Learning Debiased Models with Dynamic Gradient Alignment and Bias-conflicting Sample Mining

Bowen Zhao\textsuperscript{1} Chen Chen\textsuperscript{2,50} Qi Ju\textsuperscript{2} Shutao Xia\textsuperscript{1,3,50}
\textsuperscript{1}Tsinghua University \textsuperscript{2}Tencent TEG AI \textsuperscript{3}Peng Cheng Laboratory
zbwl8@mails.tsinghua.edu.cn, \{beckhamchen, damonju\}@tencent.com, xlast@sz.tsinghua.edu.cn

Abstract

Deep neural networks notoriously suffer from dataset biases which are detrimental to model robustness, generalization and fairness. In this work, we propose a two-stage debiasing scheme to combat against the intractable unknown biases. Starting by analyzing the factors of the presence of biased models, we design a novel learning objective which cannot be reached by relying on biases alone. Specifically, debiased models are achieved with the proposed Gradient Alignment (GA) which dynamically balances the contributions of bias-aligned and bias-conflicting samples (refer to samples with/without bias cues respectively) throughout the whole training process, enforcing models to exploit intrinsic cues to make fair decisions. While in real-world scenarios, the potential biases are extremely hard to discover and prohibitively expensive to label manually. We further propose an automatic bias-conflicting sample mining method by peer-picking and training ensemble without prior knowledge of bias information. Experiments conducted on multiple datasets in various settings demonstrate the effectiveness and robustness of our proposed scheme, which successfully alleviates the negative impact of unknown biases and achieves state-of-the-art performance.

1. Introduction

Deep Neural Networks (DNNs) have achieved significant progress in various visual tasks. In common, DNNs are anticipated to accomplish target tasks by relying on essential causal cues from training data. However, enormous previous studies \cite{2,17,19,26,30,35,41,45} have shown that DNNs may establish spurious correlations between targets and easy-to-learn shortcut cues rather than essential ones. For instance, since birds often appear in blue sky, DNNs might resort the frequently co-occurred but not innate feature (i.e., bias cues) “blue sky” to recognize birds, instead of the intrinsic characteristics (i.e. intrinsic cues) of birds. For another example, as shown in Fig. 1, color “pink” can be the bias cue to recognize digit “2” because “pink” is highly correlated with digit “2” in this biased training set. The models blinded by biased datasets usually perform poor in out-of-distribution situations (indoor birds cannot be correctly recognized and a digit “3” colored in pink may be falsely classified as “2”). Worse still, models with racial or gender bias, etc. can cause severe negative social impacts. Therefore, debiasing is urgently demanded when deploying AI systems in realistic applications.

As mentioned in \cite{36,39}, two main factors lead to severe bias issue during model training: (i) the training objective can be reached by only relying on bias cues; (ii) bias cues are much easier to learn for models and optimizers than intrinsic cues. Taking Fig. 1 as an example, models can accomplish the vanilla objective of empirical risk minimization relying on color “pink” since most training images hold the bias cue. Besides, studies \cite{7,16} indicate that texture information is usually easier to capture for deep models. These two facts result in a biased model towards image texture rather than object shapes. Inspired by the above analysis, our main idea is to interrupt the first condition by designing a new learning objective that models cannot achieve when depending on bias cues alone. We propose Gradient Alignment (GA) which prevents models from being biased by balancing the gradients between bias-aligned and bias-conflicting samples (refer to samples with and without bias cues respectively). Specifically, the gradient information is served as an indicator to up-weight bias-conflicting samples and down-weight bias-aligned samples,
which achieves a dynamic balance during the whole learning process, yielding debiased models. However, GA is based on the assumption of bias information availability which often needs expert knowledge by meticulous analysis (bias type) and painstaking manual annotations (detailed bias labels). Consequently, to address the real-world scene where the bias information is totally unknown, we propose a bias-conflicting sample mining method integrated with ensemble strategy and peer-picking mechanism to improve the quality of pseudo bias labels. The whole debiasing pipeline with unknown biases is illustrated in Fig. 1.

Compared to other debiasing techniques, the proposed solution (i) neither rely on detailed bias annotations [8, 17, 26, 35, 41, 53] nor a pre-defined bias type [2, 4, 7, 42, 45]; (ii) does not require disentangled representations [2, 18, 19, 41], so that it still works well in complex scenarios where disentangled features are difficult to extract; (iii) does not introduce any data augmentations [7, 8, 18, 19], avoiding additional training complexity like in generative models; (iv) does not involve any modification of model architectures [19, 46], making it easy to be applied on any networks; (v) significantly improves the quality of bias-conflicting sample mining. In summary, the main contributions of our work include:

• We propose a Gradient Alignment training paradigm to balance gradients from bias-aligned and bias-conflicting samples throughout the whole training stage, enforcing model to focus on intrinsic cues instead of bias cues (Sec. 3.1).
• To deal with unknown bias issue, a robust peer-picking and epoch-ensemble based bias-conflicting sample mining algorithm is designed, which can annotate pseudo labels with satisfactory precision and recall (Sec. 3.2).
• Comprehensive experiments (on both synthesized and realistic vision datasets) are conducted to compare several debiasing methods in a fair manner, in which the proposed scheme achieves state-of-the-art performance (Sec. 4).

2. Preliminary

2.1. Bias Cues and Biased Models

We consider a training dataset \( D = \{(x^i, y^i) \}_{i=1}^N \) where each example consists of an image \( x \) and its label \( y \). Let \( \pi \) represents the intrinsic cues for recognizing image \( x \) as its label \( y \). Normally, the model \( f(\cdot) \), which takes \( x \) as input and outputs \( C \)-dimensional logits, can learn the correlations between image \( x \) and label \( y \) by using intrinsic cues \( \pi \). However, it has been shown that if there exists potential bias cues \( v^i \), models may be biased to learn the correlations between \( v \) and labels, deteriorating generalization performance. The image that contains bias cues \( v \) is denoted as a bias-aligned sample \( x \), and vice versa a bias-conflicting sample \( \tau \). For example, when training a model to recognize digits on Colored MNIST [17] (which is constructed by injecting color into the MNIST dataset [24] as shown in Fig. 2), the intrinsic cues \( \pi \) are digit shapes and the bias cue \( v \) can be color information. The vanilla model on Colored MNIST can be heavily biased: the accuracy on bias-conflicting test samples is quite poor as shown in Fig. 3.

2.2. Debiasing Techniques

Here we briefly introduce several debiasing mechanisms which can be roughly classified into explicit (with known biases) and implicit (with unknown biases) ways.

Debiasing with Known Biases. A majority of approaches require explicit bias labels for each training sample. The classic approach is reweighting/resampling training samples based on sample number or loss of different groups [25, 26, 35, 36], or even synthesizing samples from minority groups [1]. Another large group of strategies aims at disentangling bias and intrinsic cues in feature domain [17, 29, 34, 41, 53], e.g., EnD [41] designs regularizers to disentangle representations with the same bias label and entangle features with the same target label; and some other studies learn disentangled representation by mutual information minimization [17, 34, 53]. Besides, Sagawa et al. [35] and Goel et al. [8] aim to improve the worst-group performance by group distributionally robust optimization [9] and CycleGAN [52] based data augmentation, respectively. Domain-independent classifiers are introduced in [46], which complete target tasks in each known bias situation.

To alleviate expensive bias annotation cost, some bias-tailored methods relax the demands by requiring only predefined bias types [2, 7, 45]. Bahng et al. [2] elaborately design specific networks according to the bias type for obtaining biased representations on purpose (e.g., using 2D CNNs for extracting static bias in action recognition task). Then the debiased representation is learned by encouraging it to be independent with the biased one. Wang et al. [45] try to project the model’s representation onto the subspace orthogonal to the texture-biased representation. In addition, ensemble approach that consists of a naive biased model and a robust debiased model is widely employed in natural language processing related problems [3–5, 13, 42].
Debiasing with Unknown Biases. Despite effectiveness of the above techniques, the assumptions limit their applications, as manually discovering the biases heavily relies on experts’ knowledge and labeling bias attributes for each training sample is even more laborious. Therefore, recent studies [6, 12, 18–20, 22, 23, 30, 33, 37, 40, 43, 49] manage to obtain debiased models with unknown biases which is more realistic. Nam et al. [30] mine bias-conflicting samples with generalized cross entropy (GCE) loss [50] and employ a designed relative difficulty score to emphasize them. Lee et al. [19] further synthesize diverse bias-conflicting samples based on the identified ones by feature-level data augmentation and Kim [18] et al. directly generate them with SwapAE [32]. Besides GCE loss, feature clustering [40], early-stopping [27], forgettable examples [49] and limited network capacity [37, 43] are resorted to identify bias-conflicting samples which are emphasized to train debiased models afterwards [4]. Furthermore, [6] and [22] alternatively infer dataset partitions and enhance domain-invariant feature learning by min-max adversarial training. In addition to the identify-emphasize paradigm, [33] introduces a novel regularization method aiming at decoupling feature learning dynamics to improve model robustness. Our proposed debiasing algorithm with unknown biases belongs to the popular identify-emphasize paradigm, but holds a more robust and principled bias-conflicting sample mining algorithm and straightforward dynamic gradient balance training pipeline.

3. Methodology

We first present the detailed motivation and derivation of learning with gradient alignment (Sec. 3.1), then introduce how to mine bias-conflicting samples with epoch-ensemble and peer-picking (Sec. 3.2).

3.1. Gradients Alignment

Generally, the bias cues $v$ lead to biased models when the following two conditions [36, 39] are satisfied. Condition I: within $\mathcal{D}$, the commonly used learning objective: empirical risk minimization with cross-entropy loss $\ell$:

$$R_{ERM} = \sum_{i=1}^{N} \ell(f(x^i), y^i)$$

(1)

can be optimized relying on bias cues $v$ alone, which means $v$ is highly correlated with the target labels in $\mathcal{D}$, e.g., most digits “2” are pink in Colored MNIST. Condition II: $v$ is easier to be learned by model $f(\cdot)$ compared to the intrinsic cues $\tau$, e.g., color “pink” is easier to learn than the shape of “2” for common deep models. Actually, DNNs are inclined to learn simple functions even without explicit regularization [31, 44, 48]. Such characteristic makes DNNs generalize remarkably well even in strongly over-parameterized regime, while might cause DNNs only to focus on superficial features instead of digging out core features in some scenarios, leading to biased models.

We focus on the important precondition of the presence of biased models: Condition I, and try to make bias cues do not have enough ability to accomplish a new designed learning objective. The most straightforward inspiration is employing a cost-sensitive learning method — reweighting to intentionally rebalance sample contributions from different domains during training. A sample-number-based reweighting strategy for debiasing is put forward in [36] as

$$R_{Rew} = \sum_{i=1}^{N} \gamma \cdot \ell(f(x^i), y^i) + \sum_{j=1}^{N} \ell(f(x^j), y^j)$$

(2)

where $\overline{N}$ and $\overline{N}$ are the number of bias-conflicting and bias-aligned samples respectively, $\gamma \in (0, \infty)$ is a reserved hyperparameter to conveniently adjust the tendency: when $\gamma \rightarrow 0$, models intend to exploit bias-aligned samples more and when $\gamma \rightarrow \infty$, the behavior is reversed. As depicted in Fig. 3, assisted with reweighting, bias-conflicting accuracy skyrocket in the beginning, indicating that the model tends to learn intrinsic features in the first few epochs, while declines gradually, manifesting that the model is inclined to be biased progressively (adjusting hyperparameter $\gamma$ can not reverse the tendency).

Therefore, the static ratio between $\overline{N}$ and $\overline{N}$ may not be a good indicator of how balanced the training is, as the influence of samples can be fluctuated during training. Accordingly, we are inspired to directly choose gradient statistics as a metric to indicate whether the training is overwhelmed by bias-aligned samples. Let us revisit the commonly used cross-entropy loss: $\ell(f(x), y) = -\sum_{k=1}^{C} \delta_{y_k} \log [p(k|f(x))]$, where $p(k|f(x)) = \sum_{i=1}^{C} \epsilon_{f_k}(x)$ and $f_k(x)$ denotes the $k$th index of $f(x)$. For an image of class $c$, the gradients on logit $f_c(x)$ is given by $\frac{\partial \ell(f(x), y)}{\partial f_c(x)} = p(c|f(x)) - 1$. Assuming within the $t$th iteration where $t \in [0, T - 1]$, the minibatch is composed of $\mathcal{B}$ bias-aligned and $\mathcal{B}$ bias-conflicting samples ($\mathcal{B}$ samples in total). Considering the images of a given class $c$, the accumulated gradients on the $c$th logit generated
by the two groups are given as
\[ g^t_i = \frac{1}{N} \sum_{i=1}^{N} \prod_{y'=c} \frac{\partial \ell(f^t(x_i), y')}{\partial f^t(y')}, \quad \hat{g}^t_c = \frac{1}{N} \sum_{j=1}^{N} \prod_{y'=-c} \frac{\partial \ell(f^t(x_j), y')}{\partial f^t(y')} \]
respectively. Under our concerned circumstance: \( B \gg \hat{B} \), the contributions from bias-aligned samples overwhelm that from bias-conflicting ones for class \( c \).

The basic idea of gradient alignment is to rebalance bias-aligned and bias-conflicting samples according to their currently produced gradients. We define \( r^t \) as the ratio of gradients of bias-conflicting samples to those of bias-aligned samples within the \( t \)th iteration:
\[ r^t = \frac{\sum_{k=1}^{C} \hat{g}^t_k}{\gamma \cdot \sum_{k=1}^{C} \hat{g}^t_k} = \frac{\sum_{j=1}^{B} [p(y_j | f^t(x_j)) - 1]}{\gamma \cdot \sum_{j=1}^{B} [p(y_j | f^t(x_j)) - 1]}, \tag{3} \]
where \( \gamma \) is a adjusting hyperparameter as the same function in Eq. (2). As bias-conflicting samples is exceedingly scarce, it is difficult and unrealistic to ensure that every category of them can be sampled in one mini-batch. Thus we define \( r^t \) by just considering samples whether hold bias cues, so that all classes will share the same ratio. After calculating \( r^t \), we apply it to the learning trajectory in the \( t \)th iteration:
\[ \mathcal{R}_{GA}^t = \sum_{i=1}^{B} r^t \cdot \ell(f^t(x_i), y_i) + \sum_{j=1}^{\hat{B}} \ell(f^t(x_j), y_j). \tag{4} \]

Different from static reweighting loss function in Eq. (2), Eq. (4) proposes a dynamic rebalancing training strategy with aligned gradients during learning process (\( r^t \) is adaptively calibrated in each iteration), which enforces models to dive into intrinsic cues instead of superficial bias cues. The impact of GA on learning trajectory compared to other methods are presented in Sec. 4.4.

Evidently, the above discussions are based on the hypothesis that bias labels are explicitly provided. In order to address the unknown bias circumstance, we first distinguish bias-aligned and bias-conflicting samples by leveraging only image \( x \) and its corresponding category label \( y \). Accordingly, a pseudo-label assignment algorithm will be delineated in Sec. 3.2, by which a probability value \( s(x, y) \in [0, 1] \) called b-c score is calculated and then binarized with threshold \( \tau \), i.e., if \( s(x, y) > \tau \), \( x \) will be assigned as a bias-conflicting sample, and vice versa.

### 3.2. Bias-Conflicting Sample Mining

Preceding implicit debiasing studies [18, 19, 27, 30] automatically separate bias-aligned and bias-conflicting samples based on the assumption that network is prone to fit shortcut bias cues rather than complicate intrinsic cues, thus they resort the output probability on target class: \( s(x, y) = 1 - p(y|f(x)) \) to look for a distinct classification boundary.

---

#### Algorithm 1: Bias-Conflicting Sample Mining

| Input: | training set \( \{x_j, y_j\}_{j=1}^{N} \); initial models \( f^0, f^0; \) iterations \( T \); batch size \( B \); threshold \( \eta \). |
| --- | --- |
| for \( i = 1 \) to \( N \) do |
| for \( t = 0 \) to \( T - 1 \) do |
| \( B = \{ (x_j, y_j) \}_{j=1}^{B} \) ← FetchMiniBatch(\( D \)); |
| \( \{ p(y_j | f^t(x_j)) \} \) ← Forward(\( B, f^t \)); |
| \( \{ p(y_j | f^{t+1}(x_j)) \} \) ← Forward(\( B, f^{t+1} \)); |
| \( \hat{\ell} ← 0; \quad \ell ← 0 \); |
| for \( j = 1 \) to \( B \) do |
| if \( f^t, f^{t+1} \) s.t. Eq. (5) then |
| \( \hat{\ell} ← \hat{\ell} - \frac{1}{B} \log p(y_j | f^t(x_j)) \); |
| \( \ell ← \ell - \frac{1}{B} \log p(y_j | f^{t+1}(x_j)) \); |
| else if \( f^t, f^{t+1} \) s.t. Eq. (7) then |
| \( \hat{\ell} ← \hat{\ell} + \frac{1}{B} \log p(y_j | f^t(x_j)) \); |
| \( \ell ← \ell + \frac{1}{B} \log p(y_j | f^{t+1}(x_j)) \); |
| \( f^{t+1} ← \) Backward(\( f^t, \hat{\ell} \)); |
| \( f^{t+1} ← \) Backward(\( f^{t+1}, \ell \)); |
| if \( (t+1)\%T' = 0 \) then |
| for \( i = 1 \) to \( N \) do |
| \( s^{t+1} ← 1 - \frac{p(y_j | f^{t+1}(x_j)) + p(y_j | f^{t+1}(x_j))}{2} \); |
| \( s ← s + \frac{T'}{T} s^{t+1}; \) |
| Output: b-c scores \( \{ s^t \}_{i=1}^{N} \). |

---

Note that by using a computationally more efficient implementation in practice, we can get rid of the for loop which is adopted for clarity here.
less, a few number of bias-conflicting samples can still be overfitted and the memorization will be strengthened if using them to update models. Thus, we introduce a co-training-like [11] paradigm during the auxiliary biased models’ training process, leading to peer-picking.

Our method maintains two auxiliary biased models \( \hat{f} \) and \( \tilde{f} \) (identical structure) simultaneously. In one mini-batch with \( B \) samples, each peer model divides samples into confident and unconfident parts with the output probabilities on target classes (e.g., \( p(y_j | \hat{f}(x_j)) \)) and threshold \( \eta \). As shown in Fig. 4, four clusters are formed. For the red cluster, \( \hat{f}(x_j) \) and \( \tilde{f}(x_j) \) both satisfy \( p(y_j | \hat{f}(x_j)) > \eta \) and \( p(y_j | \tilde{f}(x_j)) > \eta \), since two peer biased models are both confident on them. It is reasonable to believe that they are indeed bias-aligned samples, so we pick up them to update model via gradient descent as usual (Lines 9, 10, 15 and 16). While the gray cluster, \( \hat{f}(x_j) \) and \( \tilde{f}(x_j) \) both satisfy \( p(y_j | \hat{f}(x_j)) < \eta \) and \( p(y_j | \tilde{f}(x_j)) < \eta \), on which both two peer models are unconfident, will be dropped directly as they are likely to be bias-conflicting. The remaining purple clusters, \( \hat{f}(x_j) \) and \( \tilde{f}(x_j) \) do not meet any of the following two conditions:

\[
\begin{align*}
 p(y_j | \hat{f}(x_j)) > \eta & \quad \text{and} \quad p(y_j | \tilde{f}(x_j)) \leq \eta, \\
 p(y_j | \hat{f}(x_j)) \leq \eta & \quad \text{and} \quad p(y_j | \tilde{f}(x_j)) > \eta,
\end{align*}
\]

mean that some samples may be bias-conflicting, but one of the two auxiliary models have memorized them. Inspired by the work in handling noisy labels [10], we endeavour to make the corresponding model forget the memorized suspicious samples via gradient ascent (Lines 12, 14, 15 and 16). The intuition is to give models opportunities to rectify their false memorization, and regain capability to capture bias cues. Such peer-picking strategy can lead to heavily biased models \( \hat{f} \) and \( \tilde{f} \), on which we average the output results to obtain b-c scores (Line 19).

Consequently, GA can be employed based on the pseudo labels produced by the auxiliary extremely biased models, leading to method B+GA to deal with unknown biases.

4. Experiments

We introduce several synthesized and real-world datasets in Sec. 4.1, describe the unified experimental implemen-

tation in Sec. 4.2, present comprehensive quantitative results compared with various state-of-the-art debiasing approaches in Sec. 4.3 and further analysis in Sec. 4.4.

4.1. Datasets

**Colored MNIST.** The Colored MNIST dataset injects color into the standard MNIST dataset, each digit (0 - 9) being linked with one pre-defined color as shown in Fig. 2. The task is to recognize digits, in which shapes are intrinsic cues \( \varpi \) and color can be bias cue \( \nu \). For images of target class \( c \), we dye them by the corresponding color with probability \( \rho \in \{0.95, 0.98, 0.99, 0.995\} \), and by any other color with probability \( 1 - \rho \). A higher value of \( \rho \) indicates a more frequent co-occurrence of target classes and bias cues.

**Corrupted CIFAR-10.** By degrading the CIFAR-10 [21] dataset with different types of corruption [15], each object class holds a spurious correlation with a specific corruption type in Corrupted CIFAR-10 as shown in Fig. 5a. Two sets of corruption protocols are utilized, leading to two biased datasets: Corrupted CIFAR-10\(^1\) and CIFAR-10\(^2\) with \( \rho \in \{0.95, 0.98, 0.99, 0.995\} \).

**Biased Waterbirds (BWaterbirds).** BWaterbirds is a composite dataset which superimposes foreground bird images from the CUB dataset [47] onto background environment images from the Places dataset [51], in which “waterbirds” and “landbirds” are highly correlated with “wet” and “dry” habitats as exhibited in Fig. 5b. Consequently, the task aiming to distinguish images of birds as “waterbird” or “landbird” is inclined to be influenced by background information. To focus on the problem of eliminating shortcut reliance, we additionally balance the number of images per class (95% samples are bias-aligned here).

**Biased CelebA (BCelebA).** The CelebA [28] dataset is realistically established for face recognition in which each image contains 40 attributes. When the goal is to classify the hair color as “blond” or “non-blond”, the information

![Figure 4](image1.png) Illustration of peer-picking. Best seen in color.

![Figure 5](image2.png) Examples of training datasets. \( \varpi \) denotes bias-aligned samples and \( \nu \) indicates bias-conflicting samples.
of gender, i.e., “male” or “female” can be a bias cue [30]: as shown in Fig. 5c, blond hair mostly appears on women’ heads and non-blond hair mostly appears on men’ heads in this dataset. Similar to Biased Waterbirds, we additionally balance the number of images per class to focus on shortcut cues (99% bias-aligned samples here).

4.2. Implementation

The previous debiasing methods usually conduct their experiments on different datasets with different network architectures and training schedules. In order to do fair comparisons, we re-implement the representative methods described in the following Sec. 4.3 with the same configurations and meticulously tune the hyperparameters.

For Colored MNIST, we employ an MLP neural network with three-hidden layers (each hidden layer consists of 100 hidden units)4. For Corrupted CIFAR-101 and Corrupted CIFAR-102, ResNet-20 [14] is employed. For BWaterbirds and BCelebA, ResNet-18 is utilized. Exhaustive experimental details are delineated in appendix.

4.3. Quantitative Comparisons

Baselines. We carefully choose various kinds of previous methods as comparison methods:

- Vanilla training with standard ERM.
- State-of-the-art implicit methods without bias information: GEOGRE [40], LfF [30], DFA [19] and SD [33].
- Bias-tailored methods with only bias-type prior knowledge: REBIAS [2], ERew [4] and PoE [4] (we combine our auxiliary biased models with ERew and PoE, converting them into implicit methods B+ERew and B+PoE).
- Explicit methods that know complete bias types and labels: Rew [36], EnD [41] and GDRO [35].

In terms of debiasing technique, Rew, ERew and LfF employ reweighting strategy based on sample number of different groups or output score of biased model; REBIAS, EnD and DFA disentangle bias and intrinsic features; PoE applies ensemble technique (product of experts) to enhance fairness of the debiased model; GDRO and GEOGRE resort to distributionally robust optimization to emphasize on worst-group performance with explicitly provided group annotations or feature-cluster-based pseudo groups; and SD adds a painstakingly designed regularization term to prevent model from only learning spurious features.

Metric. For experiments on Colored MNIST and Corrupted CIFAR-10, we evaluate models on the unbiased test sets in which the bias cues are independent of the target labels, and report the overall unbiased accuracy. For BWaterbirds and BCelebA, to evaluate the unbiased accuracy with the official test sets which are biased and imbalanced, the accuracies of each (target, bias) group are calculated separately, and then averaged to form the overall accuracy. We further provide the accuracies on bias-aligned and bias-conflicting test samples separately in appendix.

Main Results. The overall accuracy is reported in Tab. 1. Models trained with vanilla way commonly fail to obtain acceptable results on unbiased test set, and the phenomenon is aggravated as the increase of value \( \rho \). Different debiasing methods moderate bias propagation with varying degrees of capabilities. When compared with other SOTA methods, our proposed approach gains competitive results on Corrupted CIFAR-101 and noticeable improvements on other datasets across all values of \( \rho \), especially when training set is extremely biased (i.e., with a high value of \( \rho \)). For instance, the vanilla model (\( \rho = 0.995 \)) only achieves 14.25% accuracy on unbiased Corrupted CIFAR-102 test set, indicating that it is heavily biased by spurious cues. While, employing B+GA leads to 42.18% accuracy, and exceeds other prevailing debiasing methods by 6%-25%. When it comes

| \( \rho \) | Colored MNIST | Corrupted CIFAR-10 | Corrupted CIFAR-10 | BWaterbirds | BCelebA |
|------|--------------|------------------|------------------|-------------|--------|
|      | 0.95 | 0.98 | 0.99 | 0.995 | 0.95 | 0.98 | 0.99 | 0.995 | 0.95 | 0.99 |
| Vanilla | 75.93 | 61.52 | 48.81 | 35.02 | 41.29 | 26.60 | 19.68 | 14.81 | 39.26 | 24.89 | 18.16 | 14.25 |
| Rew [36] | 84.75 | 71.70 | 60.06 | 48.40 | 54.57 | 44.66 | 36.24 | 30.75 | 53.33 | 44.38 | 37.44 | 26.63 |
| REBIAS [2] | 96.61 | 92.79 | 84.41 | 74.65 | 43.20 | 27.53 | 18.17 | 15.02 | 39.99 | 24.57 | 17.57 | 14.90 |
| EnD [41] | 84.21 | 73.15 | 61.10 | 46.59 | 38.60 | 23.48 | 19.17 | 14.12 | 36.17 | 23.28 | 17.68 | 14.53 |
| GDRO [35] | 84.02 | 76.11 | 68.65 | 63.12 | 50.22 | 38.75 | 30.24 | 27.60 | 57.32 | 45.81 | 38.30 | 33.44 |
| GEOGRE [40] | 81.64 | 65.10 | 56.44 | 36.04 | 43.29 | 28.83 | 19.15 | 15.12 | 42.49 | 25.21 | 18.93 | 15.10 |
| LfF [30] | 82.70 | 79.53 | 75.42 | 65.01 | 58.65 | 49.88 | 40.59 | 34.67 | 59.21 | 46.94 | 40.08 | 32.85 |
| DFA [19] | 86.65 | 81.31 | 74.75 | 63.89 | 59.35 | 49.06 | 42.81 | 37.01 | 57.23 | 47.79 | 41.96 | 36.08 |
| SD [33] | 79.33 | 61.02 | 47.16 | 32.04 | 41.14 | 26.14 | 19.33 | 14.73 | 37.76 | 25.49 | 18.85 | 14.01 |
| B+ERew [4] | 83.62 | 78.24 | 61.57 | 46.61 | 57.17 | 45.51 | 33.47 | 25.24 | 61.70 | 46.01 | 32.97 | 18.41 |
| B+PoE [4] | 67.79 | 53.68 | 48.66 | 38.16 | 53.12 | 50.02 | 45.22 | 36.79 | 46.07 | 38.80 | 37.63 | 30.33 |
| GA | 88.48 | 81.67 | 78.78 | 70.68 | 59.67 | 50.30 | 44.41 | 35.36 | 63.01 | 54.93 | 49.81 | 43.23 |
| B+GA | 87.57 | 82.93 | 77.24 | 71.17 | 60.34 | 50.45 | 42.98 | 33.43 | 63.25 | 55.14 | 49.84 | 42.18 |

Except a convolution neural network for method REBIAS.
to the real-world dataset BCelebA, the results also show the superiority of the proposed scheme which surpasses all the other SOTA algorithms, demonstrating our method can effectively deal with subtle real-world biases. Furthermore, it is worth mentioning that performing gradient alignment with our bias-conflicting sample mining method produces very similar results to that with known bias annotations, which reveals that it is possible to discard bias annotations and achieve debiasing with unknown biases.

**Evaluation on Pseudo Label Assignment.** We further verify the effectiveness of our bias-conflicting sample mining method (Algorithm 1) by ablation studies and comparisons with other widely-used strategies. Good mining strategy should be robust and of high performance, leading to superior Precision-Recall curves over b-c scores \( \{ s_i^k \}^N_{i=1} \). We compare different mining approaches (a)-(f) as summarized in Tab. 2. PR-curves and values of average precision are obtained with \( \rho = 0.98 \). The presented results of Colored MNIST and Corrupted CIFAR-10 are obtained with \( \rho = 0.98 \), complete results are provided in appendix.

**4.4. Further Analysis**

**Impact of GA on Learning Trajectory.** We revisit the experiments on Colored MNIST and present the statistics of \( \{ g_{k}^T \}_{T=0}^{T-1} \) and \( \{ \sum_{k=1}^{C} g_{k}^T \}_{T=0}^{T-1} \) learned with Eq. (1) (vanilla), Eq. (2) (reweighting) and Eq. (4) (gradient alignment) in Fig. 7 (a), (b) and (c), respectively. For vanilla training, the gradients of bias-aligned samples overwhelm that of bias-conflicting samples at the beginning, then the model becomes biased towards spurious cues progressively. Even though at the late stage, the gradients gap between

| (a) | (b) | (c) | (d) | (e) | (f) |
|-----|-----|-----|-----|-----|-----|
| early-stopping [27] | - | ✓ | - | - | - |
| epoch-ensemble | - | - | ✓ | ✓ | ✓ | ✓ |
| GCE loss [30] | - | - | ✓ | ✓ | ✓ | ✓ |
| confident-picking | - | - | - | ✓ | ✓ |
| peer models | - | - | - | - | ✓ |

Table 2. Different bias-conflicting sample mining strategies.

| | Colored MNIST | Corrupted CIFAR-10 | Corrupted CIFAR-10 | Biased CelebA | Biased CelebA |
|---|---|---|---|---|---|
| Precision (%) | 99.90 | 95.54 | 98.63 | 79.11 | 77.66 |
| Recall (%) | 84.15 | 94.30 | 93.90 | 50.00 | 65.18 |

Table 3. Precision, recall of our mined bias-conflicting samples.

Figure 6. Precision-Recall curves and Average Precision (%) of the mined bias-conflicting samples. Best seen in color.

Figure 7. Statistics of \( \{ g_{k}^T \}_{T=0}^{T-1} \) (pink) and \( \{ \sum_{k=1}^{C} g_{k}^T \}_{T=0}^{T-1} \) (blue) on Colored MNIST. To observe clearly, results in the late stage are enlarged and shown in each figure. Best seen in color.
bias-aligned and bias-conflicting samples shrinks, it is hard to rectify the already biased model, resulting in low bias-conflicting accuracy as shown in Fig. 3. For reweighting, as the contributions of the up-weighed bias-conflicting samples and the down-weighed bias-aligned samples are close at beginning, both of them are well learned. Nonetheless, as the bias-conflicting samples are memorized rapidly due to their small quantity, the gradients from the up-weighed bias-conflicting samples become smaller than that of the down-weighed bias-aligned samples, and the model becomes biased gradually again. The proposed gradient alignment dynamically balances the contributions of bias-conflicting and bias-aligned samples throughout the training process, obtaining optimal and balanced predictions as demonstrated in multiple challenging datasets and settings.

**Strong Regularization is Not Necessary.** As shown in [35], Group DRO and Reweighting need strong regularizations, *e.g.*, early-stopping, to achieve satisfactory results, which means the final results are always inferior to that of early-stopping (ES) as listed in Tab. 4. However, it is troublesome to decide when to stop, especially in the absence of bias prior knowledge. We argue that such strong regularization maybe not necessary if network can achieve dynamic balance throughout training process. Therefore, as shown in Tab. 4, without such tricks, models trained with GA do not suffer the degeneration of performance at late stage. On the contrary, they continue to be improved and achieve stable and better performance.

**Visualization of CAM.** We visualize the activation maps using Grad-CAM [38] in Fig. 8. Vanilla training usually activates regions related to biases when making predictions, *e.g.*, the background “water” or “land” in BWaterbirds, the gender characteristics like faces and beards in BCelebA. LfF can focus attention on key areas in some situations, but there are still some deviations. Meanwhile, B+GA mostly utilizes compact essential features to make decisions. A more complete analysis is presented in appendix.

5. Conclusions and Limitations

Biased models can cause poor out-of-distribution performance and even negative social impacts. In this work, we manage to achieve debiasing with unknown biases, which is urgently required in realistic applications. We derive a new learning objective with the idea of Gradient Alignment, which dynamically balances the gradients produced from bias-aligned and bias-conflicting samples throughout learning process. To further extend GA into more common scenarios where biases are unknown, we propose a bias-conflicting sample mining method equipped with epoch-ensemble and peer-picking. Extensive experiments on synthetic and real-world datasets demonstrate the superiority of our proposed debiasing scheme when compared with other state-of-the-art methods in a fair manner.

Despite the achieved promising results, the debiasing method can be further improved in some aspects. *First*, our method and many previous approaches (such as LfF [30] and DFA [19] *etc.*) based on the assumption that there exists bias-conflicting samples in our training set. Although the assumption is in line with most actual situations, it should be noted that there are some cases where the collected training sets are completely biased (*i.e.*, $\rho = 1$), in which our method is not applicable. For these cases, we should pay attention to methods that aim to directly prevent model from only pursuing easier features, such as SD [33]. *Second*, we observe that the bias-conflicting sample mining

| Method               | Colored MNIST | Corrupted CIFAR-10 | Corrupted CIFAR-10 | Biased Waterbirds | Biased CelebA |
|----------------------|---------------|--------------------|--------------------|-------------------|---------------|
| GDRO ES Final        | 76.11         | 38.75              | 45.81              | 82.00             | 82.10         |
| GDRO Rew Final       | 72.67         | 36.06              | 41.74              | 79.99             | 70.72         |
| GDRO GA Final        | 79.99         | 44.71              | 48.70              | 85.87             | 90.30         |
| GDRO ES Final        | 79.94         | 44.66              | 44.38              | 79.44             | 90.83         |

Table 4. Overall accuracy of different methods w/o strong regularization (early-stopping). The best results are in bold.

---

4 The presented results of Colored MNIST and Corrupted CIFAR-10 are under setting $\rho = 0.98$, complete results are provided in appendix.
is not trivial, especially in complex datasets — the precision and recall achieved by our method on Biased Waterbirds and CelebA are still significantly lower than that on simple datasets like Colored MNIST and Corrupted CIFAR-10. Therefore, a better bias-conflicting sample mining method can further reduce the gap between B+GA and GA.

References

[1] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9690–9698, 2020.

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

[3] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. Advances in neural information processing systems, 32:841–852, 2019.

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

[5] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Learning to model and ignore dataset bias with mixed capacity ensembles. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 3031–3045, 2020.

[6] Elliot Creager, Jorn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In International Conference on Machine Learning, pages 2189–2200. PMLR, 2021.

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

[8] Karan Goel, Albert Gu, Yixuan Li, and Christopher Re. Model patching: Closing the subgroup performance gap with data augmentation. In International Conference on Learning Representations, 2021.

[9] Joel Goh and Melvyn Sim. Distributionally robust optimization and fairness: Closing the subgroup performance gap with data augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14992–15001, 2021.

[10] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. SIGUA: Forgetting may make learning with noisy labels more robust. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 4006–4016. PMLR, 13–18 Jul 2020.

[11] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In NeurIPS, 2018.

[12] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In International Conference on Machine Learning, pages 1929–1938. PMLR, 2018.

[13] He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), pages 132–142, 2019.

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

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In International Conference on Learning Representations, 2018.

[16] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. Advances in Neural Information Processing Systems, 33, 2020.

[17] Byungui Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9012–9020, 2019.

[18] Eunbyeup Kim, Ji hyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailed swapping augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14992–15001, 2021.

[19] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pages 247–254, 2019.

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

[21] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. Fairness without demographics through adversarially reweighted learning. arXiv preprint arXiv:2006.13114, 2020.

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

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[24] Ningwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In Proceedings of the European Conference on Computer Vision (ECCV), pages 513–528, 2018.

[25] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9572–9581, 2019.

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

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

[28] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. Advances in Neural Information Processing Systems, 31:9084–9093, 2018.

[29] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaecho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In Advances in Neural Information Processing Systems, 2020.

[30] Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Deep neural networks are biased towards simple functions. In NeurIPS, 2019.
[32] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. arXiv preprint arXiv:2007.00653, 2020.

[33] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. arXiv preprint arXiv:2011.09408, 2020.

[34] Ruggero Ragonesi, Riccardo Volpi, Jacopo Cavazza, and Vittorio Murino. Learning unbiased representations via mutual information backpropagation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2729–2738, 2021.

[35] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In International Conference on Learning Representations, 2020.

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

[37] Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. Learning from others’ mistakes: Avoiding dataset biases without modeling them. In International Conference on Learning Representations, 2020.

[38] 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.

[39] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. Advances in Neural Information Processing Systems, 33, 2020.

[40] Nimit Sohoni, Jared Dummon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. Advances in Neural Information Processing Systems, 33, 2020.

[41] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15038–15057, 2021.

[42] P. Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. In Proceedings of the 58th Conference of the Association for Computational Linguistics: Main Volume, pages 3319–3332, 2021.

[43] Zihiu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018.

[44] Lei Wu, Zhanxing Zhu, and E Weinan. Towards understanding generalization of deep learning: Perspective of loss landscapes. ArXiv, abs/1706.10239, 2017.

[45] Yadollah Yaghoobzadeh, Soroush Mehr, Remi Tachet des Combes, Timothy J Hazen, and Alessandro Sordoni. Increasing robustness to spurious correlations using forgettable examples. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3319–3332, 2021.

[46] Wei Zhu, Haitian Zheng, Haofo Liao, Weijian Li, and Jiebo Luo. Learning bias-invariant representation by cross-sample mutual information minimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 15002–15012, 2021.
In this supplementary, we present detailed dataset and training statistics (Sec. A), more comprehensive analysis of compared baselines (Sec. B), more experimental result explanations (Sec. C, Sec. D and Sec. E), hyperparameters sensitivity analysis (Sec. F) and more CAM visualization results (Sec. G). For experimental reproduction, we also provide all codes and README in zip file.

A. Dataset and Training Details

For Corrupted CIFAR-10\(^1\), it is constructed with corruptions Snow, Frost, Fog, Brightness, Contrast, Spatter, Elastic, JPEG, Pixelate and Saturate. For Corrupted CIFAR-10\(^2\), it is constructed with corruptions Gaussian Noise, ShotNoise, ImpulseNoise, SpeckleNoise, Gaussian Blur, DefocusBlur, GlassBlur, MotionBlur, ZoomBlur and Original. For Biased Waterbirds and CelebA, as mentioned in Sec. 4.1, to focus on the problem of shortcut biases, we balance the number of training images per class. After that, the used training set of Biased Waterbirds contains 1,057 (waterbird, water) samples, 1,057 (landbird, land) samples, 56 (waterbird, land) samples and 56 (landbird, water) samples, i.e., \(\rho \approx 0.95\); for the class balanced training set of Biased CelebA, there are 22,880 samples in the target-bias group (blond, female), 22,880 samples in group (non-blond, male), 231 samples in group (blond, male) and 231 samples in group (non-blond, female), i.e., \(\rho \approx 0.99\).

We implement all methods with Pytorch and run them on a Tesla V100 GPU. For experiments on Colored MNIST, we use Adam optimizer to train models for 200 epochs with learning rate 0.001, batch size 256, without any data augmentation techniques. For Corrupted CIFAR-10\(^1\) and Corrupted CIFAR-10\(^2\), models are trained for 200 epochs with Adam optimizer, learning rate 0.001, batch size 256, image augmentation including only random crop and horizontal flip. For Biased Waterbirds and CelebA, models are trained starting from imagenet pretrained weights (Pytorch torchvision version) for 100 epochs with Adam optimizer, learning rate 0.0001, batch size 256 and horizontal flip augmentation technique.

B. Compared Debiasing Methods

We further present comprehensive analysis of state-of-the-art debiasing approaches (refer to Tab. 1) as follows based on technique categories.

- **Reweighting-based strategies.** Rew [36] is a straightforward static reweighting strategy based on group sample number. Both LfF [30] and ERew [4] reassign sample weights assisted with a biased model but differ on weight assignment functions. ERew is also a static reweighting approach which employs output scores of a pre-trained biased model as the weight indicator. LfF applies dynamic weight adjustments during training with the assistance of GCE loss. LfF is implemented referring to https://github.com/alinlab/LfF and ERew is implemented referring to https://github.com/UKPLab/emnlp2020-debiasing-unknown. But all the above methods ignore the essential gradient balance issue, which results in inferior performance with respect to our gradient alignment approach, especially on extreme biased scenarios, e.g., \(\rho = 0.995\) on Colored MNIST and Corrupted CIFAR-10 datasets.

- **Feature disentanglement.** REBIAS [2] designs specific networks according to the bias type for obtaining biased representations intentionally (for our experiments, we employ CNNs with smaller receptive fields for capturing texture bias according to original paper). Then the debiased representation is learned by encouraging it to be independent with the biased one, during which Hilbert-Schmidt Independence Criterion (HSIC) is employed to measure the degree of independence between the two representations. We implement it referring to https://github.com/clovaai/rebias. Besides, relying on explicit bias labels, EnD [41] directly designs regularizers to disentangle representations with the same bias label and entangle features with different biases but belonging to the same target label, during which the cosine similarity is utilized to measure the relationship between different features. We implement it referring to https://github.com/EIDOSlab/Learning-Entangling-Disentangling-Bias. Both REBIAS and EnD try to explicitly extract disentangled feature representations, however, it is difficult to be achieved in complex datasets and tasks, leading to unsatisfactory results as shown in Tab. 1, e.g., on Corrupted CIFAR-10, BWaterbirds and BCelebA datasets. Building on LfF, DFA [19] further introduces disentangled representations to augment bias-conflicting samples in the feature-level, which is implemented referring to https://github.com/kakaoenterprise/Learning-Debiased-Disentangled. DFA is the algorithm whose results are closest to ours, especially on simple datasets (Colored MNIST and Corrupted CIFAR-10), but it still underperforms on complicate datasets such as BWaterbirds and BCelebA since feature disentanglement on such datasets is more challenge so that the augmented feature maybe not reasonable.

- **Distributionally robust optimization (DRO).** Many previous works resort to DRO to achieve model fairness. GDRO [35] employs DRO to emphasize on worst-group performance by leveraging on explicitly annotated group labels, which is implemented referring to https://github.com/kohpangwei/group_DRO. Experimental results in Tab. 1 reveal that DRO can achieve competitive overall accuracy with respect to our method.
Furthermore, to adapt it for situations where the bias information is not already available, GEOGRE [40] performs clustering based on the feature representations of the auxiliary biased models first, and then expects to obtain fair models by using DRO with the pseudo groups. While, we find that clustering with the features extracted from vanilla biased models is not robust and accurate due to overfitting, resulting in much worse performance when performing GDRO based on the imprecise clusters as shown in Tab. 1. We adopt the clustering methods utilized in GEOGRE referring to https://github.com/HazyResearch/hidden-stratification.

• Ensemble approaches. Product-of-Experts (PoE [4]) is widely adopted in NLP-related debiasing task, which tries to train a debiased model in an ensemble manner with an auxiliary biased model, by combining the softmax outputs produced from the biased and debiased models. We also compare such strategy with our GA method based on our auxiliary biased models for bias-conflicting sample mining, and the results in Tab. 1 still demonstrate that GA outperforms PoE over all experiment setups. We implement it referring to https://github.com/UKPLab/emnlp2020-debiasing-unknown.

• Regularization methods. In addition, SD [33] directly replaces the common $l_2$ regularization with an $l_2$ penalty on the model’s logits, we implement it referring to https://github.com/mohammadpz/Gradient_Starvation. While, the optimal strength of the regularization term is hard to search, which may be very different for various datasets and tasks. Similar challenge also arise in methods like EnD, etc.

C. Detailed Results

Bias-Aligned and Bias-Conflicting Accuracy. We provide the accuracies measured on the bias-aligned and bias-conflicting test samples separately in Fig 9. GA and B+GA can achieve high bias-conflicting accuracy as well as bias-aligned accuracy mostly, leading to superior overall unbiased performance as shown in Sec. 4.3. Note that, too high bias-aligned accuracy is not always good, as shown in Fig. 8, Fig. 12 and Fig. 13. Though the model e.g., vanilla model can obtain a very high illusory bias-aligned accuracy assisted with bias cues, it actually does not learn any intrinsic features, which leads to extremely poor out-of-distribution generalization.

Precision and Recall. We provide the complete results (precision, recall) of our mined bias-conflicting samples in Tab. 5 (part of the results have presented in Tab. 3).

Results of B+GA with Different Bias-Conflicting Sample Mining Strategies. We have evaluated different bias-conflicting sample mining methods (a)-(f) and showed PR curves of them in Sec. 4.3. Further, we provide the final debiasing results of B+GA with these different mining methods in Tab. 6. Apparently, B+GA with our proposed mining method (f) attains the best results.

D. Connection to Curriculum Learning

Curriculum learning shows that using easy samples first can be superior, on the contrary, anti-curriculum learning claims that employing hard samples first is useful in some situations. We investigate the strategies of ordered learning in the context of debiasing. As presented in Tab. 7, both learning easy and hard samples first lead to inferior results than performing GA throughout training process. Therefore, it is important to achieve balance between bias-conflicting and bias-aligned samples in the whole learning stage.

---

Table 5. Precision, recall of our mined bias-conflicting samples.

|            | $\rho$ | Precision | Recall |
|------------|--------|-----------|--------|
| Colored MNIST | 0.95  | 99.58     | 79.40  |
|            | 0.98  | 99.90     | 84.15  |
|            | 0.99  | 99.81     | 86.29  |
|            | 0.995 | 99.62     | 87.67  |
| Corrupted CIFAR-10$^1$ | 0.95  | 97.73     | 91.12  |
|            | 0.98  | 95.54     | 94.30  |
|            | 0.99  | 93.32     | 92.20  |
|            | 0.995 | 86.67     | 93.60  |
| Corrupted CIFAR-10$^2$ | 0.95  | 99.49     | 93.52  |
|            | 0.98  | 98.63     | 93.90  |
|            | 0.99  | 97.56     | 95.80  |
|            | 0.995 | 94.02     | 94.40  |
| Biased Waterbirds | 0.95  | 77.66     | 65.18  |
| Biased CelebA  | 0.99  | 79.11     | 50.00  |

Table 6. Results of B+GA with different bias-conflicting sample mining methods on Colored MNIST ($\rho = 0.99$) and Biased CelebA.

| Colored MNIST | Biased CelebA |
|---------------|---------------|
| P | R | Unbiased Accuracy | P | R | Unbiased Accuracy |
| (a) | 98.25 | 9.36 | 47.28 | 77.78 | 1.52 | 78.89 |
| (b) | 99.17 | 20.07 | 58.59 | 79.17 | 20.56 | 86.16 |
| (c) | 98.28 | 38.29 | 73.28 | 79.79 | 16.23 | 82.77 |
| (d) | 99.65 | 47.83 | 72.85 | 79.25 | 27.27 | 86.16 |
| (e) | 99.68 | 52.68 | 72.73 | 79.22 | 39.61 | 87.93 |
| (f) | 99.81 | 86.29 | 77.24 | 79.11 | 50.00 | 89.69 |
| Vanilla | 48.81 | 72.06 |
(a) On Colored MNIST with $\rho = 0.95, 0.98, 0.99, 0.995$ from left to right.

(b) On Corrupted CIFAR-10$^1$ with $\rho = 0.95, 0.98, 0.99, 0.995$ from left to right.

(c) On Corrupted CIFAR-10$^2$ with $\rho = 0.95, 0.98, 0.99, 0.995$ from left to right.

(d) On Biased Waterbirds

(e) On Biased CelebA

Figure 9. Bias-aligned accuracy (horizontal-axis) and bias-conflicting accuracy (vertical-axis).
Table 7. Performing GA with ordered learning on Colored MNIST ($\rho = 0.98$). “Easy” or “Hard” means only using bias-aligned or bias-conflicting training samples to update model.

| epochs  | Unbiased Accuracy |
|---------|-------------------|
| 1-100   | 61.52             |
| 101-200 | 71.09             |
| Easy GA | 75.83             |
| Hard GA | 81.67             |

Table 8. Unbiased Accuracy on Colored MNIST with few bias-conflicting samples.

| rho     | #bias-conflicting samples | Vanilla | Rew  | GA    |
|---------|---------------------------|---------|------|-------|
| 0.997   | 180                       | 25.85   | 40.97| 66.47 |
| 0.999   | 60                        | 16.94   | 23.18| 45.58 |
| 0.9995  | 30                        | 14.12   | 20.37| 48.18 |

E. Results of GA when $\rho \to 1$

As discussed in Sec. 5, if the collected training set is completely biased (i.e., $\rho = 1$), GA is not applicable. So, we want to know how GA performs when there are only a few bias-conflicting samples (i.e., $\rho \to 1$). The results are provided in Tab. 8, from which we find GA can achieve noticeable improvement even with few bias-conflicting samples. These results demonstrate that the proposed scheme is robust on various situations.

F. Hyperparameters

In this subsection, we analytically and experimentally determine the hyperparameters introduced in our whole pipeline (balance ratio $\gamma$ in GA, confidence threshold $\eta$ in peer-picking and threshold $\tau$ in bias-conflicting sample determination). We find that the hyperparameters are not sensitive around their empirical values, which further demonstrates the robustness of our proposed approach.

- **Balance ratio $\gamma$.** As mentioned in Sec. 3.1, when $\gamma \to 0$, models generally intend to exploit bias-aligned samples more and when $\gamma \to \infty$, the behavior is reversed. The empirical value is exact 1, and we conduct B+GA experiments with different $\gamma$ on Corrupted CIFAR-10$^2$ ($\rho = 0.995$). As shown in Fig. 10, the proposed method is not particularly sensitive to $\gamma \in [1.0, 2.0]$ which is reasonable, as $\gamma$ in such region make the contributions from bias-conflicting samples close to that from bias-aligned samples. In our experiments, $\gamma$ is set to 1.6 for all settings on Colored MNIST, Corrupted CIFAR-10$^1$ and Corrupted CIFAR-10$^2$, and 1.0 for all settings on Biased Waterbirds and Biased CelebA.

- **Confidence threshold $\eta$.** We investigate the impact of the confidence threshold $\eta$ which is employed to pick up confident samples to train auxiliary biased models as described in Sec. 3.2. When $\eta \to 0$, most of training samples will be used including the bias-conflicting ones, resulting in low recall of the bias-conflicting samples; when $\eta \to 1$, most of training samples will be discarded including the hard but bias-aligned ones, leading to low precision of the mined bias-conflicting samples. The determination of threshold $\eta$ is related to the number of categories of interest and the difficulty of specific task. For example, in our experiments, $\eta$ is set to 0.05 for all settings on Colored MNIST, Corrupted CIFAR-10$^1$ and Corrupted CIFAR-10$^2$ (10-class classification tasks), and 0.9 for all settings on Biased Waterbirds and Biased CelebA (2-class classification tasks). As shown in Fig. 11, the proposed bias-conflicting sample mining method is not...
in inferior unbiased performance finally.

G. More Visualization

We provide more activation maps of different training methods in Fig. 12 (on Biased Waterbirds) and Fig. 13 (on Biased CelebA).

![Precision-Recall curves and Average Precision (%) of the mined bias-conflicting samples with varying \( \eta \). Best seen in color.](image)

![Table 9. Results of B+GA with varying \( \tau \) on Colored MNIST (\( \rho = 0.98 \)).](table)

| \( \tau \) | Precision | Recall | Unbiased Accuracy |
|---------|-----------|--------|------------------|
| 0.999   | 100.00    | 55.35  | 79.29            |
| 0.80    | 99.90     | 84.15  | 82.93            |
| 0.75    | 99.90     | 84.40  | 82.94            |
| 0.70    | 99.90     | 84.65  | 83.61            |
| 0.05    | 50.51     | 95.16  | 72.41            |
| Vanilla |           |        | 61.52            |

particularly sensitive to \( \eta \).

- **Threshold \( \tau \).** As mentioned in Sec. 4.3, high precision of the mined bias-conflicting samples guarantees that gradient alignment can work well, and high recall further increases the diversity of the up-weighted samples. Therefore, to ensure the precision first, \( \tau \) is typically set to 0.8 for all experiments. Here, we study B+GA with different threshold \( \tau \) on Colored MNIST (\( \rho = 0.98 \)). As shown in Tab. 9, B+GA is not particularly sensitive to \( \tau \), however, a too high or too low value of \( \tau \) can cause low recall or low precision of the mined bias-conflicting samples, resulting
Figure 12. Further visualized activate maps of different models on Biased Waterbirds. Best seen in color.
| Input | Vanilla | LfF | EnD | GDRO | Rew | B+GA | GA |
|-------|---------|-----|-----|------|-----|------|----|
| ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) | ![Image](image5.png) | ![Image](image6.png) | ![Image](image7.png) | ![Image](image8.png) |
| ![Image](image9.png) | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) | ![Image](image16.png) |
| ![Image](image17.png) | ![Image](image18.png) | ![Image](image19.png) | ![Image](image20.png) | ![Image](image21.png) | ![Image](image22.png) | ![Image](image23.png) | ![Image](image24.png) |
| ![Image](image25.png) | ![Image](image26.png) | ![Image](image27.png) | ![Image](image28.png) | ![Image](image29.png) | ![Image](image30.png) | ![Image](image31.png) | ![Image](image32.png) |
| ![Image](image33.png) | ![Image](image34.png) | ![Image](image35.png) | ![Image](image36.png) | ![Image](image37.png) | ![Image](image38.png) | ![Image](image39.png) | ![Image](image40.png) |
| ![Image](image41.png) | ![Image](image42.png) | ![Image](image43.png) | ![Image](image44.png) | ![Image](image45.png) | ![Image](image46.png) | ![Image](image47.png) | ![Image](image48.png) |
| ![Image](image49.png) | ![Image](image50.png) | ![Image](image51.png) | ![Image](image52.png) | ![Image](image53.png) | ![Image](image54.png) | ![Image](image55.png) | ![Image](image56.png) |
| ![Image](image57.png) | ![Image](image58.png) | ![Image](image59.png) | ![Image](image60.png) | ![Image](image61.png) | ![Image](image62.png) | ![Image](image63.png) | ![Image](image64.png) |

Figure 13. Further visualized activate maps of different models on Biased CelebA. Best seen in color.