Denoising after Entropy-Based Debiasing a Robust Training Method for Dataset Bias with Noisy Labels

Sumyeong Ahn, Se-Young Yun

Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea
{sumyeongahn, yunseyoung}@kaist.ac.kr

Abstract

Improperly constructed datasets can result in inaccurate inferences. For instance, models trained on biased datasets perform poorly in terms of generalization (*i.e., dataset bias*). Recent debiasing techniques have successfully achieved generalization performance by underestimating easy-to-learn samples (*i.e., bias-aligned samples*) and highlighting difficult-to-learn samples (*i.e., bias-conflicting samples*). However, these techniques may fail owing to noisy labels, because the trained model recognizes noisy labels as difficult-to-learn and thus highlights them. In this study, we find that earlier approaches that used the provided labels to quantify difficulty could be affected by the small proportion of noisy labels. Furthermore, we find that running denoising algorithms before debiasing is ineffective because denoising algorithms reduce the impact of difficult-to-learn samples, including valuable bias-conflicting samples. Therefore, we propose an approach called denoising after entropy-based debiasing, *i.e.,* `DENEB`, which has three main stages. (1) The prejudice model is trained by emphasizing (bias-aligned, clean) samples, which are selected using a Gaussian Mixture Model. (2) Using the per-sample entropy from the output of the prejudice model, the sampling probability of each sample that is proportional to the entropy is computed. (3) The final model is trained using existing denoising algorithms with the mini-batches constructed by following the computed sampling probability. Compared to existing debiasing and denoising algorithms, our method achieves better debiasing performance on multiple benchmarks.

1 Introduction

Deep neural networks (DNNs) have achieved human-like performance in various tasks, such as image classification (He et al. 2016), image generation (Goodfellow et al. 2014), and object detection (He et al. 2017), but require well-organized training datasets for success. For example, the trained model might have prejudices when its training dataset is biased. In real life, we often encounter *dataset bias problems* (Bahng et al. 2020; Nam et al. 2020; Lee et al. 2021; Kim, Lee, and Choo 2021). For example, as shown in Figure 1, the dataset for classifying camels in images could be very biased, as most camel images are captured against a desert background (*i.e.,* bias-aligned); only a few images are captured against other backgrounds, such as forests (*i.e.,* bias-conflicting).

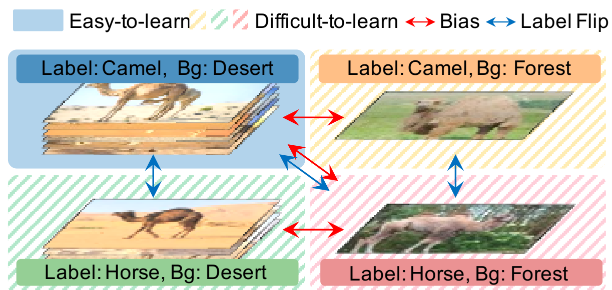


Figure 1: Examples of biased dataset with noisy labels. (1) Blue: bias-aligned, clean label samples. (2) Orange: bias-conflicting, clean labels samples. (3) Green: bias-aligned, noisy labels. (4) Red: bias-conflicting, noisy labels. The dashed background represents difficult-to-learn samples. Therefore, except for (bias-aligned, clean) case, other cases are difficult-to-learn. To mitigate dataset bias with noisy labels, training directions for each type differ. For example, (bias-aligned, noisy) case must be discarded or cleansed, while (bias-conflicting, clean) cases have to be emphasized.

This unintended bias causes the trained model to infer erroneously based on shortcuts (*i.e.,* background). To mitigate such dataset bias, previous researches have used the fact that bias-conflicting samples are more difficult-to-learn than bias-aligned samples (Nam et al. 2020). Various approaches have been proposed, such as adjusting the loss function (Bahng et al. 2020; Nam et al. 2020; Creager, Jacobsen, and Zemel 2021), feature disentanglement (Lee et al. 2021), creating mixed-attribute samples (Kim, Lee, and Choo 2021), or reconstructing balanced datasets (Liu et al. 2021; Ahn, Kim, and Yun 2023) to emphasize difficult-to-learn samples.

In addition to dataset bias, noisy labels are caused by many reasons (Yu et al. 2018; Nicholson, Sheng, and Zhang 2016) and are known to degrade training mechanisms. For example, in Figure 1, the *camel* in the bottom row can be labeled *horse* by human error. Numerous studies have focused on alleviating the impact of noisy labels or directly correcting them. Some (Bahri, Jiang, and Gupta 2020; Zhang et al. 2020; Veit et al. 2017; Ren et al. 2018; Hendrycks et al. 2018) deal with the problem of noisy labels by assuming clean data to set training guidelines (*i.e.,* purifying a given

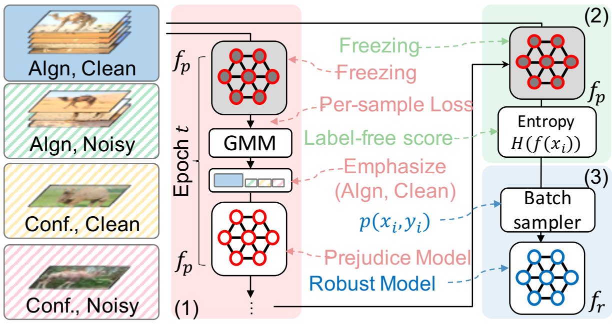


Figure 2: Overview of DENEB. It is composed of three steps. (1) Train by emphasizing (bias-aligned, clean) samples. (2) Compute label-free score, *i.e.,* entropy. (3) Train the final robust model based on the batch sampler.

corrupted dataset by using a model trained on a small clean dataset). Recently, various methods have been proposed to relax this strong clean subset assumption by taking advantage of the characteristic that noisy labels are more difficult-to-learn than clean samples. For example, such difficult-to-learn samples are guided by regularizers (Liu et al. 2020; Cao et al. 2019, 2020), giving lower weights (Wang et al. 2019; Zhang and Sabuncu 2018), cleansing out (Mirzasoleiman, Cao, and Leskovec 2020; Wu et al. 2020; Pleiss et al. 2020; Han et al. 2018; Yu et al. 2019), or utilizing a semi-supervised learning algorithm by considering them as unlabeled samples (Kim et al. 2021; Li, Socher, and Hoi 2019).

Although dataset bias and noisy labels can occur simultaneously and independently, few studies (Creager, Jacobsen, and Zemel 2021)[1] have addressed both problems at once. This is because the fundamental solutions of each problem are exact opposites. Difficult-to-learn samples have to be *emphasized* to mitigate dataset bias (Nam et al. 2020; Lee et al. 2021), while their influence should be *reduced* for denoising (Han et al. 2018; Zhang and Sabuncu 2018). Dataset bias and noisy labels can occur concurrently in the real-world. Therefore, both problems must be handled together.

Contribution. We present a training method that addresses *dataset bias with noisy labels (DBwNL)*. We first empirically analyze why existing debiasing and denoising algorithms fail to achieve their respective objectives. In this regard, we discover two facts. (1) Existing debiasing methods using *given labels* suffer from problems when the training dataset contains noisy labels because they determine the degree of being emphasized based on a given label (e.g., per-sample loss). (2) Denoising methods eliminate valuable bias-conflicting samples (*i.e.,* samples that should be emphasized for debiasing). This is because their denoising mechanism cleans or discards difficult-to-learn samples without considering whether a sample is bias-conflicting or noisy label.

Based on these findings, we propose an algorithm, coined as DENEB, which **<u>den</u>**oising after **<u>e</u>**ntropy-based de**<u>b</u>**iasing (see Figure 2). The proposed method consists of three stages. The first stage trains a prejudice model biased toward (bias-aligned, clean) data. To this aim, DENEB uses the Gaussian Mixture Model (GMM) based on per-sample losses to split (bias-aligned, clean) and the others at the beginning of each epoch. In the next stage, DENEB measures the entropy for each sample from the prejudice model. From the per-sample entropy, DENEB calculates the sampling probability of each sample in proportion to the entropy. The key intuition of the sampling probability is that (bias-aligned, noisy) samples are predicted to have low entropy by the prejudice model because it mainly learns (bias-aligned, clean) samples, but the excluded bias-conflicting samples will have a large entropy prediction. Note that the sampling probability is obtained without using the given labels as they might be corrupted. Finally, DENEB trains the ultimate robust model on the sampled mini-batches, where the sampling probabilities are obtained during Step 2.

We demonstrate the efficacy of DENEB on a variety of biased datasets, including Colored MNIST (Nam et al. 2020; Bahng et al. 2020; Kim, Lee, and Choo 2021; Lee et al. 2021), Corrupted CIFAR-10 (Nam et al. 2020; Lee et al. 2021), Biased Action Recognition (BAR) (Nam et al. 2020; Kim, Lee, and Choo 2021), and Biased Flickr-Faces-HQ (Lee et al. 2021; Kim, Lee, and Choo 2021), with symmetric noisy labels. Compared to the existing debiasing, denoising, and naive combination of both algorithms, the proposed method achieves a successful debiasing performance for all benchmarks. For example, DENEB improves the unbiased test accuracy from 39.24% to 91.81% on a colored MNIST dataset with 1% bias scenario with 10% noisy labels and BAR from 54.37% to 62.30% compared to the vanilla model.

2 Dataset Bias with Noisy Labels (DBwNL)

In this section, we define the dataset bias problem and the noisy label separately. Subsequently, we describe *dataset bias with noisy labels*, which is when dataset bias and noisy label problems occur in conjunction.

Dataset Bias. Consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ in which each input is x_i and its corresponding truth label $y_i = \{1, ..., C\}$. Each sample can be described by a set of attributes. For example, the images in Figure 1 can have `background`, `object`, and so on. For convenience of explanation, we look at the top of Figure 1 (Blue and Orange cases), without the noisy label case. The objective, *i.e.,* the *target attribute*, is to classify the "camel." When most of the samples have attributes that are strongly correlated with the target, we call the phenomenon *dataset bias* and these attributes *bias attributes*. In Figure 1, the bias-attribute "desert background," and the target attribute "camel" are highly correlated, *i.e.,* almost "camel" images are captured against "desert background." We call samples whose bias attribute is highly correlated (weakly correlated) with the target attribute *bias-aligned* (*bias-conflicting*) samples. This dataset bias problem is quite harmful when the bias attribute is easier to learn than the target attributes, because the model loses the motivation to learn the target attribute given its sufficiently low loss. We focus on the case where the bias attribute is easier to learn than the target. For convenience, we denote the set of

[1]EIIL aimed to study dataset bias problem without human supervision. They partially analyzed the impact of noisy labels in their synthetic benchmark only.

bias-conflicting and bias-aligned samples by $\mathcal{D}_c$ and $\mathcal{D}_a$ as they are clearly distinct, but both sets need not be strictly separable. Note that the portion of bias-conflicting sample is called the bias conflict ratio α, which is defined as:

$$\alpha = \frac{|\mathcal{D}_c|}{|\mathcal{D}|}.$$

Noisy Labels. Collected labels may be corrupted. If a person labels the image x, the provided label can be corrupted *i.e.,* $y_{\text{given}} \neq y$, even though the true label is y. We call samples whose labels are $y_{\text{given}} = y$ and $y_{\text{give}} \neq y$ *clean label* and *noisy label*, respectively. As shown in Figure 1, the lower row represents noisy label cases. For example, although the images in Figure 1 of the bottom boxes are "camel", they are labeled as "horse". We denote the portion of the samples whose labels are flipped as the noise ratio η. For convenience, we focus on the cases where the label corruption occurs symmetrically.

$$y_{\text{given}} = \begin{cases} \tilde{y} \sim \text{Uniform}(C). & \text{with probability } \eta \\ y & \text{with probability } 1-\eta \end{cases},$$

DBwNL. DBwNL cases occur sequentially, gathering images x and labeling y. As mentioned above, the biased dataset has a small portion of bias-conflicting samples, *i.e.,* α is small. Therefore, most of the samples in the given training dataset are bias-aligned. Training a robust model on the DBwNL dataset emphasizes bias-conflicting samples while discarding or reducing the impact of the noisy labels.

3 Failure to Debias on a DBwNL Dataset

In this section, we briefly summarize existing debiasing methods and demonstrate that they are vulnerable to noisy labels.
Brief summary of the previous methods. In previous debiasing algorithms, bias-conflicting samples are highlighted based on each proposed score. Almost all previous approaches train a biased model f_b on the given training dataset, and a debiased model f_d is trained with emphasis in the following ways:

- Relative difficulty score (`LfF` (Nam et al. 2020), `Disen` (Lee et al. 2021))

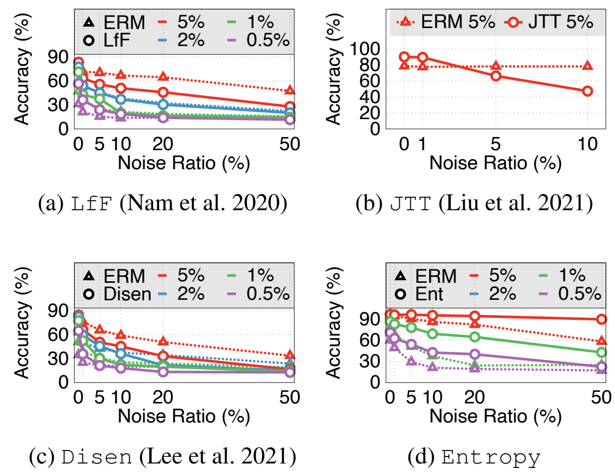
$$\mathcal{W}(x, \underline{\boldsymbol{y}_{\text{given}}}) = \frac{\mathcal{L}_{\text{CE}}(f_b(x), \underline{\boldsymbol{y}_{\text{given}}})}{\mathcal{L}_{\text{CE}}(f_b(x), \underline{\boldsymbol{y}_{\text{given}}}) + \mathcal{L}_{\text{CE}}(f_d(x), \underline{\boldsymbol{y}_{\text{given}}})}, \quad (1)$$

  where $\mathcal{L}_{\text{CE}}$ denotes conventional cross-entropy loss and $f_b(\cdot)$ and $f_d(\cdot)$ are softmax outputs of biased and debiased models, respectively.

- Per-sample accuracy (`JTT` (Liu et al. 2021))

$$\mathcal{D}_{\text{error-set}} = \{(x, y) \text{ s.t. } \underline{\boldsymbol{y}_{\text{given}}} \neq \arg\max_c f_b(x)[c]\}, \quad (2)$$

  where $f_b(x)[c]$ denotes the softmax output of logit c. The ultimate debiased model is trained on $\mathcal{D}_{\text{train}}$ composed of λ_{up} times $\mathcal{D}_{\text{error-set}}$ and the other $\mathcal{D}_{\text{corr-set}} = \mathcal{D} \setminus \mathcal{D}_{\text{error-set}}$.



(a) `LfF` (Nam et al. 2020) (b) `JTT` (Liu et al. 2021)

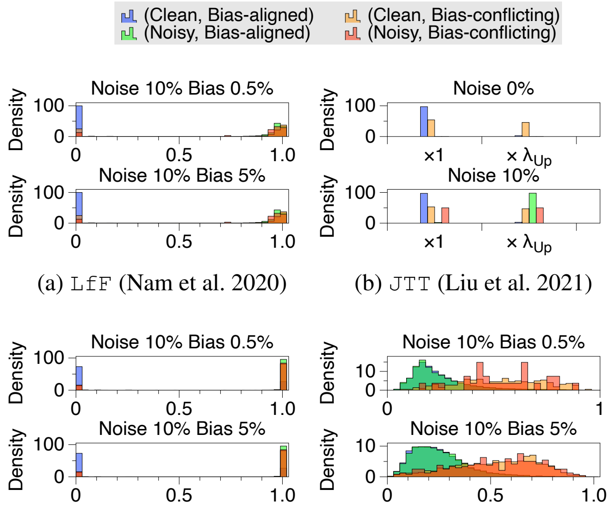
(c) `Disen` (Lee et al. 2021) (d) `Entropy`

Figure 3: Performance when label corruption occurs. In the case of `LfF` and `Disen`, it is the unbiased test accuracy of colored MNIST, and `JTT` is the worst case test performance of the waterbird dataset. The triangle-dotted lines are the vanilla results, and the circle-solid lines represent the result of each algorithm. All algorithms except for entropy case perform worse than vanilla as noise ratio η increases.

3.1 Debiasing Meets Noisy Labels

As in (1) and (2), all previous techniques are based on the given label y_{given}. Here, we refer to methods using (x, y_{given}) and only (x) respectively as "*label-based debiasing*" and "*label-free debiasing*." We observe the ultimate performance of previous methods when noisy labels are injected. We used the settings offered by their official repositories[2,3,4], such as dataset, implementation, and hyperparameters except for label flipping. For comparison, we include entropy-based debiasing, which highlights samples with proportion to the per-sample entropy score. It does not require the given label, *i.e.,* label-free method. Detail description about entropy-based setting is described in Appendix.
Label-based debiasing is prone to noisy labels. As shown in Figure 3, the performances of the label-based techniques are lower than those of the vanilla case; a small noise ratio occurs. However, as demonstrated in Figure 3d, the label-free method performs better than in the vanilla case. This is because label-based methods make incorrect emphasis, $\mathcal{W}(x, y_{\text{given}})$ and $\mathcal{D}_{\text{error-set}}$, when y_{given} is corrupted.
Why do label-based methods suffer side-effects? As in Figure 4a 4b, and 4c, the label-based scores of the (conflicting, clean) and (aligned, noisy) are entangled. This means that noisy labels are also emphasized when we run the labe-based algorithms. By contrast, the label-free method, *i.e.,* `Entropy` in Figure 4d, shows that the bias-conflicting and bias-aligned samples are easily distinguished regardless

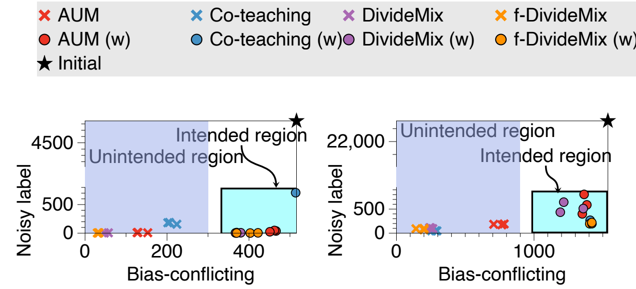
[2] https://github.com/alinlab/LfF
[3] https://github.com/anniesch/jtt
[4] https://github.com/kakaoenterprise/Learning-Debiased-Disentangled



(a) LfF (Nam et al. 2020) (b) JTT (Liu et al. 2021)

(c) Disen (Lee et al. 2021) (d) Entropy

Figure 4: Score histogram of each methodology. As LfF and Disen operate online, we report the weight histogram right after the last epoch of training. Except for the Entropy case, LfF, JTT, and Disen shows entangled histograms between (noisy, aligned), (clean, conflicting), and (noisy, conflicting). By contrast, the histogram of entropy case indicates that it is clustered not according to label corruption but bias.



(a) $\alpha = 1\%, \eta = 10\%$ (b) $\alpha = 5\%, \eta = 50\%$

Figure 5: Number of remaining noisy labels and bias-conflicting samples after denoising is conducted. Star $\star$ mark represents the number of samples before cleansing, and ✗ and ● marks indicate with or without weighted training results. Since bias-conflicting samples is precious for debiasing, bias-conflicting samples have to be protected. Therefore, the region loses bias-conflicting samples (left, blue) is the unintended region. On the other hand, the region ignores noisy labels without losing the bias-conflicting samples (right, cyan) is the intended behavior.

of whether their labels are noisy. In conclusion, a label-free method is needed to handle DBwNL.

4 DENoising after Entropy-based DeBiasing

In Section 3, we verified that the debiasing algorithms do not work properly for DBwNL alone. In this section, we analyze how debiasing algorithms should be combined with denoising algorithms, and finally propose DENEB.

4.1 How Denoising Algorithms Work in the DBwNL

We first check how the denoising algorithms work in the DBwNL dataset by observing two cases.

Observation Setting. Five denoising methods are used for the analysis: AUM (Pleiss et al. 2020), Co-teaching (Han et al. 2018), DivideMix (Li, Socher, and Hoi 2019), and f-DivideMix (Kim et al. 2021). We measure the number of samples of (noisy, aligned) and (clean, conflicting) after running the denoising algorithms. We run two types of tests. (1) Without modification (✗): to check how the denoising algorithms handle bias-conflicting samples. (2) Manually weighted training (●): we assign $\times 50$ weights to bias-conflicting samples, *i.e.,* $50 \times \mathcal{L}_{\text{CE}}(x, y), \forall (x, y) \in \mathcal{D}_c$. The second case is unrealistic as we cannot know which sample is bias-conflicting, but the result can convey the following argument: if we want to protect bias-conflicting samples from the discarding by the denoising mechanism, make the bias-conflicting samples easy-to-learn.

Valuable bias-conflicting samples can be deemed noisy. As illustrated in Figure 5, all denoising methods sufficiently differentiate noisy samples. For example, $\star$ in Figure 5a represents that the initial number of noisy samples is almost $5,000$, but almost all methods dropped to near 0 after denoising (✗ marks). However, crucial bias-conflicting samples are also eliminated, *i.e.,* ✗ marks are in the "unintended region." Therefore, utilizing denoising algorithms before debiasing can discard valuable bias-conflicting samples. As in Figure 3, because the number of bias-conflicting samples is critical, removing the bias-conflicting samples prior to debiasing can cause performance degradation.

Preventing bias-conflicting samples from being discarded. Bias-conflicting samples are considered noisy labels because the trained model thinks that they are difficult-to-learn. As illustrated in Figure 5 using ● marks, when we sufficiently highlight bias-conflicting samples, noisy labels can be eliminated by reducing the loss of bias-conflicting samples. Therefore, we can conclude that denoising should be performed after highlighting bias-conflicting samples.

4.2 Designing the Algorithm for DBwNL

Based on the previous experimental results, two inferences can be drawn in designing a debiasing algorithm for the DBwNL datasets. (1) No label-based: label-based debiasing emphasizes noisy labels. (2) Debiasing before denoising: denoising algorithms should be run after debiasing emphasizes bias-conflicting samples. We summarize our intuitions in Figure 6. For the DBwNL dataset, the main consideration is whether to apply debiasing or denoising first. If denoising is applied first, the bias-conflicting samples is erased, which is burdensome for debiasing (see the lower α cases in Figure 3). Conversely, if debiasing is conducted first, we can choose label-based or label-free. If a label-based algorithm is selected, the noisy labels are enlarged and a burden is placed on the denoising algorithm (see the higher η cases in Figure 3).

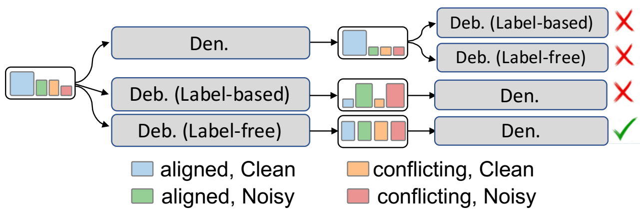


Figure 6: Case study of designing algorithm for DBwNL.

In other words, emphasizing the bias-conflicting sample and proceeding with denoising without emphasizing the noisy label through the label-free algorithm is the correct order.

4.3 Denoising after Entropy-Based Debiasing

Based on the case study, we propose denoising after entropy-based debiasing, `DENEB`, which is composed of three steps.

Step 1: Train the prejudice model f_p. The key aim while training the prejudice model f_p is that regardless of label corruption, the model should comprehensively learn the bias-aligned samples so that it can identify the bias-conflicting samples in the next steps. However, it is difficult to detach (bias-aligned, noisy) from the bias-conflicting samples. Therefore, `DENEB` trains the prejudice model on (bias-aligned, clean) only. Intuitively speaking, if the model is trained using only (bias-aligned,clean) samples, (bias-aligned,noisy) samples also can be regarded as easy-to-learn thanks to the bias attributes. `DENEB` finds (bias-aligned, clean) by using the GMM, similar with (Li, Socher, and Hoi 2019; Kim et al. 2021). Step 1 consists of two sub-steps. At first, f_p is trained on $\mathcal{D}$ with conventional cross-entropy loss, until the warm-up epoch e_w. After the warm-up phase, `DENEB` splits $\mathcal{D}$ and obtains $\bar{\mathcal{D}}$ at the beginning of each epoch. To do so, `DENEB` dynamically fits a GMM on per-sample losses and obtains $\bar{\mathcal{D}}$ whose probability of GMM $g(x_i, y_i)$ is higher than the threshold p_t at the beginning of each epoch:

$$\bar{\mathcal{D}} = \{(x_i, y_i) | g(x_i, y_i) > p_t, \text{ where } (x_i, y_i) \in \mathcal{D}\}, \quad (3)$$

Note that, unlike `DivideMix` and `f-DivideMix`, `DENEB` does not use the samples whose $g(x_i, y_i) \leq p_t$, to deepen bias, *i.e.,* it ignores every-types except for (bias-aligned,clean).

Step 2: Calculate sampling probability. Based on the trained prejudice model f_p, we extract the entropy score for each sample:

$$H_\tau(x) = -\sum_{c}^{C} f_p(x, \tau)[c] \times \log f_p(x, \tau)[c], \quad (4)$$

where $f_p(x, \tau)[j]$ is the temperature-scaled softmax for class c with temperature parameter τ, *i.e.,* $f_p(x, \tau)[j] = \frac{\exp(q_p(x)[j]/\tau)}{\sum_c \exp(q_p(x)[c]/\tau)}$ with logit $q_p(x)$. Based on $H_\tau(x)$, we find the sampling probability of each instance as follows:

$$\mathcal{P}(x_i, y_i) = \frac{H_\tau(x_i)}{\sum_{(x_j, y_j) \in \mathcal{D}} H_\tau(x_j)}. \quad (5)$$

The reason why $\mathcal{P}(x_i, y_i)$ is proportional to the entropy score is because f_p is sufficiently biased and thus the larger entropy samples are the bias-conflicting samples (see Figure 4d).

Step 3: Train the robust model f_r. To train the robust model f_r, mini-batches are constructed based on the sampling probability in equation 5. As mini-batches contain sufficient bias-conflicting samples, the main purpose of this step is to mitigate the impact of noisy labels. To this end, we inherit previous denoising algorithms by simply modifying the mini-batches. Note that `DENEB` can utilize any given denoising algorithm, $\mathcal{A}_{\text{den}}$, but we report based on `GCE` which performs better than the others. As analysis of various denoising algorithms is reported in Section 5.

5 Experiment

The effectiveness of the proposed algorithm is analyzed quantitatively and qualitatively. We compare `DENEB` to earlier debiasing and denoising techniques in four biased datasets, *i.e.,* Colored-MNIST (CMNIST), Corrupted CIFAR-10 (CCIFAR), Biased Action Recognition (BAR), and Biased FFHQ (BFFHQ). To test the generalization performance, we report the unbiased test accuracy. Details of the implementation and datasets are given in Appendix.

Dataset	train/valid/test	#class	Target	Bias
CMNIST	54K / 6K / 10K	10	Shape	Color
CCIFAR	45K / 5K / 10K	10	Object	Blur
BAR	1,746 / 195 / 654	6	Action	Place
BFFHQ	17,280/1,920/1,000	2	Gender	Age

Table 1: Benchmark Summary

5.1 Experimental Settings

Baselines. We report the performance of the debiasing and denoising algorithms. As debiasing algorithms, we use recent methods that are officially available, such as `LfF`, `JTT`[5], `EIIL`, and `Disen`. We utilize denoising algorithms `GCE`, `SCE`, `ELR+`, `Co-teaching`, `DivideMix`, and `f-DivideMix`. All implementations are reproduced following the official codes. The implementation and hyperparameters are reported in Appendix.

Datasets. We employ four benchmarks: CMNIST, CCIFAR, BAR, and BFFHQ. The target and bias attributes are summarized in Table 1. For the CMNIST and CCIFAR datasets, two pairs of bias-ratio $(\alpha, \eta) = \{(1\%, 10\%), (5\%, 50\%)\}$ are utilized. The other datasets are tested on $\eta = 10\%$. We summarize in detail the construction recipe in Appendix.

CMNIST and CCIFAR. CMNIST and CCIFAR datasets have a bias attribute, which is injected manually. The goal of CMNIST is to classify the target attribute, `digit shape`, when the bias attribute is `color`. This dataset comes from (Nam et al. 2020; Bahng et al. 2020; Lee et al. 2021; Kim, Lee, and Choo 2021). In CCIFAR, the target attribute

[5]As (Liu et al. 2021) assume that a balanced validation dataset, it is unfair to directly compare with the other algorithms. However, since `JTT` can be tuned using noisy biased validation dataset, we report the behavior of `JTT` tuned by using a biased noisy vallidation.

Algorithm	Colored MNIST		Corrupted CIFAR-10	
	$\alpha = 1\%, \eta = 10\%$	$\alpha = 5\%, \eta = 50\%$	$\alpha = 1\%, \eta = 10\%$	$\alpha = 5\%, \eta = 50\%$
Vanilla	39.24 $\pm$ 1.91%	70.13% $\pm$ 3.42 %	25.43% $\pm$ 0.84 %	31.86% $\pm$ 0.96 %
Debiasing				
LfF	29.87 $\pm$ 1.36%	57.97% $\pm$ 1.79 %	24.51% $\pm$ 1.30 %	29.68% $\pm$ 2.63 %
JTT	63.24 $\pm$ 2.60%	77.16% $\pm$ 1.15 %	23.75% $\pm$ 0.61 %	24.52% $\pm$ 0.98 %
EIIL	24.53 $\pm$ 0.31%	42.25% $\pm$ 1.43 %	20.30% $\pm$ 1.08 %	22.66% $\pm$ 1.94 %
Disen	31.49 $\pm$ 5.44%	69.20% $\pm$ 4.13 %	22.52% $\pm$ 0.38 %	28.35% $\pm$ 4.49 %
Denoising				
GCE	19.52 $\pm$ 1.98%	73.45% $\pm$ 7.62 %	24.96% $\pm$ 1.53 %	30.72% $\pm$ 0.74 %
SCE	30.95 $\pm$ 2.87%	62.10% $\pm$ 5.02 %	23.34% $\pm$ 1.73 %	29.87% $\pm$ 1.00 %
ELR+	24.76 $\pm$ 0.90%	49.38% $\pm$ 3.74 %	22.10% $\pm$ 0.37 %	30.84% $\pm$ 0.43 %
AUM	23.89 $\pm$ 2.60%	49.51% $\pm$ 6.62 %	23.55% $\pm$ 1.10 %	28.06% $\pm$ 2.38 %
Co-teaching	41.89 $\pm$ 1.45%	76.64% $\pm$ 5.52 %	25.14% $\pm$ 0.27 %	26.84% $\pm$ 0.52 %
DivideMix	20.48 $\pm$ 1.94%	33.66% $\pm$ 2.91 %	18.86% $\pm$ 0.28 %	22.03% $\pm$ 0.59 %
f-DivideMix	22.06 $\pm$ 1.70%	39.92% $\pm$ 3.26 %	19.67% $\pm$ 0.25 %	27.60% $\pm$ 0.54 %
DENEB				
DENEB	**91.81 $\pm$ 0.84%**	**94.55% $\pm$ 0.22 %**	**26.05% $\pm$ 0.54 %**	**35.32% $\pm$ 1.03 %**

Table 2: Unbiased test accuracy on CMNIST and CCIFAR. Best-performing results are marked in bold. All results are averaged on three independent runs. DENEB represents *i.e.,* $\mathcal{A}_{\text{den}} = \text{GCE}$.

Algorithm	BAR	BFFHQ
	$\eta = 10\%$	$\eta = 10\%$
Vanilla	54.37 $\pm$ 1.10%	71.38 $\pm$ 0.58%
LfF	53.62 $\pm$ 1.81%	54.35 $\pm$ 0.91%
JTT	55.67 $\pm$ 2.16%	70.18 $\pm$ 1.47%
Disen	55.80 $\pm$ 3.05%	67.44 $\pm$ 2.57%
GCE	56.39 $\pm$ 0.95%	68.45 $\pm$ 2.98%
Co-teaching	54.99 $\pm$ 1.28%	69.28 $\pm$ 1.24%
DivideMix	52.01 $\pm$ 1.51%	72.20 $\pm$ 0.58%
DENEB	**62.30 $\pm$ 0.91%**	**75.24 $\pm$ 0.68%**

Table 3: Unbiased test accuracy on BAR and BFFHQ. Best performing results are marked in bold. All results are averaged on three independent runs.

is objective such as {airplane, car,...} with the bias attribute corruption like {blur, ...}. We generate CCIFAR following the bias injection mechanism of (Nam et al. 2020; Lee et al. 2021; Hendrycks and Dietterich 2018).

BAR and BFFHQ. BAR and BFFHQ are consists of real-world images. These benchmarks are biased when selecting samples by seeing the multiple attributes. BAR (Nam et al. 2020) aims to classify actions such as {racing, climbing,...} with background bias. For example, (Climbing, Rockwall) are bias-aligned samples, while (Climbing, Ice-cliff) are the bias-conflicting ones. BFFHQ (Kim, Lee, and Choo 2021; Lee et al. 2021) aims to classify gender when its age is biased. For example, the training dataset is made up of (Female, Young (age ranging from 10 to 29)) and (Male, Old (age ranging from 40 to 59)) and very few of (Female, Old) and (Male, Young) samples.

Implementation details. For the Colored MNIST, we use a Simple-ConvNet with three convolutional layers, ReLU activation function (Agarap 2018), batch normalization (Ioffe and Szegedy 2015) and dropout (Srivastava et al. 2014). ResNet-18 (He et al. 2016) pre-trained on ImageNet is used as a backbone network for the rest. Based on grid searches, we find hyperparameters for all algorithms using 90% and 10% training and validation split. This implies that validation datasets contain noisy labels and bias-conflicting samples. The search space and searched hyperparameters are described in Appendix. For all experiments, we report the case where DENEB uses GCE as $\mathcal{A}_{\text{den}}$, which achieves the best performance among all the denoising algorithms.

5.2 Experimental Results

CMNIST and CCIFAR. Table 2 presents comparisons of the accuracy of the unbiased test. Among the debiasing baselines, the accuracy-based algorithm, *i.e.,* JTT, is better than the vanilla model in the CMNIST case. However, all debiasing baselines obtain worse performance than the vanilla model in the CCIFAR case because, as mentioned earlier, the debiasing algorithms highlight noisy labels that should not be emphasized. By contrast, denoising algorithms fail to debias, as they do not have a module to highlight bias-conflicting samples. DENEB achieves the best performance for all injected bias cases. For example, the unbiased test accuracy of CMNIST with $\alpha = 1\%$ and $\eta = 10\%$ shows that DENEB obtains 52.57% gain compared to the Vanilla model.

BAR and BFFHQ. The performances of DENEB in real-world image benchmarks is also better than the other baselines. BAR shows 7.92% improvements over vanilla and 6.5% improvement over Disen, which has the best performance among the debiasing algorithms. DENEB also shows 5.91% performance gain over GCE. Similarly, BFFHQ shows 3.86% improvement over vanilla, 5.06% over JTT and 3.04% over DivideMix. Thus, entropy-based debiases when trained on a more complex raw image dataset.

Combination of Debiasing and Denoising. In order to study how the other debiasing algorithms work with denoising algorithms, *i.e.,* debiasing $\rightarrow$ denoising similarly to DENEB, we report pairwise performance in Figure 7 for

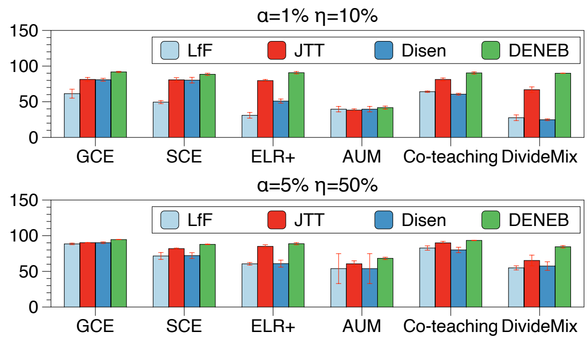


Figure 7: Combination result of Colored MNIST benchmark. All cases are the performances of Debiasing → Denoising, *i.e.,* obtain per-sample weights from `DENEB`and then run `GCE` for `DENEB`→`GCE` case.

CMNIST. As `Disen` and `LfF` are an online algorithms, we multiply the per-sample weight at the end of debiasing by denoising loss. Details are provided in Appendix. As shown in Figure 7, `DENEB` performs better than the other debiasing algorithms for all combinations. `DENEB` has better performance because the side-effects of focusing on the noisy sample are minimized when using the label-free entropy.

6 Related Work

Noisy labels. (Ghosh, Kumar, and Sastry 2017) had proposed the mean absolute error (MAE), and (Zhang and Sabuncu 2018) claim that MAE suffers poor robustness with DNN and suggested another type of cross-entropy (CE) loss, called generalized cross-entropy (GCE). The authors of (Wang et al. 2019) propose symmetry cross-entropy (SCE) loss, which is a combination of conventional CE and reverse cross-entropy. Lukasik et al. (Lukasik et al. 2020) use label smoothing techniques for noisy labels. Recently, studies on the early learning phase have been a topic of extensive interest. These works claim that DNNs memorize difficult-to-learn samples in the later phase and learn common features in the early learning phase. Based on this fact, (Liu et al. 2020) propose the early learning regularizer (ELR) to prohibit memorizing noisy labels. Some works handle noisy labels by detecting and cleansing. To do so, the co-training method, *i.e.,* teaching each other, is mainly used. In (Han et al. 2018) and (Yu et al. 2019) utilize loss and disagreement are utilized to construct a clean subset. (Pleiss et al. 2020) proposes a new metric, area under margin (AUM), to cleanse the dataset. (Li, Socher, and Hoi 2019) look at the noisy label problem as a semi-supervised learning (SSL) approach by dividing the training dataset into clean labeled and noisy unlabeled sets, and running the SSL algorithm (Berthelot et al. 2019). FINE (Kim et al. 2021) uses the alignment of the eigenvector to distinguish clean and noisy samples.

Debiasing with human supervision. (Goyal et al. 2017, 2020) construct a debiased dataset with the human hand. (Alvi, Zisserman, and Nellåker 2018; Kim et al. 2019; McDuff et al. 2019; Singh et al. 2020; Li, Li, and Vasconcelos 2018; Li and Vasconcelos 2019) use bias labels to mitigate the impact of bias labels when classifying target labels. EnD (Tartaglione, Barbano, and Grangetto 2021) proposes to entangle the target attribute and disengle the biased attributes. Multi-expert approaches (Alvi, Zisserman, and Nellåker 2018; Kim et al. 2019; Teney et al. 2021) use a shared feature extrator with multiple FC layers to classify multiple attributes independently. (McDuff et al. 2019; Ramaswamy, Kim, and Russakovsky 2021) use conditional generator to determine if the trained classifier is biased. (Singh et al. 2020) proposes overlap loss, which is measured based on the class activation map. (Li and Vasconcelos 2019) employs bias type to detect bias-conflicting samples and reconstruct balanced dataset. On the other hand, (Geirhos et al. 2018; Wang et al. 2018; Lee et al. 2019) using prior knowledge of the bias context to mitigate dataset bias. (Liu et al. 2021) use bias labels for validation datasets to tune the hyperparameters.

Debiasing without human supervision. To reduce human intervention, (Le Bras et al. 2020; Kim, Lee, and Choo 2021; Idrissi et al. 2022) utilize per-sample accuracy. They regard the inaccurate samples as bias-conflicting. (Lee et al. 2021; Nam et al. 2020) use the loss to calculate the weight. In this case, samples with higher loss from the biased model are overweighted when training the debiased model. (Creager, Jacobsen, and Zemel 2021; Sohoni et al. 2020) infer the bias-conflicting labels and use the predicted labels to mitigate dataset bias problem. (Darlow, Jastrzkebski, and Storkey 2020) generate the samples whose loss becomes large using VAE. (Zhang, Lopez-Paz, and Bottou 2022) propose an initialization point for enlarging the features.

7 Conclusion

Dataset bias with noisy labels can degrade prior debiasing algorithms. To overcome this issue, we propose `DENEB`comprising three stpes. First, the prejudice model is trained on the clean bias-aligned samples. To do so, we utilize a GMM model to select clean bias-aligned samples. After training the prejudice model, `DENEB` compute the entropy score for each sample. This entropy score does not require labels, which can mislead the algorithm into detecting bias-conflicting samples. Based on the obtained entropy score, we compute a sampling probability proportional to the entropy score. To train the final robust model, mini-batches are constructed with sampling probabilities and existing denoising algorithms are run based on the sampled mini-batches. Through extensive experiments across multiple datasets, such as Colored MNIST, Corrupted CIFAR-10, BAR, and BFFHQ, we show that `DENEB`consistently obtains substantial performance improvements compared to the other algorithms for the debiasing, denoising, or naïvely combined method. For future work, we plan to adapt this algorithm to other domains such as NLP, VQA, and so on. We hope that this study opens the door of training a robust model on DBwNL dataset.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%) and the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 90%)

References

Agarap, A. F. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Ahn, S.; Kim, S.; and Yun, S.-Y. 2023. Mitigating Dataset Bias by Using Per-Sample Gradient. In *The Eleventh International Conference on Learning Representations*.

Alvi, M.; Zisserman, A.; and Nellåker, C. 2018. 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)*, 0–0.

Bahng, H.; Chun, S.; Yun, S.; Choo, J.; and Oh, S. J. 2020. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, 528–539. PMLR.

Bahri, D.; Jiang, H.; and Gupta, M. 2020. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, 540–550. PMLR.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32.

Cao, K.; Chen, Y.; Lu, J.; Arechiga, N.; Gaidon, A.; and Ma, T. 2020. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Creager, E.; Jacobsen, J.-H.; and Zemel, R. 2021. Environment inference for invariant learning. In *International Conference on Machine Learning*, 2189–2200. PMLR.

Darlow, L.; Jastrzkebski, S.; and Storkey, A. 2020. Latent Adversarial Debiasing: Mitigating Collider Bias in Deep Neural Networks. *arXiv preprint arXiv:2011.11486*.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Ghosh, A.; Kumar, H.; and Sastry, P. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goyal, A.; Yang, K.; Yang, D.; and Deng, J. 2020. Rel3D: A Minimally Contrastive Benchmark for Grounding Spatial Relations in 3D. *Advances in Neural Information Processing Systems*, 33.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. 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*, 6904–6913.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; and Dietterich, T. G. 2018. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.

Hendrycks, D.; Mazeika, M.; Wilson, D.; and Gimpel, K. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31.

Idrissi, B. Y.; Arjovsky, M.; Pezeshki, M.; and Lopez-Paz, D. 2022. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, 336–351. PMLR.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. PMLR.

Kim, B.; Kim, H.; Kim, K.; Kim, S.; and Kim, J. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9012–9020.

Kim, E.; Lee, J.; and Choo, J. 2021. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14992–15001.

Kim, T.; Ko, J.; Choi, J.; Yun, S.-Y.; et al. 2021. FINE Samples for Learning with Noisy Labels. *Advances in Neural Information Processing Systems*, 34.

Le Bras, R.; Swayamdipta, S.; Bhagavatula, C.; Zellers, R.; Peters, M.; Sabharwal, A.; and Choi, Y. 2020. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, 1078–1088. PMLR.

Lee, J.; Kim, E.; Lee, J.; Lee, J.; and Choo, J. 2021. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34.

Lee, K.; Lee, K.; Shin, J.; and Lee, H. 2019. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *International Conference on Learning Representations*.

Li, J.; Socher, R.; and Hoi, S. C. 2019. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *International Conference on Learning Representations*.

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

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

Liu, E. Z.; Haghgoo, B.; Chen, A. S.; Raghunathan, A.; Koh, P. W.; Sagawa, S.; Liang, P.; and Finn, C. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, 6781–6792. PMLR.

Liu, S.; Niles-Weed, J.; Razavian, N.; and Fernandez-Granda, C. 2020. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33: 20331–20342.

Lukasik, M.; Bhojanapalli, S.; Menon, A.; and Kumar, S. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, 6448–6458. PMLR.

McDuff, D.; Ma, S.; Song, Y.; and Kapoor, A. 2019. Characterizing bias in classifiers using generative models. In *Advances in Neural Information Processing Systems*, 5404–5415.

Mirzasoleiman, B.; Cao, K.; and Leskovec, J. 2020. Coresets for robust training of neural networks against noisy labels. *arXiv preprint arXiv:2011.07451*.

Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684.

Nicholson, B.; Sheng, V. S.; and Zhang, J. 2016. Label noise correction and application in crowdsourcing. *Expert Systems with Applications*, 66: 149–162.

Pleiss, G.; Zhang, T.; Elenberg, E.; and Weinberger, K. Q. 2020. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33: 17044–17056.

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

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, 4334–4343. PMLR.

Singh, K. K.; Mahajan, D.; Grauman, K.; Lee, Y. J.; Feiszli, M.; and Ghadiyaram, D. 2020. 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*, 11070–11078.

Sohoni, N.; Dunnmon, J.; Angus, G.; Gu, A.; and Ré, C. 2020. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33: 19339–19352.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Tartaglione, E.; Barbano, C. A.; and Grangetto, M. 2021. EnD: Entangling and Disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13508–13517.

Teney, D.; Abbasnejad, E.; Lucey, S.; and Hengel, A. v. d. 2021. Evading the Simplicity Bias: Training a Diverse Set of Models Discovers Solutions with Superior OOD Generalization. *arXiv preprint arXiv:2105.05612*.

Veit, A.; Alldrin, N.; Chechik, G.; Krasin, I.; Gupta, A.; and Belongie, S. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 839–847.

Wang, H.; He, Z.; Lipton, Z. C.; and Xing, E. P. 2018. Learning Robust Representations by Projecting Superficial Statistics Out. In *International Conference on Learning Representations*.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 322–330.

Wu, P.; Zheng, S.; Goswami, M.; Metaxas, D.; and Chen, C. 2020. A topological filter for learning with label noise. *Advances in neural information processing systems*, 33: 21382–21393.

Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; and Sugiyama, M. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, 7164–7173. PMLR.

Yu, X.; Liu, T.; Gong, M.; and Tao, D. 2018. Learning with biased complementary labels. In *Proceedings of the European conference on computer vision (ECCV)*, 68–83.

Zhang, J.; Lopez-Paz, D.; and Bottou, L. 2022. Rich Feature Construction for the Optimization-Generalization Dilemma. *arXiv preprint arXiv:2203.15516*.

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

Zhang, Z.; Zhang, H.; Arik, S. O.; Lee, H.; and Pfister, T. 2020. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9294–9303.