# Navigate Beyond Shortcuts: Debiased Learning through the Lens of Neural Collapse

Supplementary Material

## A. Detailed Theoretical Justification

### A.1. Analysis of Vanilla Training

To illustrate why vanilla models tend to pursue shortcut learning, we follow the analysis of previous works [36, 43] and re-examine the issue of biased classification from the perspective of gradients.

Following the definition in Section 3.1, we denote $\mathbf{x}_{k,i}$ as the $i$-th sample of the $k$-th class, $\mathbf{z}_{k,i} \in \mathbb{R}^d$ as its corresponding last-layer feature, and $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K]$ as the weights of classifier. In vanilla training, the cross-entropy loss is defined as:

$$\mathcal{L}_{\mathrm{CE}}(\mathbf{z}_{k,i}, \mathbf{W}) = -\log\left(\frac{\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k'=1}^{K}\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_{k'})}\right) \tag{A.1}$$

**Gradient *w.r.t* classifier weights.** To analyze the learning behavior of the classifier, we first compute the gradient of $\mathcal{L}_{\mathrm{CE}}$ *w.r.t* the classifier weights:

$$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial \mathbf{w}_k} = \underbrace{\sum_{i=1}^{n_k} -(1 - p_k(\mathbf{z}_{k,i}))\mathbf{z}_{k,i}}_{\text{pulling part}} + \underbrace{\sum_{k' \neq k}^{K}\sum_{j=1}^{n_{k'}} p_k(\mathbf{z}_{k',j})\mathbf{z}_{k',j}}_{\text{forcing part}} \tag{A.2}$$

where $n_k$ denotes the number of training samples in the $k$-th class, and $p_k(\mathbf{z})$ is the predicted probability of $\mathbf{z}$ belongs to the $k$-th class, which is calculated with the softmax function:

$$p_k(\mathbf{z}) = \frac{\exp(\mathbf{z}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k'=1}^{K}\exp(\mathbf{z}^{\mathrm{T}}\mathbf{w}_{k'})}, 1 \leq k \leq K \tag{A.3}$$

In Eq. A.2, we decompose the gradient *w.r.t* the classifier weight $\mathbf{w}_k$ into two parts, *the pulling part* and *the forcing part*. The pulling part contains the effects of features from the same class (i.e., $\mathbf{z}_{k,i}$), which pulls the classifier weight towards the $k$-th feature cluster and each feature has an influence of $1 - p_k(\mathbf{z}_{k,i})$. Meanwhile, the forcing part of the gradient contains the features from other classes to push $\mathbf{w}_k$ away from the wrong clusters, and each feature has an influence of $p_k(\mathbf{z}_{k',j})$. When the vanilla model is trained on biased datasets, the prevalent bias-aligned samples of the $k$-th class will dominate the pulling part of the gradient. The classifier weight $\mathbf{w}_k$ will be pulled towards the center of the bias-aligned features, which have a strong correlation between the class label $k$ and a bias attribute $b_k$. The biased feature space based on shortcut will thus be formed at the early period of training, as the result of the imbalanced magnitude of gradients across different attributes. Similarly, the forcing part of the gradient is also guided by the bias-aligned samples of other classes, further reinforcing the tendency of shortcut learning. It confirms our observation in Section 3.2 that the model's pursuit of shortcut correlation leads to a biased, non-collapsed feature space, which is hard to rectify in the subsequent training steps.

**Gradient *w.r.t* features.** Furthermore, we compute the gradient of $\mathcal{L}_{\mathrm{CE}}$ *w.r.t* the last-layer features:

$$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial \mathbf{z}_{k,i}} = \underbrace{-(1 - p_k(\mathbf{z}_{k,i}))\mathbf{w}_k}_{\text{pulling part}} + \underbrace{\sum_{k' \neq k}^{K} p_{k'}(\mathbf{z}_{k,i})\mathbf{w}_{k'}}_{\text{forcing part}} \tag{A.4}$$

In Eq. A.4, the gradient *w.r.t* features is also considered as the combination of *the pulling part* and *the forcing part*. The pulling part represents the pulling effect of the classifier weight from the same class (i.e., $\mathbf{w}_k$), which will guide the features to align with the prediction behavior of the classifier. The forcing part, on the contrary, represents the pushing effect of other classifier weights. As we discussed before, the model's reliance on simple shortcuts is formed at the early stage of training, due to the misled classifier weights toward the centers of bias-aligned features. It results in a biased decision rule of the classifier, which directly affects the formation of the last-layer feature space. Consider the bias-conflicting samples of the $k$-th class, although the pulling part of the gradient supports its convergence towards the right classifier weight $\mathbf{w}_k$,

the established shortcut correlation is hard to reverse and eliminate, which has been demonstrated with the metrics of Neural Collapse in Section 3.2.

### A.2. Analysis of ETF-Debias

In light of the analysis of vanilla training, our proposed debiasing framework, ETF-Debias, turns the easy-to-follow shortcut into the prime features, which guides the model to skip the active learning of shortcuts and directly focus on the intrinsic correlations. We have provided a brief theoretical justification of our method in Section 4.3, and the detailed illustrations are as follows.

When trained on biased datasets, we assume each class $[1, ..., K]$ is strongly correlated with a bias attribute $[b_1, ..., b_K]$. Based on the mechanism of prime training, we denote the $i$-th feature of the $k$-th class as $\widetilde{\mathbf{z}}_{k,i} = [\mathbf{z}_{k,i}, \mathbf{m}_{i,b}] \in \mathbb{R}^{2\times d}$, where $\mathbf{z}_{k,i} \in \mathbb{R}^d$ represents the learnable features, and $\mathbf{m}_{i,b} \in \mathbb{R}^d$ represents the prime features retrieved based on the bias attribute $b$ of the input sample. With the definition, a bias-aligned sample of the $k$-th class $\mathbf{x}_{k,i}$ will have its feature in form of $\widetilde{\mathbf{z}}_{k,i} = [\mathbf{z}_{k,i}, \mathbf{m}_{b_k}]$, and a bias-conflicting sample will have its feature as $\widetilde{\mathbf{z}}_{k,i} = [\mathbf{z}_{k,i}, \mathbf{m}_{b_{k'}}], k' \neq k$. To keep the same form, we denote the classifier weight as $\widetilde{\mathbf{w}}_k = [\mathbf{w}_k, \mathbf{a}_k] \in \mathbb{R}^{2\times d}$, where $\mathbf{w}_k \in \mathbb{R}^d$ represents the weight for intrinsic correlations and $\mathbf{a}_k \in \mathbb{R}^d$ is the weight for shortcut features. In the detailed convergence result of Neural Collapse (Section C), we observe that, due to the fixed prime features and their strong correlation with the bias attributes, $\mathbf{a}_k$ will quickly collapse into its bias-correlated prime feature $\mathbf{m}_{b_k}$, and can be viewed as constant after just a few steps of training. Thus the cross-entropy loss can be re-written as:

$$\mathcal{L}_{\mathrm{CE}}(\widetilde{\mathbf{z}}_{k,i}, \widetilde{\mathbf{W}}) = -\log\left(\frac{\exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}}\widetilde{\mathbf{w}}_k)}{\sum_{k'=1}^{K}\exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k'})}\right) = -\log\left(\frac{\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_k, \mathbf{a}_k])}{\sum_{k'=1}^{K}[\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_{k'}, \mathbf{a}_{k'}]}\right) \tag{A.5}$$

**Gradient *w.r.t* classifier weights.** With the avoid-shortcut learning framework, we justify that the introduced prime mechanism implicitly plays the role of re-weighting, which weakens the mutual convergence between bias-aligned features and the classifier weights, and amplifies the learning of intrinsic correlations. We first analyze the gradient of $\mathcal{L}_{\mathrm{CE}}$ *w.r.t* the classifier weights $\widetilde{\mathbf{W}}$:

$$\begin{aligned}
\frac{\partial\mathcal{L}_{\mathrm{CE}}}{\partial\widetilde{\mathbf{w}}_k} &= \sum_{i=1}^{n_k} -(1 - p_k(\widetilde{\mathbf{z}}_{k,i}))\widetilde{\mathbf{z}}_{k,i} + \sum_{k'\neq k}^{K}\sum_{j=1}^{n_{k'}} p_k(\widetilde{\mathbf{z}}_{k',j})\widetilde{\mathbf{z}}_{k',j} \\
&= \sum_{i=1}^{n_k} -\left(1 - \frac{\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_k, \mathbf{a}_k])}{\sum_{k'=1}^{K}\exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k'})}\right)\widetilde{\mathbf{z}}_{k,i} + \sum_{k'\neq k}^{K}\sum_{j=1}^{n_{k'}}\frac{\exp([\mathbf{z}_{k',j}, \mathbf{m}_{j,b'}]^{\mathrm{T}}[\mathbf{w}_k, \mathbf{a}_k])}{\sum_{k'=1}^{K}\exp(\widetilde{\mathbf{z}}_{k',j}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k'})}\widetilde{\mathbf{z}}_{k',j} \qquad \text{(A.6)} \\
&\leq \sum_{i=1}^{n_k} -\left(1 - \frac{\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k'=1}^{K}\exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k'})} - \frac{\exp(\mathbf{m}_{i,b}^{\mathrm{T}}\mathbf{a}_k)}{\sum_{k'=1}^{K}\exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k'})}\right)\widetilde{\mathbf{z}}_{k,i} \\
&\quad + \sum_{k'\neq k}^{K}\sum_{j=1}^{n_{k'}}\left(\frac{\exp(\mathbf{z}_{k',j}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k''=1}^{K}\exp(\widetilde{\mathbf{z}}_{k',j}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k''})} + \frac{\exp(\mathbf{m}_{j,b'}^{\mathrm{T}}\mathbf{a}_k)}{\sum_{k''=1}^{K}\exp(\widetilde{\mathbf{z}}_{k',j}^{\mathrm{T}}\widetilde{\mathbf{w}}_{k''})}\right)\widetilde{\mathbf{z}}_{k',j} \qquad \text{(A.7)} \\
&\leq \underbrace{\sum_{i=1}^{n_k} -(1 - p_k^{(b)}(\mathbf{m}_{i,b}) - p_k^{(l)}(\mathbf{z}_{k,i}))\widetilde{\mathbf{z}}_{k,i}}_{\text{pulling part}} + \underbrace{\sum_{k'\neq k}^{K}\sum_{j=1}^{n_{k'}}(p_k^{(b)}(\mathbf{m}_{j,b'}) + p_k^{(l)}(\mathbf{z}_{k',j}))\widetilde{\mathbf{z}}_{k',j}}_{\text{forcing part}} \qquad \text{(A.8)}
\end{aligned}$$

The predicted probabilities both satisfy $0 < (1 - p_k(\widetilde{\mathbf{z}}_{k,i})) < 1$ and $0 < p_k(\widetilde{\mathbf{z}}_{k',j}) < 1$, and the scaling step from Eq. A.6 to Eq. A.7 is based on the Jensen inequality. The terms $p_k^{(l)}$ and $p_k^{(b)}$ are respectively the predicted probabilities for class labels and bias attributes, calculated with softmax:

$$p_k^{(l)}(\mathbf{z}_{k,i}) = \frac{\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k'=1}^{K}\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_{k'}, \mathbf{a}_{k'}])} \tag{A.9}$$

$$p_k^{(b)}(\mathbf{m}_{i,b}) = \frac{\exp(\mathbf{m}_{i,b}^{\mathrm{T}}\mathbf{a}_k)}{\sum_{k'=1}^{K}\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_{k'}, \mathbf{a}_{k'}])} \tag{A.10}$$

Based on the gradient *w.r.t* classifier weights in Eq. A.8, the probability $p_k^{(b)} \propto \exp(\mathbf{m}_b^{\mathrm{T}}\mathbf{a}_k)$ re-weights the influence of different samples during optimization. Consider the features of the $k$-th class, a bias-aligned sample with its feature

$\widetilde{\mathbf{z}}_{k,i} = [\mathbf{z}_{k,i}, \mathbf{m}_{b_k}]$ will have a much higher probability $p_k^{(b)}$, compared to a bias-conflicting sample of the same class with a feature as $\widetilde{\mathbf{z}}_{k,j} = [\mathbf{z}_{k,j}, \mathbf{m}_{b_{k'}}], k' \neq k$. Therefore, according to the influence of each feature in the pulling part of gradient $(1 - p_k^{(b)}(\mathbf{m}_{i,b}) - p_k^{(l)}(\mathbf{z}_{k,i}))$, the bias-aligned samples will have their influence down-weighted, while the influence of bias-conflicting samples are up-weighted, which weakens the dominance of the prevalent bias-aligned samples in the pulling effect and prevents the formation of shortcut correlations. Meanwhile, according to the forcing part of the gradient *w.r.t* $\mathbf{w}_k$, samples from other classes but have the bias-correlated attribute $b_k$ will have a relatively strong forcing effect on the weight $\mathbf{w}_k$, which also disturbs the learning of simple shortcuts and redirect the model's attention to the intrinsic, generalizable relations.

**Gradient *w.r.t* features.** With the implicit re-weighting mechanism of gradients, the classifier weights are directed away from the simple shortcuts since the beginning of model training. We also compute the gradient of $\mathcal{L}_{\text{CE}}$ *w.r.t* the feature $\widetilde{\mathbf{z}}_{k,i}$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{CE}}}{\partial \widetilde{\mathbf{z}}_{k,i}} &= -(1 - p_k(\widetilde{\mathbf{z}}_{k,i}))\mathbf{w}_k + \sum_{k' \neq k}^{K} p_{k'}(\widetilde{\mathbf{z}}_{k,i})\mathbf{w}_{k'} \\
&= -\left(1 - \frac{\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_k, \mathbf{a}_k])}{\sum_{k'=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k'})}\right)\mathbf{w}_k + \sum_{k' \neq k}^{K} \frac{\exp([\mathbf{z}_{k,i}, \mathbf{m}_{i,b}]^{\mathrm{T}}[\mathbf{w}_{k'}, \mathbf{a}_{k'}])}{\sum_{k''=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k''})}\mathbf{w}_{k'} \qquad \text{(A.11)} \\
&\leq -\left(1 - \frac{\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_k)}{\sum_{k'=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k'})} - \frac{\exp(\mathbf{m}_{i,b}^{\mathrm{T}}\mathbf{a}_k)}{\sum_{k'=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k'})}\right)\mathbf{w}_k \\
&\quad + \sum_{k' \neq k}^{K} \left(\frac{\exp(\mathbf{z}_{k,i}^{\mathrm{T}}\mathbf{w}_{k'})}{\sum_{k''=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k''})} + \frac{\exp(\mathbf{m}_{i,b}^{\mathrm{T}}\mathbf{a}_{k'})}{\sum_{k''=1}^{K} \exp(\widetilde{\mathbf{z}}_{k,i}^{\mathrm{T}} \widetilde{\mathbf{w}}_{k''})}\right)\mathbf{w}_{k'} \qquad \text{(A.12)} \\
&\leq \underbrace{-(1 - p_k^{(b)}(\mathbf{m}_{i,b}) - p_k^{(l)}(\mathbf{z}_{k,i}))\mathbf{w}_k}_{\text{pulling part}} + \underbrace{\sum_{k' \neq k}^{K} (p_{k'}^{(b)}(\mathbf{m}_{i,b}) + p_{k'}^{(l)}(\mathbf{z}_{k,i}))\mathbf{w}_{k'}}_{\text{forcing part}} \qquad \text{(A.13)}
\end{aligned}$$

where the scaling step from Eq. A.11 to Eq. A.12 is based on the Jensen inequality. Similar to the analysis before, the predicted probability for bias attributes $p_k^{(b)} \propto \exp(\mathbf{m}_b^{\mathrm{T}}\mathbf{a}_k)$ also performs re-weighting on the gradient *w.r.t* features. With the prime feature $\mathbf{m}_{b_k}$, a bias-aligned sample of the $k$-th class will have a down-weighted influence in the pulling part towards the classifier weight $\mathbf{w}_k$, which avoids the dominance of bias-aligned features around the weight centers. In contrast, a bias-conflicting feature will be emphasized and have a magnified pulling effect towards the weight $\mathbf{w}_k$, which is crucial for the learning of intrinsic correlations. As to the forcing part of the gradient, a bias-conflicting sample with prime feature $\mathbf{m}_{b_{k'}}$ will obtain a strong pushing effect from the weight $\mathbf{w}_{k'}$, which forces it away from the wrong weight center and mitigate the spurious correlations. The implicitly adjusted pulling and forcing effects both contribute to the feature's convergence according to the intrinsic correlations.

To sum up, through the theoretical justification from the perspective of gradients, the introduced prime mechanism implicitly re-weights the pulling and forcing effect of gradients. The down-weighted influence of bias-aligned samples weakens the trend of pursuing simple shortcuts, and the up-weighted influence of bias-conflicting samples enhanced the focus on intrinsic correlations, thus leading to an unbiased feature space with improved generalization capability.

## B. Dataset Description and Implementation Details

### B.1. Datasets

We use two synthetic datasets (i.e., Colored MNIST and Corrupted CIFAR-10) as well as three real-world datasets (i.e., Biased FFHQ, Dogs & Cats, and BAR) to evaluate the debiasing performance of our proposed method. The illustrative examples of the datasets are displayed in Fig. B.1, B.2, B.3.

**Colored MNIST [15].** As a modified version of the MNIST dataset [39], Colored MNIST introduces the color of digits as the bias attribute. The ten classes of digits are each correlated with a certain color (e.g., red for digit 0). We conduct experiments with the dataset used in [13, 18, 19, 23] and set the ratio of bias-conflicting training samples as $\{0.5\%, 1\%, 2\%, 5\%\}$. The unbiased test set contains an equal number of bias-aligned and bias-conflicting samples.

**Corrupted CIFAR-10 [10].** Also modified from the CIFAR-10 dataset [17], Corrupted CIFAR-10 apply different types of

corruptions to the original images, with each class correlated with a certain type of corruption. Following the previous studies [13, 30], we adopt the corruptions of *Brightness, Contrast, Gaussian Noise, Frost, Elastic Transform, Gaussian Blur, Defocus Blur, Impulse Noise, Saturate* and set the ratio of bias-conflicting training samples as $\{0.5\%, 1\%, 2\%, 5\%\}$. The unbiased test set contains an equal number of bias-aligned and bias-conflicting samples.

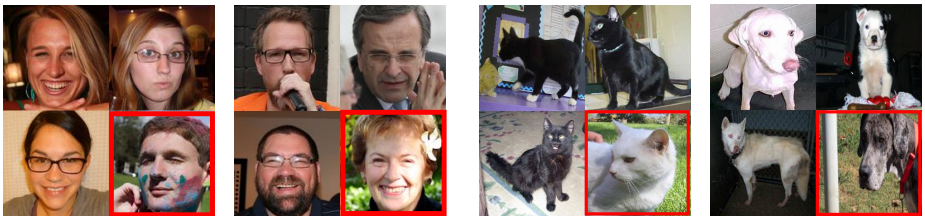
**Biased FFHQ [18].** Based on the Flickr-Faces-HQ dataset [14] of face images, Biased FFHQ (BFFHQ) is constructed with the target attribute of age and the bias attribute of gender. The bias-aligned samples contain images of young females (i.e., ages ranging from 10 to 29) and old males (i.e., ages ranging form 40 to 59), and the bias-conflicting samples contain old females and young males. We set the ratio of bias-conflicting training samples as $\{0.5\%, 1\%, 2\%, 5\%\}$. The unbiased test set contains only bias-conflicting samples.

**Dogs & Cats [15].** First used in [15], the Dogs & Cats dataset is constructed with the target attribute of animal and the bias attribute of color (i.e., bright or dark color). The bias-aligned samples contain images of dark cats and bright dogs, while the bias-conflicting samples contain bright cats and dark dogs. We set the ratio of bias-conflicting training samples as $\{1\%, 5\%\}$, and the unbiased test set contains only bias-conflicting samples.

**BAR [23].** The Biased Action Recognition (BAR) dataset contains six classes of actions (i.e., *Climbing, Diving, Fishing, Vaulting, Racing, Throwing*), each correlated with a bias attribute of place (i.e., *Rockwall, Underwater, WaterSurface, APavedTrack, PlayingField, Sky*). The bias-conflicting samples contain images with rare action-place pairs. We set the ratio of bias-conflicting training samples as $\{1\%, 5\%\}$, and the unbiased test set contains only bias-conflicting samples.

(a) Colored MNIST (b) Corrupted CIFAR-10

Figure B.1. Illustrative examples of synthetic datasets, (a) Colored MNIST with the bias attribute of color and (b) Corrupted CIFAR-10 with the bias attribute of different types of corruption (e.g. augmentations like Gaussian Blur and Pixelate, etc). Each column indicates a class in the dataset, the first row shows bias-aligned samples and the second row with red border shows bias-conflicting samples.



(a) BFFHQ: *young* (b) BFFHQ: *old* (c) Dogs & Cats: *cat* (d) Dogs & Cats: *dog*

Figure B.2. Illustrative examples of (a)-(b) BFFHQ with the bias attribute of gender and (c)-(d) Dogs & Cats with the bias attribute of color. (a) and (b) respectively indicate the class of young and old in BFFHQ, while (c) and (d) respectively indicate the class of cat and dog in Dogs & Cats. The first three images in each sub-figure show bias-aligned samples and the last one with red border shows bias-conflicting samples.

(a) *Climbing* (b) *Diving* (c) *Fishing* (d) *Racing* (e) *Throwing* (f) *Vaulting*

Figure B.3. Illustrative examples of BAR with the bias attribute of different places. (a)-(f) respectively indicates the classes of six actions. The first three images in each sub-figure show bias-aligned samples and the last one with red border shows bias-conflicting samples.

## B.2. Implementation Details

During training, we set the batch size of 256 for Colored MNIST and Corrupted CIFAR-10, and 64 for BFFHQ, Dogs & Cats and BAR. For all the baseline methods, we use the configuration in their official repositories. For ETF-Debias, we set the learning rate of 1e-3 and weight decay of 1e-5 for Colored MNIST, the learning rate of 1e-2 and weight decay of 2e-4 for Corrupted CIFAR-10, and the learning rate of 1e-4 and weight decay of 1e-6 for all the real-world datasets. The dimensions of learnable features and prime features are set as 100 and 64 for MLP and ResNet-20 respectively. The hyper-parameter $\alpha$ is set as 0.8, and all the evaluated models are trained for 200 epochs.

# C. More Convergence Results of Neural Collapse

In Fig. 2, we display the trajectory of NC metrics when trained with ETF-Debias on the Corrupted CIFAR-10 dataset. Eliminating the obstacle of misled shortcut features, the debiased model exhibits better convergence properties during training, which forms a more symmetric feature space as on balanced datasets. We provide more convergence results of NC metrics on the synthetic biased dataset (Colored MNIST) and the real-world biased dataset (Dogs & Cats) in Fig. C.1 and Fig. C.2. By applying ETF-Debias on various datasets with different types of bias attributes (**blue lines**), we observe not only the better generalization capability on test sets, but also the more collapsed and robust structure of feature spaces (as indicated by the NC metrics), which provides empirical support for the effective guidance towards the intrinsic correlations.

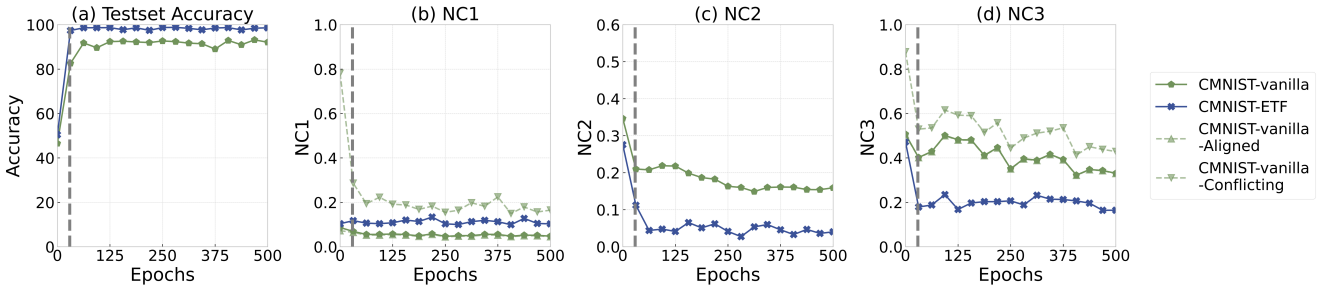



Figure C.1. Comparison of (a) testset accuracy and (b-d) Neural Collapse metrics on Colored MNIST with the bias ratio of 2.0%. All vanilla models are trained with the standard cross-entropy loss for 500 epochs. The postfix *-Aligned* and *-Conflicting* indicate the results of bias-aligned and bias-conflicting samples respectively. The vertical dashed line at the epoch of 60 divides the two stages of training. All models are trained on ResNet-20.

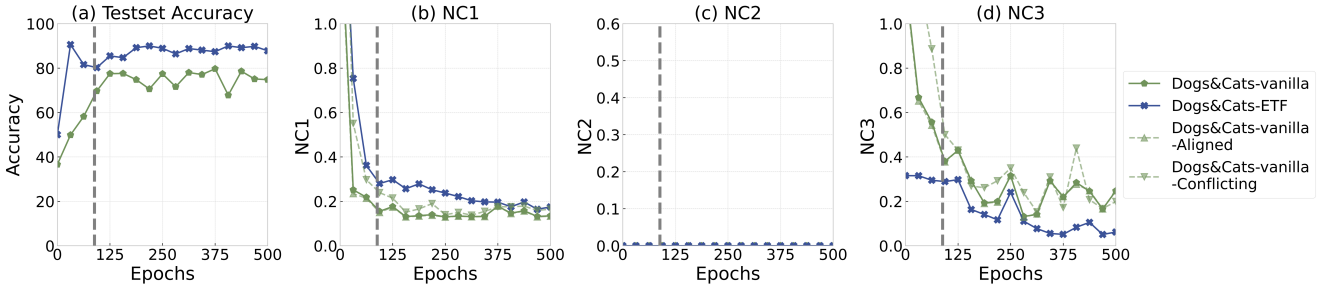



Figure C.2. Comparison of (a) testset accuracy and (b-d) Neural Collapse metrics on Dogs & Cats with the bias ratio of 5.0%. All vanilla models are trained with the standard cross-entropy loss for 500 epochs. The postfix *-Aligned* and *-Conflicting* indicate the results of bias-aligned and bias-conflicting samples respectively. The vertical dashed line at the epoch of 90 divides the two stages of training. Note that the Dogs & Cats dataset is designed for the binary classification task, which means any classifier weights will follow the ETF structure (i.e., NC2 $\rightarrow$ 0), and the detailed theoretical support is provided in [31].

## D. Detailed Results of BAR dataset

Following the previous study [23], we further report the class-wise accuracy on the unbiased test set of BAR. The debiasing performance on BAR with the bias ratio of 1.0% and 5.0% are displayed in Tab. D.1 and Tab. D.2 respectively. The compared baselines are the same as before.

Table D.1. Class-wise accuracy on the unbiased test set of BAR. The ratio of bias-conflicting training sample is 1.0%. Best performances are marked in bold, and the number in brackets indicates the improvement compared to the best result in baselines.

| Action | Climbing | Diving | Fishing | Vaulting | Racing | Throwing | Average |
|---|---|---|---|---|---|---|---|
| Vanilla | 75.46±1.68 | 52.59±4.51 | 75.00±2.78 | 71.15±3.40 | 85.47±0.85 | 48.36±0.81 | 68.00±0.43 |
| LfF [23] | 72.52±8.73 | 67.16±8.96 | 69.44±5.56 | 71.14±3.18 | 79.77±1.98 | 40.84±5.08 | 68.30±0.97 |
| LfF+BE [19] | 82.78±3.86 | 62.96±5.88 | 62.96±4.24 | 82.07±6.79 | 82.62±3.85 | 43.42±2.27 | 71.70±1.33 |
| EnD [30] | 79.85±2.29 | 50.62±3.08 | 73.15±3.21 | 74.51±0.49 | 82.05±0.00 | 49.29±4.88 | 68.25±0.19 |
| SD [42] | 79.49±2.29 | 58.12±1.96 | 71.29±3.21 | 80.67±2.22 | 82.05±0.86 | 34.74±2.15 | 67.73±0.35 |
| DisEnt [18] | 79.85±4.57 | 54.56±0.85 | 75.00±5.56 | 73.67±4.15 | 87.17±2.57 | 44.13±2.15 | 69.30±1.27 |
| Selecmix [13] | 69.60±1.68 | 59.26±9.46 | 79.63±4.24 | 66.95±5.47 | 87.18±2.96 | 56.34±5.64 | 69.83±1.02 |
| **ETF-Debias** | 83.15±1.27 | 63.70±4.86 | 76.85±4.25 | 74.79±5.11 | 87.46±1.98 | 50.23±4.53 | **72.79**±0.21 **(+1.09)** |

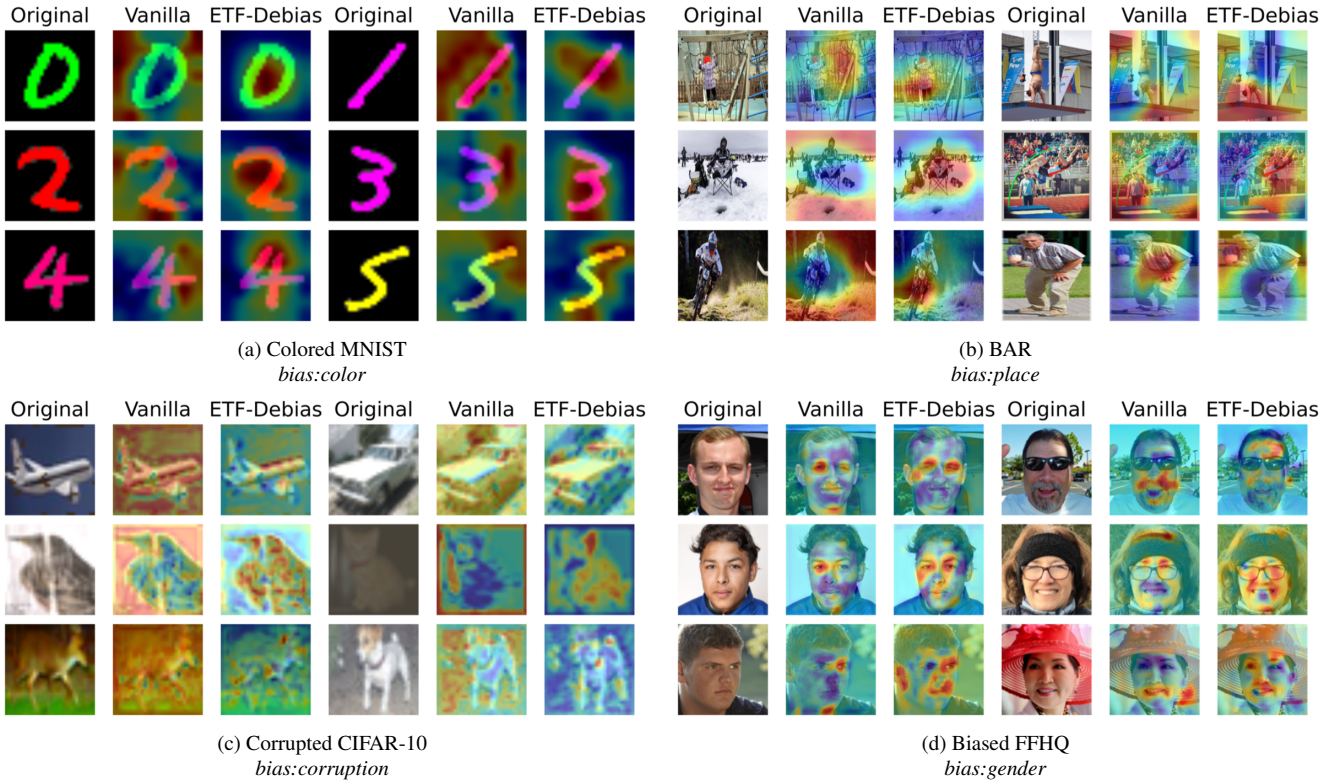
Table D.2. Class-wise accuracy on the unbiased test set of BAR. The ratio of bias-conflicting training sample is 5.0%. Best performances are marked in bold, and the number in brackets indicates the improvement compared to the best result in baselines.

| Action | Climbing | Diving | Fishing | Vaulting | Racing | Throwing | Average |
|---|---|---|---|---|---|---|---|
| Vanilla | 86.44±0.64 | 85.19±0.75 | 79.63±3.20 | 82.63±1.75 | 89.85±0.19 | 52.11±6.13 | 79.34±0.19 |
| LfF [23] | 84.24±2.54 | 84.69±4.93 | 75.92±8.93 | 80.67±2.22 | 89.74±0.86 | 52.58±6.50 | 80.25±1.27 |
| LfF+BE [19] | 86.77±2.92 | 85.96±1.91 | 78.30±5.33 | 82.11±5.76 | 90.16±1.29 | 55.70±1.95 | 82.00±1.24 |
| EnD [30] | 85.71±1.10 | 86.42±1.13 | 84.26±1.61 | 78.15±0.84 | 92.59±1.98 | 46.01±0.81 | 78.86±0.36 |
| SD [42] | 84.25±0.64 | 82.91±4.12 | 82.41±1.60 | 86.27±1.28 | 88.03±0.86 | 50.70±1.41 | 79.10±0.42 |
| Selecmix [13] | 84.98±5.08 | 79.51±2.38 | 81.48±5.78 | 85.43±0.97 | 92.02±0.50 | 49.30±3.73 | 78.79±0.52 |
| DisEnt [18] | 84.98±1.68 | 89.13±5.66 | 77.77±0.00 | 80.39±2.95 | 90.59±1.71 | 48.82±1.63 | 81.19±0.70 |
| **ETF-Debias** | 90.48±1.68 | 87.65±6.39 | 85.18±1.61 | 93.28±2.22 | 92.31±2.26 | 53.05±0.81 | **83.66**±0.21 **(+1.66)** |

## E. More Visualization Results

In Section 5.4, we provide the comparison of visualization results between vanilla training and ETF-Debias on Corrupted CIFAR-10, Dogs & Cats, and BFFHQ. To further demonstrate the rectified attention of debiased models, we supplement more visualization results in Fig. E.1(a)-(d).

By explicitly visualizing the model's attention regions, we observe that ETF-Debias does encourage the model to focus on the intrinsic correlations rather than shortcuts. On synthetic biased datasets like Colored MNIST and Corrupted CIFAR-10, the vanilla models are easily misled by artificial shortcuts and focus on the non-essential parts of images, as shown in Fig. E.1(a) & (c). In constrast, when the shortcut learning is suppressed by prime features, the debiased model trained with ETF-Debias will pay attention to the digits and objects themselves and make unbiased classification. On real-world biased datasets with diverse bias attributes, the vanilla models tend to focus on the background or the distinctive part of the bias attributes. For example, the vanilla models rely on the distinctive part of gender such as the beard of a male or the accessories of a female in Fig. E.1(d). The pursuit of shortcut correlations has also been prevented with ETF-Debias, which guides the debiased model to focus more on facial features and predict the target attribute.




(a) Colored MNIST
*bias:color*

(b) BAR
*bias:place*

(c) Corrupted CIFAR-10
*bias:corruption*

(d) Biased FFHQ
*bias:gender*

Figure E.1. More comparison of visualization results on bias-conflicting samples. We display the result of CAM [44] on (a) Colored MNIST, with the bias attribute as the color of digits, (b) BAR, with the bias attribute as the background of images, (c) Corrupted CIFAR-10, with the bias attribute as different types of corruptions on the entire image, (d) Biased FFHQ, with the bias attribute as gender. The original images in (a)-(c) represent the first six classes in each dataset. The original images in the first column of (d) are from the class of *young* in BFFHQ, while the original images in the fourth column are from the class *old*. All models are trained on ResNet-20 to obtain the CAM results.