A Simple Remedy for Dataset Bias via Self-Influence: A Mislabeled Sample Perspective

Yeonsung Jung[*1], **Jaeyun Song**[*1], **June Yong Yang**[1]
Jin-Hwa Kim[2,3], **Sung-Yub Kim**[1], **Eunho Yang**[1,4]
[1]Korea Advanced Institute of Science and Technology, [2]NAVER AI Lab
[3]Seoul National University, [4]AITRICS
{ys.jung, mercery}@kaist.ac.kr[1]

Abstract

Learning generalized models from biased data is an important undertaking toward fairness in deep learning. To address this issue, recent studies attempt to identify and leverage bias-conflicting samples free from spurious correlations without prior knowledge of bias or an unbiased set. However, spurious correlation remains an ongoing challenge, primarily due to the difficulty in precisely detecting these samples. In this paper, inspired by the similarities between mislabeled samples and bias-conflicting samples, we approach this challenge from a novel perspective of mislabeled sample detection. Specifically, we delve into Influence Function, one of the standard methods for mislabeled sample detection, for identifying bias-conflicting samples and propose a simple yet effective remedy for biased models by leveraging them. Through comprehensive analysis and experiments on diverse datasets, we demonstrate that our new perspective can boost the precision of detection and rectify biased models effectively. Furthermore, our approach is complementary to existing methods, showing performance improvement even when applied to models that have already undergone recent debiasing techniques.

1 Introduction

Deep neural networks have demonstrated remarkable performance in various fields of machine learning tasks comparable to or superior to humans on well-curated benchmark datasets [7, 4, 60, 14]. Nevertheless, the efficacy of these models trained on unfiltered, real-world data remains an open question. In this scenario, a significant concern arises due to the presence of *dataset bias* [52], where task-irrelevant attributes are spuriously correlated with labels only in the training set. This can lead to models that rely on misleading correlations rather than learning the task-related features, resulting in biased models with poor generalization performance [62, 10].

To prevent models from learning detrimental bias, various methods are proposed to encourage models to prioritize learning task-relevant features. Recent studies enhance task-related features by first identifying bias-conflicting (unbiased) samples through loss [40, 36], gradients [1], or bias prediction techniques [35] during training, using an auxiliary biased model trained with Empirical Risk Minimization (ERM). Then, they amplify bias-conflicting samples by counteracting the bias-aligned (biased) samples through loss weighting [40] or weighted sampling [35]. The effectiveness of such methods largely depends on their precision of bias-conflicting sample detection. Specifically, there is a risk of erroneously amplifying malignant bias instead of task-relevant features when bias-aligned samples are inaccurately identified as bias-conflicting. Due to the limited detection performance of previous methods [40, 36, 1, 35], it presents a crucial challenge that remains unresolved.

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1383

In this paper, we address this challenge from a novel perspective of mislabeled sample detection. Inspired by the similarities between mislabeled samples and bias-conflicting samples, we delve into Influence Functions (IF;[27]), one of the standard methods for mislabeled sample detection [55, 57, 29], to identify bias-conflicting samples and propose a simple yet effective approach for biased models by leveraging them.

We first conduct a comprehensive analysis to explore the efficacy of Self-Influence (SI) [27], a variant of IF, in biased datasets. SI estimates how removing a specific training sample during training influences the prediction of the sample itself with the trained model (see Section. 2.2). By measuring SI, we can identify a minority sample that, if removed from the training set, increases the likelihood of incorrect predictions of itself by the trained model due to their discrepancies with the majority samples. In this context, leveraging SI to biased datasets is promising as bias-conflicting samples constitute the minority and contradict the dominant malignant bias learned by the model. However, we observe that unlike in mislabeled settings, directly applying SI to biased datasets is not as effective (Figure 1(a)-1(d)). Therefore, we investigate the differences between mislabeled samples and bias-conflicting samples and reveal the essential conditions for SI to effectively identify bias-conflicting samples. Note that we denote SI under found conditions as Bias-Conditioned Self-Influence (BCSI).

Building on our analysis, we propose a simple yet effective method for rectifying biased models through fine-tuning. We construct a small pivotal subset with a higher proportion of bias-conflicting samples using BCSI. While not perfect, this pivotal set can serve as an effective alternative to an unbiased set. Leveraging this pivotal set, we rectify a biased model through fine-tuning with only a few additional iterations. Extensive experiments demonstrate that our method can effectively rectify even after models are already debiased by recent methods.

Our contributions are threefold:

- We conduct a comprehensive analysis to explore the efficacy of SI in biased datasets and reveal the essential conditions for SI to accurately differentiate bias-conflicting samples, leading to Bias-Conditioned Self-Influence (BCSI).
- We propose a simple yet effective remedy through fine-tuning that utilizes a pivotal set constructed using BCSI to rectify biased models across varying bias severities.
- Our method is complementary to existing methods, capable of further rectifying models that have already undergone recent debiasing techniques.

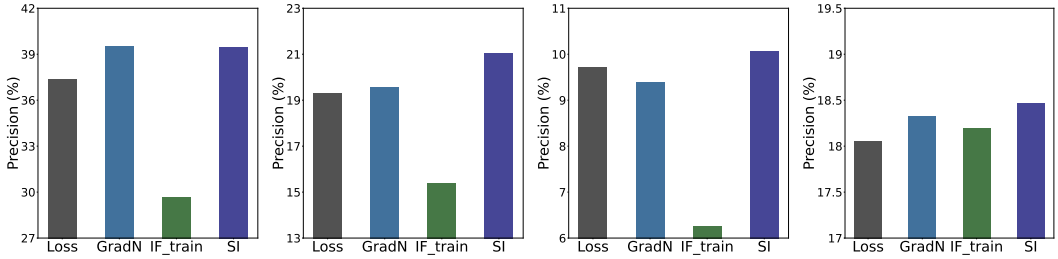
2 Background

2.1 Learning from biased data

We consider a supervised learning setting with training data $D = \{z_n\}_{n=1}^{N}$ sampled from the data distribution $\mathbf{Z} := (X, Y)$, where the input X is comprised of $X = (S, B, O)$ where S is the task-related signal, B is a task-irrelevant bias, and O is the other task-independent feature. Also, Y is the target label of the task, where the label is $y \in \{1, \dots, C\}$. When the dataset is unbiased, ideally, a model learns to predict the target label using the task-relevant signal: $P_\theta(Y|X) = P_\theta(Y|S, B, O) = P_\theta(Y|S)$. However, when the dataset is biased, the task-irrelevant bias B is highly correlated with the task-relevant features S with probability p_y, *i.e.*, $P(B = b_y|S = s_y) = p_y$, where $p_y \geq \frac{1}{C}$. Under this relationship, a data sample $x = (s, b, o)$ is *bias-aligned* if $(b = b_y) \wedge (s = s_y)$ and, *bias-conflicting* otherwise, where $\wedge$ denotes the logical conjunction. When B is easier to learn than S for a model, the model may discover a shortcut solution to the given task, learning to predict $P_\theta(Y|X) = P(Y|B)$ instead of $P_\theta(Y|X) = P(Y|S)$. However, debiasing a model inclines the model towards learning the true task-signal relationship $P_\theta(Y|X) \approx P(Y|S)$.

2.2 Influence Functions

Influence Function (IF; [11, 27]) estimates the impact of an individual sample from the training set on the model parameters, which in turn influences model predictions. A brute-force approach to assess the influence of a sample is to exclude the data point from the training set and retrain the model to compare differences in performance, referred to as leave-one-out (LOO) retaining. However, performing LOO retraining for all samples is computationally challenging; as an alternative, influence functions have been introduced as an efficient approximated method.



(a) CMNIST (1%). (b) CIFAR10C (1%). (c) BFFHQ. (d) Waterbird.

Figure 1: Precision of detecting bias-conflicting samples among Loss, Gradient Norm, Influence function on training set (IF_{train}), and Self-Influence (SI). The precision is evaluated with the ground truth number of bias-conflicting samples. The average precision of loss value, gradient norm, SI, and IF are presented in bars across three runs.

Here, we review the formal definition of influence function. Given a training dataset $D = \{z_n\}_{n=1}^N$ where $z_n = (x_n, y_n)$, model parameters θ are learned with a loss function $\mathcal{L}$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(D, \theta) = \underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \ell(z_n, \theta)$$

where $\ell(z_n, \theta) = -\log(P_\theta(y_n|x_n))$ is the cross-entropy loss for z_n with parameter θ.

To measure the impact of a single training sample z on model parameters θ, we consider the retrained parameter $\theta^*_{z,\epsilon}$ obtained by up-weighting the loss of z by ϵ:

$$\theta^*_{z,\epsilon} = \underset{\theta}{\operatorname{argmin}}(\mathcal{L}(D, \theta) + \epsilon \cdot \ell(z, \theta)).$$

Then, IF, the impact of z on another sample z', is defined as the deviation of the retrained loss $\ell(z', \theta^*_{z,\epsilon})$ from the original loss $\ell(z', \theta^*)$:

$$\mathcal{I}_\epsilon(z, z') = \ell(z', \theta^*_{z,\epsilon}) - \ell(z', \theta^*)$$

For infinitesimally small ϵ, we have

$$\mathcal{I}(z, z') = \left.\frac{d\mathcal{I}_\epsilon(z, z')}{d\epsilon}\right|_{\epsilon=0} = \nabla_\theta \ell(z', \theta^*)^\top H^{-1} \nabla_\theta \ell(z, \theta^*) \tag{1}$$

where $H := \nabla^2_\theta \mathcal{L}(D, \theta^*) \in \mathbb{R}^{P \times P}$ is the Hessian of the loss function with respect to the model parameters at θ^*. Intuitively, the influence $\mathcal{I}(z, z')$ estimates the effect of z on z' through the learning process of the model parameters. Note that IF is commonly computed once a model has converged since Equation 1 approximates more accurately when the average gradient norm of the training set is sufficiently small.

Influence function also can be calculated on itself to measure the Self-influence of z:

$$\mathcal{I}_{\texttt{self}}(z) \approx \nabla_\theta \ell(z, \theta^*)^\top H^{-1} \nabla_\theta \ell(z, \theta^*),$$

which approximates the difference in loss of z when z itself is excluded from the training set. This metric is commonly used for detecting mislabeled training samples in the noisy label setting [27, 51, 55, 57, 29] or important samples in data pruning for efficient training [49, 59].

3 An analysis of Self-Influence in bias-conflicting sample detection

In this section, we conduct a comprehensive analysis to delve into the efficacy of SI in bias-conflicting sample detection. First, we examine the process of identifying bias-conflicting sample detection through the perspective of mislabeled sample detection (Section 3.1). Next, we introduce essential conditions required for SI to effectively identify bias-conflicting samples by analyzing the differences between mislabeled and bias-conflicting samples (Section 3.2). We term the SI calculated under these conditions as Bias-Conditioned Self-Influence (BCSI) and demonstrate that BCSI outperforms SI in detecting bias-conflicting samples.

https://doi.org/10.52202/079017-1383

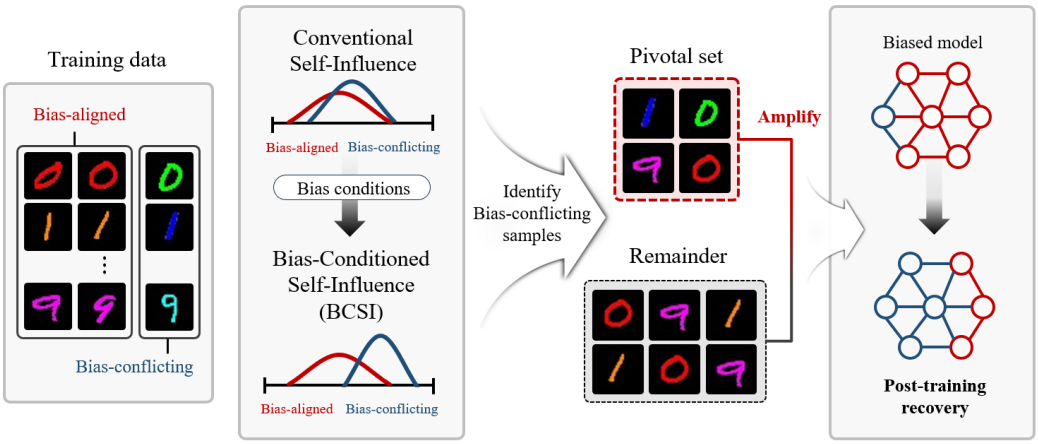


Figure 2: The overview of our method. We compute Bias-Conditioned Self-Influence (BCSI) of the training data and construct a small but concentrated pivotal set with a high ratio of bias-conflicting samples. Then, we remedy biased models through fine-tuning that utilizes the pivotal set and remaining samples.

3.1 Bias-conflicting sample detection from the perspective of mislabeled sample detection

IF is one of the standard methods for mislabeled sample detection [27]. The use of influence functions for mislabeled sample detection generally involves two approaches: computing influence scores using a clean validation set or computing self-influence scores. The former, $\mathcal{I}(z_i, \mathcal{V})$, utilizes a validation set $\mathcal{V}$ free of mislabeled samples to measure the impact on validation loss, identifying samples whose removal reduces this loss as likely mislabeled. The latter, Self-influence $\mathcal{I}(z_i, z_i)$, estimates how the loss of a sample z_i changes when it is removed from the training set. If removing a sample significantly increases its own loss, it indicates that the sample is likely mislabeled, as normal samples can still be correctly predicted using the remaining samples. For instance, in a task classifying dogs and cats, if a dog image is mislabeled as a cat, removing this mislabeled sample from the training set decreases the likelihood of correctly predicting it as a cat.

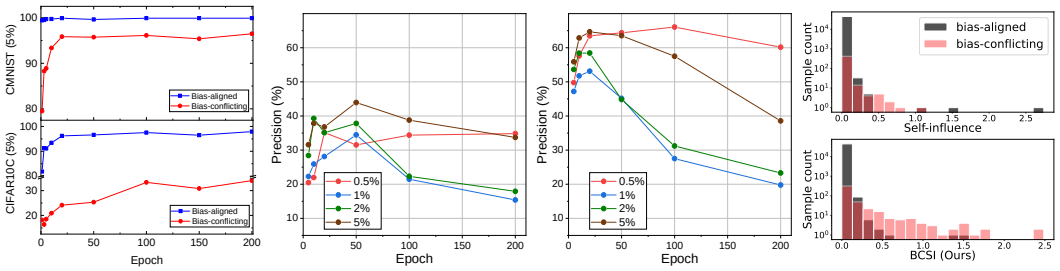
In this context, mislabeled samples and bias-conflicting samples share a key characteristic that both are minority samples contradicting the dominant features learned by the model. Mislabeled samples have incorrect labels that conflict with the learned features, making them easily identifiable through SI. Similarly, bias-conflicting samples contradict the malignant bias that a model learns from a biased dataset. Despite the different contexts, both types of samples can be detected through the same underlying principle of IF.

In summary, given the similarities between mislabeled samples and bias-conflicting samples, it is promising to leverage the perspective and methodology of mislabeled sample detection to identify bias-conflicting samples. However, in real-world scenarios, preemptively identifying malignant bias and constructing an unbiased validation set to mitigate the bias problem is impractical. Therefore, using self-influence offers a more feasible and practical solution for addressing bias-conflicting samples instead of using influence scores on a validation set. Consequently, we center our approach on SI to effectively detect bias-conflicting samples.

3.2 Bias-Conditioned Self-Influence (BCSI)

To validate Self-Influence (SI) in detecting bias-conflicting samples, we conduct experiments on benchmark datasets with diverse bias types and severities: Colored MNIST, Corrupted CIFAR10, Biased FFHQ (BFFHQ), and Waterbird. These datasets feature bias related to color, synthetic corruption, gender, and place background, respectively (details in Appendix N.1). In contrast to the mislabeled setting, we observe that directly applying SI to detect bias-conflicting samples in biased datasets often fails. In Figure 1, the detection precision of SI is significantly low, mostly below 25%. Note that since an unbiased validation set is unavailable in our target problem, we additionally estimate the influence score on the training set, indicated as IF_{train} in Figure 1.

https://doi.org/10.52202/079017-1383



(a) Classification Acc. (b) Detection precision of IF_{train} (c) Detection precision of SI (d) Histograms of SI and BCSI.

Figure 3: A comprehensive analysis of Influence function on the training set (IF_{train}) and Self-Influence (SI) in biased datasets. Figure 3(a) shows the classification accuracy of bias-aligned and bias-conflicting samples over training epochs. Figure 3(b) and 3(c) depict the detection precision of IF_{train} and SI across training epochs for varying ratios of bias-conflicting samples in CIFAR10C. Figure 3(d) shows histograms of sample distribution in CIFAR10C (1%) and each bar indicates the number of samples within a specific range.

This is due to the inherent differences between mislabeled samples and bias-conflicting samples. While mislabeled samples strongly conflict with the dominant features learned by the model due to their incorrect labels, bias-conflicting samples share task-related features with bias-aligned samples. For instance, in a biased dataset where seagulls are spuriously correlated with sea backgrounds, a seagull image against a desert background still retains the features of a seagull. Despite the dominance of malignant bias, these features are still partially utilized. Therefore, bias-conflicting samples do not exhibit a clear contrast with the dominant features of a biased model, posing a challenge for using SI.

To address this challenge, we introduce essential conditions that enable SI to accurately detect bias-conflicting samples. The key concept is to restrict the model from learning task-related features and instead induce the model to focus more on the malignant bias to achieve better separation. A simple but effective way to attain this is by leveraging models in the early stages of training, since malignant bias is learned first, followed by task-related features later, according to Nam et al. [40]. In Figure 3(a), experiments on CIFAR10C and CMNIST demonstrate that the classification accuracy of bias-aligned samples increases rapidly, while that of bias-conflicting samples shows a slower rise. In addition, as shown in Figure 3(b) and 3(c), our experiments on CIFAR10C with diverse ratios of bias-conflicting samples (0.5%, 1%, 2%, and 5%) demonstrate a significant decline in detection precision of IF and SI as training epochs increase, since the model gradually learns task-related features. Therefore, computing SI with models in the early stages of training can achieve better separation. Formally, given a model parameterized by θ at an early epoch t, we compute the self-influence $\mathcal{I}_{\texttt{self}}(z)$ as:

$$\mathcal{I}_{\texttt{self}}(z) = \nabla_{\theta_t}\ell(z,\theta_t)^\top H_t^{-1}\nabla_{\theta_t}\ell(z,\theta_t), \tag{2}$$

where H_t is the Hessian of the loss function at the parameter θ_t.

To further enhance the separation of SI, we employ Generalized Cross Entropy (GCE) [61] to induce the model to focus more on the easier-to-learn bias-aligned samples, resulting in a more biased model. GCE emphasizes samples that are easier to learn, thereby amplifying the model's bias by tending to give more weight to bias-aligned samples in the training set.

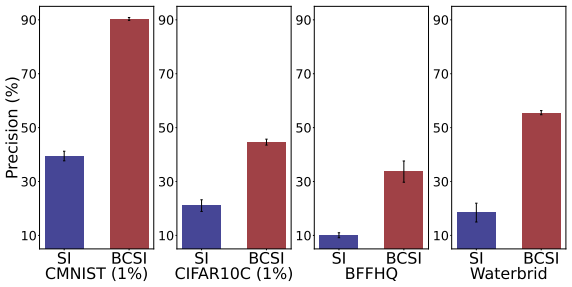


Figure 4: Comparison of average precision between SI and BCSI across diverse datasets over three runs.

Consequently, we employ the model trained under these conditions to measure SI and refer to SI estimated by this heavily biased model with Equation 2 as Bias-Conditioned Self-Influence (BCSI). Since we induce the model to heavily exploit bias and discourage the model from learning task-related features, BCSI can effectively detect bias-conflicting samples. To avoid the impracticality of manually searching epoch t for each dataset, we base our method on the well-known findings of Frankle et al.

https://doi.org/10.52202/079017-1383

(a) Bias-conflicting with high BCSI (b) Bias-conflicting with low BCSI (c) Bias-aligned with high BCSI (d) Bias-aligned with low BCSI

Figure 5: Example images from BFFHQ ranked within the top 100 by BCSI score. (a) and (b) are bias-conflicting samples with high and relatively lower BCSI scores, respectively. (c) is a bias-aligned sample with a high BCSI score, while (d) is a bias-aligned sample with a low BCSI score.

[9] that the primary directions of the model's parameter weights are determined during 500 to 2,000 iterations. Thus, we set epoch t within this range according to the mini-batch size of each dataset. Specifically, we used t=5 for all datasets to ensure practicability and consistency across experiments, but fine-tuning the epoch t for each dataset can yield further improvement.

We validate the efficacy of BCSI in detecting bias-conflicting samples. Since calculating $H^{-1} := (\nabla^2_\theta L(D, \theta^*))^{-1}$ is computationally expensive for large networks due to their extensive number of parameters, we calculate H^{-1} and the loss gradient of the sample z, $\nabla_\theta \ell(z, \theta^*)$, by using the last layer of the model following Koh and Liang [27], Pruthi et al. [43]. In Figure 4, BCSI outperforms conventional SI in detection precision.

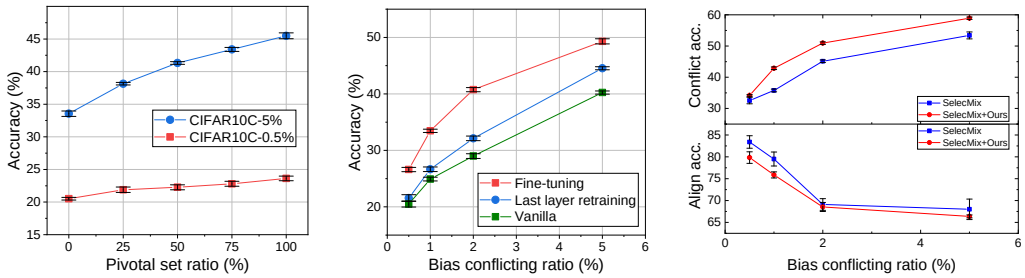
Additionally, Figure 3(d) demonstrates that BCSI has a notable tendency for bias-conflicting samples to exhibit larger scores compared to bias-aligned samples, in contrast to SI. This trend is also observed in other biased datasets, as shown in Appendix A and C. These experimental results support that BCSI can serve as an effective indicator for identifying bias-conflicting samples. To further analyze the qualitative characteristics of bias-conflicting and bias-aligned samples within the top 100 samples ranked by BCSI, we examine BCSI on BFFHQ, as illustrated in Figure 5. In BFFHQ, gender serves as the bias attribute and age as the target attribute, leading to spurious correlations between 'young' and 'woman' as well as 'old' and 'man'. For bias-conflicting samples, Figure 5(a) shows that BCSI assigns high scores to clear counterexamples, such as boys or very elderly women. In contrast, Figure 5(b) exhibits relatively lower BCSI scores for cases like slightly older young men or elderly women who appear younger, indicating that BCSI prioritizes samples with stronger opposition to spurious correlations. A similar trend is observed for bias-aligned samples in Figure 5(c) and Figure 5(d), enhancing that BCSI effectively distinguishes between varying degrees of alignment with the spurious correlations.

4 Remedy biased models through fine-tuning

In this section, we propose a simple but effective remedy that first utilizes BCSI to construct a concentrated pivotal subset abundant in bias-conflicting samples and then employs it for rectifying biased models via fine-tuning without leveraging the supervision of bias or an unbiased validation set. Our method is complementary to existing methods, capable of rectifying models that have already undergone other debiasing techniques. The overall pipeline is described in Figure 2.

Constructing a pivotal subset. We select the top-k subset of samples from each class, based on their BCSI scores, to form a pivotal subset of bias-conflicting samples as follows: $\mathbf{Z}_{\mathrm{P}} =$

https://doi.org/10.52202/079017-1383



(a) Acc. of last layer retraining. (b) Acc. of fine-tuning. (c) Performance gain.

Figure 6: Test accuracy under varying bias-conflicting ratios. Figure 6(a) shows the accuracy for last layer retraining across varying bias ratios in pivotal sets. Figure 6(b) depicts performance changes of last layer retraining and fine-tuning under diverse bias ratios. In Figure 6(c), our performance gains are provided. We present the average accuracy with the error bars indicating the standard error across three runs.

$\bigcup_{c=1}^{C} \left\{ z_{\text{BCSI-rank}(n,c)} \right\}_{n=1}^{k}$, where C is the number of classes and $\text{BCSI-rank}(n, c)$ is the dataset index of the n-th training sample of class c sorted by BCSI score. Due to the unknown ratio of bias-conflicting samples beforehand, determining a proper k through hyper-parameter tuning for each dataset is computationally expensive. To mitigate this, we repeat the selection process three times with different randomly initialized models and use the intersection of the resulting sets as the pivotal set. This ensures a high likelihood of selecting bias-conflicting samples, as they are consistently identified by three runs. We provide the resulting bias-conflicting ratios of the pivotal sets across various datasets in Appendix E and confirm that this process improves detection precision by 16.11% on average. Note that we set $k = 100$ across all datasets in our experiments. For computational costs, since we only train models for five epochs, this iterative approach incurs negligible cost compared to full training, as commonly done in previous studies [40, 31, 19]. We confirm that in Appendix G. The detailed filtering process is outlined in Algorithm 1, which can be found in Appendix B.

Efficient remedy via fine-tuning. Recent works [26, 32] show that retraining specific layers of a model using a small unbiased set can effectively mitigate bias in biased models, overcoming the inefficiency of retraining models from scratch [40, 31, 19]. However, preemptively identifying the bias and curating an unbiased set is very costly, making it an impractical solution. Instead, our method leverages the pivotal set which has a high proportion of bias-conflicting samples, as a practical alternative. While not perfect, our method can efficiently remedy biased models with just a few additional training iterations without the need for prior knowledge of bias or unbiased datasets. As shown in Figure 6(a), even without a perfect pivotal set, its concentration facilitates its applicability in fine-tuning. Note that contrary to the claims of Kirichenko et al. [26], we observed that in highly biased scenarios, the feature extractor also becomes biased, making last-layer retraining insufficient, as demonstrated in Figure 6(b). Therefore, we fine-tune all the parameters in the models.

In addition, we formulate a counterweight cross-entropy loss by drawing a mini-batch from the remaining training set. In real-world scenarios, the unpredictability of bias severity necessitates robustness across a wide range of bias severities. However, previous methods often assume a sufficient presence of bias-aligned samples in the training set, which limits their performance in low-bias scenarios. Despite its significance, the study on both low and high-bias scenarios has been underexplored, and to the best of our knowledge, we are the first to bring up this issue.

We then train the model using both the cross-entropy loss on the pivotal subset and the counterweight loss on the remaining training set:

$$\mathcal{L}(\mathbf{Z}_{\text{P}}, \mathbf{Z}_{\text{R}}) := \mathcal{L}_{\text{CE}}(\mathbf{Z}_{\text{P}}) + \lambda \mathcal{L}_{\text{CE}}(\mathbf{Z}_{\text{R}}),$$

where $\mathbf{Z}_{\text{P}}$ is the pivotal subset, $\mathbf{Z}_{\text{R}} \sim \mathbf{Z} \setminus \mathbf{Z}_{\text{P}}$ a randomly drawn mini-batch from the remaining training set, and $\mathcal{L}_{\text{CE}}$ is the mean cross-entropy loss.

To this end, our method efficiently remedies bias through fine-tuning that utilizes a pivotal set constructed via BCSI across varying bias severities. Additionally, our approach complements existing methods, capable of further rectifying models that have already undergone recent debiasing techniques. The overall process is described in Algorithm 2, which is included in Appendix B.

https://doi.org/10.52202/079017-1383

Table 1: The average and the standard error over three runs. *Ours* indicates our method applied to a model initially trained with the prefix method. The best accuracy is annotated in **bold**. ✓ indicates that a given method uses bias information while ✗ denotes that a given model does not use any bias information.

Method	Bias	CMNIST				CIFAR10C				BFFHQ
	Info	0.5%	1%	2%	5%	0.5%	1%	2%	5%	0.5%
GroupDRO	✓	79.57	90.50	94.89	97.54	33.44	38.30	45.81	57.32	54.80
ERM	✗	71.76±1.84	86.47±0.61	93.87±0.32	96.28±0.29	20.50±0.54	24.91±0.33	28.99±0.42	40.24±0.28	53.53±2.05
LfF	✗	89.06±1.87	89.50±2.88	85.74±4.37	94.30±0.67	25.28±2.89	31.15±1.67	38.64±0.39	46.15±0.54	55.33±2.69
DFA	✗	84.71±1.66	90.20±1.29	92.31±0.77	94.33±1.23	27.13±1.66	31.26±2.71	37.96±0.71	44.99±0.84	52.07±1.91
BPA	✗	73.34±2.37	87.21±0.30	89.42±3.37	97.13±0.15	25.50±1.03	26.86±0.69	27.47±1.46	34.29±2.20	51.40±2.98
DCWP	✗	85.16±7.75	89.68±6.95	89.42±4.23	95.17±1.75	31.27±0.24	34.87±0.63	41.47±0.06	52.86±1.24	57.33±1.75
SelecMix	✗	84.46±0.58	94.51±0.53	95.75±1.34	98.09±0.13	37.63±0.81	40.14±0.42	47.54±0.59	54.86±0.76	63.07±2.32
Ours ERM	✗	75.87±1.60	89.69±0.41	95.08±0.17	96.79±0.13	26.61±0.38	33.47±0.29	40.75±0.37	49.30±0.46	56.00±1.07
Ours LfF	✗	**90.79**±1.13	94.10±1.08	92.95±1.17	95.59±0.53	27.63±1.00	35.29±1.21	43.36±0.78	51.95±0.29	57.13±2.46
Ours DFA	✗	88.39±0.28	92.85±067	95.67±0.12	97.52±0.06	25.66±0.85	33.53±2.01	42.80±0.81	52.61±0.54	56.60±2.83
Ours SelecMix	✗	87.63±1.20	**95.35**±0.17	**97.15**±0.48	**98.13**±0.17	**38.74**±0.36	**46.18**±0.33	**52.70**±0.40	**59.66**±0.31	**65.80**±3.12

5 Experiments

In this section, we present experiments applying our method to models trained with ERM and recent debiasing methods. We validate our method and its individual components by following prior conventions. Below, we provide a brief overview of our experimental setting in Section 5.1, followed by empirical results presented in Section 5.2, 5.3, and 5.4.

5.1 Experimental settings

We now describe datasets, baselines, and evaluation protocol. Detailed descriptions about these are provided in Appendix N.

Datasets. For a fair evaluation, we follow the conventions of using benchmark biased datasets [40]. Colored MNIST dataset (CMNIST) is a synthetically modified MNIST [6], where the labels are correlated with colors. We conduct benchmarks on bias ratios of $r \in \{0.5, 0.1, 0.2, 5\}$. CIFAR10C is a synthetically modified CIFAR10 [30] dataset with common corruptions. To evaluate our method in low-bias scenarios, we expand our scope and conduct experiments with varying bias ratios $r \in \{0.5, 0.1, 0.2, 5, 20, 30, 50, 70, 90(\text{unbiased})\}$. Biased FFHQ (BFFHQ) [31] is a curated Flickr-Faces-HQ (FFHQ) [22] dataset, which consists of facial images where ages and genders exhibit spurious correlation. The Waterbirds dataset [53] consists of bird images, to classify bird types, but their backgrounds are correlated with bird types. Non-I.I.D. Image dataset with Contexts (NICO) [16] is a natural image dataset for out-of-distribution classification. We follow the setting of [54], inducing long-tailed bias proportions within each class, simulating diverse bias ratios in a single benchmark. Additionally, to demonstrate the effectiveness of our method on NLP datasets, we conduct experiments on CivilComments [2, 28] and MultiNLI [58, 45], as detailed in appendix F.

Baselines. Since our goal is addressing the dataset bias without leveraging any prior knowledge of bias or an unbiased set, we evaluate our method with such baselines. GroupDRO [44] uses bias supervision to debias models. LfF [40], BPA [48], and DCWP [41] adjust the loss function to amplify the learning signals for bias-conflicting samples. DFA [31] and SelecMix [19] augment samples possessing various biases different from the original data.

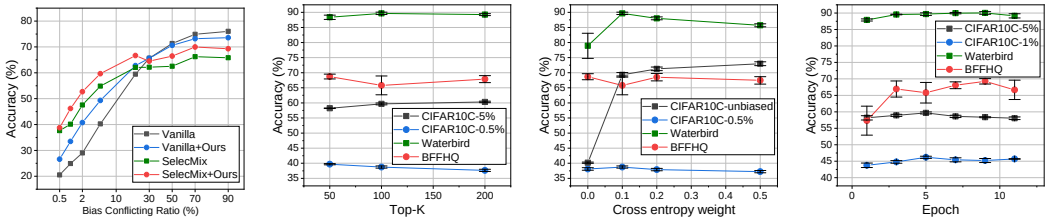
Evaluation protocol. Following other baselines, we calculate the accuracy for unbiased test sets in CMNIST, CIFAR10C, and NICO. We measure the minority-group accuracy in BFFHQ, and the worst-group accuracy in Waterbird. Note that we use the models from the final epoch for all

Table 2: The average and the standard error over three runs on low-bias scenarios.

Method	CIFAR10C				
	20%	30%	50%	70%	90%(unbiased)
ERM	59.47±0.59	65.64±0.51	71.33±0.09	**74.90**±0.25	**76.03**±0.26
LfF	59.78±0.85	60.56±0.96	60.35±0.37	62.52±0.49	63.42±0.63
DFA	60.34±0.46	64.24±0.44	65.97±1.80	64.97±0.20	66.59±5.20
SelecMix	62.05±1.26	62.17±0.35	62.52±1.54	66.23±0.09	65.81±0.96
Ours ERM	62.78±0.67	65.61±0.77	70.61±0.62	73.20±0.35	73.57±0.16
Ours LfF	64.46±0.29	64.40±0.27	65.82±0.15	67.29±0.17	68.15±0.76
Ours DFA	66.30±0.48	**68.13**±0.45	**72.79**±0.38	73.56±0.15	70.36±4.08
Ours SelecMix	**66.67**±0.43	64.51±1.44	66.45±0.28	69.97±0.21	69.29±0.75

Table 3: Accuracy on Waterbirds, NICO

Method	Waterbird	NICO
ERM	68.74±2.65	39.56±1.77
LfF	75.27±2.12	34.56±1.47
DFA	77.57±1.60	44.59±0.33
SelecMix	74.72±1.14	33.87±1.27
Ours ERM	87.64±1.30	43.54±0.50
Ours LfF	87.85±0.68	40.18±0.91
Ours DFA	87.12±0.68	**45.69**±1.12
Ours SelecMix	**89.67**±0.38	44.33±0.55



(a) Acc. on bias ratios. (b) Acc. on varying k. (c) Acc. on varying λ. (d) Acc. on epochs.

Figure 7: Figure 7(a) displays the test accuracy for SelecMix and our method at different bias ratios. Figure 7(b), 7(c), and 7(d) depict the unbiased evaluation under varying the size k in the pivotal set, λ, the number of epochs for detection models, respectively. We present the average accuracy with the error bars indicating the standard error across three runs.

experiments to evaluate performance. We report the average value and the error bars denote standard errors across three runs.

5.2 Results on highly biased scenarios

We evaluate our method to measure the degree of rectification of baseline models when combined with ours on benchmark datasets. In Table 1, we significantly enhance the performance of baselines on the majority of datasets under various experimental settings. Ours $_{\text{SelecMix}}$ achieves state-of-the-art accuracy on CIFAR10C. Also, we observe that performance gain is larger as the ratio of bias-conflicting samples increases in CIFAR10C. We conjecture that fine-tuning becomes more effective in CIFAR10C (2%) and (5%) since the bias-conflicting sample purity of the pivotal set increases, as shown in Section 4. In Figure 6(c), performance gain mainly stems from the increased performance of bias-conflicting samples.

5.3 Results on low-bias scenarios

We validate the baselines on CIFAR10C under varying ratios of bias-conflicting samples in Table 2 and 3. Baselines exhibit performance deterioration compared to ERM when the bias-conflicting ratio is high. In contrast, our method can significantly rectify remaining bias within a model, even in mildly biased datasets except for ERM. Albeit there is a slight decrease in performance for ERM, the accuracy gap is much lower than other baselines. Since the innate nature of fine-tuning can minimize friction by training from pre-trained parameters, our method can remedy bias in a wider range of bias ratios, as in Figure 7(a). The results for other methods are provided in Appendix D.

5.4 Ablation study

We examine the sensitivity of hyperparameters such as the number of selected samples per class (k) in the pivotal set, the weight for the remaining data in fine-tuning (λ), and the number of epochs

https://doi.org/10.52202/079017-1383

to train detection models. In Figure 7(b), there is a slight performance decrease as k increases in CIFAR10C (0.5%). In contrast, the accuracy in CIFAR10C (5%) increases. Since there are a few bias-conflicting samples per class in CIFAR10C (0.5%), additional usage of samples dilutes the ratio of bias-conflicting data in the pivotal set, leading to a performance drop. In Figure 7(c), we observe a marginal accuracy drop as λ increases in CIFAR10C (0.5%), CIFAR10C (90%) experiences a performance increase. These results indicate that learning the remaining samples is beneficial in CIFAR10C (90%), fostering the model to capture task-relevant signals. For the number of epochs used to train the model for the detection, we compute the final performance when combining SelecMix in Figure 7(d). Except for insufficiently trained 1 epoch, the performance is not sensitive to the number of epochs between 3 and 11 epochs. We note that the analysis for intersections is provided in Appendix H.

6 Related work

Debiasing deep neural networks. Research on mitigating bias has centered on modulating task-related information and malignant bias during training. Early works relied on human knowledge through direct supervision or implicit information of bias [44, 33, 18, 12], which is often impractical due to its acquisition cost. Thus, several studies have focused on identifying and utilizing bias-conflicting samples without relying on human knowledge. Loss modification methods [40, 36, 41] amplify the learning signals of (estimated) bias-conflicting samples by modifying the learning objective. Sampling methods [35, 1] overcome dataset bias by sampling (estimated) bias-conflicting data more frequently. Data augmentation approaches [31, 34, 20, 19] synthesize samples with various biases distinct from the inherent bias of the original data. Recently, based on the observation that bias in classification layers is severe compared to feature extractors, several approaches focus on rectifying the last layer [23, 39, 26]. Similarly, Lee et al. [32] demonstrated that selectively fine-tuning a subset of the layers with an unbiased dataset can match or even surpass the performance of commonly used fine-tuning methods. However, identifying the bias and curating an unbiased set is very costly, making it an impractical essential condition.

Influence functions. Influence function (IF) and its approximations [43, 46, 24] have been utilized in various deep learning tasks by measuring the importance of training samples and the relationship between them. One application of IF is in quantifying memorization by self-influence, which is the increase in loss when a training sample is excluded [43, 8]. IF can be used to estimate the significance of samples, enabling the reduction of less important ones for efficient training [49, 59]. Recent works utilize IF to identify and relabel mislabeled samples in noisy label settings [27, 51, 55, 57, 29]. Furthermore, IF has also been applied in 3D domains like NeRF, where it measures pixel-wise distraction caused by unexpected objects, aiding in the identification and mitigation of such distractions [21].

7 Conclusion

In this work, we introduce a novel perspective of mislabeled sample detection on biased datasets. By conducting a comprehensive analysis of Self-Influence in detecting bias-conflicting samples, we discover essential conditions required for SI to effectively identify these samples, which we denote as Bias-Conditioned Self-Influence (BCSI). Building on our analysis, we propose a simple yet effective remedy for biased models through fine-tuning that utilizes a small but concentrated pivotal set constructed via BCSI. Our method is not only capable of further rectifying models that have already undergone recent debiasing techniques but also demonstrates better generalization on a wide range of bias severities compared to previous studies.

Limitations. In this work, we rectify biased models via a simple fine-tuning approach. However, this is the basic method; more sophisticated techniques such as sample weighting or curriculum learning are possible. We believe that our introduction of this novel perspective will pave the way for more advanced future work.

Broader impact. Our work aims to learn unbiased deep learning models without bias annotations. Since filtering every training data under every given circumstance, the social impact of the ability to debias a biased deep learning model after its training is much needed in terms of fairness.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grants (No.2022-0-00713 Meta-learning applicable to real-world problems, No.2022-0-00984 Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation, No.RS-2019-II190075 Artificial Intelligence Graduate School Program(KAIST)), National Research Foundation of Korea (NRF) grants (RS-2023-00209060 A Study on Optimization and Network Interpretation Method for Large-Scale Machine Learning) funded by the Korea government (MSIT), and KAIST-NAVER Hypercreative AI Center.

References

[1] Sumyeong Ahn, Seongyoon Kim, and Se-Young Yun. Mitigating dataset bias by using per-sample gradient. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7mgUec-7GMv.

[2] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[3] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[6] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[8] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Conference on Neural Information Processing Systems (NeurIPS)*, 33:2881–2891, 2020.

[9] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkl1iRNFwS.

[10] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020.

[11] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[12] Xiaochuang Han and Yulia Tsvetkov. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4398–4409, 2021.

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

 https://doi.org/10.52202/079017-1383

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE Computer Society, 2016.

[16] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021.

[17] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *CoRR*, abs/1903.12261, 2019. URL `http://arxiv.org/abs/1903.12261`.

[18] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26449–26461. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/de8aa43e5d5fa8536cf23e54244476fa-Paper.pdf`.

[19] Inwoo Hwang, Sangjun Lee, Yunhyeok Kwak, Seong Joon Oh, Damien Teney, Jin-Hwa Kim, and Byoung-Tak Zhang. Selecmix: Debiased learning by contradicting-pair sampling. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 14345–14357, 2022.

[20] Yeonsung Jung, Hajin Shim, June Yong Yang, and Eunho Yang. Fighting fire with fire: Contrastive debiasing without bias-free data via generative bias-transformation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15435–15450. PMLR, 2023.

[21] Yeonsung Jung, Heecheol Yun, Joonhyung Park, Jin-Hwa Kim, and Eunho Yang. PruNeRF: Segment-centric dataset pruning via 3D spatial consistency. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22696–22709. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/jung24b.html`.

[22] 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*, pages 4401–4410, 2019.

[23] Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL `https://openreview.net/forum?id=bcxmUnTPwny`.

[24] SungYub Kim, Kyungsu Kim, and Eunho Yang. GEX: A flexible method for approximating influence via geometric ensemble. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. URL `https://openreview.net/forum?id=tz4ECtAu8e`.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL `http://arxiv.org/abs/1412.6980`.

[26] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=Zb6c8A-Fghk`.

[27] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/koh17a.html`.

[28] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[29] Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=EskfH0bwNVn.

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

[31] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 25123–25133, 2021.

[32] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=APuPRxjHvZ.

[33] Yi Li and Nuno Vasconcelos. REPAIR: removing representation bias by dataset resampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9572–9581. Computer Vision Foundation / IEEE, 2019.

[34] Jongin Lim, Youngdong Kim, Byungjai Kim, Chanho Ahn, Jinwoo Shin, Eunho Yang, and Seungju Han. Biasadv: Bias-adversarial augmentation for model debiasing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 3832–3841. IEEE, 2023.

[35] Evan Zheran 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 (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 2021.

[36] Sheng Liu, Xu Zhang, Nitesh Sekhar, Yue Wu, Prateek Singhal, and Carlos Fernandez-Granda. Avoiding spurious correlations via logit correction. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=5BaqCFVh5qL.

[37] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[38] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pages 6543–6553. PMLR, 2020.

[39] Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=jphnJNOwe36.

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

[41] Geon Yeong Park, Sangmin Lee, Sang Wan Lee, and Jong Chul Ye. Training debiased subnetworks with contrastive weight pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7929–7938. IEEE, 2023.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

 https://doi.org/10.52202/079017-1383

[43] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Conference on Neural Information Processing Systems (NeurIPS)*, 33:19920–19930, 2020.

[44] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=ryxGuJrFvS`.

[45] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryxGuJrFvS`.

[46] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8179–8186, 2022.

[47] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[48] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Unsupervised learning of debiased representations with pseudo-attributes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16721–16730. IEEE, 2022.

[49] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[50] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36, 2023.

[51] Daniel Ting and Eric Brochu. Optimal subsampling with influence functions. *Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[52] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528. IEEE Computer Society, 2011.

[53] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[54] Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3091–3100, 2021.

[55] Tianyang Wang, Jun Huan, and Bo Li. Data dropout: Optimizing training data for convolutional neural networks. In *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece*, pages 39–46. IEEE, 2018.

[56] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330, 2019.

[57] Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. Less is better: Unweighted data subsampling via influence function. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 6340–6347. AAAI Press, 2020.

[58] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

[59] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=4wZiAXD29TQ.

[60] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7829–7833. IEEE, 2020.

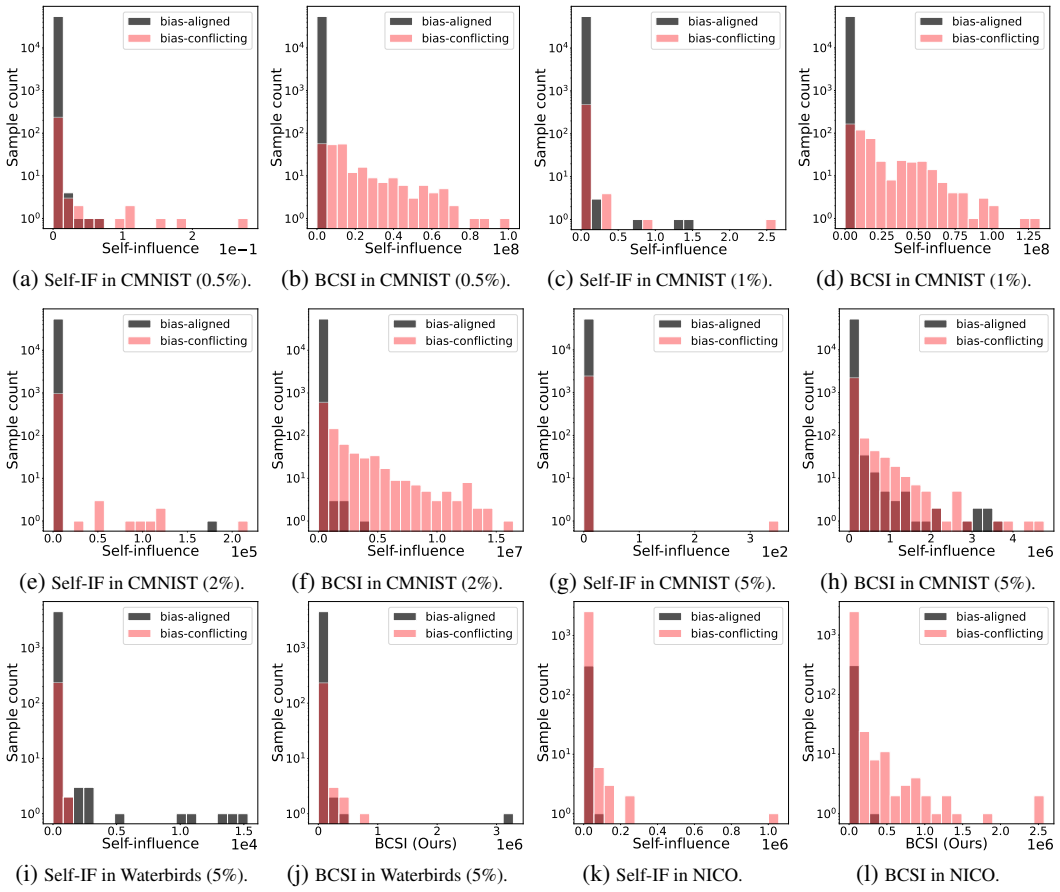
[61] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf.

[62] Zhuotun Zhu, Lingxi Xie, and Alan L. Yuille. Object recognition with and without objects. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3609–3615. ijcai.org, 2017.

 https://doi.org/10.52202/079017-1383

A Distribution of Self-Influence and bias-conditioned Influence

In Figure 1(c) and Figure 1(d) of the main paper, we have shown the influence histogram of naive self-influence and bias-conditioned self-influence (Ours) for the training set of CIFAR10C (1%). In this section, we show the histograms of self-influence and bias-conditioned self-influence for the training sets of an extended variety of bias ratios and datasets. Figure 8 shows the influence histograms for CIFAR10C, and BFFHQ. Figure 9 shows the influence histograms of CMNIST, Waterbird, and NICO. In accordance with the main paper, we observe that bias-conditioned self-influence generally exhibits better separation compared to naive self-influence, deeming it a better option to detect bias-conflicting samples.



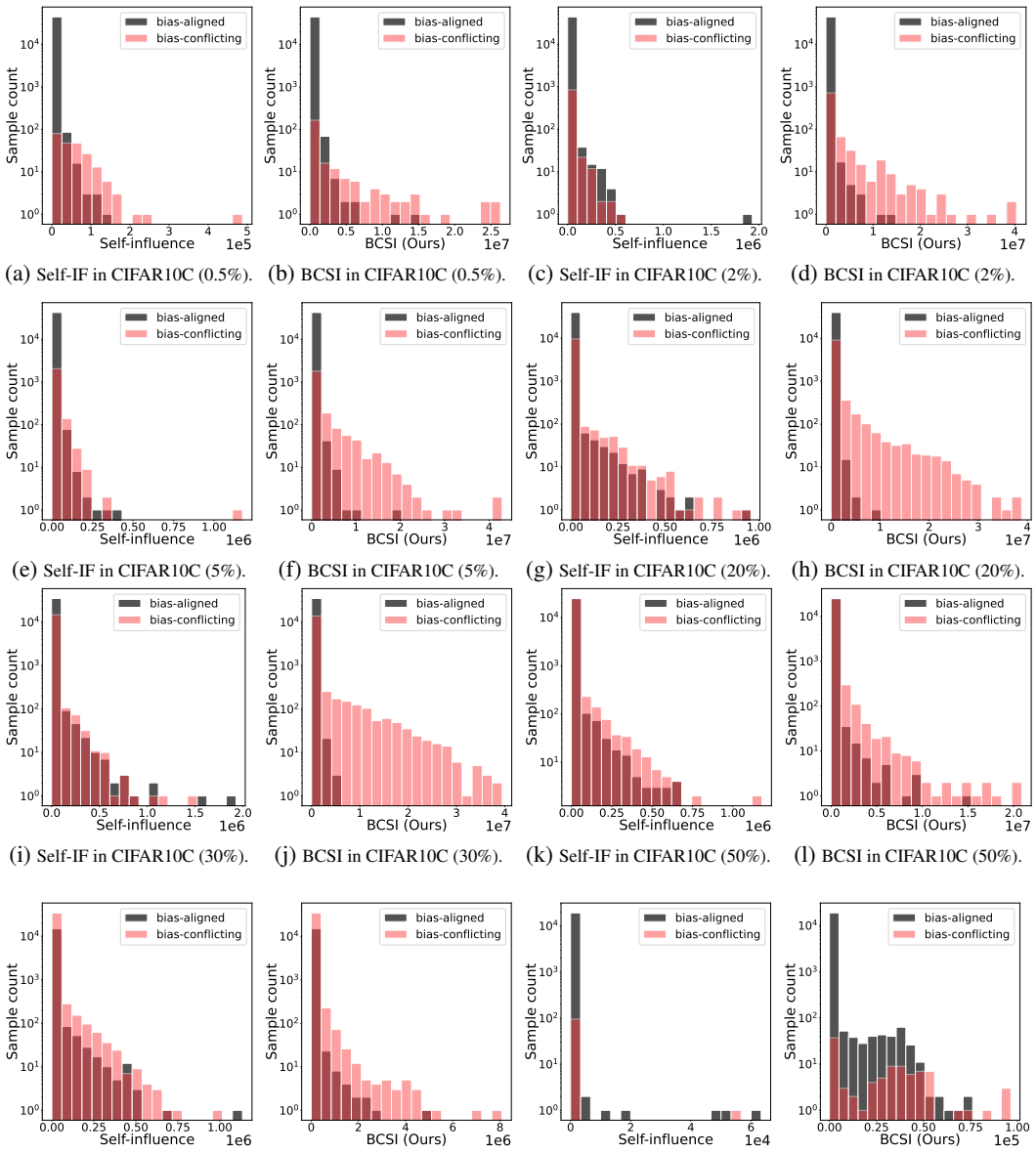
(a) Self-IF in CMNIST (0.5%). (b) BCSI in CMNIST (0.5%). (c) Self-IF in CMNIST (1%). (d) BCSI in CMNIST (1%).

(e) Self-IF in CMNIST (2%). (f) BCSI in CMNIST (2%). (g) Self-IF in CMNIST (5%). (h) BCSI in CMNIST (5%).

(i) Self-IF in Waterbirds (5%). (j) BCSI in Waterbirds (5%). (k) Self-IF in NICO. (l) BCSI in NICO.

Figure 8: Histogram of self-influence and bias-conditioned self-influence for CMNIST, Waterbird, and NICO.

https://doi.org/10.52202/079017-1383



(a) Self-IF in CIFAR10C (0.5%). (b) BCSI in CIFAR10C (0.5%). (c) Self-IF in CIFAR10C (2%). (d) BCSI in CIFAR10C (2%).

(e) Self-IF in CIFAR10C (5%). (f) BCSI in CIFAR10C (5%). (g) Self-IF in CIFAR10C (20%). (h) BCSI in CIFAR10C (20%).

(i) Self-IF in CIFAR10C (30%). (j) BCSI in CIFAR10C (30%). (k) Self-IF in CIFAR10C (50%). (l) BCSI in CIFAR10C (50%).

(m) Self-IF in CIFAR10C (70%). (n) BCSI in CIFAR10C (70%). (o) Self-IF in BFFHQ (0.5%). (p) BCSI in BFFHQ (0.5%).

Figure 9: Histogram of self-influence and bias-conditioned self-influence for the CIFAR10C dataset with varying bias-conflicting ratio and BFFHQ.

https://doi.org/10.52202/079017-1383

B Algorithm

Algorithm 1 Construct a pivotal set

Input: model parameters θ, GCE $\mathcal{L}_{\text{GCE}}$, number of epochs n_{epoch}, learning rate ρ, number of classes C, train set $\mathbf{Z}$, number of topk n_{topk}
Initialize: Model parameter θ.
for $i = 0, 1, 2, \cdots, n_{\text{epoch}}$ **do**
  $\theta \leftarrow \theta - \rho \nabla_\theta \mathcal{L}_{\text{GCE}}(\mathbf{Z}, \theta)$
end for
# Select samples with high self-influence
$\mathbf{Z}_{\text{P}} \leftarrow \emptyset$
for $c = 0, 1, 2, \cdots, C$ **do**
  $\mathbf{Z}_c \leftarrow \{(x, y) \in \mathbf{Z} | y = c\}$
  for $j = 0, 1, 2, \cdots, n_{\text{topk}}$ **do**
    $z_{\text{highest}} \leftarrow \text{argmax}_{z \in \mathbf{Z}_c} \mathcal{I}_{\texttt{self}}(z)$
    $\mathbf{Z}_c \leftarrow \mathbf{Z}_c \setminus \{z_{\text{highest}}\}$
    $\mathbf{Z}_{\text{P}} \leftarrow \mathbf{Z}_{\text{P}} \cup \{z_{\text{highest}}\}$
  end for
end for
Output: $\mathbf{Z}_{\text{P}}$

Algorithm 2 Post-training with the pivotal set

Input: pre-trained model parameters θ^*, CE $\mathcal{L}_{\text{CE}}$, number of iterations n_{iter}, learning rate ρ, train set $\mathbf{Z}$, pivotal set $\mathbf{Z}_{\text{P}}$, weight of remaining set λ
Initialize: Last-layer of model $\theta^*_{\text{last-layer}}$.
$\mathbf{Z}_{\text{R}} \leftarrow \mathbf{Z} \setminus \mathbf{Z}_{\text{P}}$
$n_{\text{P}} \leftarrow |\mathbf{Z}_{\text{P}}|$
for $i = 0, 1, 2, \cdots, n_{\text{iter}}$ **do**
  # Sample data from remaining samples
  $\mathbf{Z}_{\text{S}} \leftarrow \emptyset$
  for $j = 0, 1, 2, \cdots, n_{\text{P}}$ **do**
    $z \sim \mathbf{Z}_{\text{R}}$
    $\mathbf{Z}_{\text{S}} \leftarrow \mathbf{Z}_{\text{S}} \cup \{z\}$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}(\mathbf{Z}_{\text{P}}, \theta^*)$
  $\mathcal{L} \leftarrow \mathcal{L} + \lambda \mathcal{L}_{\text{CE}}(\mathbf{Z}_{\text{S}}, \theta^*)$
  $\theta^* \leftarrow \theta^* - \rho \nabla_\theta \mathcal{L}$
end for
Output: θ^*

C Detection precision for other datasets

We now describe the detailed experimental setting used in Figure 4 of the main paper. We first train ResNet18 [15] for five epochs and then compute self-influence, and bias-conditioned self-influence. Note that we only use the last layer when computing self-influence, and bias-conditioned self-influence. Subsequently, we sort the training data in descending order based on the values obtained by each method, selecting samples ranging from the highest to the k-th sample, where k is the number of total bias-conflicting samples in the training set. We then calculate the precision in detecting bias-conflicting samples within the selected data.

To further demonstrate the effectiveness of bias-conditioned self-influence in detecting bias-conflicting samples, we compare bias-conditioned self-influence with self-influence on other datasets including CMNIST (0.5%, 2%, 5%), CIFAR10C (0.5%, 2%, 5%, 20%, 50%), NICO. As shown in Table 4, bias-conditioned self-influence exhibits superior performance or is comparable to self-influence in most cases. This observation is consistent with the result in the main paper.

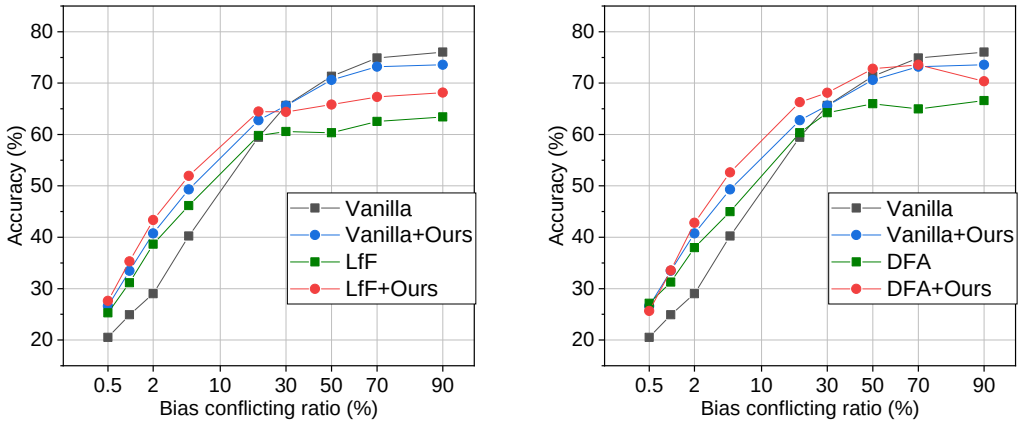
Table 4: Comparison of bias-conflicting sample detection precisions between self-influence (SI), and bias-conditioned self-influence (BCSI) across various datasets. The average and the standard error of precision over three runs are provided.

Method	CMNIST			CIFAR10C					NICO
	0.5%	2%	5%	0.5%	2%	5%	20%	50%	
SI	31.94$_{\pm 2.85}$	39.35$_{\pm 1.38}$	37.91$_{\pm 0.84}$	**63.33**$_{\pm 1.75}$	20.19$_{\pm 2.22}$	42.17$_{\pm 1.74}$	41.11$_{\pm 0.08}$	58.73$_{\pm 0.30}$	89.37$_{\pm 0.21}$
BCSI	**92.08**$_{\pm 0.24}$	**82.86**$_{\pm 1.38}$	**44.00**$_{\pm 3.77}$	38.33$_{\pm 2.52}$	**50.45**$_{\pm 0.34}$	**60.48**$_{\pm 1.91}$	**69.48**$_{\pm 0.39}$	**71.78**$_{\pm 0.29}$	**90.86**$_{\pm 0.39}$

https://doi.org/10.52202/079017-1383

D Performance with respect to the bias-conflicting ratio

In Figure 4.2 of the main paper, we showed the unbiased accuracy trends of the CIFAR10C dataset with respect to the bias-conflicting ratio for SelecMix and SelecMix with our method. In Figure 10, we provide the CIFAR10C accuracy trends of LfF [40] and DFA [31] alone and with our method.



(a) The performance of LfF and Ours.

(b) The performance of DFA and Ours.

Figure 10: Performance of the other baselines and Ours on the CIFAR10C dataset with varying bias ratio. The performance of LfF [40] is shown in Figure 10(a). Figure 10(b) displays the performance of DFA [31].

E Bias-conflicting ratio of the pivotal set

We provide the resulting bias-conflicting ratios (*i.e.*bias-conflicting detection precisions) of the pivotal set produced across a variety of datasets. Table 5 and Table 6 show the bias-conflicting ratios for CMNIST (0.5%, 1%, 2%, 5%), CIFAR10C (0.5%, 1%, 2%, 5%, 20%, 30%, 50%, 70%), BFFHQ, Waterbirds, and NICO.

Table 5: The average and the standard error of detection precision over three runs. Note that we compute the precision of the pivotal sets across varying ratios of bias-conflicting samples in CIFAR10C.

	CIFAR10C							
	0.5%	1%	2%	5%	20%	30%	50%	70%
Accuracy	$45.57_{\pm 1.63}$	$68.18_{\pm 0.96}$	$86.13_{\pm 1.18}$	$96.60_{\pm 0.11}$	$99.94_{\pm 0.06}$	$99.88_{\pm 0.12}$	$98.30_{\pm 0.30}$	$85.81_{\pm 2.05}$

Table 6: The average and the standard error of detection precision over three runs. Note that we compute the precision of the pivotal sets on CMNIST, BFFHQ, Waterbirds, and NICO.

	CMNIST				BFFHQ	Waterbirds	NICO
	0.5%	1%	2%	5%	0.5%	5%	
Accuracy	$68.19_{\pm 4.57}$	$84.61_{\pm 1.86}$	$97.27_{\pm 0.51}$	$73.24_{\pm 6.55}$	$66.32_{\pm 3.94}$	$64.26_{\pm 1.93}$	$94.01_{\pm 2.57}$

F Results on natural language processing datasets

To verify the effectiveness of our method on NLP datasets, we conduct experiments on two widely used benchmarks, CivilComments and MultiNLI. CivilComments contains user-generated comments labeled as either toxic or non-toxic. The spurious attribute in this dataset indicates whether a comment mentions one of the protected attributes, such as male, female, LGBT, black, white, Christian, Muslim, or other religions. These attributes are disproportionately associated with toxic comments, creating

https://doi.org/10.52202/079017-1383

a spurious correlation. Similarly, the MultiNLI dataset comprises pairs of sentences with labels denoting their relationship as contradiction, entailment, or neutral. The spurious attribute in MultiNLI is the presence of negation words, which are more frequently observed in the contradiction class. Both datasets are structured into groups based on combinations of the label y and the spurious attribute s, resulting in four groups in *CivilComments* and six in *MultiNLI*.

As shown in table 7, our method demonstrates its effectiveness in further rectifying models previously debiased with JTT, achieving increases in worst-group accuracy of 3.4% and 14.8% on MultiNLI and CivilComments, respectively.

Table 7: The average and the worst-group accuracy on NLP datasets.

Method	MultiNLI		CivilComments	
	Avg.	Worst-group	Avg.	Worst-group
ERM	82.4	67.9	92.6	57.4
JTT	80.0	70.2	92.6	63.7
Ours $_{\text{JTT}}$	79.8	**73.6**	86.9	**78.5**

G Comparison of time costs

In this section, we analyze the time cost of our method and compare it with the other baselines. For a practical and tangible comparison, we measure the wall clock time for the CIFAR10C (0.5%) dataset. We run our experiments with a machine equipped with Intel Xeon Gold 5215 (Cascade Lake) processors, 252GB RAM, Nvidia GeForce RTX2080ti (11GB VRAM), and Samsung 860 PRO SSD. For self-influence calculation, we utilize the JAX [3] library for fast Hessian vector product calculation. For all other deep learning functionalities, we utilize Pytorch [42]. In Table 8, the wall-clock duration of each component of our method is shown. We observe that the self-influence calculation step takes a longer time compared to the fine-tuning step due to the intersection process. However, this can be executed in parallel, which reduces the time cost of self-influence calculation approximately threefold. In Table 9, a wall-clock time comparison with the other baselines is shown. Our method consumes a significantly lesser amount of time, dropping to less than half the time of ERM full training when the self-influence calculation is executed in parallel. Reflecting on these results, we assert that the time cost of our method is rather small or even negligible compared to the full training time of other baselines.

Table 8: The average and the standard error of computational costs over three runs. We measure the computing time for full training as the wall-clock time of each component. Self-influence (parallel) represents calculating the bias-conditioned self-influence in GPU-parallel. Note that † indicates that corresponding methods use JAX while others utilize PyTorch.

Component	Self-influence	Self-influence (parallel)	Fine-tuning
Time (min.)	$11.46^{\dagger}{}_{\pm 0.08}$	$3.86^{\dagger}{}_{\pm 0.03}$	$1.08_{\pm 0.04}$

Table 9: The average and the standard error of computational costs over three runs. We measure the computing time for full training as the wall-clock time of each method. Ours (parallel) presents our method which computes bias-conditioned self-influence in GPU-parallel. Note that † indicates that corresponding methods use JAX while others utilize PyTorch.

Method	ERM	LfF	DFA	SelecMix	Ours	Ours (parallel)
Time (min.)	$22.55_{\pm 0.32}$	$33.64_{\pm 0.34}$	$53.18_{\pm 2.55}$	$352.53_{\pm 5.13}$	$12.54^{\dagger}{}_{\pm 0.08}$	$4.94^{\dagger}{}_{\pm 0.03}$

H Analysis of intersections within pivotal sets

In this section, we analyze the effects of intersections between pivotal sets obtained from various random initializations of models. For the comparison, we provide the number of samples, detection precision, and performance after fine-tuning models across different numbers of the intersections

in Table 10, Table 11, and Table 12. We observe that the detection precision increases as the number of intersections rises, while the number of samples in the pivotal set decreases. For the performance, a higher number of intersections shows effectiveness in the highly-biased scenarios, as bias-conflicting samples are scarce, and intersections reduce the size of the pivotal set. In contrast, a fewer intersections exhibit superior performance in low-based scenarios as there are abundant bias-conflicting samples. Note that, to observe the trend across varying ratios of bias-conflicting samples, we conduct experiments on CIFAR10C (0.5%, 1%, 2%, 5%, 20%, 30%, 50%, 70%).

Table 10: The average and the standard error of **the number of pivotal sets** over three runs considering numbers of intersections.

Number of	CIFAR10C							
Intersections	0.5%	1%	2%	5%	20%	30%	50%	70%
1	1000	1000	1000	1000	1000	1000	1000	1000
2	322.67$\pm$3.38	386.67$\pm$11.98	503.67$\pm$39.75	577.00$\pm$16.46	554.00$\pm$63.38	421.67$\pm$21.17	309.67$\pm$40.03	290.00$\pm$70.32
3	201.67$\pm$4.91	267.00$\pm$8.50	388.33$\pm$19.06	430.33$\pm$30.66	452.00$\pm$65.09	281.00$\pm$11.02	144.67$\pm$30.99	141.67$\pm$30.99

Table 11: The average and the standard error of **detection precision** over three runs considering numbers of intersections.

Number of	CIFAR10C							
Intersections	0.5%	1%	2%	5%	20%	30%	50%	70%
1	13.27$\pm$0.50	24.90$\pm$0.87	47.07$\pm$0.65	76.27$\pm$0.50	97.10$\pm$0.95	97.60$\pm$1.20	91.27$\pm$2.02	80.83$\pm$1.15
2	31.68$\pm$1.61	52.83$\pm$1.80	75.17$\pm$3.66	92.01$\pm$0.80	99.77$\pm$0.16	99.02$\pm$0.74	96.00$\pm$0.47	83.68$\pm$0.99
3	45.57$\pm$1.63	68.18$\pm$0.96	86.13$\pm$1.18	96.60$\pm$0.11	99.94$\pm$0.06	99.88$\pm$0.12	98.30$\pm$0.30	85.81$\pm$2.05

Table 12: The average and the standard error of **classification accuracy** of 'Ours+SelecMix' over three runs considering numbers of intersections.

Number of	CIFAR10C							
Intersections	0.5%	1%	2%	5%	20%	30%	50%	70%
1	36.44$\pm$0.34	40.76$\pm$0.03	49.57$\pm$0.41	59.31$\pm$0.15	**67.99**$\pm$0.33	**67.04**$\pm$0.65	**67.39**$\pm$0.79	**70.09**$\pm$0.28
2	**38.85**$\pm$0.62	43.47$\pm$0.21	51.43$\pm$0.53	**60.22**$\pm$0.19	66.96$\pm$0.25	65.90$\pm$0.81	66.77$\pm$0.40	69.92$\pm$0.53
3	38.74$\pm$0.36	**46.18**$\pm$0.33	**52.70**$\pm$0.40	59.66$\pm$0.31	66.66$\pm$0.43	64.51$\pm$1.44	66.45$\pm$0.28	69.97$\pm$0.21

I Improving performance in low-bias settings

In CIFAR-10C, as the bias severity decreases from 30% to 90%, the dataset gradually transitions into the low-bias domain, approaching an unbiased state at 90%. This reduction undermines the assumption that the bias is sufficiently malignant, reducing the effectiveness of debiasing methods and allowing ERM to achieve better performance. In this context, to improve the performance of our method when applied to ERM—which leverages a large number of conflicting samples—it is necessary to increase the size of the pivotal set, thereby expanding the number of conflicting samples. As shown in Table 13, expanding the pivotal set can improve performance in low-bias settings. This result implies that we could further enhance performance by adjusting the top-k value if we had access to information regarding bias severity.

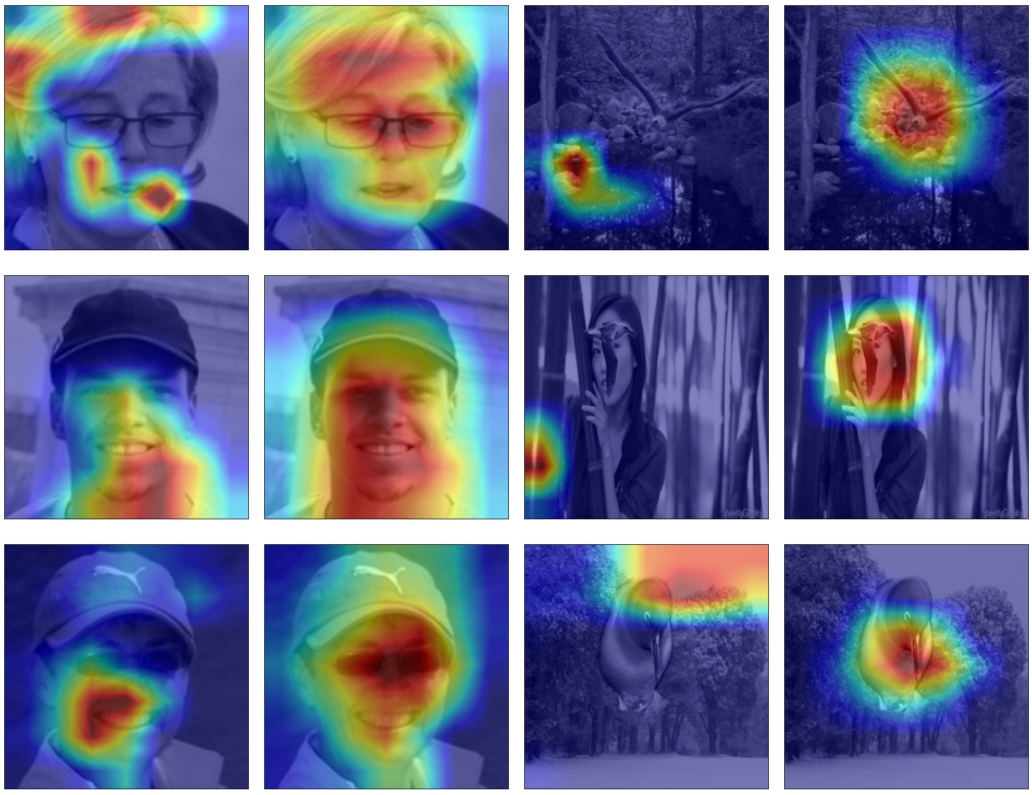
https://doi.org/10.52202/079017-1383

Table 13: The average and the standard error of classification accuracy over three runs.

Method	CIFAR10C			
	30%	50%	70%	90%
ERM	65.64$\pm$0.51	71.33$\pm$0.09	74.90$\pm$0.25	76.03$\pm$0.26
Ours ERM (k=100)	65.61$\pm$0.77	70.61$\pm$0.62	73.20$\pm$0.35	73.57$\pm$0.16
Ours ERM (k=2000)	**71.25**$\pm$0.34	**74.46**$\pm$0.34	**75.84**$\pm$0.33	**76.14**$\pm$0.23

J Qualitative analysis using Grad-CAM

This section provides qualitative results of our method using Grad-CAM [47] on BFFHQ and Waterbird. For BFFHQ, the target attributes are {young, old} and the bias attributes are {man, woman}. For Waterbird, the target attributes are {waterbird, landbird}, and the bias attributes are {water, land}. In Figure 11 (a) and (c), ERM focuses on biased features such as gender and background. However, ERM combined with our method tends to focus more on task-relevant features including age-related facial features and the birds themselves. This implies that our approach effectively guides the model in prioritizing target attributes over biased ones.



(a) ERM (b) Ours ERM (c) ERM (d) Ours ERM

Figure 11: The Grad-CAM of ERM and ERM+Ours on BFFHQ and Waterbird. (a-b) show results on BFFHQ, while (c-d) display results on Waterbird. (a) and (c) represent the Grad-CAMs for ERM, and (b) and (d) correspond to Grad-CAMs for ERM combined with our method.

K Ablation study on the loss function of the detection model

In this section, we conduct an ablation study on the learning objectives of the detection model. Our method uses Generalized Cross Entropy (GCE), a commonly adopted loss function in debiasing tasks to acquire biased models. However, conceptually, our method can be applied to any loss function to obtain biased models. To demonstrate the generality of our approach with different loss functions, we evaluate it on BFFHQ and Waterbird using alternative objectives, such as GCE, SCE [56], and NCE+RCE [38], which are designed for handling noisy label environments. In Table 14, both SCE and NCE+RCE demonstrate performance comparable to GCE. Since these objectives encourage models to focus more on the majority samples, our method combined with these loss functions also achieves similar results. Note that, naive cross-entropy, which does not promote majority sample utilization, fails on the BFFHQ dataset.

Table 14: The average and the standard error over three runs.

Method	BFFHQ	Waterbirds
	0.5%	5%
SelecMix	63.07$_{\pm 2.32}$	74.72$_{\pm 1.14}$
Ours w/ CE SelecMix	62.73$_{\pm 3.71}$	88.73$_{\pm 0.45}$
Ours w/ GCE SelecMix	65.80$_{\pm 3.12}$	89.67$_{\pm 0.38}$
Ours w/ SCE SelecMix	66.20$_{\pm 0.53}$	89.46$_{\pm 0.36}$
Ours w/ NCE+RCE SelecMix	**67.73**$_{\pm 1.99}$	**89.72**$_{\pm 0.41}$

L Ablation study on Influence estimation methods

We conduct an ablation study on other influence estimation methods. We leverage the fundamental form of Influence Functions to demonstrate the generalizability of our approach. However, other estimation methods are compatible. To further show this, we evaluate our method on BFFHQ and Waterbird using MoSo [50], TracIn [43], and Arnoldi [46]. As shown in Table 15, TracIn outperforms the basic IF, while MoSo and Arnoldi exhibit comparable performance. These results indicate that our method can enhance performance across various estimation approaches.

Table 15: The average and the standard error of detection precision over three runs.

Method	BFFHQ	Waterbirds
	0.5%	5%
SelecMix	63.07$_{\pm 2.32}$	74.72$_{\pm 1.14}$
Ours SelecMix	65.80$_{\pm 3.12}$	89.67$_{\pm 0.38}$
Ours w/ MoSo SelecMix	63.13$_{\pm 3.27}$	89.72$_{\pm 1.12}$
Ours w/ TracIn SelecMix	**69.20**$_{\pm 0.50}$	**90.39**$_{\pm 0.70}$
Ours w/ Arnoldi SelecMix	66.40$_{\pm 3.12}$	71.08$_{\pm 3.12}$

M Evaluation with fairness metrics

To further demonstrate the effectiveness of our method, we evaluate it using fairness metrics including demographic parity (DP) [5] and equalized odds equal opportunity (EOP) [13] on Waterbird. In Table 16, our method significantly improves performance in both DP and EOP. It indicates that our method also addresses the fairness problem.

https://doi.org/10.52202/079017-1383

Table 16: The average and the standard error of demographic parity (DP) and equalized odds equal opportunity (EOP) over three runs.

Method	Waterbirds	
	DP (↓)	EOP (↓)
ERM	$0.1826_{\pm 0.0044}$	$0.2731_{\pm 0.0187}$
SelecMix	$0.1146_{\pm 0.0004}$	$0.1885_{\pm 0.0100}$
Ours SelecMix	$\mathbf{0.0242}_{\pm 0.0053}$	$\mathbf{0.0099}_{\pm 0.0064}$

N Experimental settings

N.1 A detailed description of benchmark datasets

(a) Colored MNIST. (b) Corrupted CIFAR10.

Figure 12: Example images of CMNIST and CIFAR10C. Images in the first and second rows are *bias-aligned* and images in the third row are *bias-conflicting*.

(a) Biased FFHQ. (b) Waterbirds.

Figure 13: Example images of BFFHQ and Waterbirds. The red-bordered images are *bias-aligned* and the blue-bordered images are *bias-conflicting*.

Colored MNIST. Colored MNIST (CMNIST) is a synthetically modified version of MNIST [6], where the digit is the label and the color is the bias. For example, an image of digit `0` is correlated with the color `red`. We use the following bias-conflicting ratios: $r \in \{0.5\%, 1\%, 2\%, 5\%\}$. The images are in `28 x 28` resolution and are resized to `32 x 32`. There are approximately 55,000 training, 5,000 validation, and 10,000 test samples. Examples are shown in Figure 12(a).

Corrupted CIFAR10. Corrupted CIFAR10 (CIFAR10C) is a synthetically modified version of CIFAR10 [30] proposed by Hendrycks and Dietterich [17] with the following common corruptions as the bias: `{Snow, Frost, Fog, Brightness, Contrast, Spatter, Elastic transform, JPEG, Pixelate and Saturate}`. We use the following bias-conflicting ratios: $r \in \{0.5\%, 1\%, 2\%, 5\%, 20\%, 30\%, 50\%, 90\%(\text{unbiased})\}$. The images are in `32 x 32` resolution. There are approximately 45,000 training, 5,000 validation, and 10,000 test samples. Examples are shown in Figure 12(b).

Biased FFHQ. Biased FFHQ (BFFHQ) [31] is a curated Flickr-Faces-HQ (FFHQ) [22] dataset, which consists of images of human faces. The designated task label is the age `{young, old}` while the bias attribute is the gender `{man, woman}`. The bias-conflicting ratio is $r \in \{0.5\%\}$. The images

are in `128 x 128` resolution and are resized to `224 x 224`. There are approximately 20,000 training, 1,000 validation, and 1,000 test samples. Examples are shown in Figure 13(a).

Waterbirds. Waterbirds is proposed by Sagawa et al. [44], which synthetically combines bird images from the Caltech-UCSD Birds-200-2011 (CUB) with place background as bias. It consists of bird images to classify bird types `{waterbird, landbird}`, but their backgrounds `{water, land}` are correlated with bird types. The bias-conflicting ratio is $r \in \{5\%\}$. The images are in varying resolutions and are resized to `224 x 224`. There are approximately 5,000 training, 1,000 validation, and 6,000 test samples. Examples are shown in Figure 13(b).

NICO. NICO is a dataset designed to evaluate non I.I.D. classification by simulating arbitrary distribution shifts. To evaluate debiasing methods, a subset composed of *animal* classes label is utilized, as in [54]. The class labels (*e.g.* `"dog"`) are correlated to spurious contexts (*e.g.* `"on grass"`, `"in water"`, `"in cage"`, `"eating"`, `"on beach"`, `"lying"`, `"running"`) which exhibits a long-tail distribution. The images are in varying resolutions and are resized to `224 x 224`. There are approximately 3,000 training, 1,000 validation, and 1,000 test samples.

N.2 Baselines

We validate our method by combining various debiasing approaches. ERM is the model trained by cross-entropy loss. GroupDRO [44] minimizes the worst-group loss by exploiting group labels directly. LfF [40] detects bias-conflicting samples and allocates large loss weights on them. DFA [31] augments diverse features by swapping the features obtained from the biased model and concatenating the feature from the debiased model with the exchanged feature. BPA [48] utilizes a clustering method to identify pseudo-attributes using a clustering approach and adjusts loss weights according to the cluster size and its loss. DCWP [41] debiases a network by pruning biased neurons. SelecMix [19] identifies and mixes a bias-contradicting pair within the same class while detecting and mixing a bias-aligned pair from different classes. Note that we adopt SelecMix+LfF rather than SelecMix since SelecMix+LfF exhibits superior performance than SelecMix [19].

N.3 Evaluation protocol

We provide experimental setups for evaluation. We use JAX [3] and PyTorch [42] for the experiments. We conduct our experiments with a machine equipped with Intel Xeon Gold 5215 (Cascade Lake)594 processors, 252GB RAM, Nvidia GeForce RTX2080ti (11GB VRAM) (or Nvidia GeForce RTX3090 (24GB VRAM)), and Samsung 860 PRO SSD. In constructing pivotal sets, we adopt ResNet18 [15] as the base architecture for all datasets. For optimization, we employ the Adam optimizer [25] with a learning rate of 0.001, and train the models for 5 epochs. To calculate self-influence, we only utilize the last layer of the models. In fine-tuning, we deploy ResNet18 for CMNIST, CIFAR10C, BFFHQ, and NICO while ResNet50 is used for Waterbirds as following other baselines [40, 31, 36]. We adopt the Adam optimizer for CMNIST, CIFAR10C, BFFHQ, NICO while SGD is used for Waterbirds. For the learning rate, we use 0.001 for CMNIST, CIFAR10C, Waterbirds, and 10^{-4} for BFFHQ. We apply cosine annealing [37] to decay the learning rate to 10^{-3} of the initial value. We utilize weight decay of 10^{-4} for all datasets. We fine-tune the pre-trained models for 100 iterations. We set $\lambda = 0.1$ for all experiments. For baselines [31, 19], we use the officially released codes. For our method, we adopt $k = 100, \lambda = 0.1$ for all datasets.

N.4 Licenses for existing assets

Flickr-Faces-HQ (FFHQ) [22] is a high-quality image dataset of human faces, originally created as a benchmark for generative adversarial networks (GAN). The individual images were published in Flickr by their respective authors under either Creative Commons BY 2.0, Creative Commons BY-NC 2.0, Public Domain Mark 1.0, Public Domain CC0 1.0, or U.S. Government Works license. NICO [16] dataset does not own the copyright of images. Only researchers and educators who wish to use the images for non-commercial researches and/or educational purposes, have access to NICO. JAX [3] has Apache License.

 https://doi.org/10.52202/079017-1383

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We reflect all our assertions and contributions on Abstach and Introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not include theoretical results in our paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the experimental details in Section 5.1 and Appendix N.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-1383

Answer: [Yes]

Justification: We use benchmark datasets that are open to the public and provide codes with supplements.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the experimental details in Section 5.1 and Appendix N.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars and use standard errors. We also state them in Section 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide machine information in Appendix N.3 and computational costs in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read and agree with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Section 7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-1383

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We conduct experiments on recognition models and use benchmark datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the source of data and models in Section 5.1, 6 and Appendix N.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We would not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We use benchmark datasets and compute accuracy or precision on them.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We use benchmark datasets and compute accuracy or precision on them.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1383