Learning Debiased Representations via Conditional Attribute Interpolation
Supplemental Material

Yi-Kai Zhang, Qi-Wei Wang, De-Chuan Zhan, Han-Jia Ye
State Key Laboratory for Novel Software Technology, Nanjing University
{zhangyk,wangqiwei,zhandc,yehj}@lamda.nju.edu.cn

Contents

1. Related Work 1

2. Implementation Details 2
   2.1. How to capture and visualize the training dynamic of Figure 2 in the main paper 2
   2.2. Dataset construction 2
   2.3. Pre-processing 3
   2.4. Training details
      2.4.1 χ-shape matching pattern 4
      2.4.2 χ-structured metric learning 4

3. Additional Experiments 5
   3.1. Results with error bars 5
   3.2. More observations and results in the first stage 5
   3.3. Ablation study of χ-structured metric learning 5
   3.4. Robustness of the χ²-model with varying identification thresholds of BC sample 6

4. Overall Algorithm 6

5. Discussion About the Limitations 6

Abstract

From the main paper, 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. Then we rectify the representation with a χ-structured metric learning objective.

In this supplementary material, we present more related work of learning a debiased model from what bias information is provided in advance.

1. Related Work

There are various methods of learning a debiased model from what bias information is provided in advance.

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. [16] leverage bias clues to minimize the mutual information between the representation and the bias attributes with gradient reversal layers [7]. Similarly, Li & Vasconcelos [22] perform RGB vector as color side information to conduct the minimax bias mitigation. [5, 31] utilize the auxiliary bias instruction to train the relevant independent models and ensemble their predictions. [9, 27] balance the performance of bias subgroups over distribution shift. [4, 28] 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 [8, 12, 23]. Based on such observations, several approaches adopt the bias type to build a bias-capturing module. Wang et al. [30] remove texture bias through latent space projection with gray-level co-occurrence matrix [19].

Bahng et al. [2] 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 [1, 3].

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. [25] 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 [33] 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. [20] 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 [6, 14, 17, 18, 24, 35].

In the first stage, we consider the similarity of a sample to the BC one to assist in debiasing. It is agreement in querying high quality data for training. Some methods in Active Learning establish on the notions of uncertainty in classification. They try to find the hard samples heuristically such as selecting by the highest entropy [15] or the lowest confidence [21]. Similarly, the mislabeled sample identification [26] can be modified to mine the special BC samples. Further, some of debiasing methods can be abstracted to the first stage, e.g., matching the loss in the work of Nam et al. [25] and mining with peer-picking of Zhao et al. [34].

2. Implementation Details

2.1. How to capture and visualize the training dynamic of Figure 2 in the main paper

To visualize the 2D attribute boundary, we first add an extra linear projection layer \( w_{proj} \in \mathbb{R}^{d \times 2} \) behind the feature extraction network and correspondingly modify the top-layer classifier \( w_c \) to classify on 2D features. After training is completed, we directly present the 2D features of the data and the top-layer classifier in the Figure 2 of the main paper. 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 bias aligned (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 in purple and the 8 in green are BA samples. In contrast, the samples 2 in green and the 8 in purple are bias conflicting (BC) samples.

2.2. Dataset construction

Colored MNIST. Following most of the previous work [13, 20, 25], 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 [20] 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 main paper [25]. 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 - 1 : 1, 098\}. The BA samples is \{1 - 0 : 18, 279\} and \{0 - 1 : 53, 577\}.

Biased NICO. The Biased NICO dataset is dedicatedly sampled in NICO [11], which is originally designed for Non-IID. 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 running, CAR on track, TRAIN at station. The quantitative details of each class are shown in Table 1. Similarly, The details divided by bias attribute are shown in Table 2. 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%.
we first resize the images to a size of $224 \times 224$. In the real-world datasets like Biased CelebA, we direct normalize the data from Colored MNIST and Corrupted CIFAR-10 by the mean and standard deviation of both datasets, $\mu$ and $\sigma$, respectively. We feed the original images into the model and do not use data augmentation during training and testing. We apply the RandomHorizontalFlip transformation. As for the Biased NICO dataset, following most of the previous works [32], we append the RandomHorizontalFlip, ColorJitter, RandomGrayscale transformations after the RandomResizedCrop to $224 \times 224$. 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 $(0.229, 0.224, 0.225)$.

### 2.4. Training details

Our code is based on the PyTorch library. Following the previous work [13], We use the four-layer convolutional neural network with kernel size $7 \times 7$ for the Colored MNIST dataset and ResNet-18 [10] 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.
Table 4. Ablation study of $\chi$-structured 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 $\gamma$ is relatively high.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio (%)</td>
<td>99.9</td>
<td>99.5</td>
<td>99.0</td>
<td>95.0</td>
</tr>
<tr>
<td>$\chi^2$-model</td>
<td>66.91</td>
<td>88.73</td>
<td>92.15</td>
<td>97.87</td>
</tr>
<tr>
<td>$\gamma$-structured metric learning objective</td>
<td>61.99</td>
<td>85.84</td>
<td>90.23</td>
<td>97.94</td>
</tr>
<tr>
<td>$\gamma$-structured metric learning objective</td>
<td>57.26</td>
<td>86.59</td>
<td>92.14</td>
<td>97.33</td>
</tr>
</tbody>
</table>

Table 5. 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 $\circ$ as those without any information. The best result is in bold, while the second-best is with underlines.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
<th>Colored MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio (%)</td>
<td>99.9</td>
<td>99.5</td>
<td>99.0</td>
<td>95.0</td>
</tr>
<tr>
<td>Vanilla</td>
<td>$\circ$ 28.94$\pm$1.33</td>
<td>58.75$\pm$0.64</td>
<td>71.66$\pm$2.24</td>
<td>88.91$\pm$1.72</td>
</tr>
<tr>
<td>LFF</td>
<td>$\circ$ 32.98$\pm$2.20</td>
<td>69.44$\pm$3.15</td>
<td>85.78$\pm$7.32</td>
<td>95.79$\pm$0.99</td>
</tr>
<tr>
<td>$\chi^2$-model</td>
<td>$\circ$ 68.04$\pm$1.22</td>
<td>90.37$\pm$1.33</td>
<td>93.21$\pm$0.91</td>
<td>98.30$\pm$0.35</td>
</tr>
</tbody>
</table>

2.4.1 $\chi$-shape matching pattern

- As stated in the paper, we design two exponential $\chi$-shape functions in Equation 4 to capture the ideal training dynamics of BC or BA samples for the first stage. Considering the model predictions throughout the training process, we take the *forgetting statistics* [29] per sample to adapt the matching factor $A_1$ and $A_2$ in the Equation 3 of the main paper. We compute the number of incorrect-to-correct or correct-to-incorrect predictions for every sample, denoted as prediction fluctuations as above. A higher prediction fluctuation leads to a higher factor for more likely $\chi$-shape matching and vice versa. The maximum value of the factor primarily influences the exponential function. In the paper, we adopt $A_1$ equals 0.1 and $A_2$ equals 1.2. We analyze the relevant ablation studies in Figure 6(c), Table 6 and Table 7.

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 4.

In addition, for the IASs importance verification experiments in Table 1 of the main paper, the “step-wise” setting indicates we apply uniformly higher and lower sampling weights to BC and BA samples. As described in the paper, the unified weights are related to the BA ratio $\rho$ in the whole dataset, *i.e.*, the weight on BC samples is $\rho$ and on the BA ones is $1 - \rho$.

2.4.2 $\chi$-structured metric learning

- In the second stage, we construct the data pools $D_B$ and $D_A$ 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 3.4 and Table 10.

We then construct different ratios of bias bags $\{B_{\gamma}, B_{1 - \gamma}\}$ and mixed prototypes $\{P_{\gamma}, P_{1 - \gamma}\}$ by bootstrapped sampling a batch containing almost the same number of BA and BC samples using the first stage $\chi$-pattern score (described in subsection 3.3 in the main paper). The more numerous part is the one that contains all the samples in that part of the batch, *e.g.*, for a large $\gamma$ with a majority of the BC part, $B_{\gamma}$ 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 $P_{\gamma}$ and $P_{1 - \gamma}$ are extracted and constructed similarly. The mixed ratios $\gamma$ are from $\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$.

As shown in the paper, the computation of $L_{CE}(P_{\gamma}, B_{1 - \gamma})$ and $L_{CE}(P_{1 - \gamma}, B_{\gamma})$ will yield different ratios of mixed prototypes interacting with BA or BC samples.
Table 6. Ablation studies on the influence of different matching factor $A_1$ (as in Equation 4 in the main paper) and $A_2$ (fixed at 1.2) to the top-ranking mean accuracy (in %) on 99.5% BA ratio Colored MNIST dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Colored MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor $A_1$</td>
<td>1.5 1.4 1.3 1.2 1.1 1.0 0.9 0.8</td>
</tr>
<tr>
<td>$\chi$-shape performance</td>
<td>93.78 94.10 94.43 94.43 94.73 95.03 95.06</td>
</tr>
<tr>
<td>Factor $A_1$</td>
<td>0.7 0.6 0.5 0.4 0.3 0.2 0.1</td>
</tr>
<tr>
<td>$\chi$-shape performance</td>
<td>95.43 95.43 95.8 95.84 96.18 95.84 95.84</td>
</tr>
<tr>
<td>Average from 1.5 to 0.1</td>
<td>95.13±0.35</td>
</tr>
</tbody>
</table>

Table 7. Ablation studies on the influence of $A_1$ (fixed at 0.1) and different matching factor $A_2$ (as in Equation 4 in the main paper) to the top-ranking mean accuracy (in %) on 99.5% BA ratio Colored MNIST dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Colored MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor $A_2$</td>
<td>1.5 1.4 1.3 1.2 1.1 1.0 0.9 0.8</td>
</tr>
<tr>
<td>$\chi$-shape performance</td>
<td>95.84 95.84 96.18 95.84 95.84 95.58 95.58 95.55</td>
</tr>
<tr>
<td>Factor $A_2$</td>
<td>0.7 0.6 0.5 0.4 0.3 0.2 0.1</td>
</tr>
<tr>
<td>$\chi$-shape performance</td>
<td>95.25 94.95 94.99 94.69 94.69 94.69 94.39</td>
</tr>
<tr>
<td>Average from 1.5 to 0.1</td>
<td>95.33±0.27</td>
</tr>
</tbody>
</table>

In this process, we set the temperature $\tau$ in the metric-based prediction of mixed prototypes $p_\gamma$ or $p_{1-\gamma}$ as Equation 7 of the paper 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 LiF [25].

3. Additional Experiments

3.1. Results with error bars

We run our methods and the comparison methods like vanilla method and Learning from Failure (LfF) [25] multiple times and report error bars. We present the full results with both 95% confidence interval as Table 5 and the standard deviation in Figure 3.

3.2. 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 of the main paper with four typical samples of BC samples, intermediate attribute samples, and BA samples. Here we show more observations on the whole training set in statistical significance.

- As shown in Figure 1, 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 main paper, in fact, this is another interpretation of the right half of Figure 4 in the paper.

- The right two columns of Figure 1 also represent more statistical information at the dataset level, e.g., the third column shows the $\chi$-shaped prediction of BC samples over the whole dataset as training epoch increases. This corresponds to the left half of 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 $\chi$-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 8. Moreover, we plot the PR curves of various methods in the first stage on Colored MNIST and Biased NICO datasets in Figure 2. The results show that our method maintains excellent performance.

3.3. Ablation study of $\chi$-structured metric learning

In order to verify whether the effectiveness of our method is indeed derived from our $\chi$-structured metric learning objective. We first remove one of the mixed prototypes and bias bag losses as $-\mathcal{L}_{CE}(p_\gamma, B_1-\gamma)$ in Table 4. This substantially lose the metric-based push relationship between the BA
samples and the high BC ratio prototypes $p_{\gamma}$. 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 $p_{1-\gamma}$. This reduces the debiasing capability using the general properties of the Figure 5 in the main paper. The results show that our method with $\chi^2$-structured 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 $\chi^2$-structured can further improve the model performance and overcome the debiasing problem comprehensively.

3.4. Robustness of the $\chi^2$-model with varying identification thresholds of BC sample

For $\chi^2$-model, we use the BC identification thresholds to split $D_{||}$ and $D_{\perp}$. We show the influence of different thresholds in the Table 10, 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 $\chi$-model. From this result, we can find that the model is less affected by the thresholds. Furthermore, since the Bias Bag $\{ B_{\gamma}, B_{1-\gamma}\}$ 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 $\chi^2$-pattern scores.

4. Overall Algorithm

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

5. Discussion About the Limitations

In this paper, we adopt a new two-stage $\chi^2$-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 approach may emphasize the weaker one, but retain the effects of the stronger one.
Algorithm 1 Training for $\chi^2$-model

**Require:** Biased training data $D_{train} = \{(x_i, y_i)\}_{i=1}^N$.

1. **First stage:** $\chi$-shape pattern.
2. Train a vanilla model $\theta$ on $D_{train}$ with cross entropy loss as mentioned in Equation 1 in the main paper:
   $$
   \mathcal{L}_{CE} = \mathbb{E}_{(x_i, y_i) \sim D_{train}} \left[- \log \Pr(h_\theta(x_i) = y_i \mid x_i)\right].
   $$
3. Consider the $T$ epochs change on ground-truth label $y_i$ and bias label $b_i$ ($x_i$, $h_\theta$):
   $$
   \mathcal{L}_{CE}(x_i) = \left(\mathcal{L}_{CE}^{gt}(x_i) = \left\{- \log \Pr^t(y_i \mid x_i)\right\}_t^T \mathcal{L}_{CE}(x_i) = \left\{- \log \Pr^t(b_i \mid x_i, h_\theta) \mid x_i\right\}_t^T \right).$
4. Capture the BC sample with two exponential $\chi$-shape functions:
   $$
   \chi_{shape} = \left\{p_\theta = \left\{e^{-At}\right\}_t^T, p^b = \left\{e^{At}\right\}_t^T\right\}.
   $$
5. Compute the ranking score $s(x_i)$ with the inner product over two curves as Equation 4 in the paper:
   $$
   s(x_i) = \langle L_{CE}(x_i), \chi_{shape} \rangle = \langle L_{CE}^{gt}(x_i), p_\theta^t \rangle + \langle L_{CE}(x_i), p^b \rangle
   $$
   $$
   = \sum_{t=1}^T -e^{-At} \log \Pr(h_\theta(x_i) = y_i \mid x_i) - e^{At} \log \Pr(h_\theta(x_i) = b_i \mid x_i, h_\theta) \mid x_i).
   $$
6. **Second stage:** $\chi$-structured metric learning objective.
7. for each step $\gamma$
   8. Construct multiple bias bags $B_\gamma$ with bootstrapping as Equation 5 in the paper:
   $$
   B_\gamma = \{(x_i, y_i) \mid \text{NUM}(D_{+}) : \text{NUM}(D_{\parallel}) = \gamma\}
   $$
9. where the ratio of BC samples is $\gamma$.
10. Build the prototype $p$ for class $c$ based on $B_\gamma$ as Equation 6 in the paper:
   $$
   p_{\gamma, c} = \frac{1}{K} \sum_{(x_i, y_i) \in B_\gamma} f_\phi(x_i) \cdot [y_i = c].
   $$
11. Consider a high $\gamma$:
   12. for all samples $x_i \in B_{1-\gamma}$
   13. Classify with $p_\gamma$ as Equation 7 in the paper:
   $$
   \Pr(y_i \mid x_i) = \frac{\exp (-d(f_\phi(x_i), p_{\gamma, y_i})/\tau)}{\sum_{c \in [C]} \exp (-d(f_\phi(x_i), p_{\gamma, c})/\tau)}.
   $$
14. Compute $\mathcal{L}_{CE}(p_{\gamma}, B_{1-\gamma})$.
15. end for
16. for all samples $x_i \in B_\gamma$
17. Classify with $p_{1-\gamma}$ as mentioned before.
18. Compute $\mathcal{L}_{CE}(p_{1-\gamma}, B_\gamma)$.
19. end for
20. Compute $\nabla_\phi \mathcal{L}_{CE}(p_{\gamma}, B_{1-\gamma}) + \mathcal{L}_{CE}(p_{1-\gamma}, B_\gamma)$.
21. Update $\phi$ with $\nabla_\phi$.
22. end for

**References**

Figure 1. The more observation in the first \( \chi \)-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.

Figure 2. The Precision-Recall curves of the BC samples identification on Colored MNIST dataset (as above C-MNIST) over various ratios. Best view in colors.
Figure 3. 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 4. Example samples of Colored MNIST and Corrupted CIFAR-10 datasets.

(iii) Biased CelebA

(iv) Biased NICO

Figure 5. Example samples of Biased CelebA and Biased NICO datasets.