MITIGATING DATASET BIAS BY USING PER-SAMPLE GRADIENT

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of deep neural networks is strongly influenced by the training dataset setup. In particular, when attributes having a strong correlation with the target attribute are present, the trained model can provide unintended prejudgments and show significant inference errors (*i.e., the dataset bias problem*). Various methods have been proposed to mitigate dataset bias, and their emphasis is on weakly correlated samples, called *bias-conflicting samples*. These methods are based on explicit bias labels provided by human. However, such methods require human costs. Recently, several studies have tried to reduce human intervention by utilizing the output space values of neural networks, such as feature space, logits, loss, or accuracy. However, these output space values may be insufficient for the model to understand the bias attributes well. In this study, we propose a debiasing algorithm leveraging gradient called PGD (**P**er-sample **G**radient-based **D**ebiasing). PGD comprises three steps: (1) training a model on uniform batch sampling, (2) setting the importance of each sample in proportion to the norm of the sample gradient, and (3) training the model using importance-batch sampling, whose probability is obtained in step (2). Compared with existing baselines for various datasets, the proposed method showed state-of-the-art accuracy for the classification task. Furthermore, we describe theoretical understandings of how PGD can mitigate dataset bias.

1 INTRODUCTION

Dataset bias (Torralba & Efros, 2011; Shrestha et al., 2021), is a bad training dataset problem that occurs when unintended easier-to-learn attributes *(i.e., bias attributes)*, having a high correlation with the target attribute, are present (Shah et al., 2020; Ahmed et al., 2020). This is due to the fact that the model can infer outputs by focusing on the bias features, which could lead to testing failures. For example, most "camel" images include a "desert background," and this unintended correlation can provide a false shortcut for answering "camel" on the basis of the "desert." In (Nam et al., 2020; Lee et al., 2021), samples of data that have a strong correlation (like "desert background" in "camel" class images) are called "bias-aligned samples," while samples of data that have a weak correlation (like "camel on the grass" images) are termed "bias-conflicting samples."

To reduce the dataset bias, initial studies (Kim et al., 2019; McDuff et al., 2019; Singh et al., 2020; Li & Vasconcelos, 2019) have frequently assumed a case where labels with bias attributes are provided, but these additional labels provided through human effort are expensive. Alternatively, the bias-type, such as "background" is assumed in (Lee et al., 2019; Geirhos et al., 2018; Bahng et al., 2020; Cadene et al., 2019; Clark et al., 2019). However, assuming biased knowledge from humans is still unreasonable since even humans cannot predict the type of bias that may exist in a large dataset (Schäfer, 2016). Data for deep learning is typically collected by web-crawling without thorough consideration of the dataset bias problem.

Recent studies (Le Bras et al., 2020; Nam et al., 2020; Kim et al., 2021; Lee et al., 2021; Seo et al., 2022; Zhang et al., 2022) have replaced human intervention with DNN results. They have identified bias-conflicting samples by using empirical metrics for output space (*e.g.,* training loss and accuracy). For example, Nam et al. (2020) suggested a "relative difficulty" based on per-sample training loss and thought that a sample with a high "relative difficulty" was bias-conflicting sample. Most of the previous research has focused on the output space, such as feature space (penultimate layer

output) (Lee et al., 2021; Kim et al., 2021; Bahng et al., 2020; Seo et al., 2022; Zhang et al., 2022), loss (Nam et al., 2020), and accuracy (Le Bras et al., 2020; Liu et al., 2021). However, this limited output space can impose restrictions on describing the data in detail.

Recently, as an alternative, model parameter space (*e.g.,* gradient (Huang et al., 2021; Killamsetty et al., 2021b; Mirzasoleiman et al., 2020)) has been used to obtain high-performance gains compared to output space approaches for various target tasks. For example, Huang et al. (2021) used gradient-norm to detect out-of-distribution detection samples and showed that the gradient of FC layer $\in \mathbb{R}^{h \times c}$ could capture joint information between feature and softmax output, where h and c are the dimension of feature and output vector, respectively. Since the gradients of each data point $\in \mathbb{R}^{h \times c}$ constitute high dimensional information, it is much more informative than the output space, such as logit $\in \mathbb{R}^{c}$ and feature $\in \mathbb{R}^{h}$. However, there is no approach to tackle the dataset bias problem using a gradient norm-based metric.

In this paper, we present a resampling method from the perspective of the per-sample gradient norm to mitigate dataset bias. Furthermore, we theoretically justify that the gradient-norm-based resampling method can be an excellent debiasing approach. Our key contributions can be summarized as follows:

- We propose PGD, **P**er-sample **G**radient-norm based **D**ebiasing, a simple and efficient gradient-norm-based debiasing method. PGD is motivated by prior research demonstrating (Mirzasoleiman et al., 2020; Huang et al., 2021; Killamsetty et al., 2021b) that gradient is effective at finding rare samples, and it is also applicable to finding the bias-conflicting samples in the dataset bias problem (See Section 3 and Appendix C).
- PGD outperforms dataset bias on various benchmarks, such as colored MNIST, multi-bias MNIST, corrupted CIFAR, biased action recognition (BAR), biased FFHQ (BFFHQ), CelebA and CivilComments-WILD. In particular, for the colored MNIST case, the proposed method yielded higher unbiased test accuracies compared with the vanilla and the best method by 35.94% and 2.32%, respectively. (See Section 4)
- We provide theoretical evidence of the superiority of PGD. To this aim, we first explain that minimizing the trace of inverse Fisher information is a good objective to mitigate dataset bias. In particular, PGD, resampling based on the gradient norm computed by the biased model, is a possible optimizer for mitigating the dataset bias problem. (See Section 5)

2 Dataset Bias Problem

Classification model. We first describe the conventional supervised learning setting. Let us consider the classification problem when a training dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ with input image x_i and corresponding label y_i is given. Assuming that there are $c \in \mathbb{N} \setminus \{1\}$ classes, y_i is assigned to the one element in set $C = \{1, ..., c\}$. Note that we focus on a situation where dataset $\mathcal{D}_n$ does not have noisy samples, for example, noisy labels or out-of-distribution samples (e.g., SVHN samples when the task is CIFAR-10). When input x_i is given, $f(y_i|x_i, \theta)$ represents the softmax output of the classifier for label y_i. It is derived from the model parameter $\theta \in \mathbb{R}^{d}$. The cross-entropy (CE) loss $\mathcal{L}_{\text{CE}}$ is frequently used to train the classifier, and it is defined as $\mathcal{L}_{\text{CE}}(x_i, y_i; \theta) = -\log f(y_i|x_i, \theta)$.

Dataset bias. Let us suppose that a training set $\mathcal{D}_n$ is comprised of images, as shown in Figure 1, and that the objective is to classify the `digits`. Each image can be described by a set of attributes, (*e.g.,* for the first image in Figure 1, it can be {`digit 0`, `red`, `thin`,...}). The purpose of the training classifier is to find a model parameter θ that correctly predicts the target attributes, (*e.g.,* `digit`). Notably, the target attributes are also interpreted as *classes*. However, we focus on a case wherein another attribute that is strongly correlated to the target exists, and we call these attributes *bias attributes*. For example, in Figure 1, the bias attribute is `color`. Furthermore, samples whose bias attributes are highly correlated to the target attributes are called *bias-aligned* (top three rows in Figure 1). Conversely, weakly correlated samples are called *bias-conflicting* (see the bottom row of Figure 1). Therefore, our main scope is that the training dataset which have samples whose bias and target attributes are misaligned.[1] According to (Nam et al., 2020), when the bias attributes

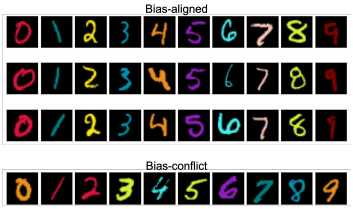


Figure 1: Target and bias attribute: `digit` `shape`, `color`.

[1]Note that bias-alignment cannot always be strictly divisible in practice. For ease of explanation, we use the notations bias-conflicting/bias-aligned.

are easier-to-learn than the target attributes, dataset bias is problematic as the trained model may prioritize the bias attributes over the target attributes. For example, for a model trained on the images in Figure 1, the model can output class 4 when the (`Orange`, 0) image (*e.g.,* left bottom image) is given, due to the wrong priority, `color` which is an easier-to-learn attribute (Nam et al., 2020).

3 PGD: Per-sample Gradient-Norm-Based Debiasing

In this section, we propose a novel debiasing algorithm, coined as PGD. PGD consists of two models, biased f_b and debiased f_d with parameters θ_b and θ_d, respectively. Both models are trained sequentially. Obtaining the ultimately trained debiased model f_d involves three steps: (1) train the biased model, (2) compute the sampling probability of each sample, and (3) train the debiased model. These steps are described in Algorithm 1.

Algorithm 1 PGD: Per-sample Gradient-norm based Debiasing

Input: dataset $\mathcal{D}_n$, learning rate η, iterations T_b, T_d, Batch size B, Data augmentation operation $\mathcal{A}(\cdot)$, Initial parameter θ_0, GCE parameter α

/ STEP 1: Train f_b **/**

for $t = 1, 2, \cdots, T_b$ **do**

  Construct a mini-batch $\mathcal{B}_t = \{(x_i, y_i)\}_{i=1}^{B} \sim U$.

  Update θ_t as:

  $\theta_{t-1} - \frac{\eta}{B} \nabla_\theta \sum_{(x,y)\in B_t} \mathcal{L}_{\text{GCE}}(\mathcal{A}(x), y; \theta_{t-1}, \alpha)$

end for

/ STEP 2: Calculate h **/**

Calculate $h(x_i, y_i)$ for all $(x_i, y_i) \in \mathcal{D}_n$, (1).

/ STEP 3: Train f_d based on h **/**

for $t = 1, 2, \cdots, T_d$ **do**

  Construct a mini-batch $\mathcal{B}'_t = \{(x_i, y_i)\}_{i=1}^{B} \sim h$.

  Update θ_{T_b+t} as:

  $\theta_{T_b+t-1} - \frac{\eta}{B} \nabla_\theta \sum_{(x,y)\in \mathcal{B}'_t} \mathcal{L}_{\text{CE}}(\mathcal{A}(x), y; \theta_{T_b+t-1})$

end for

Step 1: Training the biased model. In the first step, the biased model is trained on the mini-batches sampled from a uniform distribution U, similar to conventional SGD-based training, with data augmentation $\mathcal{A}$. The role of the biased model is twofold: it detects which samples are bias-conflicting and calculates how much they should be highlighted. Therefore, we should make the biased model uncertain when it faces bias-conflicting samples. In doing so, the biased model f_b is trained on the generalized cross-entropy (GCE) loss $\mathcal{L}_{\text{GCE}}$ (Nam et al., 2020; Lee et al., 2021). For an input image x and the corresponding true class y, $\mathcal{L}_{\text{GCE}}$ is defined as $\mathcal{L}_{\text{GCE}}(x, y; \theta, \alpha) = \frac{1-f(y|x,\theta)^\alpha}{\alpha}$. Note that $\alpha \in (0, 1]$ is a hyperparameter that controls the degree of emphasizing the easy-to-learn samples, namely bias-aligned samples. Remark that when $\alpha \to 0$, the GCE loss $\mathcal{L}_{\text{GCE}}$ is exactly the same as the conventional CE loss $\mathcal{L}_{\text{CE}}$. We set $\alpha = 0.7$ as done by the authors of Zhang & Sabuncu (2018), Nam et al. (2020) and Lee et al. (2021).

Step 2: Compute the gradient-based sampling probability. In the second step, the sampling probability of each sample is computed from the trained biased model. Since rare samples have large gradient norm compared to the usual samples at the biased model (Hsu et al., 2020), the sampling probability of each sample is computed to be proportional to its gradient norm so that bias-conflicting samples are over-sampled. Before computing the sampling probability, the per-sample gradient with respect to $\mathcal{L}_{\text{CE}}$ for all $(x_i, y_i) \in \mathcal{D}_n$ is obtained from the biased model. We propose the following sampling probability of each sample $h(x_i, y_i)$ which is proportional to their gradient norm as follows:

$$h(x_i, y_i) = \frac{\|\nabla_\theta \mathcal{L}_{\text{CE}}(x_i, y_i; \theta_b)\|_s^r}{\sum_{(x_i,y_i)\in\mathcal{D}_n} \|\nabla_\theta \mathcal{L}_{\text{CE}}(x_i, y_i; \theta_b)\|_s^r}, \tag{1}$$

where $\|\cdot\|_s^r$ denotes r square of the L_s norm, and θ_b is the result of step 1. Since, $h(x_i, y_i)$ is the sampling probability on $\mathcal{D}_n$, $h(x_i, y_i)$ is the normalized gradient-norm. Note that computing the gradient for all samples requires huge computing resources and memory. Therefore, we only extract the gradient of the final FC layer parameters. This is a frequently used technique for reducing the computational complexity (Ash et al., 2019; Mirzasoleiman et al., 2020; Killamsetty et al., 2021b;a; 2020). In other words, instead of $h(x_i, y_i)$, we empirically utilize $\hat{h}(x_i, y_i) = \frac{\|\nabla_{\theta_{\text{fc}}} \mathcal{L}_{\text{CE}}(x_i, y_i; \theta_b)\|_s^r}{\sum_{(x_i,y_i)\in\mathcal{D}_n} \|\nabla_{\theta_{\text{fc}}} \mathcal{L}_{\text{CE}}(x_i, y_i; \theta_b)\|_s^r}$, where θ_{fc} is the parameters of the final FC layer. We consider $r = 1$ and $s = 2$ (*i.e.,* L_2), and deliver ablation studies on various r and s in Section 4.

Step 3: Ultimate debiased model training. Finally, the debiased model f_d is trained using mini-batches sampled with the probability $h(x_i, y_i)$ obtained in stage 2. Note that, as described in Algorithm 1, our debiased model inherits the model parameters of the biased model θ_{T_b}. However, Lee et al. (2021) argued that just oversampling bias-conflicting samples does not successfully debias, and this unsatisfactory result stems from the data diversity, *i.e.,* data augmentation techniques are required. Hence, we used simple randomized augmentation operations $\mathcal{A}$ such as random rotation and random color jitter to oversample the bias-conflicting samples.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness of PGD for multiple benchmarks compared with previous proposed baselines. Detail analysis not in this section, *e.g.,* training time, unbiased case study, easier to learn target attribute, sampling probability analysis, reweighting with PGD are described in the Appendix C.

4.1 BENCHMARKS

To precisely examine the debiasing performance of PGD, we used the Colored MNIST, Multi-bias MNIST, and Corrupted CIFAR datasets as synthetic datasets, which assume situations in which the model learns bias attributes first. BFFHQ, BAR, CelebA, and CivilComments-WILDS datasets obtained from the real-world used to observe the situations in which general algorithms have poor performance due to bias attributes. Note that BFFHQ and BAR are biased by using human prior knowledge, while CelebA and CivilComments-WILDS are biased datasets by nature. A detailed explanation of each benchmark are presented in Appendix A.

Colored MNIST (CMNIST). Colored MNIST is a modified version of MNIST dataset (LeCun et al., 2010), where `color` are the biased attribute and `digit` serving as the target. We randomly selected ten colors that will be injected into the digit. Evaluation was conducted for various ratios $\rho \in \{0.5\%, 1\%, 5\%\}$, where ρ denotes the portion of bias-conflicting samples. Note that colored MNIST has only one bias attribute: `color`.

Multi-bias MNIST (MB-MNIST). The authors of (Shrestha et al., 2021) stated that colored MNIST is too simple to examine the applicability of debiaising algorithms for complex bias cases. However, the dataset that Shrestha et al. (2021) generated is also not complex since they did not use an official dataset (*e.g.,* MNIST) and used simple patterns (*e.g.,* straight line and triangle). Therefore, we generated a multi-bias MNIST; we used benchmark images to better reflect the real-world than (Shrestha et al., 2021). MBMNIST consists of eight attributes: digit (LeCun et al., 2010), alphabet (Cohen et al., 2017), fashion object (Xiao et al., 2017), Japanese character (Clanuwat et al., 2018), digit color, alphabet color, fashion object color, Japanese character color. Among the eight attributes, the target attribute is digit shape and the others are the bias attributes. To construct MB-MNIST, we follow the CMNIST protocol, which generates bias by aligning two different attributes (*i.e.,* digit and color) with probability $(1 - \rho)$. MB-MNIST dataset is made by independently aligning the digit and seven other attributes with probabity $(1 - \rho)$. Note that the most rare sample is generated with probability ρ^7. When ρ is set as the CMNIST case, it is too low to generate sufficient misaligned samples. Therefore, we use $\rho \in \{10\%, 20\%, 30\%\}$ to ensure the trainability.

Corrupted CIFAR (CCIFAR). CIFAR10 (Krizhevsky et al., 2009) comprises ten different objects, such as an `airplane` and a `car`. Corrupted CIFAR are biased with ten different types of texture bias (*e.g.,* frost and brightness). The dataset was constructed by following the design protocol of (Hendrycks & Dietterich, 2019), and the ratios $\rho \in \{0.5\%, 1\%, 5\%\}$ are used.

Biased action recognition (BAR). Biased action recognition dataset was derived from (Nam et al., 2020). It comprised six classes for action, (*e.g.,* `climbing` and `diving`), and each class is biased with place. For example, `diving` class pictures are usually taken `underwater`, while a few images are taken from the `diving pool`.

Biased FFHQ (BFFHQ). BFFHQ dataset was constructed from the facial dataset, FFHQ (Karras et al., 2019). It was first proposed in (Kim et al., 2021) and was used in (Lee et al., 2021). It comprises two gender classes, and each class is biased with age. For example, most female pictures are *young* while male pictures are *old*. This benchmark is $\rho = 0.5\%$.

CelebA. CelebA (Liu et al., 2015) is a common real world face classification dataset. The goal is classifying the hair color ("blond" and "not blond") of celebrities which has a spurious correlation with the gender ("male" or "female") attribute. Almost hair color of female images are blond. We report the average accuracy and the worst-group accuracy on the test dataset.

CivilComments-WILDS. CivilComments-WILDS (Borkan et al., 2019) is a dataset to classify whether an online comment is toxic or non-toxic. The mentions of certain demographic identities (male, female, White, Black, LGBTQ, Muslim, Christian, and other religion) cause the supurious correlation with the label.

Table 1: Average test accuracy and standard deviation (three runs) for experiments with the MNIST variants under various bias conflict ratios. The best accuracy is indicated in **bold** for each case.

Dataset	ρ	Vanilla	LfF	JTT	Disen	PGD (Ours)
CMNIST	0.5%	60.94$\pm$ 0.97	91.35$\pm$ 1.83	85.84$\pm$ 1.32	94.56$\pm$ 0.57	**96.88$\pm$ 0.28**
	1%	79.13$\pm$ 0.73	96.88$\pm$ 0.20	95.07$\pm$ 3.42	96.87$\pm$ 0.64	**98.35$\pm$ 0.12**
	5%	95.12$\pm$ 0.24	98.18$\pm$ 0.05	96.56$\pm$ 1.23	98.35$\pm$ 0.20	**98.62$\pm$ 0.14**
MBMNIST	10%	25.23$\pm$ 1.16	19.18$\pm$ 4.45	25.34$\pm$ 1.45	25.75$\pm$ 5.38	**61.38$\pm$ 4.41**
	20%	62.06$\pm$ 2.45	65.72$\pm$ 6.23	68.02$\pm$ 3.23	61.62$\pm$ 2.60	**89.09$\pm$ 0.97**
	30%	87.61$\pm$ 1.60	89.89$\pm$ 1.76	85.44$\pm$ 3.44	88.36$\pm$ 2.06	**90.76$\pm$ 1.84**
CCIFAR	0.5%	23.06$\pm$ 1.25	28.83$\pm$ 1.30	25.34$\pm$ 1.00	29.96$\pm$ 0.71	**30.15$\pm$ 1.22**
	1%	25.94$\pm$ 0.54	33.33$\pm$ 0.15	33.62$\pm$ 1.05	36.35$\pm$ 1.69	**42.02$\pm$ 0.73**
	5%	39.31$\pm$ 0.66	50.24$\pm$ 1.41	45.13$\pm$ 3.11	51.19$\pm$ 1.38	**52.43$\pm$ 0.14**

Table 2: Average test accuracy and standard deviation (three runs) for experiments with the raw image benchmarks: BAR and BFFHQ. The best accuracy is indicated in **bold** and for the overlapped best performance case is indicated in Underline.

Dataset	Vanilla	LfF	JTT	Disen	PGD (Ours)
BAR	63.15$\pm$ 1.06	64.41$\pm$ 1.30	63.62$\pm$1.33	64.70$\pm$ 2.06	**65.39$\pm$ 0.47**
BFFHQ	77.77$\pm$ 0.45	82.13$\pm$ 0.38	77.93$\pm$ 2.16	82.77$\pm$ 1.40	**84.20$\pm$ 1.15**

4.2 Implementation.

Baselines. We select baselines available for the official code from the respective authors among debiasing methods without prior knowledge on the bias. Our baselines comprise eight methods on the various tasks: vanilla network, LfF (Nam et al., 2020), JTT (Liu et al., 2021)[2], Disen (Lee et al., 2021), GEORGE (Sohoni et al., 2020), EIIL (Creager et al., 2021), BPA(Seo et al., 2022) and CNC (Zhang et al., 2022).

Implementation details. We use three types of networks: two types of simple convolutional networks (SimConv-1 and SimConv-2) and ResNet18 (He et al., 2016). Network imeplementation is described in Appendix B. Colored MNIST is trained on SGD optimizer, batch size 128, learning rate 0.02, weight decay 0.001, momentum 0.9, learning rate decay 0.1 every 40 epochs, 100 epochs training, and GCE parameter α 0.7. For Multi-bias MNIST, it also utilizes SGD optimizer, and 32 batch size, learning rate 0.01, weight decay 0.0001, momentum 0.9, learning rate decay 0.1 with decay step 40. It runs 100 epochs with GCE parameter 0.7. For corrupted CIFAR and BFFHQ, it uses ResNet18 as a backbone network, and exactly the same setting presented by Disen (Lee et al., 2021).[3] For CelebA, we follows experimental setting of (Seo et al., 2022) which uses ResNet18 as a backbone network. For CivilComments-WILDS, we utilize exactly the same hyperparameters of (Liu et al., 2021) and utilize pretrained BERT. Detail hyperparameters for CIFAR, BFFHQ, CelebA, CivilComments-WILDS are described in Appendix B.

To reduce the computational complexity in extracting the per-sample gradients, we use only a fully connected layer, similar to (Ash et al., 2019; Mirzasoleiman et al., 2020; Killamsetty et al., 2021b;a; 2020).

4.3 Results and Empirical Analysis

Accuracy results. In Table 1, we present the comparisons of the image classification accuracy for the unbiased test sets. The proposed method outperforms the baseline methods for all benchmarks and for all different ratios. For example, our model performance is 35.94% better than that of the vanilla model for the colored MNIST benchmark with a ratio $\rho = 0.5\%$. For the same settings, PGD performs better than Disen by 2.32%.

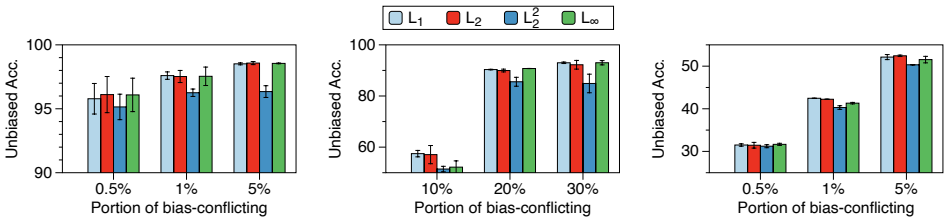
As pointed out in (Shrestha et al., 2021), colored MNIST is too simple to evaluate debiasing performance on the basis of the performance of baselines. In Multi-bias MNIST case, other models fail to obtain higher unbiased test results, even though the ratio is high, *e.g.,* 10%. In this complex setting,

[2]In the case of JTT (Liu et al., 2021), although the authors used bias label for validation dataset (especially, bias-conflicting samples), we tune the hyperparameters using a part of the biased training dataset for fair comparison. Considering that JTT does not show significant performance gain in the results, it is consistent with the existing results that the validation dataset is important in JTT, as described in (Idrissi et al., 2022).

[3]Lee et al. (2021) only reported bias-conflicting case for BFFHQ, but we report the unbiased test result.

Table 3: Average and worst test accuracy with the raw image benchmark: **CelebA** and raw NLP task: **CivilComments-WILDS**. The results of comparison algorithms for CelebA † and CivilComments-WILDS‡ are the results reported in (Seo et al., 2022) and (Zhang et al., 2022), respectively. The best worst accuracy is indicated in **bold**.

		Vanilla	LfF	GEORGE	BPA	EIIL	JTT	CNC	Ours
CelebA†	Avg.	80.52	84.89	83.13	90.18	-	-	-	89.27
	Worst	41.02	57.96	65.45	82.54	-	-	-	**82.73**
CivilComments‡	Avg.	92.1	92.5	-	-	90.5	91.1	81.7	92.1
	Worst	58.6	58.8	-	-	67.0	69.3	68.9	**70.6**



(a) Colored MNIST (b) Multi-bias MNIST (c) Corrupted CIFAR

Figure 2: Average PGD results for various of norms, $\{L_1, L_2, L_2^2, L_\infty\}$, for the feature-injected benchmarks. The error bars represent the standard deviation of three independent trials.

PGD shows superior performance over other methods. For example, its performance is higher by 36.15% and 35.63% compared with the performance of vanilla model and Disen for the ratio of 10%.

Similar to the results for the bias-feature-injected benchmarks, as shown in Table 2 and Table 3, PGD shows competitive performance among all the debiasing algorithms on the raw image benchmark (BAR, BFFHQ, and CelebA). For example, for the BFFHQ benchmark, the accuracy of PGD is 1.43% better than that of Disen. As in Table 3, PGD outperforms the other baselines on CivilComments-WILDs, much more realistic NLP task. Therefore, we believe PGD also works well with transformer, and it is applicable to the real-world.

Unbiased test accuracy on various norms. Since, gradient norm can have various candidates, such as order of the norm, we report four configurations of gradient norms. As shown in Figure 2, all norms have significant unbiased test performance. Amongst them, the L_2-norm square case shows lower unbiased performance than the other cases. Therefore, it is recommended that any first power of $L_{\{1,2,\infty\}}$-norms be used in PGD for overcoming the dataset bias problem. This is quite different from the results with (Huang et al., 2021), which suggested that L_1-norm is the best choice in the research field of out-of-distribution detection.

Ablation study. Table 4 shows the importance of each module in our method: generalized cross entropy, and data augmentation modules. We set the ratio ρ as 0.5% and 10% for colored MNIST and multi-bias MNIST, respectively. We observe that GCE is more important that data augmentation for colored MNIST. However, data augmentation shows better performance than GCE for multi-bias MNIST. For all cases, the case where both are utilized outperforms the other cases.

Table 4: Ablation studies on GCE and data augmentation (✓for applied case).

GCE	Aug.	Colored (0.5%)	Multi-bias (10%)
		84.93 ± 0.79	40.58 ± 3.39
✓		93.18 ± 1.07	45.70 ± 2.91
	✓	91.19 ± 0.97	46.70 ± 1.10
✓	✓	**96.88** ± 0.28	**61.38** ± 4.41

5 Mathematical Understanding of PGD

This section provides a theoretical analysis of per-sample gradient-norm based debiasing. We first briefly summarize the maximum likelihood estimator (MLE) and Fisher information (FI) which are ingredients of this section. We then interpret the debiasing problem as a min-max problem and deduce that solving the min-max problem can be phrased as minimizing the trace of the inverse FI. Since handling the trace of the inverse FI is very difficult owing to its inverse computation, we look at a glance by relaxing it into a one-dimensional toy example. In the end, we conclude that the gradient-norm based re-sampling method is an attempt to solve the dataset bias problem.

5.1 Preliminary

Training and test joint distributions. The general joint distribution $\mathcal{P}(x, y|\theta)$ is assumed to be factored into the parameterized conditional distribution $f(y|x, \theta)$ and the marginal distribution $\mathcal{P}(x)$, which is independent of the model parameter θ, *i.e.,* $\mathcal{P}(x, y|\theta) = \mathcal{P}(x)f(y|x, \theta)$. We refer to the

model $f(y|x,\theta^\star)$ that produces the exact correct answer, as an oracle model, and to its parameter $\theta^\star$ as the oracle parameter. The training dataset $\mathcal{D}_n$ is sampled from $\{(x_i,y_i)\}_{i=1}^n \sim p(x)f(y|x,\theta^\star)$, where the training and test marginal distributions are denoted by $p(x)$ and $q(x)$, respectively. Here, we assume that both marginal distributions are defined on the marginal distribution space $\mathcal{M} = \{\mathcal{P}(x) | \int_{x\in\mathcal{X}} \mathcal{P}(x)\,dx = 1\}$, where $\mathcal{X}$ means the input data space, *i.e.,* $p(x),\ q(x) \in \mathcal{M}$.

The space $\mathcal{H}$ of sampling probability h. When the training dataset $\mathcal{D}_n$ is given, we denote the sampling probability as $h(x)$ which is defined on the probability space $\mathcal{H}$[4]:

$$\mathcal{H} = \{h(x) \,|\, \textstyle\sum_{(x_i,y_i)\in\mathcal{D}_n} h(x_i) = 1\,,\ h(x_i) \geq 0 \quad \forall (x_i,y_i) \in \mathcal{D}_n\}. \tag{2}$$

Maximum likelihood estimator (MLE). When $h(x)$ is the sampling probability, we define MLE $\hat{\theta}_{h(x),\mathcal{D}_n}$ as follows:

$$\hat{\theta}_{h(x),\mathcal{D}_n} = \arg\min_\theta \ -\textstyle\sum_{(x_i,y_i)\in\mathcal{D}_n} h(x_i)\log f(y_i|x_i,\theta).$$

Note that MLE $\hat{\theta}_{h(x),\mathcal{D}_n}$ is a variable controlled by two factors: (1) a change in the training dataset $\mathcal{D}_n$ and (2) the adjustment of the sampling probability $h(x)$. If $h(x)$ is a uniform distribution $U(x)$, then $\hat{\theta}_{U(x),\mathcal{D}_n}$ is the outcome of empirical risk minimization (ERM).

Fisher information (FI). FI, denoted by $I_{\mathcal{P}(x)}(\theta)$, is an information measure of samples from a given distribution $\mathcal{P}(x,y|\theta)$. It is defined as:

$$I_{\mathcal{P}(x)}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{P}(x)f(y|x,\theta)}[\nabla_\theta \log\ f(y|x,\theta)\nabla_\theta^\top \log\ f(y|x,\theta)]. \tag{3}$$

FI provides a guideline for understanding the test cross-entropy loss of MLE $\hat{\theta}_{U(x),\mathcal{D}_n}$. When the training set is sampled from $p(x)f(y|x,\theta^\star)$ and the test samples are generated from $q(x)f(y|x,\theta^\star)$, we can understand the test loss of MLE $\hat{\theta}_{U(x),\mathcal{D}_n}$ by using FI as follows.

Theorem 1. *Suppose Assumption 1 in Appendix D and Assumption 2 in Appendix E hold, then for sufficiently large $n = |\mathcal{D}_n|$, the following holds with high probability:*

$$\mathbb{E}_{(x,y)\sim q(x)f(y|x,\theta^\star)}\left[\mathbb{E}_{\mathcal{D}_n\sim p(x)f(y|x,\theta^\star)}\left[-\log f(y|x,\hat{\theta}_{U(x),\mathcal{D}_n})\right]\right] \leq \tfrac{1}{2n}\operatorname{Tr}\left[I_{p(x)}(\hat{\theta}_{U(x),\mathcal{D}_n})^{-1}\right]\operatorname{Tr}\left[I_{q(x)}(\theta^\star)\right]. \tag{4}$$

Here is the proof sketch. The left-hand side of (4) converges to the Fisher information ratio (FIR) $\operatorname{Tr}\left[I_{p(x)}(\theta^\star)^{-1}I_{q(x)}(\theta^\star)\right]$ related term. Then, FIR can be decomposed into two trace terms with respect to the training and test marginal distributions $p(x)$ and $q(x)$. Finally, we show that the term $\operatorname{Tr}[I_{p(x)}(\theta^\star)^{-1}]$ which is defined in the oracle model parameter can be replaced with $\operatorname{Tr}[I_{p(x)}(\hat{\theta}_{U(x),\mathcal{D}_n})^{-1}]$. The proof of Theorem 1 is in Appendix D. Note that Theorem 1 means that the upper bound of the test loss of MLE $\hat{\theta}_{U(x),\mathcal{D}_n}$ can be minimized by reducing $\operatorname{Tr}[I_{p(x)}(\hat{\theta}_{U(x),\mathcal{D}_n})^{-1}]$.

Empirical Fisher information (EFI). In practice, the exact FI (3) cannot be computed since we do not know the exact data generation distribution $\mathcal{P}(x)f(y|x,\theta)$. For practical reasons, the empirical Fisher information (EFI) is commonly used (Jastrzębski et al., 2017; Chaudhari et al., 2019) to reduce the computational cost of gathering gradients for all possible classes when x is given. In this study, we used a slightly more generalized EFI that involved the sampling probability $h(x) \in \mathcal{H}$ as follows:

$$\hat{I}_{h(x)}(\theta) = \textstyle\sum_{(x_i,y_i)\in\mathcal{D}_n} h(x_i)\nabla_\theta \log\ f(y_i|x_i,\theta)\nabla_\theta^\top \log\ f(y_i|x_i,\theta). \tag{5}$$

Note that the conventional EFI is the case when $h(x)$ is uniform. EFI provides a guideline for understanding the test cross-entropy loss of MLE $\hat{\theta}_{h(x),\mathcal{D}_n}$.

5.2 Understanding dataset bias problem via min-max problem

Debiasing formulation from the perspective of min-max problem. Improving generalization is understandable as reducing *(min)* the loss of test samples that are difficult *(max)* to infer. There was a trial for formulating the dataset bias problem when multiple training datasets from different training distributions are given (Arjovsky et al., 2019). The problem is seen differently when we can only control the sampling probability $h(x) \in \mathcal{H}$ on the given one training dataset $\mathcal{D}_n$. The problem, in our view, can be defined from the left-hand side of (4) in a practical point of view as follows:

[4]Note that for simplicity, we abuse the notation $h(x,y)$ used in Section 3 as $h(x)$. This is exactly the same for a given dataset $\mathcal{D}_n$ situation.

Definition 1. When the training dataset $\mathcal{D}_n \sim p(x)f(y|x,\theta^\star)$ is given, the debiasing objective is

$$\min_{h(x)\in\mathcal{H}} \max_{q(x)\in\mathcal{M}} \mathbb{E}_{(x,y)\sim q(x)f(y|x,\theta^\star)} \left[-\log f(y|x,\hat{\theta}_{h(x),\mathcal{D}_n}) \right]. \tag{6}$$

The meaning of Definition 1 is that we have to train the model $\hat{\theta}_{h(x),\mathcal{D}_n}$ so that the loss of the worst case test samples ($\max_{q(x)}$) is minimized by controlling the sampling probability $h(x)$ ($\min_{h(x)}$). Note that since we cannot control the given training dataset $\mathcal{D}_n$ and test marginal distribution $q(x)$, the only controllable term is the sampling probability $h(x)$. Therefore, from Theorem 1 and EFI, we design a practical objective function for the dataset bias problem as follows:

$$\min_{h(x)\in\mathcal{H}} \mathrm{Tr}\left[\hat{I}_{h(x)}(\hat{\theta}_{h(x),\mathcal{D}_n})^{-1} \right]. \tag{7}$$

5.3 Meaning of PGD in terms of (7).

In this section, we present an analysis of PGD with respect to (7). To do so, we try to understand (7), which is difficult to directly solve. It is because computing the trace of the inverse matrix is computationally expensive. Therefore, we intuitively understand (7) in the one-dimensional toy scenario.

One-dimensional example. We assume that $\mathcal{D}_n$ comprises sets M and m such that elements in each set share the same loss function $\frac{1}{2}(\theta+a)^2$ for all elements in M and $\frac{1}{2}(\theta-a)^2$ for all elements in m with a given constant a. In other words, there are only two distinguishable points, and we construct the training data from only these two points. We also assume that each sample of M and m has the set dependent probability mass $h_M(x)$ and $h_m(x)$, respectively. With these settings, our objective is to determine $h^\star(x) = \arg\min_{h(x)\in\mathcal{H}} \mathrm{Tr}[\hat{I}_{h(x)}(\hat{\theta}_{h(x),\mathcal{D}_n})^{-1}]$. Thanks to the model's simplicity, we can easily find $h^\star(x)$ in a closed form with respect to the gradient at $\hat{\theta}_{U(x),\mathcal{D}_n}$ for each set, *i.e.*, $g_M(\hat{\theta}_{U(x),\mathcal{D}_n})$ and $g_m(\hat{\theta}_{U(x),\mathcal{D}_n})$.

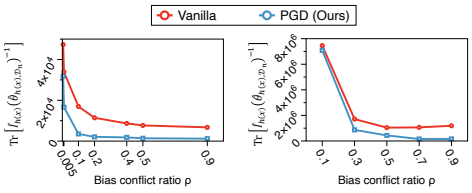
Theorem 2. *Under the above setting, the solution of* $(h^\star_M(x), h^\star_m(x)) = \arg\min_{h(x)\in\mathcal{H}} \mathrm{Tr}[\hat{I}_{h(x)}(\hat{\theta}_{h(x),\mathcal{D}_n})^{-1}]$ *is:*

$$h^\star_M(x) = |g_M(\hat{\theta}_{U(x),\mathcal{D}_n})|/Z, \quad h^\star_m(x) = |g_m(\hat{\theta}_{U(x),\mathcal{D}_n})|/Z,$$

where $Z = |M||g_M(\hat{\theta}_{U(x),\mathcal{D}_n})| + |m||g_m(\hat{\theta}_{U(x),\mathcal{D}_n})|$, *and* $|M|$ *and* $|m|$ *denote the cardinality of* M *and* m, *respectively.*

The proof of Theorem 2 is provided in Appendix E. Note that $h^\star_M(x)$ and $h^\star_m(x)$ are computed using the trained biased model with batches sampled from the uniform distribution $U(x)$. It is the same with the second step of PGD.

PGD tries to minimize (7). Theorem 2 implies that (7) can be minimized by sampling in proportion to their gradient norm. Because the fundamental of PGD is oversampling based on the gradient norm from the biased model, we can deduce that PGD strives to satisfy (7). Furthermore, we empirically show that PGD reduces the trace of the inverse of EFI in the high-dimensional case, as evident in Figure 3.



(a) Colored MNIST (b) Multi-bias MNIST

Figure 3: Target objective $\mathrm{Tr}[\hat{I}_h(\hat{\theta}_{h(x),\mathcal{D}_n})^{-1}]$. PGD : $h(x) = \hat{h}(x)$, and vanilla: $h(x) = U(x)$.

6 Related Work

Debiasing with bias label. In (Goyal et al., 2017; 2020), a debiased dataset was generated using human labor. Various studies (Alvi et al., 2018; Kim et al., 2019; McDuff et al., 2019; Singh et al., 2020; Teney et al., 2021) have attempted to reduce dataset bias using *explicit bias labels*. These studies (Alvi et al., 2018; Kim et al., 2019; McDuff et al., 2019; Singh et al., 2020; Li et al., 2018; Li & Vasconcelos, 2019), used bias labels for each sample to reduce the influence of the bias labels when classifying target labels. Furthermore, Tartaglione et al. (2021) proposed the EnD regularizer, which entangles target correlated features and disentangles biased attributes. Several studies (Alvi et al., 2018; Kim et al., 2019; Teney et al., 2021) have designed DNNs as a shared feature extractors and multiple classifiers. In contrast to the shared feature extractor methods, McDuff et al. (2019) and Ramaswamy et al. (2021) fabricated a classifier and conditional

generative adversarial networks, yielding test samples to determine whether the classifier was biased. Furthermore, Singh et al. (2020) proposed a new overlap loss defined by a class activation map (CAM). The overlap loss reduces the overlapping parts of the CAM outputs of the two bias labels and target labels. The authors of (Li & Vasconcelos, 2019; Li et al., 2018) employed bias labels to detect bias-conflicting samples and to oversample them to debias. In (Liu et al., 2021), a reconstructing method based on the sample accuracy was proposed. The authors of Liu et al. (2021) used bias labels in the validation dataset to tune the hyper-parameters. On the other hand, there has been a focus on fairness within each attribute (Hardt et al., 2016; Woodworth et al., 2017; Pleiss et al., 2017; Agarwal et al., 2018). Their goal is to prevent bias attributes from affecting the final decision of the trained model.

Debiasing with bias context. In contrast to studies assuming the explicit bias labels, a few studies (Geirhos et al., 2018; Wang et al., 2018; Lee et al., 2019; Bahng et al., 2020; Cadene et al., 2019; Clark et al., 2019) assumed that the bias context is known. In (Geirhos et al., 2018; Wang et al., 2018; Lee et al., 2019), debiasing was performed by directly modifying known context bias. In particular, the authors of (Geirhos et al., 2018) empirically showed that CNNs trained on ImageNet (Deng et al., 2009) were biased towards the image texture, and they generated stylized ImageNet to mitigate the texture bias, while Lee et al. (2019) and Wang et al. (2018) inserted a filter in front of the models so that the influence of the backgrounds and colors of the images could be removed. On the other hand, some studies (Bahng et al., 2020; Clark et al., 2019; Cadene et al., 2019), mitigated bias by reweighting bias-conflicting samples: Bahng et al. (2020) used specific types of CNNs, such as BagNet (Brendel & Bethge, 2018), to capture the texture bias, and the bias was reduced using the Hilbert-Schmidt independence criterion (HSIC). In the visual question answering (VQA) task, Clark et al. (2019) and Cadene et al. (2019) conducted debiasing using the entropy regularizer or sigmoid output of the biased model trained on the fact that the biased model was biased toward the question.

Debiasing without human supervision. Owing to the impractical assumption that bias information is given, recent studies have aimed to mitigate bias without human supervision (Le Bras et al., 2020; Nam et al., 2020; Darlow et al., 2020; Kim et al., 2021; Lee et al., 2021). Le Bras et al. (2020) identified bias-conflicting samples by sorting the average accuracy of multiple train-test iterations and performed debiasing by training on the samples with low average accuracy. In (Ahmed et al., 2020), each class is divided into two clusters based on IRMv1 penalty (Arjovsky et al., 2019) using the trained biased model, and train the debiased model so that the output of two clusters become similar. Furthermore, Kim et al. (2021) used Swap Auto-Encoder (Park et al., 2020) to generate bias-conflicting samples, and Darlow et al. (2020) proposed the modification of the latent representation to generate bias-conflicting samples by using an auto-encoder. Lee et al. (2021) and Nam et al. (2020) proposed a debiasing algorithm weighted training by using a relative difficulty score, which is measured by the per-sample training loss. Specifically, Lee et al. (2021) used feature mixing techniques to enrich the dataset feature information. Seo et al. (2022) and Sohoni et al. (2020) proposed unsupervised clustering based debiasing method. Recently, contrastive learning based method (Zhang et al., 2022) and self-supervised learning method (Kim et al., 2022) are proposed. On the other hand, there have been studies (Li & Xu, 2021; Lang et al., 2021; Krishnakumar et al., 2021) that identify the bias attribute of the training dataset without human supervision.

7 Conclusion

We propose a gradient-norm-based dataset oversampling method for mitigating the dataset bias problem. The main intuition of this work is that gradients contains abundant information about each sample. Since the bias-conflicting samples are relatively more difficult-to-learn than bias-aligned samples, the bias-conflicting samples have a higher gradient norm compared with the others. Through various experiments and ablation studies, we demonstrate the effectiveness of our gradient-norm-based oversampling method, called PGD. Furthermore, we formulate the dataset bias problem as a min-max problem, and show theoretically that it can be relaxed as minimizing the trace of the inverse Fisher information. Furthermore, we provide empirical and theoretical evidence that PGD tries to solve the minimizing the trace of the inverse Fisher information problem. Despite this successful outcome and analysis, we are still working on two future projects: release approximations such as a toy example for understanding PGD and cases where the given training dataset is corrupted, such as with noisy labels. We hope that this study will help improve understanding of researchers about the dataset bias problem.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69. PMLR, 2018.

Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. In *International Conference on Learning Representations*, 2020.

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

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

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.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019. URL http://arxiv.org/abs/1903.04561.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2018.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems*, pp. 839–850, 2019.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Kamalika Chaudhuri, Sham Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation, 2015.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

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

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

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

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

Ankit Goyal, Kaiyu Yang, Dawei Yang, and Jia Deng. Rel3d: A minimally contrastive benchmark for grounding spatial relations in 3d. *Advances in Neural Information Processing Systems*, 33, 2020.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10951–10960, 2020.

Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34, 2021.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pp. 336–351. PMLR, 2022.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

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.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. *arXiv preprint arXiv:2012.10630*, 2020.

Krishnateja Killamsetty, S Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021b.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9012–9020, 2019.

Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. *arXiv preprint arXiv:2108.10008*, 2021.

Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. *arXiv preprint arXiv:2206.10843*, 2022.

Arvindkumar Krishnakumar, Viraj Prabhu, Sruthi Sudhakar, and Judy Hoffman. Udis: Unsupervised discovery of bias in deep visual recognition models. In *British Machine Vision Conference (BMVC)*, volume 1, pp. 3, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 693–702, 2021.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, pp. 1078–1088. PMLR, 2020.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

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, 2021.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *International Conference on Learning Representations*, 2019.

Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer-Verlag, New York, NY, USA, second edition, 1998.

Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9572–9581, 2019.

Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 513–528, 2018.

Zhiheng Li and Chenliang Xu. Discover the unknown biased attribute of an image classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14970–14979, 2021.

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.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 3730–3738. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.425. URL https://doi.org/10.1109/ICCV.2015.425.

Daniel McDuff, Shuang Ma, Yale Song, and Ashish Kapoor. Characterizing bias in classifiers using generative models. In *Advances in Neural Information Processing Systems*, pp. 5404–5415, 2019.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL http://arxiv.org/abs/1802.03426. cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Jun Hyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *34th Conference on Neural Information Processing Systems (NeurIPS) 2020*. Neural Information Processing Systems, 2020.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *arXiv preprint arXiv:2007.00653*, 2020.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9301–9310, 2021.

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.

Roland Schäfer. On bias-free crawling and representative web corpora. In *Proceedings of the 10th web as corpus workshop*, pp. 99–105, 2016.

Seonguk Seo, Joon-Young Lee, and Bohyung Han. Unsupervised learning of debiased representations with pseudo-attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16742–16751, June 2022.

Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Robik Shrestha, Kushal Kafle, and Christopher Kanan. An investigation of critical issues in bias mitigation techniques. *arXiv preprint arXiv:2104.00170*, 2021.

Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11070–11078, 2020.

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.

Jamshid Sourati, Murat Akcakaya, Todd K. Leen, Deniz Erdogmus, and Jennifer G. Dy. Asymptotic analysis of objectives based on fisher information in active learning, 2016.

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.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. *arXiv preprint arXiv:2105.05612*, 2021.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

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

Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer, New York, 2004. ISBN 978-1-4419-2322-6. doi: 10.1007/978-0-387-21736-9.

Blake E. Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *COLT*, 2017. URL https://www.semanticscholar.org/paper/Learning-Non-Discriminatory-Predictors-Woodworth-Gunasekar/00cda3a1c7f716d6136f0d3c0c1fe1046e685e82.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

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 R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.