is spuriously correlated to the object classes of CIFAR-10, which are `Plane`, `Car`, `Bird`, `Cat`, `Deer`, `Dog`, `Frog`, `Horse`, `Ship`, and `Truck`. We use the samples corrupted in most severe level among five different severity, fol-

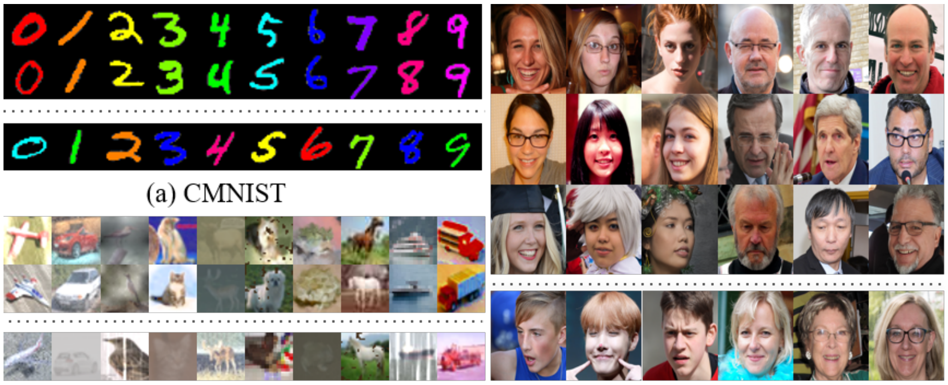


Figure 4: Example images of datasets. The images above the dotted line denote the bias-aligned samples, while the ones below the dotted line are the bias-conflicting samples. For CMNIST and CIFAR10-C, each column indicates each class. For BFFHQ, the group of three columns indicates each class.

lowing Lee et al. (2021). The exact number of (bias-aligned, bias-conflicting) samples is set to: (44,832, 228)-0.5%, (44,527, 442)-1%, (44,145, 887)-2%, and (42,820, 2,242)-5%.

BFFHQ. Each sample in this biased dataset are selected from Flickr-Faces-HQ (FFHQ) Dataset (Karras et al., 2019), where we conduct binary classifications with considering (`Age`, `Gender`) as target and spuriously correlated attribute pair following Kim et al. (2021); Lee et al. (2021). Specifically, majority of training images correspond to either young women (i.e., aged 10-29) or old men (i.e., aged 40-59). This dataset consists of 19,104 number of such bias-aligned samples and 96 number of bias-conflicting samples, i.e., old women and young men.

E.2 Simulation settings

Architecture details. We use a simple convolutional network with three convolution layers for CMNIST, with feature map dimensions of 64, 128 and 256, each followed by a ReLU activation and a batch normalization layer following Zhang et al. (2021). For CIFAR10-C and BFFHQ, we use ResNet-18 with pretrained weights provided in PyTorch `torchvision` implementations. Each convolutional network and ResNet-18 includes 1.3×10^6 and 2.2×10^7 number of parameters, respectively. We assign a pruning parameter for each weight parameter except bias in deep networks. Each of pruning parameter is initialized with value 1.5 so that the initial probability of preserving the corresponding weight is set to $\sigma(1.5) \approx 0.8$ in default.

Training details. We first train bias-capturing networks using GCE loss (q=0.7) for CMNIST and BFFHQ, with 2000 and 10000 iterations, respectively. For CIFAR10-C, we use epoch-ensemble based mining algorithms presented in Zhao et al. (2021), which selects samples cooperated with ensemble of predictions at each epoch to prevent overfitting. We use b-c score threshold $\tau = 0.8$ and confidence threshold $\eta = 0.05$ as suggested in the original paper.

Then, main networks are pretrained for 10000 iterations using an Adam optimizer with learning rate 0.01 and 0.001 for CMNIST and others, respectively.

We train pruning parameters for 2000 iterations using a learning rate 0.01, upweighting hyperparameter $\lambda_{up} = 80$ and a balancing hyperparameter $\lambda_{align} = 0.05$ for each dataset. We use a Lagrangian multiplier $\lambda_{\ell_1} = 10^{-8}$ for CMNIST, and $\lambda_{\ell_1} = 10^{-9}$ for CIFAR10-C and BFFHQ. Specifically, we set λ_{ℓ_1} by considering the size of deep networks, where we found that the value within range $\mathcal{O}(0.1 * n^{-1})$ serves as a good starting point where n is the number of parameters.

After pruning, we finetune the networks with decaying learning rate to 0.001 for CMNIST and 0.0005 for others. We use $\lambda_{align} = 0.05$ and $\lambda_{up} = 80$ for BFFHQ, and $\lambda_{up} = \{10, 30, 50, 80\}$ for CMNIST and CIFAR10-C with $\{0.5\%, 1.0\%, 2.0\%, 5.0\%\}$ of bias ratio, respectively.

Considering the pruning as a strong regularization, we did not use additional capacity control techniques such as early stopping or strong ℓ_2 regularization presented in Sagawa et al. (2020); Liu et al. (2021).

Data augmentations. We did not use any kinds of data augmentations which may implicitly enforce networks to encode invariances. For the BFFHQ dataset, we only apply random horizontal flip. For the CIFAR10-C dataset, we take 32×32 random crops from image padded by 4 pixels followed by random horizontal flip, following Nam et al. (2020). We do not use any kinds of augmentations in CMNIST.

Baselines. We use the official implementations of Rebias, LfF, DisEnt released by authors, and reproduce EnD and MRM by ourselves. For DisEnt, we use the official hyperparameter configurations provided in the original paper. We use $q = 0.7$ for LfF as suggested by authors on every experiment. For Rebias, we use the official hyperparameter configurations for CMNIST, and train for 200 epochs using Adam optimizer with learning rate 0.001 and RBF kernel radius of 1 for other datasets. For MRM, we use λ_{ℓ_1} of 10^{-8} for CMNIST following the original paper, and 10^{-9} for the others. For EnD, we set the multipliers α for disentangling and β for entangling to 1.