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

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 in Section 1. Further, we describe more implementation details in Section 2, such as the visualization of Figure 2, matching factors A_1 and A_2 of Equation 3, and the Biased NICO dataset construction of subsection 4.1 in the main paper. We also show additional experimental observations and results in Section 3, including error intervals, robustness analysis, etc.

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]. Similarity, 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 $\mathbf{w}_{proj} \in \mathbb{R}^{d\times 2}$ 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 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 2 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 $\{1\text{-}1 : 1, 098\}$. 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 [11], 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 *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%.

Table 1. 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 2. 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

2.3. 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 and standard deviation of both $(0.5, 0.5, 0.5)$. In the real-world datasets like 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 most of the previous

Table 3. 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	–	–	–	–

works [32], 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 $(0.229, 0.224, 0.225)$.

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 7×7 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 χ-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 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

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 ○ 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	○	<u>32.98</u>$_{\pm 2.20}$	<u>69.44</u>$_{\pm 3.15}$	<u>85.78</u>$_{\pm 7.32}$	<u>95.79</u>$_{\pm 0.99}$
χ^2-model	○	**68.04**$_{\pm 1.22}$	**90.37**$_{\pm 1.33}$	**93.21**$_{\pm 0.91}$	**98.30**$_{\pm 0.35}$

2.4.1 χ-shape matching pattern

- As stated in the paper, we design two exponential χ-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 χ-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 ρ in the whole dataset, *i.e.*, the weight on BC samples is ρ and on the BA ones is $1-\rho$.

2.4.2 χ-structured metric learning

- In the second stage, we construct the data pools $\mathcal{D}_{\|}$ 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 3.4 and Table 10.

  We then 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 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 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 shown in the paper, the computation of $\mathcal{L}_{\text{CE}}\left(\mathbf{p}_{\gamma}, \mathcal{B}_{1-\gamma}\right)$ and $\mathcal{L}_{\text{CE}}\left(\mathbf{p}_{1-\gamma}, \mathcal{B}_{\gamma}\right)$ 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.

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 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.

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	$95.33_{\pm 0.27}$							

In this process, we set the temperature τ in the metric-based prediction of mixed prototypes $\mathbf{p}_\gamma$ or $\mathbf{p}_{1-\gamma}$ 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 LfF [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 a 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 χ-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 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 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 χ-structured metric learning

In order to verify whether the effectiveness of our method is indeed derived from our χ-structured 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 4. This substantially lose the metric-based *push* relationship between the BA

Table 8. The *average precision (AP)* 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 [15]	96.49	83.52	77.61	64.94	30.83
Confidence [21]	96.68	85.61	80.73	69.76	31.00
Loss [25]	<u>98.06</u>	<u>98.22</u>	<u>97.03</u>	<u>91.55</u>	**39.64**
Pleiss *et al.* [26]	97.97	89.24	79.15	55.48	30.48
Zhao *et al.* [34]	98.03	96.04	93.27	81.29	34.27
χ-pattern	**98.07**	**98.44**	**97.77**	**95.96**	<u>37.79</u>

Table 9. The *mean accuracy* of BC samples identification on the Colored MNIST and NICO dataset. It is similar to the above marked table.

Dataset	**Colored MNIST**				**NICO**
Ratio (%)	99.9	99.5	99.0	95.0	86.27
Entropy [15]	91.66	78.33	72.57	63.43	34.31
Confidence [21]	91.66	80.33	76.25	68.90	34.91
Loss [25]	<u>95.00</u>	<u>94.00</u>	<u>92.80</u>	<u>87.43</u>	**41.81**
Pleiss *et al.* [26]	<u>95.00</u>	82.67	72.24	53.70	33.53
Zhao *et al.* [34]	<u>95.00</u>	90.33	88.12	79.66	39.84
χ-pattern (Ours)	**96.66**	**95.67**	**95.48**	**94.30**	<u>40.00</u>

Table 10. 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

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 $\mathbf{p}_{1-\gamma}$. This reduces the debiasing capability using the general properties of the Figure 5 in the main paper. The results show that our method with χ-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 χ-structured can further improve the model performance and overcome the debiasing problem comprehensively.

3.4. Robustness of the χ^2-model with varying identification thresholds of BC sample

For χ^2-model, we use the BC identification thresholds to split $\mathcal{D}_{\parallel}$ and $\mathcal{D}_{\perp}$. 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 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.

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 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 approach may emphasize the weaker one, but retain the effects of the stronger one.

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: χ**-shape pattern**.

Train a *vanilla* model $\boldsymbol{\theta}$ on $\mathcal{D}_{\text{train}}$ with cross entropy loss as mentioned in Equation 1 in the main paper:

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

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 χ-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 4 in the paper:

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

Second stage: χ**-structured metric learning objective**.

for each step **do**

Construct multiple *bias bags* $\mathcal{B}_{\boldsymbol{\gamma}}$ with bootstrapping as Equation 5 in the paper:

$$\mathcal{B}_{\boldsymbol{\gamma}} = \left\{ (\mathbf{x}_i, y_i) \,\middle|\, \mathrm{NUM}\left(\mathcal{D}_{\perp}\right) : \mathrm{NUM}\left(\mathcal{D}_{\parallel}\right) = \boldsymbol{\gamma} \right\} ,$$

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 6 in the paper:

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

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 7 in the paper:

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

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

References

[1] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. 1

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, pages 528–539, 2020. 1

[3] Rémi Cadène, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, pages 839–850, 2019. 1

[4] Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. Fairfil: Contrastive neural debiasing method for pretrained text encoders. In *ICLR*, 2021. 1

[5] 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*, pages 4067–4080, 2019. 1

[6] Luke Nicholas Darlow, Stanislaw Jastrzebski, and Amos J. Storkey. Latent adversarial debiasing: Mitigating collider bias in deep neural networks. *CoRR*, abs/2011.11486, 2020. 2

[7] 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. 1

[8] 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. 1

[9] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. Model patching: Closing the subgroup performance gap with data augmentation. In *ICLR*, 2021. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[11] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021. 2

[12] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, pages 793–811, 2018. 1

[13] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *NeurIPS*, pages 26449–26461, 2021. 2, 3

[14] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pages 124–140, 2020. 2

[15] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379, 2009. 2, 6

[16] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, pages 9012–9020, 2019. 1

[17] Nayeong Kim, SEHYUN HWANG, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. In *NeurIPS*, pages 18403–18415, 2022. 2

[18] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *ICLR*, 2023. 2

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

[20] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *NeurIPS*, 2021. 2

[21] Mingkun Li and Ishwar K. Sethi. Confidence-based active learning. *IEEE TPAMI*, 28(8):1251–1261, 2006. 2, 6

[22] Yi Li and Nuno Vasconcelos. REPAIR: removing representation bias by dataset resampling. In *CVPR*, pages 9572–9581, 2019. 1

[23] 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. 1

[24] 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*, pages 6781–6792, 2021. 2

[25] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, pages 20673–20684, 2020. 1, 2, 5, 6

[26] Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, pages 17044–17056, 2020. 2, 6

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2020. 1

[28] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *CVPR*, pages 13508–13517, 2021. 1

[29] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019. 4

[30] Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *ICLR*, 2019. 1

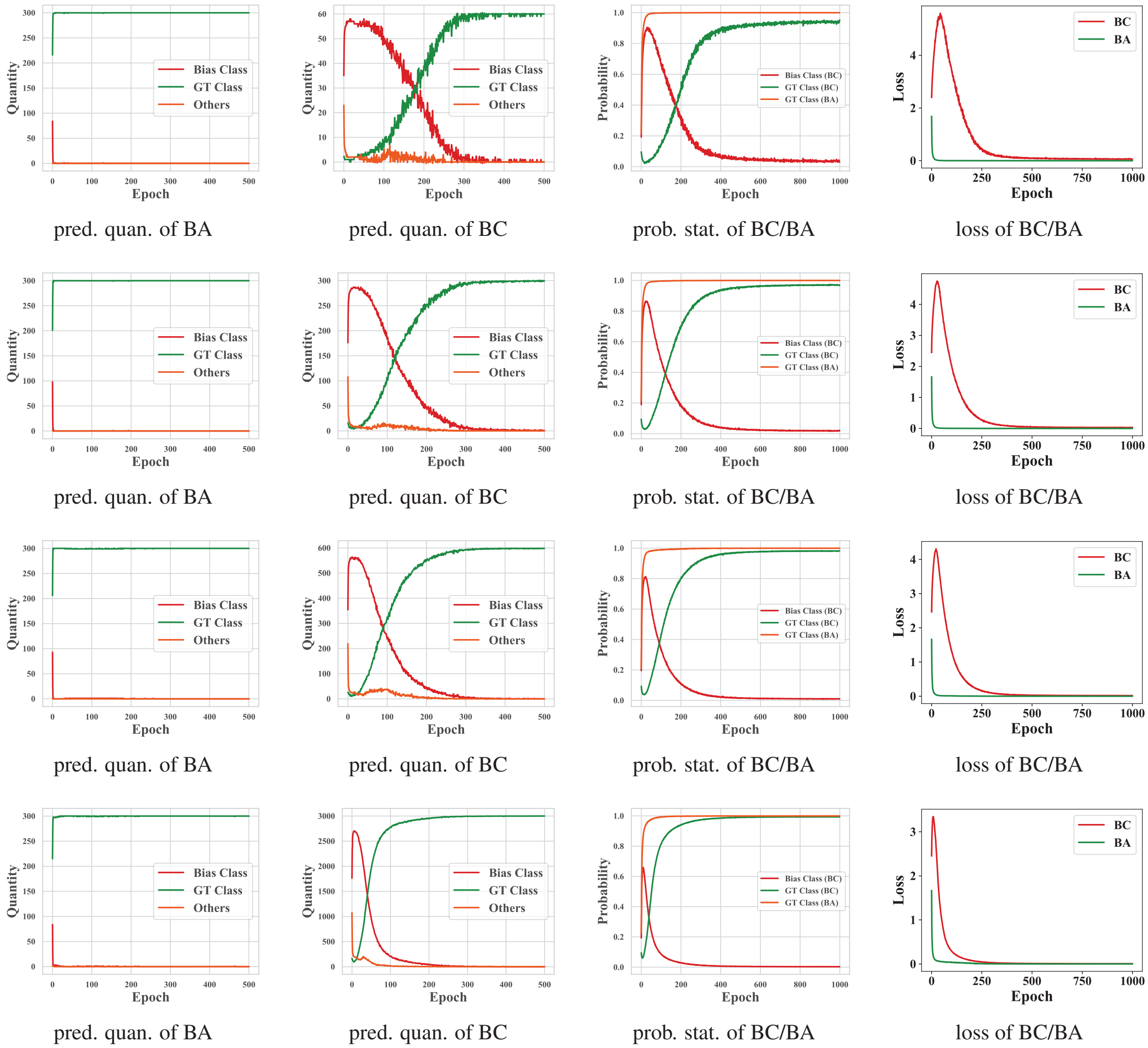
[31] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, pages 8916–8925, 2020. 1

[32] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *CVPR*, pages 5372–5382, 2021. 3

[33] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8792–8802, 2018. 1

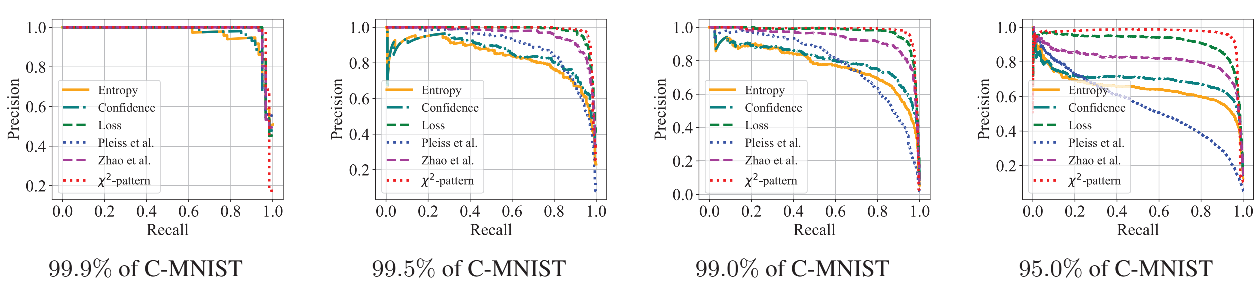
[34] Bowen Zhao, Chen Chen, Qi Ju, and Shutao Xia. Learning debiased models with dynamic gradient alignment and bias-conflicting sample mining. *CoRR*, abs/2111.13108, 2021. 2, 6

[35] Wei Zhu, Haitian Zheng, Haofu Liao, Weijian Li, and Jiebo Luo. Learning bias-invariant representation by cross-sample mutual information minimization. In *ICCV*, pages 14982–14992, 2021. 2



pred. quan. of BA | pred. quan. of BC | prob. stat. of BC/BA | loss of BC/BA

Figure 1. 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.



99.9% of C-MNIST | 99.5% of C-MNIST | 99.0% of C-MNIST | 95.0% of C-MNIST

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.

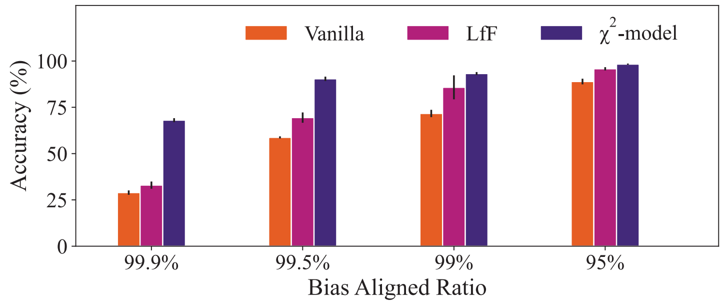


Figure 3. The classification performance with error bars on unbiased Colored MNIST test set. Error bars expressed by the black line denote the standard deviation. Best view in colors.

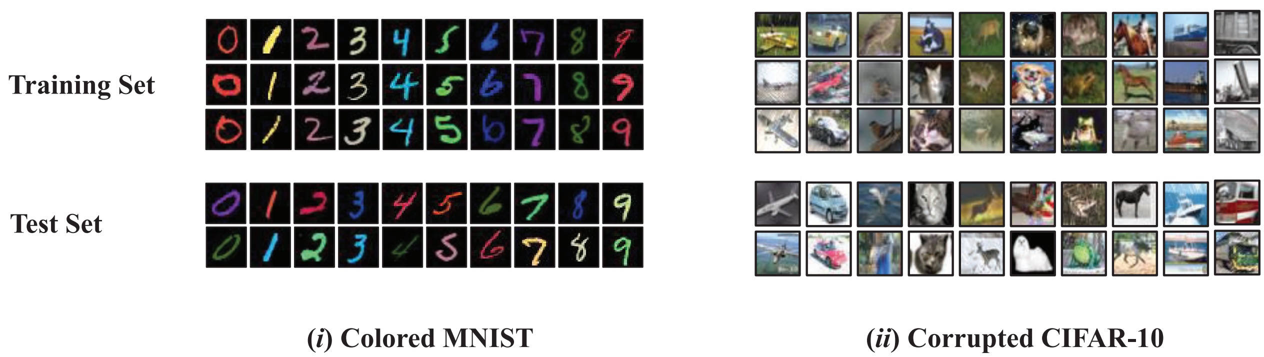


Figure 4. Example samples of Colored MNIST and Corrupted CIFAR-10 datasets.

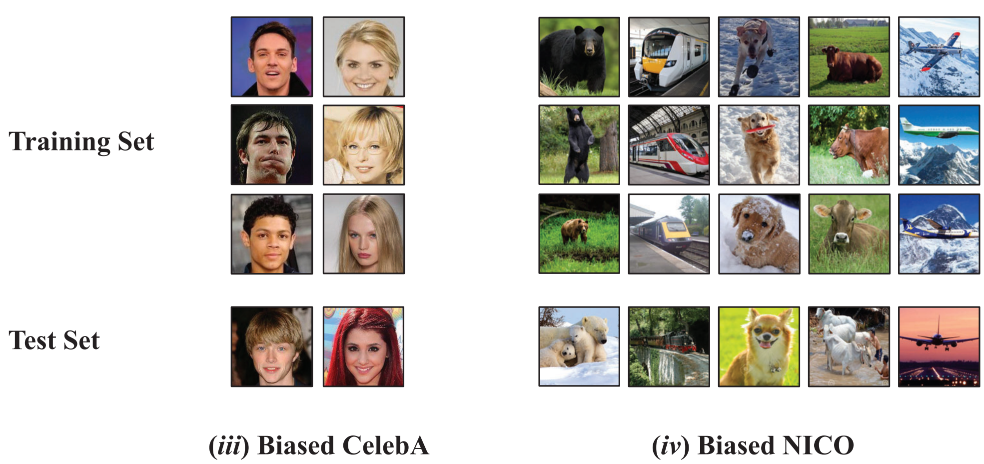


Figure 5. Example samples of Biased CelebA and Biased NICO datasets.