LEARNING DEBIASED REPRESENTATIONS VIA CONDITIONAL ATTRIBUTE INTERPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

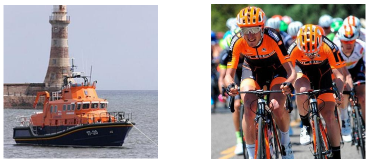
An image is usually described by more than one attribute like “shape” and “color”. When a dataset is *biased*, i.e., most samples have attributes spuriously correlated with the target label, a Deep Neural Network (DNN) is prone to make predictions by the “unintended” attribute, especially if it is easier to learn. To improve the generalization ability when training on such a biased dataset, we propose a χ^2-model to learn debiased representations. First, we design a χ-shape pattern to match the training dynamics of a DNN and find Intermediate Attribute Samples (IASs) — samples near the attribute decision boundaries, which indicate how the value of an attribute changes from one extreme to another. Then we rectify the representation with a χ-structured metric learning objective. Conditional interpolation among IASs eliminates the negative effect of peripheral attributes and facilitates retaining the intra-class compactness. Experiments show that χ^2-model learns debiased representation effectively and achieves remarkable improvements on various datasets. (*This is the modified version with a dark blue mark.)

1 INTRODUCTION

Deep neural networks (DNNs) have emerged as an epoch-making technology in various machine learning tasks with impressive performance (LeCun et al., 2015; Bengio et al., 2021). In some real applications, an object may possess multiple attributes, and some of them are only spuriously correlated to the target label. For example, in Figure 1, the intrinsic attribute of an image annotated by “lifeboats” is its *shape*. Although there are many lifeboats colored orange, a learner can not make predictions through the *color*, *i.e.*, there is a misleading correlation from attribute as *one containing “orange” color is the target “lifeboats”*. When the major training samples can be well discerned by such peripheral attribute, especially learning on it is easier than on the intrinsic one, a DNN is prone to *bias* towards that “unintended” bias attribute (Torralba & Efros, 2011; Khosla et al., 2012; Tommasi et al., 2015; Geirhos et al., 2019; Brendel & Bethge, 2019; Xiao et al., 2021; Singla & Feizi, 2022), like recognizing a “cyclist” wearing orange as a “lifeboat”. Similar spurious attribute also exists in various applications such as recommendation system (Cañamares & Castells, 2018; Morik et al., 2020; Zhang et al., 2021b) and neural language processing (Zhao et al., 2017; He et al., 2019; Selvaraju et al., 2019; Mendelson & Belinkov, 2021; Guo et al., 2022).

Given such a biased training dataset, how to get rid of the negative effect of the misleading correlations? One intuitive solution is to perform special operations on those samples highly correlated to the bias attributes, which requires additional supervision, such as the pre-defined bias type (Kim et al., 2019; Wang et al., 2020; Agarwal et al., 2020; Goel et al., 2021; Tartaglione et al., 2021; Geirhos et al., 2019; Bahng et al., 2020; Minderer et al., 2020; Li et al., 2021). Since prior knowledge of the dataset bias requires expensive manual annotations and is naturally missing in some applications, learning a debiased model without additional supervision about bias is in demand. Nam *et al*. Nam et al. (2020) identify samples with intrinsic attributes based on the observation that malignant bias attributes are often easier-to-learn than others. Then the valuable samples for a debiasing scheme could be dynamically reweighted or augmented (Geirhos et al., 2019; Minderer et al., 2020; Lee et al., 2021). However, the restricted number of such samples implies uncertain representations and limits its ability to assist in debiasing.

To leverage more valuable-for-debiasing knowledge, we take a further step in analyzing the representation space of naïvely-training dynamics, especially focusing on the discrepancies in attributes with a different learning difficulty. As we will later illustrate in Figure 2, an attribute-based DNN pushes and fits on the easier bias attribute initially. The intrinsic attribute is then forced to shift in a “lazy” manner. The bias attribute that is pushed away first leaves a large margin boundary. Since the space of the other intrinsic attribute is filled with many different samples on bias attribute, it has a large intra-class variance, like a “hollow”. The representation is biased toward one side of the “hollow”, *i.e.*, those samples aligned with the bias attribute. Without the true intra-class structure, the model becomes biased.



(a) Orange lifeboat (b) Orange cyclists

97.8%	**lifeboat**	57.0%	**lifeboat**
0.5%	beacon	13.8%	bicycle for two
0.4%	container ship	9.0%	toyshop

Figure 1: Classification of a standard ResNet-50 of **(a)** an orange lifeboat in the training set (with both *color* and *shape* attributes), and **(b)** an orange cyclist for the test (aligned with *color* attribute but conflicting with the *shape* one). Most of the lifeboats in the training set are orange. The biased model is prone to predict via the “unintended” *color* attribute rather than the intrinsic *shape*.

From the above observation, it is crucial to fill the intra-class “hollow” and remodel the representation compactness. Notice that the samples shifting to the two sides of the “hollow” have different characteristics, as aligned with the bias attribute and conflicting with it, respectively. We can find samples with an intermediate attribute state between the above two samples. We call this type of sample the *Intermediate* Attribute Samples (IASs) which are near the decision boundary. When we *condition* (fix) on the intrinsic attribute, IASs vary on the other bias attribute and are exactly located in the “hollow” with low-density structural knowledge. Further, we can mine different samples, including IASs, based on the distinct training dynamics.

To this end, we propose our two-stage χ^2-model. In the first stage, we train a vanilla model on the biased dataset and record the sample-wise training dynamics w.r.t. both the target and the most obvious non-target classes (as the bias ones) along the epochs. An IAS is often predicted as a non-target class in the beginning and then switched to its target class gradually, making its dynamics plot a χ-shape. Following this observation, we design a χ-shape pattern to match the training samples. The matching score ranks the mined samples according to the bias level, *i.e.*, how much they are biased towards the side of the bias attribute. Benefiting from the IASs, we conduct conditional attribute interpolation, *i.e.*, fixing the value of the target attribute. We interpolate the class-specific prototypes around IASs with various bias ratios. These conditional interpolated prototypes precisely “average out” on the bias attribute. From that, we design a χ-structured metric learning objective. It pulls samples close to those same-class interpolated prototypes, then intra-class samples become compact, and the influence of the bias attribute is removed. Our χ^2-model learns debiased representation effectively and achieves remarkable improvements on various datasets. Our contributions are summarized as

- We claim and verify that Intermediate Attribute Samples (IASs) distributed around attribute decision boundaries facilitate learning a debiased representation.
- Based on the diverse learning behavior of different attribute types, we mine samples with varying bias levels, especially IASs. From that, we interpolate bias attribute conditioned on the intrinsic one and compact intra-class samples to remove the negative effect of bias.
- Experiments on benchmarks and a newly constructed real-world dataset from NICO (He et al., 2021) validate the effectiveness of our χ^2-model in learning debiased representations.

2 A Closer Look at Learning with the Bias Attribute

After the background of learning on a biased dataset, we analyze the training dynamics of the model.

2.1 Problem Definition

Given a training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, each sample $\mathbf{x}_i$ is associated with a class label $y \in \{1, 2, \cdots, C\}$. We aim to find a decision rule $h_{\boldsymbol{\theta}}$ that maps a sample to its label. $h_{\boldsymbol{\theta}}$ is optimized by fitting all the training samples, *e.g.*, minimizing the cross entropy loss as follows:

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{\text{train}}} \left[-\log \Pr \left(h_{\boldsymbol{\theta}}(\mathbf{x}_i) = y_i \mid \mathbf{x}_i \right) \right] \ . \tag{1}$$

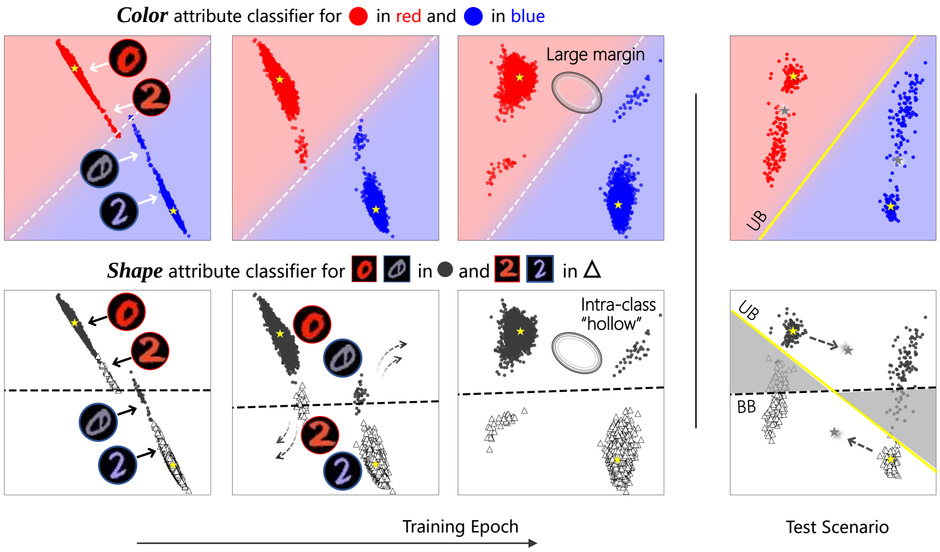

Color attribute classifier for ● in red and ● in blue
Large margin
UB
Shape attribute classifier for 🟥 ⬛ in ● and 🟥 ⬛ in Δ
Intra-class
"hollow"
UB
BB
Training Epoch
Test Scenario

Figure 2: **An illustration of the training dynamics of a naïvely-trained model on a biased dataset.** The different attribute classes are drawn with a specific *color* (top row) or *shape* (bottom row). The first three columns correspond to the sequential training progress of these two classifiers, and the final column shows the test scenario. The easier-to-learn *color* is fitted first, which leaves a large margin on decision boundary and correspondingly triggers the *shape* attribute intra-class "hollow". "UB" and "BB" are the abbreviations of "unbiased boundary" and "biased boundary". The biased model cannot generalize well as the shadow area in the last column frames the *shape*-misclassified samples. We use yellow stars to indicate the shifted class centers of the training samples and gray stars for those of the test samples.

We denote $h_{\boldsymbol{\theta}} = \arg\max_{c\in[C]} \mathbf{w}_c^\top f_{\boldsymbol{\phi}}(\mathbf{x})$, where $f_{\boldsymbol{\phi}} \to \mathbb{R}^d$ is the feature extraction network and $\{\mathbf{w}_c\}_{c\in[C]}$ is top-layer C-class classifier. The $\boldsymbol{\theta}$ represents the union of learnable parameters $\boldsymbol{\phi}$ and $\mathbf{w}$. We expect the learned $h_{\boldsymbol{\theta}}$ to have the high discerning ability over the test set $\mathcal{D}_{\text{test}}$ which has the same form as the training set $\mathcal{D}_{\text{train}}$.

In addition to its class label, a sample could be described based on various attributes. If an attribute is spuriously correlated with the target label, we name it *non-target bias* attribute a_b. The attribute that intrinsically determines the class label is the *target* attribute a_y. For example, when we draw different handwritten digits in the MNIST dataset with specific colors (Kim et al., 2019), the *color* attribute will not help in the model generalization since we need to discern digits by the *shape*, *e.g.*, "1" is like the stick. However, if almost all training images labeled "1" are in the same "yellow" color, the decision rule *image in "yellow" is digit "1"* will perform well on a such biased training set.

In the task of learning with a biased training set (Li & Vasconcelos, 2019; Kim et al., 2019; Nam et al., 2020), the bias attribute a_b on most of the same-class samples are consistent, and spuriously correlated with the target label (as example digit "1" in "yellow" above), so a model $h_{\boldsymbol{\theta}}$ that relies on a_b or the target attribute a_y will both perform well on $\mathcal{D}_{\text{train}}$. In real-world applications, it is often easier to learn to rely on a_b than on a_y, such as "background" or "texture" is easier-to-learn than the object (Shah et al., 2020; Xiao et al., 2021). Therefore a model is prone to recognize based on the a_b. Such a *simplicity bias* (Arpit et al., 2017; Palma et al., 2019; Pérez et al., 2019; Shah et al., 2020) dramatically hurts the generalization of an unbiased test set. Nam *et al.* Nam et al. (2020) also observe that the loss dynamics indicate the easier a_b is learned first, where the model is distracted and fails to learn a_y.

Based on the behaviors of the "ultimate" biased model, samples in $\mathcal{D}_{\text{train}}$ are split into two sets. Those training samples that could be correctly predicted based on the bias attribute a_b are named as *Bias-Aligned* (BA) samples (as example "yellow digit 1" above), while the remaining ones are *Bias-Conflicting* (BC) samples (as digit "1" of other colors). The number of BC samples is extremely small, and previous methods emphasize their role with various strategies (Geirhos et al., 2019; Nam et al., 2020; Minderer et al., 2020; Lee et al., 2021). For additional related methods of learning a debiased model (Li & Vasconcelos, 2019; Clark et al., 2019; Sagawa et al., 2020; Cheng et al., 2021;

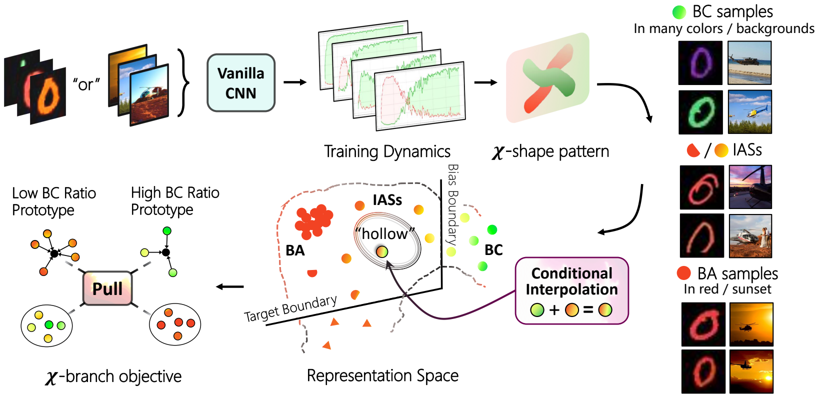


Figure 3: **An illustration of the χ^2-model.** In the first stage (top row, *left to right*), we match and mine all samples with a χ-shape pattern. As shown in the right part, the images are getting biased towards the side of peripheral bias attribute from top to bottom, *i.e.*, from various *colors* or *backgrounds* to a single *red* or *sunset*. Different shapes (○ or Δ) and colors (orange or green) drawn in the *Representation Space* indicate the target and bias attribute, respectively. In the second stage (bottom row, *right to left*), we construct prototypes by conditional interpolating around IASs with various ratios and design the χ-structured metric learning objective to pull the intra-class samples.

Hendricks et al., 2018; Wang et al., 2019; Cadène et al., 2019; Arjovsky et al., 2019; Zhu et al., 2021; Liu et al., 2021; Kim et al., 2022; Kirichenko et al., 2022) please see the Appendix B.

2.2 The Training Dynamics when Learning on a Biased Training Set

We analyze the training dynamics of a naïvely trained model in Eq. 1 on the Colored MNIST dataset. The non-target bias attribute is the *color* and the target attribute is the *shape*. For visualization shown in Figure 2, we set the output dimension of the penultimate layer as two. In addition to the learned classifier on shape attribute a_y , simultaneously, we add another linear classifier on top of the embedding to show how the decision boundary of color attribute a_b changes. More details are described in the supplementary material. Focusing on the precedence relationship for learning a_y and a_b, we have the following observations:

- **The easier-to-learn bias attribute *color* is fitted soon.** The early training stage is shown in the first column. Both color and shape attribute classifiers discern by different colors and do correctly on almost all BA samples (red "0" and blue "2", *about* 95% *of the training set*).
- **The target attribute *shape* is learned later in a "lazy" manner.** To further fit all shape labels, the model focuses on the limited BC samples (blue "0" and red "2", *correspondingly about* 5%) that cannot be perfectly classified by color. It pushes minor BC representations to the other (correct) side instead of adjusting the decision boundary.
- The ahead-*color* and lagged-*shape* learning process leaves **a large margin of *color* attribute boundary**, which further triggers **the *shape* attribute intra-class "hollow"**. Because the representation of different colors is continuously pushed away (classified) before that of the shape, the gaps between different color attribute clusters are significantly larger than that of the shape attribute.
- Since there is an intra-class "hollow" between BA and BC samples which is conditioned on a particular *shape*, **the true class representation is deviated toward *color***. The fourth column shows that the training class centers (yellow stars) and the test ones (gray stars) are mismatched. The true class center is located in the low-density "hollow" between shape-conditioned BA and BC samples.

Previous observations indicate that the before-and-latter learning process on attributes of different learning difficulties leaves the model to lose intra-class compactness, primarily when learning relying on the bias attribute is easier. To alleviate class center deviation towards the BA samples, only emphasizing the BC samples is insufficient due to their scarcity. In addition, we propose to utilize Intermediate Attribute Samples (IASs), *i.e.*, the samples near the attribute decision boundary and remodel the shifted representation. Especially when conditioned on the target attribute, the IASs vary on bias attribute and fill in the low-density intra-class "hollow" between BA and BC samples.

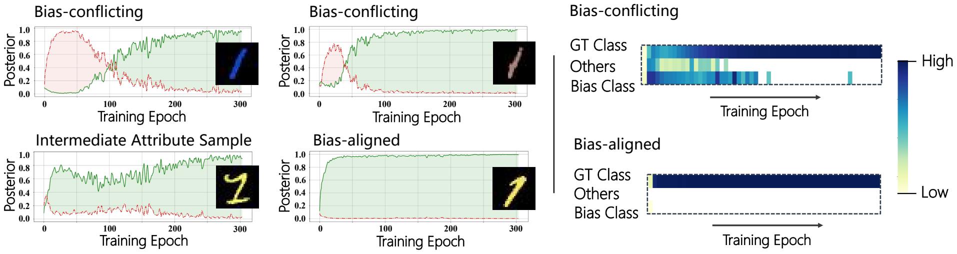


Figure 4: **Left**: The change of posterior over the GT-class (green curve) as well as the bias class (red curve) over four samples. The two curves over BC samples and IAS demonstrate a χ-shape, which is different from the curves over the BA sample. **Right**: The change of prediction frequencies of BC or BA samples along with the training epochs. The statistics are calculated over all BC or BA samples. A BA sample is easily predicted as the GT-class from the initial training stage, while a BC sample changes its prediction from the bias class to the GT-class gradually.

3 χ^2-Model

To mitigate the representation deviation and compact the intra-class "hollow", we leverage IASs to encode how the bias attribute changes from one extreme (major BA samples) to another (minor BC samples). Then, the variety of the bias attribute could be interpolated when conditioned on a particular target attribute. We propose our two-stage χ^2-model, whose notion is illustrated in Figure 3. First, the χ^2-model discovers IASs based on the training dynamics of the vanilla model in subsection 3.1. Next, we analyze where the top-ranked samples with a χ-shape pattern are as well as their effectiveness in debiasing in subsection 3.2. A conditional attribute interpolation step with IASs then fills in the low-density "hollow" to get a better estimate of the class-specific prototypes. By pulling the samples to the corresponding prototype, the χ-structured metric learning makes intra-class samples compact in subsection 3.3. Following subsection 2.2, we investigate the Colored MNIST dataset. Results on other datasets are consistent.

3.1 Scoring Samples with a χ-shape Pattern

From the observations in the previous section, we aim to collect IASs to reveal how BC samples shift and leave the intra-class "hollow" between them and BA ones. As discussed in subsection 2.2, the vanilla model fits BC samples later than BA ones, which motivates us to score the samples from their training dynamics. Once we have the score pattern to match and distinguish BA and BC samples, IASs, with intermediate scores can be extracted and available for the next debiasing stage. In the following, we denote the posterior of the Ground-Truth class (GT-class) y_i for a sample $\mathbf{x}_i$ as

$$\Pr\left(h_{\boldsymbol{\theta}}(\mathbf{x}_i) = y_i \mid \mathbf{x}_i\right) = \mathrm{softmax}(\mathbf{w}_c^\top f_{\boldsymbol{\phi}}(\mathbf{x}))_{y_i} \ , \tag{2}$$

the larger the posterior, the more confident a model predicts $\mathbf{x}_i$ with y_i. For notation simplicity, we abbreviate the posterior as $\Pr\left(y_i \mid \mathbf{x}_i\right)$. The target posterior of a BA sample reaches one or becomes much higher than other categories soon after training several epochs, while the posterior of a BC sample has a delayed increase. To sufficiently capture the clues on the change of bias attribute, we also analyze the posterior of the *most obvious non-GT* attribute, which reveals how the dataset bias influences a sample. Denote the model at the t-th epoch plus the superscript t, such as $h_{\boldsymbol{\theta}}^t$. we take the bias class for the sample $\mathbf{x}_i$ at epoch t as $b_i^t = \arg\max_{c\in[C],c\neq y_i} \left(\mathbf{w}_c^\top f_{\boldsymbol{\phi}}(\mathbf{x})\right)^t$. Then, we define the non-GT bias class as the most frequent b_i^t along all epochs, *i.e.*, $b_i = \mathrm{max\_freq}\{b_i^t\}_{t=1}^T$. A sample has a larger bias class posterior when it has low confidence in its target class and vice versa.

Taking posteriors of both y_i and b_i into account, a BA sample has larger $\Pr\left(y_i \mid \mathbf{x}_i\right)$ and small $\Pr\left(b_i \mid \mathbf{x}_i\right)$ along all its training epochs. For a BC sample, $\Pr\left(y_i \mid \mathbf{x}_i\right)$ increases gradually and meanwhile $\Pr\left(b_i \mid \mathbf{x}_i\right)$ decreases. We verify the phenomenon on Colored MNIST dataset in Figure 4 (left). For BA samples (yellow "1"), the two curves demonstrate a "rectangle", while for BC samples (blue "1"), the two curves have an obvious intersection and reveal a "χ" shape. The statistics for the change of posteriors are shown in Figure 4 (right). Therefore, how much the training dynamics match the "χ" shape reveals the probability of a sample that shifts from the major BA clusters to minor BC ones. We design a χ-shape for the dynamics of losses to capture such BC-specific properties. The change of sample-specific loss for ground-truth label and bias label over T epochs could be summarized by $\mathcal{L}_{\mathrm{CE}}$. Then, we use two exponential χ-shape functions χ_{pattern} to capture the ideal loss shape of the BC sample, *i.e.*, the severely shifted case.

$$\mathcal{L}_{\mathrm{CE}}\left(\mathbf{x}_i\right)=\left(\begin{array}{l}\mathcal{L}_{\mathrm{CE}}^{gt}\left(\mathbf{x}_i\right)=\left\{-\log \operatorname{Pr}^t\left(y_i \mid \mathbf{x}_i\right)\right\}_{t=1}^T \\ \mathcal{L}_{\mathrm{CE}}^{b}\left(\mathbf{x}_i\right)=\left\{-\log \operatorname{Pr}^t\left(b_i \mid \mathbf{x}_i\right)\right\}_{t=1}^T\end{array}\right), \ \chi_{\text{pattern}}=\left(\begin{array}{l}\mathrm{p}^{gt}=\left\{e^{-A_1 t}\right\}_{t=1}^T \\ \mathrm{p}^{b}\ =\left\{e^{A_2 t}\right\}_{t=1}^T\end{array}\right),$$

where A_1 and A_2 are the matching factors. They could be determined based on the dynamics of prediction fluctuations. For more details please see the supplementary material. The χ_{pattern} encodes the observations for the most deviated BC samples. To match the loss dynamics with the pattern, we use the inner product over the two curves:

$$\begin{aligned}\mathbf{s}(\mathbf{x}_i)=\left\langle\mathcal{L}_{\mathrm{CE}}\left(\mathbf{x}_i\right), \chi_{\text{shape}}\right\rangle &=\left\langle\mathcal{L}_{\mathrm{CE}}^{gt}\left(\mathbf{x}_i\right), \mathrm{p}^{gt}\right\rangle+\left\langle\mathcal{L}_{\mathrm{CE}}^{b}\left(\mathbf{x}_i\right), \mathrm{p}^{b}\right\rangle \\ &=\sum_{t=1}^{T}-e^{-A_1 t} \cdot \log \operatorname{Pr}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)=y_i \mid \mathbf{x}_i\right)-e^{A_2 t} \cdot \log \operatorname{Pr}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right)=b_i \mid \mathbf{x}_i\right) .\end{aligned} \tag{3}$$

The inner product $\mathbf{s}(\mathbf{x}_i)$ takes the area under the curves (AUC) into account, which is more robust w.r.t. the volatile loss changes. When $\mathbf{s}(\mathbf{x}_i)$ score goes from low to high, the sample varies on the bias level, *i.e.*, from BA samples to IASs, and then to BC samples.

3.2 Where IASs Are and Why IASs Can Help to Learn a Debiased Model?

Combining the analysis in subsection 2.2 and collecting ranked samples by $\mathbf{s}(\mathbf{x}_i)$, we find there are two types of IASs according to the representation near the target attribute decision boundary (as "0" for complex shapes in Figure 3), or that of the bias attribute (as helicopter in intermediate transitional "sunset" background). **(1)** If an IAS has an intermediate target attribute value, it may be a difficult samples and contains rich information about the target class boundaries. **(2)** If an IAS is in an intermediate state on the bias attribute, it may help to fill in the intra-class vacant "hollow" when conditioning (fixing) on the target attribute. Both types of IASs are similar to BC samples but from two directions, *i.e.*, compared to the BA samples, they contain richer semantics on target or bias attributes. In the representation space, they are scattered between BA and BC samples, compensating for the sparsity of BC samples and valuable for debiasing. We will show how χ-structured objective with IASs help to remodel the true class centers in the following subsection.

Table 1: The classification accuracy on the unbiased test sets of vanilla models. Various training sampling strategies are compared. "0-1" denotes only using BC samples. "Step-wise" denotes applying uniformly higher and lower weights on BC and BA samples. "χ-pattern" denotes sampling with scores calculated by our χ^2-model. The best result are in bold, while the second-best ones are with underlines. **C-CIFAR-10** is a similar biased dataset as **C-MNIST**.

Dataset	**C-MNIST**		**C-CIFAR-10**	
Ratio (%)	99.9	99.5	99.9	99.5
Vanilla	28.58	59.29	26.91	30.16
+ 0-1	**54.78**	70.41	18.73	25.06
+ Step-wise	41.68	<u>73.52</u>	<u>32.12</u>	<u>35.91</u>
+χ-pattern	<u>52.67</u>	**80.29**	**35.47**	**37.83**

We illustrate the importance of IASs with simple experiments on biased Colored-MNIST and Corrupted CIFAR-10 datasets. Details of the datasets are described in subsection 4.1. We investigate whether various reweighting strategies on the vanilla model improve the generalization ability over an unbiased test set. We use "0-1" to denote the strategy that utilizes only the BC samples. "step-wise" means we apply uniformly higher (ratio of BA samples) and lower weights (one minus above ratio) to BC and BA samples. Our "χ-pattern" smoothly reweights all samples with the matched scores, where BC samples as well as IASs have relatively larger weights than the remaining BA ones. The results are listed in Table 1, where we find that simple reweighting strategies easily improve the performance of a vanilla classifier, which verifies the importance of emphasizing BC-like samples. Our "χ-pattern" gets the best results in most scenarios, indicating that higher resampling weights on the IASs and BC samples assist the vanilla model to better frame the representation space.

3.3 Learning Debiased Representation from a χ-Structured Objective

Although the BA samples are severely biased towards the bias attribute, the BC samples, integrating the rich bias attribute semantics, naturally make the representation independent of the biased influence (Hong & Yang, 2021). An intuitive approach for debiasing is to average over BC samples and

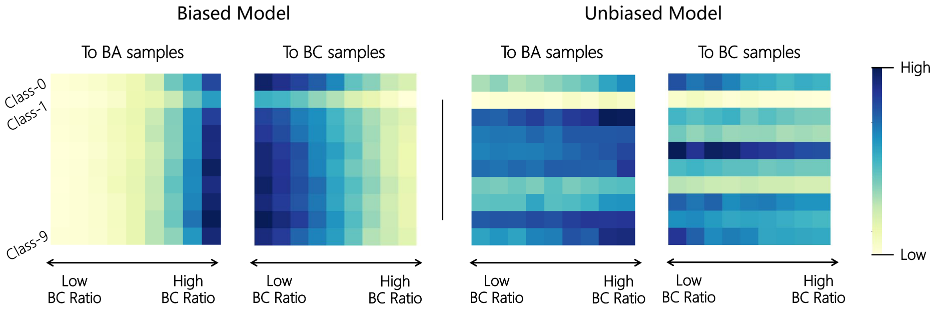


Figure 5: **Heatmap of the mean distance from the sample to its conditional interpolated prototypes.** We construct the prototype with mixing over the same-class subset but with different ratios of BC samples. Then the mean distances between the sample and those prototypes are measured. For a biased model (left two), when the BC ratio γ of interpolated prototype $\mathbf{p}_\gamma$ changes from low to high (horizontal direction), the distance of a sample to different $\mathbf{p}_\gamma$ varies hugely. The BA samples are closer to the low ratio ones, while the BC samples behave the opposite. An intra-class "hollow" exists. For an unbiased model (right two), the distances from any sample to $\mathbf{p}_\gamma$ with different ratios are almost the same.

classified by the BC class centers. However, the sparsity of BC samples induces an erratic estimation which is far from the true class center, as shown in Figure 2.

Benefiting from the analysis that BC-like IASs better estimate the intra-class structure, we target conditional interpolating around it, *i.e.*, mixing the same-class samples with different BC-like scores to remodel the intermediate samples between BA and BC samples. From that, we can construct many prototypes closer to the real class center and pull samples to these prototypes to compact the intra-class space. Combined with the soft ranking score from the χ-pattern in the previous stage, we build two pools (subsets) of the samples denoted as $\mathcal{D}_{\|}$ and $\mathcal{D}_{\perp}$. The $\mathcal{D}_{\perp}$ pool collects the top-rank sampels and most of them are BC samples and IASs. The $\mathcal{D}_{\|}$ pool is sampled from the remaining (BA) part according to the score. With the help of $\mathcal{D}_{\|}$ and $\mathcal{D}_{\perp}$, we construct multiple *bias bags* (subset) $\mathcal{B}_\gamma$ with bootstrapping where the *ratio* of BC samples is γ.

$$\mathcal{B}_\gamma = \left\{(\mathbf{x}_i, y_i) \,\middle|\, \mathrm{num}\left(\mathcal{D}_{\perp}\right) : \mathrm{num}\left(\mathcal{D}_{\|}\right) = \gamma\right\} \;, \tag{4}$$

where $\mathrm{num}\left(\mathcal{D}\right)$ equals the number of samples in $\mathcal{D}$. When γ is low to high, the $\mathcal{B}_\gamma$ contains samples ranging from the extremes of BA samples to the IASs, and then to the BC ones. Based on $\mathcal{B}_\gamma$, we compute the prototype, *i.e.*, averaged on $\mathcal{B}_\gamma$ to interpolate bias attribute conditioned on the particular target attribute. For example, the prototype conditioned on class c is formalized as $\mathbf{p}_{\gamma,c}$:

$$\mathbf{p}_{\gamma,c} = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i)\in\mathcal{B}_\gamma} f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right) \cdot \mathbb{I}\left[y_i = c\right] \;. \tag{5}$$

To further demonstrate the significance of intra-class compactness, we design the experiments to study the difference between a biased vanilla model and the unbiased oracle model (well-trained on an unbiased training set). We measure the mean distance between samples and their multiple conditional interpolated prototypes with changing ratio γ. If the prototypes are shifted with changing γ, that indicates a large intra-class deviation exists. As shown in Figure 5, for a biased model, when γ decreases, $\mathbf{p}_\gamma$ is interpolated closer to the BA samples. Opposite phenomena are observed in the BC samples. As for the unbiased oracle model, no matter how the BC ratio γ changes, such mean distance is almost unchanged and shows a lower variance. This coincides with the observation in Figure 2.

Motivated by mimicking the oracle, we adopt the conditional interpolated prototypes and construct a customized χ-structured metric learning task. Assuming γ is large, we use $\mathbf{p}_\gamma$ and $\mathbf{p}_{1-\gamma}$ to denote prototypes in bias bags $\mathcal{B}$ with high and low BC ratios. The model is required to be more concerned with pulling the majority of low BC ratio bias bag $\mathcal{B}_{1-\gamma}$ closer to $\mathbf{p}_\gamma$, which interpolated into the high BC space. Similarly, the high BC bias bag $\mathcal{B}_\gamma$ should be pulled to low BC interpolated $\mathbf{p}_{1-\gamma}$. We optimize the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ to enable the pulling operation. Concretely, the posterior via the distance $\mathrm{d}\left(\cdot,\cdot\right)$ in the representation space is formalized as:

$$\Pr\left(y_i \mid \mathbf{x}_i\right) = \frac{\exp\left(-\mathrm{d}\left(f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right), \mathbf{p}_{\gamma,y_i}\right)/\tau\right)}{\sum_{c\in[C]} \exp\left(-\mathrm{d}\left(f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right), \mathbf{p}_{\gamma,c}\right)/\tau\right)} \;, \tag{6}$$

Table 2: The classification performance on unbiased test set (in %; higher is better) evaluated on unbiased test sets of Colored MNIST and Corrupted CIFAR-10 with training on varying BA samples ratios. We denote bias pre-provided type by ○ (without any information), ◑ (bias prior knowledge), and ● (explicit bias supervision). The best result is in bold, while the second-best is with underlines.

Dataset		**Colored MNIST**				**Corrupted CIFAR-10**			
Ratio (%)		99.9	99.5	99.0	95.0	99.9	99.5	99.0	95.0
Vanilla	○	28.58	59.29	74.42	87.13	26.91	30.16	37.71	41.60
$+\,\mathbf{p}$ (Snell et al., 2017)	○	31.01	64.82	76.84	87.86	26.55	29.48	38.07	42.30
RUBi (Cadène et al., 2019)	◑	27.82	70.80	86.58	96.77	33.70	34.70	34.59	47.23
ReBias (Bahng et al., 2020)	◑	27.71	72.89	85.95	96.87	33.65	34.40	35.82	47.45
End (Tartaglione et al., 2021)	●	28.19	81.81	88.10	96.99	31.30	33.83	34.02	38.77
DI (Wang et al., 2020)	●	33.18	80.63	86.28	**98.36**	32.09	33.37	37.65	51.27
LfF (Nam et al., 2020)	○	30.24	68.90	76.69	96.81	29.89	33.68	35.28	45.38
LFA (Lee et al., 2021)	○	22.31	64.13	81.83	95.45	32.49	35.74	39.63	47.25
χ-pattern $+\,\mathbf{p}$	●	60.33	64.15	**93.53**	98.30	35.33	**39.31**	**41.32**	**53.37**
χ^2 (Ours)	○	**66.91**	**88.73**	92.15	97.87	**35.67**	37.61	40.74	49.04

where τ is a scaled temperature. One of the branches of the χ-structure classification task is optimizing the $\mathcal{L}_{\text{CE}}$ between samples in the $\mathcal{B}_{1-\gamma}$ and $\mathbf{p}_\gamma$. Similarly, the other branch is optimizing between $\mathcal{B}_\gamma$ and $\mathbf{p}_{1-\gamma}$ at the same time. As shown in Figure 3, such a high-and-low correspondence captures and compacts the intra-class "hollow". In summary, The bias bags of high BC ratios $\mathcal{B}_\gamma$ with corresponding low BC interpolated prototypes conditioning on the target attribute $\mathbf{p}_{1-\gamma}$, and $\mathcal{B}_{1-\gamma}$ with $\mathbf{p}_\gamma$ form the χ-structure crossover objective.

4 Experiments

We conduct experiments to verify whether χ^2-model has effective debiasing capability. We begin by introducing bias details in each dataset (as in subsection 4.1). We present the comparison approaches and training details. In subsection 4.2, the experiments show that χ^2-model achieves superior performance in each stage. Furthermore, we experimentally exemplify the inherent quality of the prototype-based classification for the debiasing task and offer the ablation studies in subsection 4.3.

4.1 Experimental Setups

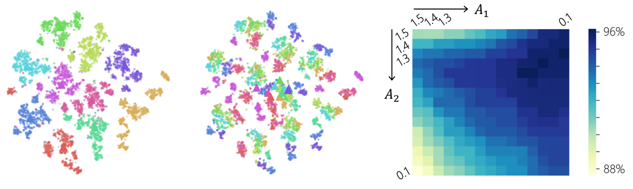
Datasets. To cover more general and challenging cases of bias impact, we validate χ^2-model in a variety of datasets, including two synthetic bias datasets (Colored MNIST (Bahng et al., 2020), Corrupted CIFAR-10 (Nam et al., 2020)) and two real-world datasets (Biased CelebA (Liu et al., 2015) and Biased NICO).

The BA samples ratio ρ in the training set is usually high (over 95%), so the bias attribute is highly correlated with the target label. For example in the Colored MNIST dataset, each digit is associated with one of the pre-defined bias *colors*. Similarly, there is an *object* target with *corruption* bias in Corrupted CIFAR-10 and a *gender* target with *hair color* bias in Biased CelebA. Following the previous works (Hong & Yang, 2021), we use the BA ratio $\rho \in \{95.0\%, 99.0\%, 99.5\%, 99.9\%\}$ for Colored MNIST and Corrupted CIFAR-10, respectively, and approximately 96% for Biased CelebA. The Biased NICO dataset is dedicatedly sampled in NICO (He et al., 2021), initially designed for OOD (Out-of-Distribution) image classification. NICO is enriched with variations in the *object* and *context* dimensions. We select the bias attribute with the highest co-occurrence frequency to the target one, *e.g.*, *helicopter* to *sunset* in training set correlates strongly (see BA samples in Figure 3). The correlation ratio is roughly controlled to 86%. For more details please see the supplementary material.

Table 3: The classification performance on the unbiased CelebA and NICO test set. The data source *BA* denotes the measurement on BA samples and *BC* is corresponding the BC samples.

Data	**Biased CelebA**			**NICO**
Source	BA	BC	All	All
LfF	73.69	**70.41**	72.05	34.44
DFA	94.01	58.98	76.50	33.10
χ^2-model	**97.66**	60.79	**79.23**	**36.99**

Baselines. We carefully select the classic and the latest trending approaches as baselines: (1) Vanilla model training with cross entropy as described in subsection 2.1. (2) Bias-tailored approaches with pre-provided bias type: RUBi, Rebias. (3) Explicit approaches under the guidance of total bias supervision: EnD and DI. (4) Implicit methods through general bias properties: LfF and DFA.



(a) t-SNE on a_t (b) t-SNE on a_b (c) ablation study of A_1, A_2

Figure 6: **(a)** and **(b)** show the t-SNE visualization of our unbiased representation in terms of *digit* (the target attribute) and *color* (the bias one) in Colored MNIST, respectively. **(c)** displays the top-300 mean accuracy of mining BC samples on Colored MNIST with 99.5% BA ratio when A_1 and A_2 are changed.

Implementation details. Following the existing popular benchmarks (Hong & Yang, 2021; Kim et al., 2021), we use the four-layer CNN with kernel size 7×7 for the Colored MNIST dataset and ResNet-18 (He et al., 2016) for Corrupted CIFAR-10, Biased CelebA, and Biased NICO datasets. For a fair comparison, we re-implemented the baselines with the same configuration. We mainly focus on unbiased test accuracy for all categories. All models are trained on an NVIDIA RTX 3090 GPU. More details are in the supplementary material.

Baselines for the first stage. To better demonstrate the effectiveness of χ-pattern, we consider related sample-specific scoring methods (Pleiss et al., 2020; Zhao et al., 2021) and report average precision, top-threshold accuracy, and the minimum samples (threshold) required for 98% accuracy. For more results, such as PR curves, please see the supplementary material.

4.2 Quantitative Evaluation

Performance of χ-shape pattern. As shown in Table 4, our χ-pattern matching achieves state-of-the-art performance on various evaluation metrics. Thus, the χ-structure metric learning objective can leverage more IASs cues to interpolate bias attribute and further learn the debiased representation.

Table 4: The performance of BC samples mining on Colored MNIST with 99.5% BA ratio. *Acc.* denotes mean accuracy of ranking with top-300. 98%-σ denotes the number of samples required to contain 98% of BC samples. AP is average precision. ↑ means higher is better, while ↓ is the opposite.

Measure	**Acc.** ↑	**98%-σ** ↓	**AP** ↑
Entropy(Joshi et al., 2009)	78.33	632	83.52
Confidence(Li & Sethi, 2006)	80.33	590	85.61
Loss(Nam et al., 2020)	94.39	418	98.22
Pleiss et al. (2020)	82.67	686	89.24
Zhao et al. (2021)	90.33	451	96.04
χ-pattern	**95.84**	**372**	**98.44**

χ^2-model in different types of bias constructions. **(1)** Synthetic bias on Colored MNIST and Corrupted CIFAR-10: From Table 2 we find that under extreme bias influence, as ρ is 99.9%, the performance of the vanilla model and other baselines decreases catastrophically. In contrast, Our χ^2-model maintains the robust and efficient debiasing capability on the unbiased dataset. Further, more results in Figure 2 present the remarkable performance of our χ^2-model compared to other methods. **(2)** Real-world bias on Biased CelebA and Biased NICO: Table 3 shows that compared to the recent methods which do not pre-provide any bias information in advance as the same as ours, our method also achieves a remarkable performance. The above experiments indicate that conditional interpolation among IASs feedback the shift of the intrinsic knowledge and facilitate learning debiased representations even in extremely biased conditions.

4.3 Further analysis

The inherent debiasing capability of prototype-based classification. We directly construct the prototype by averaging the trained representations of the vanilla model (as in Table 2 line two named "$+$ **p**"). The results show that on some datasets like Colored MNIST, the prototype-based classifier without training achieves performance improvement.

Visualize the test set representation on 2D embedding space via t-SNE. Figure 6 shows the 2D projection of the feature extracted by χ^2-model on Colored MNIST. We color the target and bias attributes separately. The representations follow the target attribute to cluster into classes which indicates that our model learns the debiased representations.

Ablation studies. We further perform the ablation analysis of the matching factors A_1, A_2 in Eq. **??**, which directly determine the χ-shape curves. The results show the first stage of χ^2-model is robust to changes in hyperparameters. For more related experiments like on different BC identification thresholds, please see the supplementary material.

5 Conclusion

Although intra-class biased samples with a "hollow" structure impede learning debiased representations, we propose the χ^2-model to leverage *Intermediate* Attribute Samples (IASs) to capture how the samples with intrinsic attribute shift. χ^2-model works in a two-stage manner, matching and ranking possible IASs based on their χ-shape training dynamics followed by a χ-branch metric-based debiasing objective with conditional attribute interpolation.

References

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal VQA: revealing and reducing spurious correlations by invariant and covariant semantic editing. In *CVPR*, pp. 9687–9695, 2020.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, volume 70, pp. 233–242, 2017.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, pp. 528–539, 2020.

Yoshua Bengio, Yann LeCun, and Geoffrey E. Hinton. Deep learning for AI. *Communications of the ACM*, 64(7):58–65, 2021.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*, 2019.

Rémi Cadène, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, pp. 839–850, 2019.

Rocío Cañamares and Pablo Castells. Should I follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems. In *SIGIR*, pp. 415–424, 2018.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. Fairfil: Contrastive neural debiasing method for pretrained text encoders. In *ICLR*, 2021.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP-IJCNLP*, pp. 4067–4080, 2019.

Luke Nicholas Darlow, Stanislaw Jastrzebski, and Amos J. Storkey. Latent adversarial debiasing: Mitigating collider bias in deep neural networks. *CoRR*, abs/2011.11486, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.

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 *ICLR*, 2019.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. In *ICLR*, 2021.

Yue Guo, Yi Yang, and Ahmed Abbasi. Auto-debias: Debiasing masked language models with automated biased prompts. In *ACL*, pp. 1012–1023, 2022.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In *EMNLP-IJCNLP Workshop*, pp. 132–142, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

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

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, pp. 793–811, 2018.

Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *NeurIPS*, pp. 26449–26461, 2021.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pp. 124–140, 2020.

Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pp. 2372–2379, 2009.

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, pp. 158–171, 2012.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, pp. 9012–9020, 2019.

Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *ICCV*, pp. 14972–14981, 2021.

Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. *CoRR*, abs/2206.10843, 2022.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *CoRR*, abs/2204.02937, 2022.

S.W.-C. Lam. Texture feature extraction using gray level gradient based co-occurence matrices. In *IEEE*, volume 1, pp. 267–271 vol.1, 1996.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *NeurIPS*, 2021.

Mingkun Li and Ishwar K. Sethi. Confidence-based active learning. *IEEE TPAMI*, 28(8):1251–1261, 2006.

Yi Li and Nuno Vasconcelos. REPAIR: removing representation bias by dataset resampling. In *CVPR*, pp. 9572–9581, 2019.

Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan L. Yuille, and Cihang Xie. Shape-texture debiased neural network training. In *ICLR*, 2021.

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 *ICML*, volume 139, pp. 6781–6792, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pp. 3730–3738, 2015.

Michael Mendelson and Yonatan Belinkov. Debiasing methods in natural language understanding make bias more accessible. In *EMNLP*, pp. 1545–1557, 2021.

Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. In *ICML*, pp. 6927–6937, 2020.

Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*, pp. 429–438, 2020.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, pp. 20673–20684, 2020.

Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. In *NeurIPS*, pp. 1962–1974, 2019.

Guillermo Valle Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *ICLR*, 2019.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, pp. 17044–17056, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2020.

Ramprasaath Ramasamy Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry P. Heck, Dhruv Batra, and Devi Parikh. Taking a HINT: leveraging explanations to make vision and language models more grounded. In *ICCV*, pp. 2591–2600, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *NeurIPS*, 2020.

Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *ICLR*, 2022.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pp. 4077–4087, 2017.

Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *CVPR*, pp. 13508–13517, 2021.

Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Pattern Recognition*, volume 9358, pp. 504–516, 2015.

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

Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *ICLR*, 2019.

Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, pp. 8916–8925, 2020.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021.

Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *CVPR*, pp. 5372–5382, 2021a.

Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*, pp. 11–20, 2021b.

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pp. 8792–8802, 2018.

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

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, pp. 2979–2989, 2017.

Wei Zhu, Haitian Zheng, Haofu Liao, Weijian Li, and Jiebo Luo. Learning bias-invariant representation by cross-sample mutual information minimization. In *ICCV*, pp. 14982–14992, 2021.

Appendix

- Appendix A: An example of the color-biased model.
- Appendix B: Additional related work (cf. subsection 2.1).
- Appendix C: Implementation details and hyper-parameter settings (cf. subsection 4.1).
- Appendix D Additional experiments, ablation studies, and robustness analysis.
- Appendix E Overall algorithm.
- Appendix F Discussion about the limitations.

A An example of the color-biased model on *orange* lifeboat

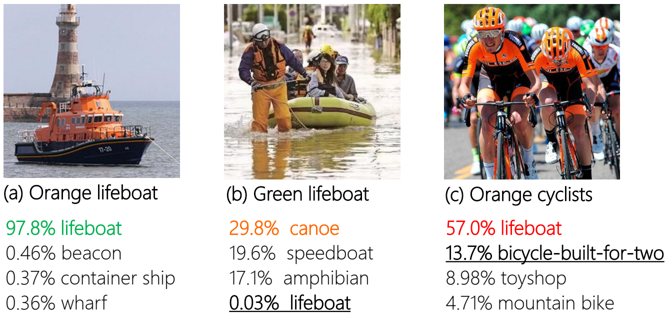

(a) Orange lifeboat (b) Green lifeboat (c) Orange cyclists

Figure 7: Classification of a standard ImageNet pre-trained ResNet-50 of (a) a lifeboat aligned with the bias attribute (colored in orange); (b) a lifeboat conflicting with the bias attribute (colored in green), and (c) cyclists aligned with the bias attribute but conflicting with the target one (wearing orange but not a lifeboat). Most of the lifeboats in the training set are orange, as in (a), and the biased model towards classification by orange leads to errors in (b) and (c).

B Related work from what bias information provided in advance

There are various methods of learning a debiased model from a biased training set.

Debiasing under the guidance of bias supervision. This thread of methods introduces full explicit bias attribute supervision and an additional branch of the model to predict the label of the bias. Kim et al. (2019) leverage bias clues to minimize the mutual information between the representation and the bias attributes with gradient reversal layers (Ganin et al., 2016). Similarly, Li & Vasconcelos (2019) perform RGB vector as *color* side information to conduct the minimax bias mitigation. (Clark et al., 2019; Wang et al., 2020) utilize the auxiliary bias instruction to train the relevant independent models and ensemble their predictions. (Sagawa et al., 2020; Goel et al., 2021) balance the performance of bias subgroups over distribution shift. (Tartaglione et al., 2021; Cheng et al., 2021) directly regularize the bias attribute to disentangle the confused bias representations.

Debiasing with bias prior knowledge. Many real-world applications limit access to sufficient bias supervision. However, a relaxed condition could be met to provide prior knowledge of the bias (*e.g.*, bias type). Many methods highlight that the *content* bias type plays an important role in CNN object recognition (Hendricks et al., 2018; Geirhos et al., 2019; Li et al., 2021). Based on such observations, several approaches adopt the bias type to build a bias-capturing module. Wang et al. (2019) remove *texture* bias through latent space projection with gray-level co-occurrence matrix (Lam, 1996). Bahng et al. (2020) encourage the debiased model to learn independent representation from a designed biased one. Other approaches mitigate the dataset bias existing in natural language processing with logits re-weighting (Cadène et al., 2019).

Debiasing through general intrinsic bias properties. Towards more practical applications, this line of methods takes full advantage of the bias property, which does not require either explicit bias supervision or pre-defined bias prior knowledge. Nam et al. (2020) make a comprehensive analysis on the properties of bias. The observations indicate a two-branch training strategy — a biased model trained with Generalized Cross-Entropy loss (Zhang & Sabuncu, 2018) amplifying its "prejudice" on BA samples, and a debiased model focuses more on samples that go against the prejudice of the biased one. Similarly, Lee et al. (2021) fit one of the encoders to the bias attribute and randomly swap the latent features to work as augmented BC samples. Other approaches also consider the model learning shortcuts revealed by the high gradients of latent vectors (Darlow et al., 2020; Huang et al., 2020).

C Implementation Details

C.1 Training dynamic visualization of Figure 2

To visualize the 2D attribute boundary, we first add an extra linear projection layer $\mathbf{w}_{proj} \in \mathbb{R}^{d\times 2}$ behind the feature extraction network and correspondingly modify the top-layer classifier $\mathbf{w}_c$ to classify on 2D features. After training is completed, we directly present the 2D features of the data and the top-layer classifier in Figure 2. Secondly, to compare different attributes and feedback on their gradients fairly, we jointly train the attribute classifier with a shared feature extraction network. This ensures their features are consistent and comparable to the classifiers with different attributes. Figure 2 shows the results of the above model trained on Colored MNIST with a BA ratio of 0.95, where the learning rate is 0.00001. The two digit (shape) classes in the figure are 2 and 8. Correspondingly, the two color classes are purple and green. The samples 2 in purple and the 8 in green are BA ones. In contrast, the samples 2 in green and the 8 in purple are BC ones. The ratio of BA and BC samples is roughly 0.95.

C.2 Datasets

Colored MNIST. Following most of the previous work (Nam et al., 2020; Hong & Yang, 2021; Lee et al., 2021), we construct the Colored MNIST by coloring each digit and keeping the background black, in other words, every target attribute *digit* in the Colored MNIST is highly correlated with a specific bias attribute *color*. The degree of severity we chose to calibrate the dataset bias difficulty was 1 as in previous works. The different bias-aligned (BA) ratios contains different BA samples, *e.g.*, in the ratio of 99.9% we have 59940 BA samples and 60 bias-conflicting (BC) samples in the training set. Similarly, the ratio of 99.5% has $\{59940, 60\}$ BA and BC samples, correspondingly. In the same way, for other ratios of BA and BC samples, the ratio of 99.0% is $\{58,402; 598\}$ and the ratio of 95.0% is $\{57,000; 3000\}$.

Corrupted CIFAR-10. For the Corrupted CIFAR dataset, we follow the earlier work (Lee et al., 2021) and choose 10 corruption types, *i.e.*, {*Snow*, *Frost*, *Fog*, *Brightness*, *Contrast*, *Spatter*, *Elastic*, *JPEG*, *Pixelate*, *Saturate*}. The *corruption* type is highly correlated with the target ones as Plane, Car, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck. Similarly, we choose severity 1 in the original paper (Nam et al., 2020). The number of BA samples and BC samples for each ratio of BA ones are: 99.9%-$\{49,950; 50\}$, 99.5%-$\{49,750; 250\}$, 99.0%-$\{49,500; 500\}$, 95.0%-$\{47500; 2,500\}$.

Biased CelebA. Following the experimental configuration of previous works, We intentionally truncated a portion of the CelebA dataset so that each target attribute *containing BlondHair or not* was skewed towards the bias attribute of *Male*. The number of target bias, *i.e.*, *BlondHair-Male* is as follows: BC samples like BlondHair equals 0 with Male equals 0 contains $1,558$ and $\{1\text{-}1 : 1,098\}$. The BA samples is $\{1\text{-}0 : 18,279\}$ and $\{0\text{-}1 : 53,577\}$.

Biased NICO. The Biased NICO dataset is dedicatedly sampled in NICO (He et al., 2021), which is originally designed for Non-I.I.D. or OOD (Out-of-Distribution) image classification. NICO is enriched with variations in the *object* and *context* dimensions. Concretely, there are two superclasses: *Animal* and *Vehicle*: with 10 classes as Bear, Bird, Cat, Cow, Dog, Elephant, Horse, Monkey, Rat and Sheep for Animal, and 9 classes as Airplane, Bicycle, Boat, Bus, Car, Helicopter, Motorcycle, Train and Truck for Vehicle. Each object class has 9 or 10 contexts. We select the bias attribute with the highest co-occurrence frequency with the target one, *i.e.*, Dog *on snow*, Bird *on grass*, Cat *eating*, Boat *on beach*, Bear *in forest*, Helicopter *in sunset*, Bus *in city*, Cow *lying*, Elephant *in river*, Motorcycle *in street*, Monkey *in water*, Truck *on road*, Rat *at home*, Bicycle *with people*, Airplane *aside mountain*, Sheep *walking*, Horse

Table 5: The number of each class in the Biased NICO training set and the bias aligned (BA) samples. We take the most occurring bias attribute in each class as the attribute of the BA samples. The correlation ratio over all classes is roughly controlled to 86.27%.

Animal	**Size**	**BA**	**Ratio** (%)	***Vehicle***	**Size**	**BA**	**Ratio** (%)
BEAR	274	247	90.15	AIRPLANE	102	75	73.53
BIRD	272	245	90.07	BICYCLE	203	176	86.70
CAT	310	283	91.29	BOAT	195	168	86.15
COW	190	163	85.79	BUS	225	201	89.33
DOG	274	247	90.15	CAR	116	89	76.72
ELEPHANT	203	176	86.70	HELICOPTER	190	163	85.79
HORSE	172	145	84.30	MOTORCYCLE	202	175	86.63
MONKEY	143	116	81.12	TRAIN	182	158	86.81
RAT	154	127	82.47	TRUCK	177	150	84.75
SHEEP	108	81	75.00				

Table 6: The number of each bias attribute in the Biased NICO training set and the bias aligned (BA) samples. These bias attributes are the most frequent in each class. In addition, a few other bias attributes appear in rare numbers, but are balanced with remaining ones in the test set.

Bias Attribute	**Size**	**BA**	**Ratio** (%)
on snow	292	247	84.59
on grass	284	245	86.27
eating	304	283	93.09
on beach	198	168	84.85
in forest	274	247	90.15
in sunset	181	163	90.06
in city	219	201	91.78
lying	181	163	90.06
in river	188	176	93.62
in street	190	175	92.11
in water	134	116	86.57
on road	162	150	92.59
at home	139	127	91.37
with people	191	176	92.15
aside mountain	84	75	89.29
walking	87	81	93.10
running	151	145	96.03
on track	92	89	96.74
at station	161	158	98.14

running, CAR *on track*, TRAIN *at station*. The quantitative details of each class are shown in Table 5. Similarly, The details divided by bias attribute are shown in Table 6, The remaining bias attributes that do not appear in the BA samples are: {*at wharf, at airport, aside traffic light, eating grass, white, in cage, in hole, in garage, cross bridge, at park, yacht, flying, aside tree, black, standing, sitting, at night, double decker, on sea, around cloud, with pilot, in sunrise, in hand, on booth, aside people, at sunset, brown, on shoulder, spotted, subway, in race, climbing, cross tunnel, velodrome, on bridge, shared, at yard, in circus, on ground, on tree, at heliport, taking off, on branch, wooden, sailboat, in zoo*}, which are few in number, about 4 of each. In the test set they are balanced with the remaining bias attributes. The training set's total correlation ratio is roughly 86.27%.

C.3 DATA PRE-PROCESSING

The image sizes of Colored MNIST and Corrupted CIFAR-10 are 28×28 and 32×32, respectively. We feed the original images into the model and do not use data augmentation transformations during training and testing. We directly normalize the data from Colored MNIST and Corrupted CIFAR-10 by the mean of $(0.5, 0.5, 0.5)$ and the standard deviation of $(0.5, 0.5, 0.5)$. In the real-world datasets, for the training phase of Biased CelebA, we first resize the images to a size of 224×224, and then apply the `RandomHorizontalFlip` transformation. As for the Biased NICO dataset, following

Table 7: Convolutional neural network for Colored MNIST dataset. The kernel is written in the form of $H \times W \times C$. BN indicates whether the batch normalization layer is applied.

Layer	Kernel	Padding	BN	Activation
Conv	$7 \times 7 \times 16$	3	$\surd$	ReLU
Conv	$7 \times 7 \times 32$	3	$\surd$	ReLU
Conv	$7 \times 7 \times 64$	3	$\surd$	ReLU
Conv	$7 \times 7 \times 128$	3	$\surd$	ReLU
AvgPool	1×1	–	–	–
Norm	–	–	–	–

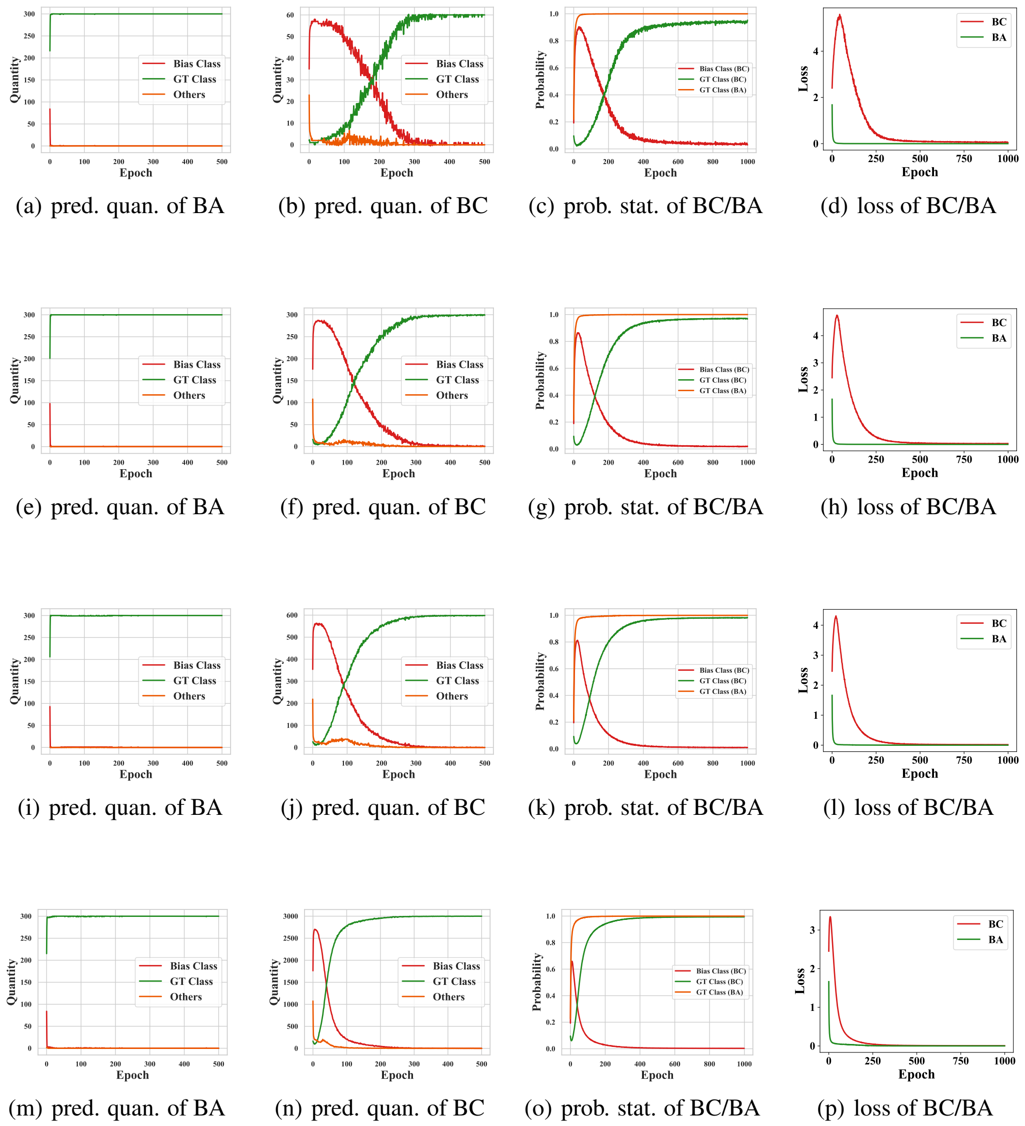
most of the previous works (Zhang et al., 2021a), we append the `RandomHorizontalFlip`, `ColorJitter`, `RandomGrayscale` transformations after the `RandomResizedCrop` to 224×224. For both of them, during the test, we only resize the images. We normalize these real-world datasets by the mean of $(0.485, 0.456, 0.406)$ and the standard deviation of $(0.229, 0.224, 0.225)$.

C.4 Training Details

Our code is based on the `PyTorch` library. Following the previous work (Hong & Yang, 2021), We use the four-layer convolutional neural network with kernel size 7×7 for the Colored MNIST dataset and ResNet-18 (He et al., 2016) for Corrupted CIFAR-10, Biased CelebA, Biased NICO datasets. For all methods and datasets, we do not consider loading any additional pretrained weights to allow the models represent the pure debiasing capability. In the training phase, we use Adam optimizer and cosine annealing learning rate scheduler. For all datasets, the batch size is selected from $\{64, 128, 256\}$. Correspondingly, the learning rate is from $\{0.0001, 0.0005, 0.001, 0.005\}$, and the smaller ones are used for training the *vanilla* model. For all methods, including the reproduced comparison ones, we train the model for 200 epochs on Colored MNIST, Corrupted CIFAR-10, while training 50 and 100 epochs on Biased CelebA and Biased NICO, respectively.

χ^2-model.

- For the first stage: We train the 1000 epochs vanilla model with a learning rate 1e-5 on Colored MNIST, 5e-3 on Corrupted CIFAR-10, 5e-5 on Biased CelebA and 1e-3 on Biased NICO to extract the training dynamics. In practice, we design the *Area Under Score* (AUS) strategy to capture the training dynamics. All comparison methods leverage epoch-specific scores, and AUS applies to these methods, *e.g.*, *Loss* is the calculated all the epoch-level loss summations. We generally use the ratio of divided BC samples as a hyperparameter. We find that a slightly larger BC ratio brings better results in our experiments, as detailed in Table 8. In addition, for the IASs importance verification experiments in Table 1, the "step-wise" setting indicates we apply uniformly higher and lower sampling weights to BC and BA samples. The unified weights are related to the BA ratio ρ in the whole dataset, *i.e.*, the weight on BC samples is ρ and on the BA ones is $1 - \rho$.
- In the second stage: χ-branch metric learning objective, we first construct the data pools $\mathcal{D}_{\parallel}$ and $\mathcal{D}_{\perp}$ with the ranking. The BC identification threshold to split those two data pools can be adjusted to a suitable value without knowing the ground-truth dataset BC ratio. To observe the IASs validity and unify the style, we report the results one level higher in $\{0.999, 0.995, 0.99, 0.95\}$ than the dataset BC ratio in the main text, *i.e.*, the threshold is 0.99 if the dataset BC ratio is 0.995. See more details in subsection D.4 and Table 14. We construct different ratios of bias bags $\{\mathcal{B}_{\gamma}, \mathcal{B}_{1-\gamma}\}$ and mixed prototypes $\{\mathbf{p}_{\gamma}, \mathbf{p}_{1-\gamma}\}$ by bootstrapped sampling a batch containing almost the same number of BA and BC samples using the first stage χ-pattern score (described in subsection 3.3). The more numerous part is the one that contains all the samples in that part of the batch, *e.g.*, for a large γ with a majority of the BC part, $\mathcal{B}_{\gamma}$ contains all the BC samples in the above batch. In this case, the remaining $1 - \gamma$ ratio of BA samples are sampled uniformly in the batch. The mixed prototype $\mathbf{p}_{\gamma}$ and $\mathbf{p}_{1-\gamma}$ are extracted and constructed similarly. The mixed ratios γ are from $\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. As described in the paper, the computation of $\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}_{\gamma}, \mathcal{B}_{1-\gamma}\right)$ and $\mathcal{L}_{\mathrm{CE}}\left(\mathbf{p}_{1-\gamma}, \mathcal{B}_{\gamma}\right)$ will yield different ratios of mixed prototypes interacting with BA or BC sample. In this process, we set the temperature τ of the mixed prototypes metric-based prediction (as in Eq. 8) from $\{0.01, 0.05, 0.1\}$. Our model's average training time with NVIDIA RTX 3090 GPU is about 1.8x faster than that of LfF (Nam et al., 2020).



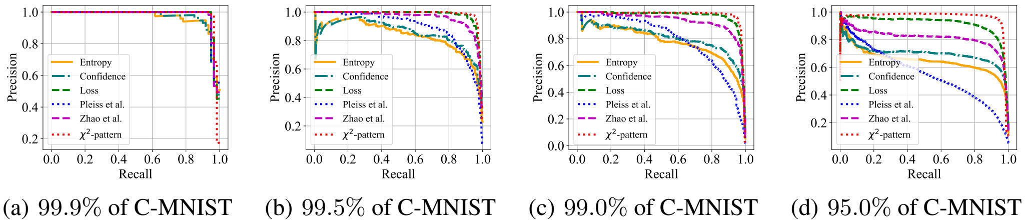
(a) pred. quan. of BA (b) pred. quan. of BC (c) prob. stat. of BC/BA (d) loss of BC/BA

(e) pred. quan. of BA (f) pred. quan. of BC (g) prob. stat. of BC/BA (h) loss of BC/BA

(i) pred. quan. of BA (j) pred. quan. of BC (k) prob. stat. of BC/BA (l) loss of BC/BA

(m) pred. quan. of BA (n) pred. quan. of BC (o) prob. stat. of BC/BA (p) loss of BC/BA

Figure 8: The more observation in the first χ-pattern stage on Colored MNIST. In these figures the general bias properties is represented over the whole dataset from a statistical perspective. The figures in the left two columns indicate that as the training epoch increases, the model prediction quantity of the BC samples and BA samples on the *GT-class*, *Bias class* or *Others* changes. The third column figures represent training dynamics of the predicted probability on BC and BA samples in different classes. The last column figures denote change of the loss BC and BA samples during training.



(a) 99.9% of C-MNIST (b) 99.5% of C-MNIST (c) 99.0% of C-MNIST (d) 95.0% of C-MNIST

Figure 9: The Precision-Recall curves of the BC samples identification on Colored MNIST dataset (as above C-MNIST) over various ratios. Best view in colors.

Table 8: Ablation study of χ-branch metric learning objective. We removed different branches of the task and reported unbiased accuracy of the Colored MNIST dataset with varying ratios of BA samples. The BC ratio γ is relatively high.

‘

Dataset	**Colored MNIST**			
Ratio (%)	99.9	99.5	99.0	95.0
χ^2-model	**66.91**	**88.73**	**92.15**	<u>97.87</u>
$-\mathcal{L}_{\text{CE}}\left(\mathbf{p}_{\gamma}, \mathcal{B}_{1-\gamma}\right)$	<u>61.99</u>	85.84	90.23	**97.94**
$-\mathcal{L}_{\text{CE}}\left(\mathbf{p}_{1-\gamma}, \mathcal{B}_{\gamma}\right)$	57.26	<u>86.59</u>	<u>92.14</u>	97.33

D Additional experiments

D.1 More observations and results in the first stage

In the main text, we have shown the change of posterior over the GT-class and the bias one in Figure 4 with four typical samples of BC samples, intermediate attribute samples, and BA samples. Here we show more observations on the whole training set in a statistical significance.

- As shown in Figure 8, the vertical axis of the left two columns figures is the quantity and the horizontal axis is the epoch of model training. Each point on the curve represents how many samples are predicted as *GT-class*, *Bias class* or *Others* by the current epoch model. The first column figures represent the prediction on BA samples, while the second column represents the prediction on BC ones. It can be found that for BC samples, even at the dataset level, the vanilla model always predicts them as *Bias class* first. It is consistent with our observation in the original paper, in fact, this is another interpretation of the right half of Figure 4 in the paper.
- The right two columns of Figure 8 also represent more statistical information at the dataset level, *e.g.*, the third column shows the χ-shaped prediction of BC samples over the whole dataset as training epoch increases. This corresponds to the left half of Figure Figure 4 in the paper. The last column figures shows the change of the loss. It can be found that the loss on the BC sample corresponds to the lower branch of the χ-shaped curve in the paper.

Further, we show more BA sample identification results of the first stage over various ratios. In Table 9, we display their top-ratio accuracy, *e.g.*, taking the top ranking with the number of BC samples in the full training set to calculate how many ground truth BC samples they contain. In addition, we also present the average precision in Table 10. Moreover, we plot the PR curves of various methods in the first stage on Colored MNIST and Biased NICO datasets in Figure 9 and Figure 10. The results show that our method maintains excellent performance.

D.2 Results with error bars

We run our methods and the comparison methods like vanilla and Learning from Failure (LfF) multiple times and report error bars. We present the full results with both 95% confidence interval as Table 13 and the standard deviation in Figure 11.

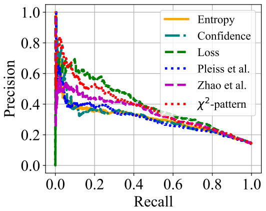


Figure 10: The Precision-Recall curves of the BC samples identification on Biased NICO dataset. Best view in colors.

D.3 Ablation study of χ-branch metric learning objective

In order to verify whether the effectiveness of our method is indeed derived from our χ-branch metric learning objective. We first remove one of the mixed prototypes and bias bag losses as "$-\mathcal{L}_{\text{CE}}\left(\mathbf{p}_{\gamma}, \mathcal{B}_{1-\gamma}\right)$" in Table 8. This substantially lose the metric-based *push* relationship between the BA samples and the high BC ratio prototypes $\mathbf{p}_{\gamma}$. Next, we also drop another branch of the prototypes training, *i.e.*, attenuate the effect of most BC samples on a low ratio of mixed prototypes

Table 9: The mean accuracy of BC samples identification on the Colored MNIST and NICO dataset. We display top-BC-ratio accuracy, *e.g.*, calculating the proportion of the mined top-ranked samples in the total BC ones.

Dataset	**Colored MNIST**				**NICO**
Ratio (%)	99.9	99.5	99.0	95.0	86.27
Entropy	91.66	78.33	72.57	63.43	34.31
Confidence	91.66	80.33	76.25	68.90	34.91
Loss	95.00	94.00	92.80	87.43	**41.81**
Pleiss et al. (2020)	95.00	82.67	72.24	53.70	33.53
Zhao et al. (2021)	95.00	90.33	88.12	79.66	39.84
χ-pattern	**96.66**	**95.67**	**95.48**	**94.30**	40.00

Table 10: The average precision (AP) of BC samples identification on the Colored MNIST and NICO dataset.

Dataset	**Colored MNIST**				**NICO**
Ratio (%)	99.9	99.5	99.0	95.0	86.27
Entropy	96.49	83.52	77.61	64.94	30.83
Confidence	96.68	85.61	80.73	69.76	31.00
Loss	98.06	98.22	97.03	91.55	**39.64**
Pleiss et al. (2020)	97.97	89.24	79.15	55.48	30.48
Zhao et al. (2021)	98.03	96.04	93.27	81.29	34.27
χ-pattern	**98.07**	**98.44**	**97.77**	**95.96**	37.79

$\mathbf{p}_{1-\gamma}$. This reduces the debiasing capability using the general properties of the Figure 5 in original paper. The results show that our method with χ-branch objective is significantly better than the single branch at 99.9%, 99.5% and 99.0%. It achieves the same superior level at 95.0%. Especially in the extreme environment, *i.e.*, when the BC samples are rare, the χ-branch can further improve the model performance and overcome the debiasing problem comprehensively.

D.4 Robustness of the χ^2-model with varying BC identification thresholds

For χ^2-model, we use the BC identification thresholds to split $\mathcal{D}_{\|}$ and $\mathcal{D}_{\perp}$. We show the influence of different thresholds in the Table 14, where the vertical axis represents the ground-truth ratio of BA samples included in the dataset. The horizontal axis represents the ratio of BA samples used as hyperparameters in the χ-model. From this result, we can find that the model is less affected by the thresholds. Furthermore, since the *Bias Bag* $\{\mathcal{B}_{\gamma}, \mathcal{B}_{1-\gamma}\}$ is constructed taking into account the presence of IASs. Based on bootstrapped sampling, the BC identification threshold learning is already embedded in the first stage χ-pattern scores.

E Overall algorithm

In Algorithm 1, we show the entire pseudo-code of this work.

F Discussion about the limitations

In this paper, we adopt a new two-stage χ^2-model. However, the first stage still requires training the long-epoch vanilla model as a weaker bias-capture mechanism. When two attributes have an equal learning difficulty and jointly determine the target label, our method may encounter difficulties.

Table 11: Ablation studies on the influence of different matching factor A_1 (as in Equation 3) and A_2 (fixed at 1.2) to the top-ranking mean accuracy (in %) on 99.5% BA ratio Colored MNIST dataset.

Dataset	**Colored MNIST**							
Factor A_1	**1.5**	**1.4**	**1.3**	**1.2**	**1.1**	**1.0**	**0.9**	**0.8**
χ-shape performance	93.78	94.10	94.43	94.43	94.73	95.03	95.06	95.06
Factor A_1	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**	
χ-shape performance	95.43	95.43	95.8	95.84	96.18	95.84	95.84	
Average from 1.5 to 0.1	$95.13_{\pm 0.35}$							

Table 12: Ablation studies on the influence of A_1 (fixed at 0.1) and different matching factor A_2 (as in Equation 3) to the top-ranking mean accuracy (in %) on 99.5% BA ratio Colored MNIST dataset.

Dataset	**Colored MNIST**							
Factor A_2	**1.5**	**1.4**	**1.3**	**1.2**	**1.1**	**1.0**	**0.9**	**0.8**
χ-shape performance	95.84	95.84	96.18	95.84	95.84	95.58	95.58	95.55
Factor A_2	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**	
χ-shape performance	95.25	94.95	94.99	94.69	94.69	94.69	94.39	
Average from 1.5 to 0.1	$95.33_{\pm 0.27}$							

Table 13: The classification performance with 95% confidence interval error bars on unbiased test set (in %; higher is better) evaluated on unbiased test sets of Colored MNIST with respect to the random seed after running experiments multiple times. We denote bias pre-provided type by ○ as those without any information. The best result is in bold, while the second-best is with underlines.

‘

Dataset		**Colored MNIST**			
Ratio (%)		99.9	99.5	99.0	95.0
Vanilla	○	$28.94_{\pm 1.33}$	$58.75_{\pm 0.64}$	$71.66_{\pm 2.24}$	$88.91_{\pm 1.72}$
LfF	○	$32.98_{\pm 2.20}$	$69.44_{\pm 3.15}$	$85.78_{\pm 7.32}$	$95.79_{\pm 0.99}$
χ^2-model	○	$\mathbf{68.04}_{\pm 1.22}$	$\mathbf{90.37}_{\pm 1.33}$	$\mathbf{93.21}_{\pm 0.91}$	$\mathbf{98.30}_{\pm 0.35}$

Table 14: The unbiased test set accuracy to verify the robustness of the χ^2-model with varying BC identification thresholds on Colored MNIST. The vertical and horizontal axes indicate the ground-truth ratio of BA samples and the ratio of BA ones fed to the χ^2-model's sampling process, respectively.

Ratio (%)	99.9	99.5	99.0	95.0
99.9	57.82	66.91	67.30	69.16
99.5	82.42	87.02	88.73	90.58
99.0	86.28	89.65	91.39	92.15
95.0	96.49	97.77	97.68	97.87

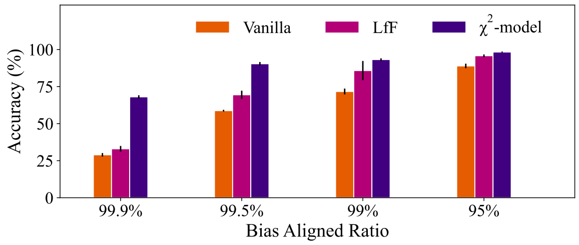
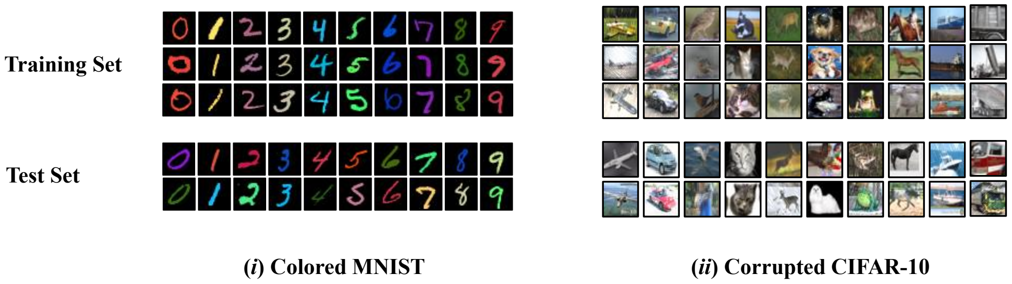


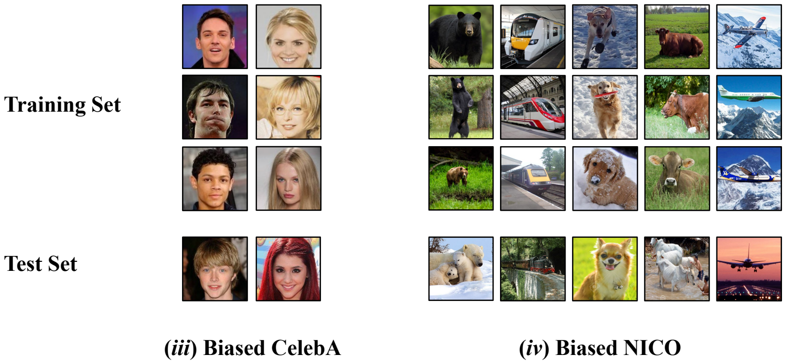
Figure 11: The classification performance with error bars on unbiased Colored MNIST test set. Error bars expressed by the black line denote the standard deviation. Best view in colors.



(*i*) Colored MNIST

(*ii*) Corrupted CIFAR-10

Figure 12: Example samples of Colored MNIST and Corrupted CIFAR-10 datasets.



(*iii*) Biased CelebA

(*iv*) Biased NICO

Figure 13: Example samples of Biased CelebA and Biased NICO datasets.

Algorithm 1 Training for χ^2-model

```
Require: Biased training data $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$.
First stage: $\chi$-shape pattern.
Train a vanilla model $\boldsymbol{\theta}$ on $\mathcal{D}_{\text{train}}$ with cross entropy loss as mentioned in Equation 1:
    $\mathcal{L}_{\text{CE}} = \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{\text{train}}} \left[ -\log \Pr \left( h_{\boldsymbol{\theta}}(\mathbf{x}_i) = y_i \mid \mathbf{x}_i \right) \right] .$
Consider the $T$ epochs change on ground-truth label $y_i$ and bias label $b_i\left(\mathbf{x}_i, h_{\boldsymbol{\theta}}\right)$:
    $\mathcal{L}_{\text{CE}}\left(\mathbf{x}_i\right) = \begin{pmatrix} \mathcal{L}_{\text{CE}}^{gt}\left(\mathbf{x}_i\right) = \left\{ -\log \Pr^t \left( y_i \mid \mathbf{x}_i \right) \right\}_{t=1}^T \\ \mathcal{L}_{\text{CE}}^{b}\left(\mathbf{x}_i\right) = \left\{ -\log \Pr^t \left( b_i\left(\mathbf{x}_i, h_{\boldsymbol{\theta}}\right) \mid \mathbf{x}_i \right) \right\}_{t=1}^T \end{pmatrix} .$
Capture the BC sample with two exponential $\chi$-shape functions:
    $\chi_{\text{shape}} = \begin{pmatrix} \mathrm{p}^{gt} = \left\{ e^{-At} \right\}_{t=1}^T \\ \mathrm{p}^{b} \ = \left\{ e^{At} \right\}_{t=1}^T \end{pmatrix} .$
Compute the ranking score $\mathbf{s}(\mathbf{x}_i)$ with the inner product over two curves as Equation 3:
    $\mathbf{s}(\mathbf{x}_i) = \left\langle \mathcal{L}_{\text{CE}}\left(\mathbf{x}_i\right), \chi_{\text{shape}} \right\rangle = \left\langle \mathcal{L}_{\text{CE}}^{gt}\left(\mathbf{x}_i\right), \mathrm{p}^{gt} \right\rangle + \left\langle \mathcal{L}_{\text{CE}}^{b}\left(\mathbf{x}_i\right), \mathrm{p}^{b} \right\rangle$
    $= \sum_{t=1}^{T} -(e^{-At}) \log \Pr \left( h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) = y_i \mid \mathbf{x}_i \right) - (e^{At}) \log \Pr \left( h_{\boldsymbol{\theta}}\left(\mathbf{x}_i\right) = b_i\left(\mathbf{x}_i, h_{\theta^t}\right) \mid \mathbf{x}_i \right) .$
Second stage: $\chi$-branch metric learning objective.
for each step do
    Construct multiple bias bags $\mathcal{B}_{\boldsymbol{\gamma}}$ with bootstrapping as Equation 4:
        $\mathcal{B}_{\boldsymbol{\gamma}} = \left\{ (\mathbf{x}_i, y_i) \,\middle|\, \text{NUM}\left(\mathcal{D}_{\perp}\right) : \text{NUM}\left(\mathcal{D}_{\parallel}\right) = \boldsymbol{\gamma} \right\} ,$
    where the ratio of BC samples is $\boldsymbol{\gamma}$.
    Build the prototype $\mathbf{p}$ for class $c$ based on $\mathcal{B}_{\boldsymbol{\gamma}}$ as Equation 5:
        $\mathbf{p}_{\boldsymbol{\gamma}, c} = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}_{\boldsymbol{\gamma}}} f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right) \cdot \mathbb{I}\left[ y_i = c \right] .$
    Consider a high $\boldsymbol{\gamma}$:
    for all samples $\mathbf{x}_i \in \mathcal{B}_{1-\boldsymbol{\gamma}}$ do
        Classify with $\mathbf{p}_{\boldsymbol{\gamma}}$ as Equation 6:
            $\Pr\left( y_i \mid \mathbf{x}_i \right) = \frac{\exp\left( -\mathrm{d}\left( f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right), \mathbf{p}_{\boldsymbol{\gamma}, y_i} \right) / \tau \right)}{\sum_{c \in [C]} \exp\left( -\mathrm{d}\left( f_{\boldsymbol{\phi}}\left(\mathbf{x}_i\right), \mathbf{p}_{\boldsymbol{\gamma}, c} \right) / \tau \right)} .$
        Compute $\mathcal{L}_{\text{CE}}\left( \mathbf{p}_{\boldsymbol{\gamma}}, \mathcal{B}_{1-\boldsymbol{\gamma}} \right)$.
    end for
    for all samples $\mathbf{x}_i \in \mathcal{B}_{\boldsymbol{\gamma}}$ do
        Classify with $\mathbf{p}_{1-\boldsymbol{\gamma}}$ as mentioned before.
        Compute $\mathcal{L}_{\text{CE}}\left( \mathbf{p}_{1-\boldsymbol{\gamma}}, \mathcal{B}_{\boldsymbol{\gamma}} \right)$.
    end for
    Compute $\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\text{CE}}\left( \mathbf{p}_{\boldsymbol{\gamma}}, \mathcal{B}_{1-\boldsymbol{\gamma}} \right) + \mathcal{L}_{\text{CE}}\left( \mathbf{p}_{1-\boldsymbol{\gamma}}, \mathcal{B}_{\boldsymbol{\gamma}} \right)$.
    Update $\boldsymbol{\phi}$ with $\nabla_{\boldsymbol{\phi}}$.
end for
```