BCE VS. CE IN DEEP FEATURE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

When training classification models, it expects that the leaned features are compact within classes, and can well separate different classes. As a dominant loss function to train classification models, the minimization of CE (Cross-entropy) loss can maximize the compactness and distinctiveness, i.e., reaching neural collapse. The recently published works show that BCE (Binary CE) loss performs also well in multi-class tasks. In this paper, we compare BCE and CE in the context of deep feature learning. For the first time, we prove that BCE can also maximize the intra-class compactness and inter-class distinctiveness when reaching its minimum, i.e., leading to neural collapse. We point out that CE measures the relative values of decision scores in the model training, implicitly enhancing the feature properties by classifying samples one-by-one. In contrast, BCE measures the absolute values of decision scores and adjust the positive/negative decision scores across all samples to uniform high/low levels. Meanwhile, the classifier bias in BCE presents a substantial constraint on the samples' decision scores. Thereby, BCE explicitly enhances the feature properties in the training. The experimental results are aligned with above analysis, and show that BCE consistently and significantly improve the classification performance and leads to better compactness and distinctiveness among sample features.

1 INTRODUCTION

Cross-entropy (CE) loss is the most commonly used loss function for classifications and feature learning. In a multi-class classification with K categories, for any sample $\boldsymbol{X}^{(k)}$ from category k, a model $\mathcal{M}$ first extracts its feature $\boldsymbol{h}^{(k)} = \mathcal{M}(\boldsymbol{X}^{(k)}) \in \mathbb{R}^d$, which is output from the penultimate hidden layer in deep model. Then a linear classifier with weight $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]^T \in \mathbb{R}^{K\times d}$ and bias $\boldsymbol{b} = [b_1, \cdots, b_K]^T \in \mathbb{R}^K$ transforms the feature into K logits/decision scores, $\{\boldsymbol{w}_j^T \boldsymbol{h}^{(k)} - b_j\}_{j=1}^K$, which are finally converted into predicted probabilities by Softmax, and computed the loss using cross-entropy,

$$\mathcal{L}_{\text{ce}}\big(\boldsymbol{z}^{(k)}\big) = -\boldsymbol{y}_k^T \cdot \log\Big(\text{Softmax}\big(\boldsymbol{z}^{(k)}\big)\Big) = \log\left(1 + \sum_{\substack{\ell=1\\ \ell\neq k}}^{K} \frac{\mathrm{e}^{\boldsymbol{w}_\ell^T \boldsymbol{h}^{(k)} - b_\ell}}{\mathrm{e}^{\boldsymbol{w}_k^T \boldsymbol{h}^{(k)} - b_k}}\right), \tag{1}$$

where $\boldsymbol{z}^{(k)} = \boldsymbol{W}\boldsymbol{h}^{(k)} - \boldsymbol{b}$ and $\boldsymbol{y}_k$ is the one-hot label, i.e., the vector with one only in the kth entry.

In the multi-class classification, binary CE (BCE) loss is deduced by decomposing the task into K binary tasks and predicting whether the sample $\boldsymbol{X}^{(k)}$ belongs to the jth category, for $\forall j \in [K]$,

$$\begin{aligned}\mathcal{L}_{\text{bce}}\big(\boldsymbol{z}^{(k)}\big) &= -\,\boldsymbol{y}_k^T \cdot \log\Big(\text{Sigmoid}\big(\boldsymbol{z}^{(k)}\big)\Big) - (\boldsymbol{1} - \boldsymbol{y}_k)^T \cdot \log\Big(\boldsymbol{1} - \text{Sigmoid}\big(\boldsymbol{z}^{(k)}\big)\Big) \\ &= \log\Big(1 + \mathrm{e}^{-\boldsymbol{w}_k^T \boldsymbol{h}^{(k)} + b_k}\Big) + \sum_{\substack{j=1\\ j\neq k}}^{K} \log\Big(1 + \mathrm{e}^{\boldsymbol{w}_j^T \boldsymbol{h}^{(k)} - b_j}\Big),\end{aligned} \tag{2}$$

which has been widely used in the multi-label classification (Kobayashi, 2023) and attracted increasing attentions in the multi-class classification (Beyer et al., 2020; Wightman et al., 2021; Touvron et al., 2022; Fang et al., 2023; Wen et al., 2022; Zhou et al., 2023).

The pre-trained classification models can be used as feature extractors for downstream tasks that request well intra-class compactness and inter-class distinctiveness across the sample features, such as person re-identification (He et al., 2021), object tracking (Cai et al., 2023), image segmentation (Guo et al., 2022), and facial recognition (Wen et al., 2022), etc. For CE, a remarkable theoretical result is that when it reaches its minimum, both the compactness and distinctiveness on the training samples will be maximized, which refers to neural collapse found by Papyan et al. (2020). Neural collapse gives peace of mind in training classification models by using CE, and it was extended to the losses satisfying contrastive property by Zhu et al. (2021) and Zhou et al. (2022), including CE, focal loss, and label smoothing loss. However, though BCE in Eq. (2) is a combination of multiple CE, it does not satisfy the contrastive property due to its classifier bias, and whether BCE can lead to neural collapse, or not, is not answered.

Besides that, in the practical training of classification models, the classifier vectors $\{\boldsymbol{w}_k\}_{k=1}^K$ play the role of proxy for each category (Wen et al., 2022). Intuitively, when the distances between the sample features and their class proxy are closer, or the *positive* decision scores between them are larger, it usually leads to better intra-class compactness. Similarly, when the distances between sample features and the proxy of different classes are farther, or the *negative* decision scores between them are smaller, it could results in better inter-class distinctiveness. However, according to Eq. (1), CE measures the *relative* value between the exponential positive and negative decision scores using Softmax and logarithmic functions, to pursue that the positive decision score is greater than all its negative ones for each sample, making it unable to explicitly and directly enhance the intra-class compactness and inter-class distinctiveness across samples. In contrast, BCE in Eq. (2) respectively measures the *absolute* values of the exponential positive decision score and the exponential negative ones using Sigmoid and logarithmic functions, which makes it could explicitly and directly enhance the compactness and distinctiveness of features in the training.

In this paper, we compare BCE and CE in deep feature learning. We primarily address two questions: **Q1**. Can BCE result in the neural collapse, i.e., maximizing the compactness and distinctiveness in theoretical? **Q2**. In practical training of classification models, does BCE perform better than CE in terms of the feature compactness and distinctiveness? Our contributions are summarized as follows.

(1) We provide the first theoretical proof that BCE can also lead to the neural collapse, i.e., maximizing the compactness and distinctiveness, even when the loss does not satisfy the contrastive property, broadening the range of losses that can lead to neural collapse.

(2) We find that BCE performs better than CE in enhancement of intra-class compactness and inter-class distinctiveness across sample features, and, BCE can explicitly enhance the feature properties, while CE only implicitly enhance them.

(3) We point out that when training models with BCE, the classifier bias plays a substantial role in enhancing the feature properties, while in the training with CE, it almost does not work.

(4) We conduct extensive experiments, and find that, compared to CE, BCE can more quickly lead to the neural collapse on the training dataset and achieves better feature compactness and distinctiveness, resulting in higher classification performance on the test dataset.

2 Related works

2.1 CE vs. BCE

The CE loss is the most popular loss used in the multi-class classification and feature learning, which has been evolved into many variants in different scenarios, such as focal loss (Lin et al., 2017), label smoothing loss (Szegedy et al., 2016), normalized Softmax loss (Wang et al., 2017), and marginal Softmax loss (Liu et al., 2016), etc. The classification models are often applied to the downstream tasks, such as image segmentation (Guo et al., 2022), person re-identification (He et al., 2021), object tracking (Cai et al., 2023), etc., which request well intra-class compactness and inter-class distinctiveness among the sample features. In the multi-class classification task, the BCE loss can be deduced by decomposing the task into K binary tasks and adding the K binary cross-entropy losses, which has been widely applied in the multi-label classification (Kobayashi, 2023).

The CE and BCE losses are expected to train the models to fit the sample distribution in the multi-class and multi-label classifications. When Wightman et al. (2021) applied BCE to the training of ResNets for a multi-class task, they considered that this loss is consistent with Mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019) augmentations, which mix multiple objects from different samples into one sample. DeiT III (Touvron et al., 2022) adopted this approach and achieved a significant improvement in the multi-class task on ImageNet-1K by using the BCE loss. Currently, though the CE loss dominates the training of multi-class and feature learning models, the BCE loss is also gaining more attention and is increasingly being applied in these fields (Fang et al., 2023; Wang et al., 2023; Xu et al., 2023; Mehta & Rastegari, 2023; Chun, 2024; Hao et al., 2024). However, none of these works reveals the essential advantages of BCE over CE.

2.2 NEURAL COLLAPSE

The neural collapse was first found by Papyan et al. (2020), which refers to four properties about the sample features $\{\boldsymbol{h}_i^{(k)}\}$ and the classifier vectors $\{\boldsymbol{w}_k\}$ at the terminal phase of training.

- **NC1**, within-class variability collapse. Each feature $\boldsymbol{h}_i^{(k)}$ collapse to its class center $\bar{\boldsymbol{h}}^{(k)} = \frac{1}{n_k}\sum_{i'=1}^{n_k}\boldsymbol{h}_{i'}^{(k)}$, indicating the *maximal intra-class compactness*
- **NC2**, convergence to simplex equiangular tight frame. The set of class centers $\{\bar{\boldsymbol{h}}^{(k)}\}_{k=1}^{K}$ form a simplex equiangular tight frame (ETF), with equal and maximized cosine distance between every pair of feature means, i.e., the *maximal inter-class distinctiveness*.
- **NC3**, convergence to self-duality. The class center $\bar{\boldsymbol{h}}^{(k)}$ is ideally aligned with the classifier vector $\boldsymbol{w}_k, \forall k \in [K]$.
- **NC4**, simplification to nearest class center. The classifier is equivalent to a nearest class center decision.

The current works about neural collapse (Kothapalli, 2023) are focused on the CE loss (Lu & Steinerberger, 2022; Graf et al., 2021; Zhu et al., 2021) and mean squared error (MSE) loss (Han et al., 2022; Tirer & Bruna, 2022). It has been proved that the models will fall to the neural collapse when the loss reaches its minimum. A comprehensive analysis (Zhou et al., 2022) for various losses, including CE loss, focal loss (Lin et al., 2017), and label smoothing loss (Szegedy et al., 2016), shows that these losses perform equally as any global minimum point of the loss satisfies the neural collapse. The neural collapse has also been investigated in the imbalanced datasets (Fang et al., 2021; Yang et al., 2022; Wang et al., 2024), out-of-distribution data (Ammar et al., 2024), and models with fixed classifiers (Yang et al., 2022; Kim & Kim, 2024). All these studies are conducted using CE or MSE losses; and whether BCE can lead to neural collapse remains unexplored.

3 MAIN RESULTS

In this section, we first theoretically prove that BCE can maximize the compactness and distinctiveness when reaching its minimums (**Q1**). Then, through in-depth analyzing the decision scores in the training with BCE and CE, we explain that BCE can better enhance the compactness and distinctiveness of sample features in practical training (**Q2**).

3.1 PRELIMINARY

Let $\mathcal{D} = \bigcup_{k=1}^{K}\bigcup_{i=1}^{n_k}\left\{\boldsymbol{X}_i^{(k)}\right\}$ be a sample set, where $\boldsymbol{X}_i^{(k)}$ denotes the ith sample of category k, n_k is the number of samples in this category, and $\boldsymbol{h}_i^{(k)} = \mathcal{M}(\boldsymbol{X}_i^{(k)})$. In classification tasks, a linear classifier with vectors $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]^T \in \mathbb{R}^{K\times d}$ and bias $\boldsymbol{b} = [b_1, \cdots, b_K]^T \in \mathbb{R}^K$ predicts the category for each sample according to its feature. For the well predication results, the CE or BCE loss is applied to tune the parameters of the model $\mathcal{M}$ and classifier.

Following the previous works (Fang et al., 2021; Lu & Steinerberger, 2022; Graf et al., 2021; Zhu et al., 2021; Han et al., 2022; Tirer & Bruna, 2022) for neural collapse, we compare CE and BCE in training of unconstrained model or layer-peeled model in this paper, i.e, treating the features $\bigcup_{k=1}^{K}\left\{\boldsymbol{h}_i^{(k)}\right\}_{i=1}^{n_k}$, classifier vectors $\{\boldsymbol{w}_k\}_{k=1}^{K}$, and classifier bias $\{b_k\}_{k=1}^{K}$ as free variables, without considering the sophisticated structure or the parameters of the model $\mathcal{M}$. Then, taking the regularization terms on the variables, the CE or BCE loss in the training is

$$f_\mu(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) := \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}_\mu\big(\boldsymbol{W}\boldsymbol{h}_i^{(k)} - \boldsymbol{b}\big) + \frac{\lambda_{\boldsymbol{W}}}{2}\left\|\boldsymbol{W}\right\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\left\|\boldsymbol{H}\right\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\left\|\boldsymbol{b}\right\|_2^2, \quad (3)$$

where $\mathcal{L}_\mu$ is presented in Eqs. (1-2), $\mu \in \{\text{ce}, \text{bce}\}$,

$$\boldsymbol{W} = \left[\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K\right]^T \in \mathbb{R}^{K \times d}, \tag{4}$$

$$\boldsymbol{H} = \left[\boldsymbol{h}_1^{(1)}, \boldsymbol{h}_2^{(1)}, \cdots, \boldsymbol{h}_{n_1}^{(1)}, \cdots, \boldsymbol{h}_1^{(K)}, \boldsymbol{h}_2^{(K)}, \cdots, \boldsymbol{h}_{n_K}^{(K)}\right] \in \mathbb{R}^{d \times (\sum_{k=1}^K n_k)}, \tag{5}$$

$$\boldsymbol{b} = [b_1, b_2, \cdots, b_K]^T \in \mathbb{R}^K, \tag{6}$$

and $\lambda_{\boldsymbol{W}}, \lambda_{\boldsymbol{H}} > 0, \lambda_{\boldsymbol{b}} \geq 0$ are weight decay parameters for the regularization terms.

3.2 Neural collapse with CE and BCE losses

On the balanced dataset, i.e., $n = n_k, \forall k \in [K]$, Zhu et al. (2021) proved that the CE loss can result in neural collapse, and in Theorem 1, Zhou et al. (2022) extended the proof to the losses satisfying the contrastive property (see Definition S-1 in supplementary), such as focal loss and label smoothing loss. Though BCE loss is a combination of CE loss, it fails to satisfy the contrastive property, as that the classifier bias parameters present substantial constraint within its components. Despite that, we find that the BCE loss can also result in the neural collapse, i.e., Theorem 2. The primary difference between BCE and CE losses lies in the bias parameter $\boldsymbol{b}$ of their classifiers.

Theorem 1 *(Zhou et al., 2022) Assume that the feature dimension d is larger than the category number K, i.e., $d \geq K-1$, and $\mathcal{L}_\mu$ is satisfying the contrastive property. Then any global minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ of $f_\mu(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ defined using $\mathcal{L}_\mu$ with Eq. (3) obeys the following properties,*

$$\|\boldsymbol{w}^\star\| = \|\boldsymbol{w}_1^\star\| = \|\boldsymbol{w}_2^\star\| = \cdots = \|\boldsymbol{w}_K^\star\|, \tag{7}$$

$$\boldsymbol{h}_i^{(k)\star} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}} \boldsymbol{w}_k^\star, \ \forall\, k \in [K],\ i \in [n], \tag{8}$$

$$\tilde{\boldsymbol{h}}_i^\star := \frac{1}{K}\sum_{k=1}^K \boldsymbol{h}_i^{(k)\star} = \boldsymbol{0}, \ \forall\, i \in [n], \tag{9}$$

$$\boldsymbol{b}^\star = b^\star \boldsymbol{1}_K, \tag{10}$$

where either $b^\star = 0$ or $\lambda_{\boldsymbol{b}} = 0$. The matrix $\boldsymbol{W}^{\star T}$ forms a K-simplex ETF in the sense that

$$\frac{1}{\|\boldsymbol{w}^\star\|_2^2}\boldsymbol{W}^{\star T}\boldsymbol{W}^\star = \frac{K}{K-1}\Big(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^T\Big), \tag{11}$$

where $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$ denotes the identity matrix, and $\boldsymbol{1}_K \in \mathbb{R}^K$ denotes the all ones vector. ∎

Theorem 2 *Assume that the feature dimension d is larger than the category number K, i.e., $d \geq K-1$. Then any global minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ of $f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ defined using $\mathcal{L}_{\text{bce}}$ with Eq. (3) obeys the properties (7) - (11), where $b^\star$ is the solution of equation*

$$0 = -\frac{K-1}{K\left(1+\exp\left(b + \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} + \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b\right)\right)} + \lambda_{\boldsymbol{b}} b, \tag{12}$$

and $\rho = \|\boldsymbol{W}^\star\|_F^2$ is the squared Frobenius norm of $\boldsymbol{W}^\star$.

Proof *The detailed proof is presented in the supplementary, i.e., Theorem S-4, which similar to that of Zhu et al. (2021); Zhou et al. (2022); Lu & Steinerberger (2022), studies lower bounds for the BCE loss in Eq. (3) and finds the conditions for achieving the lower bounds.* ∎

Theorem 2 significantly broadens the range of losses that can lead to neural collapse, i.e., the contrastive property (Zhou et al., 2022) is not necessarily satisfied.

The decision scores. According to Theorems 1 and 2, when training a classification model with CE or BCE losses, if the loss reaches its minimum and results in the neural collapse, the sample feature $\boldsymbol{h}_i^{(k)}$ will converge to its class center $\bar{\boldsymbol{h}}^{(k)} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{h}_i^{(k)}$, indicating the maximum intra-class compactness. Furthermore, the class center $\bar{\boldsymbol{h}}^{(k)}$ becomes a multiple of the corresponding classifier vector $\boldsymbol{w}_k$, and the K classifier vectors $\{\boldsymbol{w}_k\}_{k=1}^K$ will form an ETF, indicating the maximum inter-class distinctiveness. In addition, the positive and negative decision scores without the biases of all samples will respectively converge to fixed values, i.e., for $\forall j \neq k \in [K],\ i \in [n]$,

$$s_{\text{pos}}^{(kk,i)} = \boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} \rightarrow \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}} \frac{\rho}{K} \quad \text{and} \quad s_{\text{neg}}^{(jk,i)} = \boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} \rightarrow -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}} \frac{\rho}{K(K-1)}. \tag{13}$$

The classifier bias. Comparing Theorems 1 and 2, one can find that the primary difference between CE and BCE losses lies in their classifier bias parameter. According to Theorem 1, when $\lambda_{\boldsymbol{b}} > 0$, the minimum point of CE loss satisfies $\boldsymbol{b} = \boldsymbol{0}$, separating the final positive and negative scores in Eq. (13); when $\lambda_{\boldsymbol{b}} = 0$, any point that satisfies properties (7) - (11) and $\boldsymbol{b} = b^\star \boldsymbol{1}$ is a minimum point of CE loss, which implies that the minimum points of CE loss form a ridge line in term of $\boldsymbol{b}$. In contrast, the classifier bias $\boldsymbol{b}$ at the minimum points of BCE loss satisfy Eq. (12) whenever $\lambda_{\boldsymbol{b}} = 0$ or not. According to Lemma 5 in the supplementary, Eq. (12) has only one solution, indicating that the BCE loss has only one minimum point in term of $\boldsymbol{b}$. This optimal classifier bias $\boldsymbol{b} = b^\star \boldsymbol{1}$ will separate the positive and negative decision scores if it satisfies the Eq. (165) (see Lemma 6 in supplementary for details). Both of the optimal points of CE and BCE losses are associated with the unified classifier bias $\boldsymbol{b} = b^\star \boldsymbol{1}$, which is aligned with the unified bias integrated loss designed by Wen et al. (2022) and Zhou et al. (2023) for facial recognition.

3.3 The decision scores in training with BCE and CE

Though the intra-class compactness and inter-class distinctiveness of sample features can be theoretically maximized by both CE and BCE, the two losses perform very different in practical training of classification models. We here compare the decision scores in the model training by using BCE and CE, to explain their difference in enhancing the feature properties.

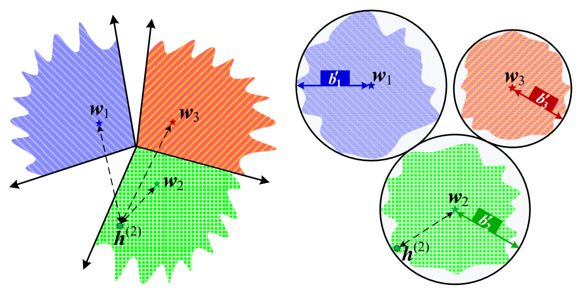
A geometric comparison for CE and BCE. In practical training with CE or BCE, to minimize the loss, it is desirable for their exponential function variables to be as small as possible, and less than zero at least. For CE in Eq. (1), it is desirable that, for $\forall \ell \neq k \in [K]$,

$$\underbrace{\boldsymbol{w}_k \boldsymbol{h}^{(k)} - b_k}_{\text{positive decision score}} > \underbrace{\boldsymbol{w}_\ell \boldsymbol{h}^{(k)} - b_\ell}_{\text{negative decision score}}, \tag{14}$$

while, for BCE in Eq. (2), it is desirable that, for $\forall j \neq k \in [K]$,

$$\boldsymbol{w}_k \boldsymbol{h}^{(k)} - b_k > 0, \tag{15}$$

$$\boldsymbol{w}_j \boldsymbol{h}^{(k)} - b_j < 0. \tag{16}$$



(a) CE loss (b) BCE loss

Figure 1: The feature distributions of CE and BCE losses in the distance space. We apply the blue, red, and green shading to indicate the feature space of three categories, respectively. The pentagrams represent their classifier vectors, and the solid dot represents a general feature vector $\boldsymbol{h}^{(2)}$ in the second category. Since the distance between two vectors is inversely proportional to their similarity/inner product, CE loss requires the distance from the feature to its classifier vector to be less than the distance to other classifier vectors, while BCE loss requires the distance to be less than its corresponding bias. Small b_k' implies large b_k in Eqs. (15-16).

As Fig. 1 shows, we apply the distance of vectors to reflect their inner product or similarity in the metric space. Without considering the bias $\boldsymbol{b}$, the CE loss push feature vector $\boldsymbol{h}^{(k)}$ closer to its classifier vector $\boldsymbol{w}_k$ compared to others $\{\boldsymbol{w}_\ell\}_{\ell \neq k}$, implying a *unbounded* feature space for each category and bad intra-class compactness. In addition, any two unbounded feature spaces introduced by CE could share the same decision boundary, indicating bad inter-class distinctiveness. In the training with CE, the bias b_k acts as a compensation to adjust the distance/decision score between the sample features and the classifier vector, introducing indirect constraint across sample features by Eq. (14). This constraint will vanish if $b_k = b_\ell$ for $\forall k, \ell \in [K]$, which could be reached at the minimum points of CE according to Theorem 1. Overall, in the training of classification models, CE does not require absolutely large positive decision scores or absolutely small negative ones, but only requires the positive one to be relatively greater than the negative ones for each sample, thereby implicitly enhancing the features' properties by correctly classifying samples one-by-one.

In contrast, for BCE, Eq. (15) requires the feature $\boldsymbol{h}^{(k)}$ to fall within a *closed* hypersphere centered at its classifier vector $\boldsymbol{w}_k$ with a "radius" of b_k, while Eq. (16) requires that any two hypersphere do not intersect, indicating well intra-class compactness and inter-class distinctiveness. In other words, BCE presents explicitly constraint across-samples in the training. While Eq. (15) requires the positive decision scores of all samples are uniformly larger than threshold $t = 0$, Eq. (16) requires the negative ones of all samples are uniformly smaller than the unified threshold, i.e.,

$$\min \bigcup_{k=1}^{K} \bigcup_{i=1}^{n} \{\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} - b_k\} > t \geq \max \bigcup_{k=1}^{K} \bigcup_{i=1}^{n} \{\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\}_{\substack{j=1 \\ j\neq k}}^{K}, \tag{17}$$

while the unified threshold t might be not exactly zero in practice. As Eq. (17), BCE expects the positive decision scores to be uniformly high and the negative ones to be uniformly low, which could result in better compactness and distinctiveness than that introduced by Eq. (14). For the kth category, the bias b_k would be absorbed into the threshold. In contrary, the bias b_k explicitly reflect the intra-class compactness of its corresponding category and the inter-class distinctiveness between its category with other different categories. Therefore, BCE can explicitly enhance the compactness and distinctiveness across sample features by learning well biases.

The decision scores in practical training. In deep learning, gradient descent and back propagation are the most commonly used techniques for the model training. We here analyze the gradients in terms of the positive decision score $(\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k)$ and negative one $(\boldsymbol{w}_j \boldsymbol{h}_i^{(k)} - b_j)$ for any sample $\boldsymbol{X}_i^{(k)}$ from category k with $\forall j \neq k$

$$\frac{\partial f_{\text{ce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial(\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k)} = \frac{\mathrm{e}^{\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}_i^{(k)} - b_\ell}} - 1, \quad \frac{\partial f_{\text{ce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial(\boldsymbol{w}_j \boldsymbol{h}_i^{(k)} - b_j)} = \frac{\mathrm{e}^{\boldsymbol{w}_j \boldsymbol{h}_i^{(k)} - b_j}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}_i^{(k)} - b_\ell}}, \text{ and} \tag{18}$$

$$\frac{\partial f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial(\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k)} = \frac{1}{1 + \mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} + b_k}} - 1, \quad \frac{\partial f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial(\boldsymbol{w}_j \boldsymbol{h}_i^{(k)} - b_j)} = \frac{1}{1 + \mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}_i^{(k)} + b_j}}. \tag{19}$$

According to Eq. (18), for any two samples $\boldsymbol{X}_i^{(k)}, \boldsymbol{X}_{i'}^{(k)}$ from the same category k with diverse initial positive scores, if their predicted probabilities are equal, i.e., $\frac{\mathrm{e}^{\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}_i^{(k)} - b_\ell}} = \frac{\mathrm{e}^{\boldsymbol{w}_k \boldsymbol{h}_{i'}^{(k)} - b_k}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}_{i'}^{(k)} - b_\ell}}$, which is somewhat likely to occur during the practical training, then their positive scores will experience the same update of amplitude during back propagation. Consequently, it will be difficult to update the positive scores to the uniformly high level, impeding the enhancement of intra-class compactness within the same category. A similar phenomenon can also occur with the negative decision scores, resulting in unsatisfactory inter-class distinctiveness in the training with CE loss.

In contrast, according to Eq. (19), during training with BCE loss, the large positive decision scores $(\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k)$ were updated for the small amplitude $1 - \frac{1}{1+\exp(-\boldsymbol{w}_k \boldsymbol{h}_i^{(k)} + b_k)}$, while the small ones were updated for the large update amplitude, facilitating a more rapid adjustment of positive scores across different samples to a uniform high level, to enhance the intra-class compactness of sample features. For the negative decision scores, similarly, the large/small score will be updated with large/samll amplitudes in the training with BCE loss to adjust them to a uniform low level, enhancing the inter-class distinctiveness.

The classifier bias in practice. During the model training, the classifier bias is also updated through the gradient descent, and the positive and negative decision scores are constrained by approaching the stable point of the bias. For CE, the gradient of bias b_k is

$$\frac{\partial f_{\text{ce}}}{\partial b_k} = \frac{1}{nK} \left(n - \sum_{j=1}^{K} \sum_{i=1}^{n} \frac{\mathrm{e}^{\boldsymbol{w}_k \boldsymbol{h}_i^{(j)} - b_k}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}_i^{(j)} - b_\ell}} \right) + \lambda_{\boldsymbol{b}} b_k \rightarrow \lambda_{\boldsymbol{b}} b. \tag{20}$$

As approaching the stable point of the bias, i.e., the points satisfying $\frac{\partial f_{\text{ce}}}{\partial b_k} = 0$, Eq. (20) presents constraint on the relative value of the exponential decision scores. This constraint will vanish as reaching the minimum of CE, and the bias gradient $\frac{\partial f_{\text{ce}}}{\partial b_k}$ approaches $\lambda_{\boldsymbol{b}} b$, according to Eq. (13). At the minimum points, the update amplitude of bias is $\eta \lambda_{\boldsymbol{b}} b$, where η denotes the learning rate. If $\lambda_{\boldsymbol{b}} = 0$, the update is zero, and the final bias can locate at any point on the ridge line $\boldsymbol{b} = b\mathbf{1}$, where b is depended on some other factors, such as the bias initial value, but not the relationship between the bias and the decision scores. If $\lambda_{\boldsymbol{b}} > 0$, one can concluded $b = 0$; however, in practice, this theoretical value might be not reached due to that $\eta\lambda_{\boldsymbol{b}}$ will be very small at the terminal phase of practical training. The above analysis implies that the classifier bias, in the training with CE, cannot provide consistent and explicit constraints on the decision scores, and thus almost does not affect the final features' properties.

In contrast, for BCE, the gradient of bias b_k is

$$\frac{\partial f_{\text{bce}}}{\partial b_k} = \frac{1}{nK}\left(n - \sum_{j=1}^{K}\sum_{i=1}^{n}\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k\boldsymbol{h}_i^{(j)}+b_k}}\right) + \lambda_{\boldsymbol{b}} b_k \rightarrow \text{RHS of Eq. (12)}, \tag{21}$$

which presents clear constraint on the absolute value of the exponential decision scores for the all samples. The constraint evolve into Eq. (12) when BCE reaches its minimum points. Therefore, as approaching the stable point, the classifier bias consistently and explicitly constrain the decision scores, regardless $\lambda_{\boldsymbol{b}} = 0$ or not, and it will separate the final positive and negative decision scores if Eq. (165) holds. In other words, the classifier bias in BCE plays a substantial role in enhancing the final features' properties.

4 Experiments

To compare CE and BCE in deep feature learning, we train deep classification models, ResNet18 (He et al., 2016), ResNet50 (He et al., 2016), and DenseNet121 (Huang et al., 2017), using the two losses respectively, on three popular datasets, including MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky et al., 2009), and CIFAR100 (Krizhevsky et al., 2009). We train the models using SGD and AdamW for 100 epochs with batch size of 128. The initial learning rate is set to 0.01 and 0.001 for SGD and AdamW, which is respectively decayed in "step" and "cosine" schedulers.

4.1 Maximizing compactness and distinctiveness by BCE and CE

We first experimentally illustrate that both BCE and CE can maximize the intra-class compactness and inter-class distinctiveness among sample features, i.e., resulting in neural collapse (NC). Similar to (Zhu et al., 2021; Zhou et al., 2022), we do not apply any data augmentation in the experiments of NC, and adopt the metrics, $\mathcal{NC}_1, \mathcal{NC}_2$, and $\mathcal{NC}_3$ (see supplementary for their definitions), to measure the properties of NC1, NC2, and NC3. The lower metrics reflect the better NC properties.

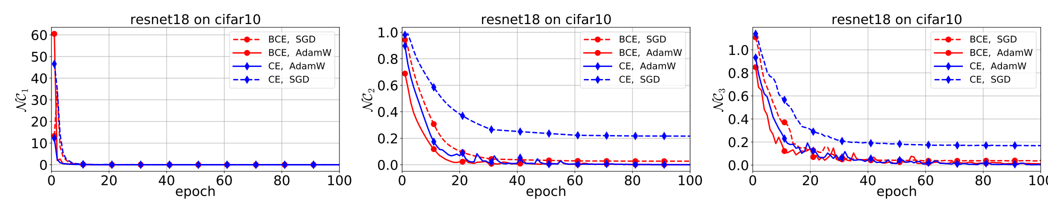


Figure 2: NC metrics of ResNet18 trained on CIFAR10 with CE and BCE using SGD and AdamW, respectively. The NC metrics of CE and BCE approach zero at the terminal phase of training, while the NC metrics of BCE decrease faster than that of CE in the first 20 epochs.

In the training, we set $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$, and no weight decay is applied on the other parameters of the model $\mathcal{M}$. Fig. 2 shows the NC results of ResNet18 trained by CE and BCE with two optimizers on CIFAR10, and the other results are presented in the supplementary. In the figure, the red curves with dot markers exhibit the evolution of the metrics $\mathcal{NC}_1, \mathcal{NC}_2, \mathcal{NC}_3$ of BCE, and the blue curves with diamond markers exhibit that of CE. All the three NC metrics consistently approach zero in the training with different losses and optimizers, which matches the conclusions of Theorem 1 and 2. In the initial training stage (the first 20 epochs), the NC metrics of BCE usually decrease faster than that of CE, implying that BCE is easier to result in NC.

The final classifier bias and decision scores. As reaching NC and maximizing the feature compactness and distinctiveness, the final classifier biases and the positive/negative decision scores will converge to fixed values. We compute their means and standard deviations for the different models on the training set. Table 1 shows the results on CIFAR10. Except for that of ResNet50 and DenseNet121 trained by SGD, the standard deviations of the final decision scores and classifier biases are very small, and less than 0.3 when the models are trained using AdamW, indicating that the diverse classifier biases are almost equal, so are all the final positive/negative decision scores. As $\lambda_{\boldsymbol{b}} > 0$, the final classifier biases are near zero ($\hat{b} \approx 0$) in the CE-trained models, while, in the BCE-trained models, the biases make that the function ($\alpha(b)$ in Table 1) on the RHS of Eq. (12) almost degenerate to zero, i.e., $\alpha(\hat{b}) \approx 0$, aligning with our analysis.

Table 1: The means and standard deviations of positive/negative decision scores (s_{pos}, s_{neg}, without bias) and the classifier biases ($\hat{b}$) of the models trained by CE and BCE on CIFAR10 with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$. The score values are computed on the training set, and $\mathcal{A}$ denotes the classification accuracy on the training set.

$\mathcal{M}$		CIFAR10			
		SGD		AdamW	
		CE	BCE	CE	BCE
ResNet18	s_{pos}	5.71 ± 0.23	6.56 ± 0.20	5.64 ± 0.06	7.50 ± 0.05
	s_{neg}	-0.64 ± 0.36	-3.46 ± 0.19	-0.63 ± 0.01	-2.36 ± 0.02
	$\hat{b}$	-0.01 ± 0.04	2.26 ± 0.07	-0.00 ± 0.00	3.29 ± 0.01
	$\alpha(\hat{b})$	—	-0.03	—	-0.01
	$\mathcal{A}$	99.99	100.0	100.0	100.0
ResNet50	s_{pos}	5.74 ± 8.21	6.56 ± 4.39	5.64 ± 0.11	7.44 ± 0.28
	s_{neg}	-0.64 ± 14.1	-3.57 ± 7.01	-0.63 ± 0.02	-2.45 ± 0.22
	$\hat{b}$	0.00 ± 0.14	2.40 ± 0.15	-0.00 ± 0.01	3.21 ± 0.03
	$\alpha(\hat{b})$	—	-0.02	—	-0.01
	$\mathcal{A}$	99.61	99.65	99.99	100.0
DenseNet121	s_{pos}	5.72 ± 1.72	6.24 ± 0.84	5.62 ± 0.29	7.67 ± 0.12
	s_{neg}	-0.63 ± 0.87	-3.62 ± 1.63	-0.62 ± 0.03	-2.17 ± 0.06
	$\hat{b}$	0.00 ± 0.04	2.09 ± 0.12	0.00 ± 0.01	3.46 ± 0.02
	$\alpha(\hat{b})$	—	-0.03	—	-0.01
	$\mathcal{A}$	99.40	99.72	99.87	100.0

The failures in NC. According to the results in Table 1, one can find that NC might be not caused by the ResNet50 and DenseNet121 trained by SGD. Though the classification accuracy ($\mathcal{A}$) of the two models are higher than 99.00% (almost 100%), their positive and negative decision scores have large standard deviations, implying that the decision scores have not converged to the fixed values and conflict with the results (Eq. (13)) of neural collapse. It still requires a long time training to reach the neural collapse, after zero classification error. More discussions can be found in the supplementary.

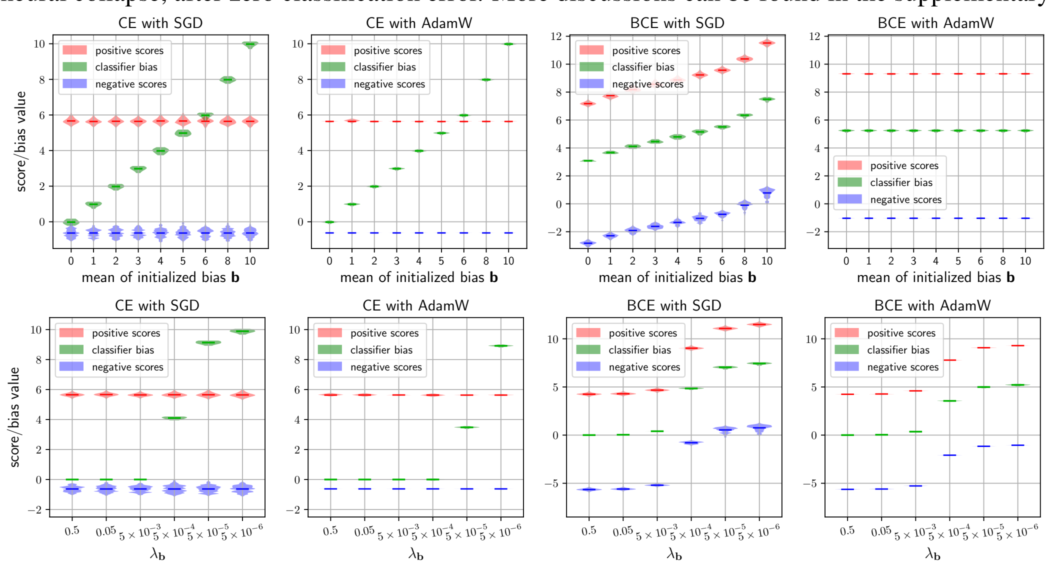


Figure 3: The distributions of the final classifier bias and positive/negative decision scores for ResNet18 trained on MNIST with fixed weight decay factor $\lambda_{\boldsymbol{b}}$ (top) and varying $\lambda_{\boldsymbol{b}}$ (bottom), while $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = 5 \times 10^{-4}$. The mean of initialized bias $\boldsymbol{b}$ is respectively set as $0, 1, 2, 3, 4, 5, 6, 8, 10$ in the experiments with fixed $\lambda_{\boldsymbol{b}} = 0$, and the bias mean is set as 10 in the experiments with varying $\lambda_{\boldsymbol{b}}$.

The bias decay factor $\lambda_{\boldsymbol{b}}$. To illustrate the different effects of classifier bias of CE and BCE on the decision scores, we conduct two groups of experiments by respectively applying fixed and varying classifier bias decay factor $\lambda_{\boldsymbol{b}}$ in the training of ResNet18 on MNIST: (1) with fixed $\lambda_{\boldsymbol{b}} = 0$ and default other hyper-parameters, respectively, setting the mean of the initialized classifier bias to $0, 1, 2, 3, 4, 5, 6, 8$, and 10; (2) with varying $\lambda_{\boldsymbol{b}} = 0.5, 0.05, 5 \times 10^{-3}, 5 \times 10^{-4}, 5 \times 10^{-5}$, and 5×10^{-6}, respectively, setting the mean of initialized classifier bias to 10. Fig. 3 shows the distributions of final classifier bias and positive/negative decision scores (without bias) using violin plots for the models in these experiments, while the numerical results are presented in Tables S-8 and S-9 in supplementary. One can find from Fig. 3(top), for the CE-trained models with $\lambda_{\boldsymbol{b}} = 0$, the final classifier bias values are almost entirely determined by their initial values, no matter which optimizer was applied. For the CE-trained models in Fig. 3(bottom), the means of the final classifier bias reach to zero from the initial mean of 10 with appropriate lager λ_b ($\geq 5 \times 10^{-3}$ for SGD and $\geq 5 \times 10^{-4}$ for AdamW), and they do not achieve the theoretical value when λ_b is too small. As a comparison, for the all CE-trained models, their final positive and negative decision scores respectively converge to around 5.64 and -0.63 (see supplementary for details). In total, in CE-trained models, the classifier bias hardly affects the decision scores, and thus almost does not affect the final feature properties.

In contrast, for the BCE-trained models in Fig. 3, their final positive and negative decision scores are always separated by the final classifier biases, no matter what the initial mean of classifier bias, λ_b, or optimizer are, and clear correlation exists between the bias and positive/negative decision scores. These results imply that, in the training with BCE, the classifier bias has a substantial impact on the sample feature distribution, thereby enhancing the compactness and distinctiveness across samples.

4.2 BCE PERFORMING BETTER THAN CE IN ENHANCING FEATURE PROPERTIES

To further demonstrate that BCE performs better than CE in enhancing the intra-class compactness and inter-class distinctiveness of sample features in the practical training, we train the three classification models by applying two different data augmentation techniques, (1) DA1: random cropping and horizontal flipping, (2) DA2: Mixup and CutMix, on CIFAR10 and CIFAR100 using SGD and AdamW, respectively. In the experiments, we take a global weight decay factor λ for the all parameters in the models, including the classifier weight and bias, and $\lambda = 5 \times 10^{-4}$ for SGD, $\lambda = 0.05$ for AdamW. The other hyper-parameters are presented in the supplementary. To compare the results of BCE and CE, besides the classification accuracy ($\mathcal{A}$), we define and apply three other metrics, uniform accuracy ($\mathcal{A}_{\text{Uni}}$), compactness ($\mathcal{E}_{\text{com}}$), and distinctiveness ($\mathcal{E}_{\text{dis}}$), seeing Eqs. (43,47,48) in supplementary for the definitions. While $\mathcal{A}_{\text{Uni}}$ is evolved from Eq. (17), it is calculated on the decision scores across samples, simultaneously reflecting the feature compactness and distinctiveness; as their name implies, $\mathcal{E}_{\text{com}}$ and $\mathcal{E}_{\text{dis}}$ respectively measure the intra-class compactness and inter-class distinctiveness among sample features.

Table 2: The classification on the test set of CIFAR10 and CIFAR100. The accuracy ($\mathcal{A}$) of most BCE-trained models is higher than that of CE-trained ones, while BCE-trained models perform consistently and significantly better than CE-trained models in terms of uniform accuracy ($\mathcal{A}_{\text{Uni}}$).

$\mathcal{D}$	$\mathcal{M}$	Loss	SGD				AdamW			
			DA1		DA1+DA2		DA1		DA1+DA2	
			$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$	$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$	$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$	$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$
CIFAR10	R18	CE	92.82	85.20	92.71	89.08	93.36	88.97	95.72	94.34
		BCE	93.22	91.92	93.64	91.87	93.95	93.37	95.57	95.16
		Δ	+0.40	+6.72	+0.93	+2.79	+0.59	+4.40	−0.15	+0.82
	R50	CE	92.69	85.23	92.74	89.58	94.48	87.86	96.00	94.31
		BCE	93.40	92.48	93.20	91.50	94.02	93.55	96.15	95.72
		Δ	+0.71	+7.25	+0.46	+1.92	−0.46	+5.69	+0.15	+1.41
	D121	CE	87.87	78.67	86.65	81.54	90.42	83.62	92.55	90.70
		BCE	88.66	87.58	87.78	84.95	90.55	89.91	92.59	91.78
		Δ	+0.79	+8.91	+1.13	+3.41	+0.13	+6.29	+0.04	+1.08
CIFAR100	R18	CE	71.16	43.21	71.76	56.66	71.69	49.17	76.53	64.43
		BCE	72.16	63.33	72.34	62.89	73.15	66.27	76.70	69.96
		Δ	+1.00	+20.1	+0.58	+6.23	+1.46	+17.1	+0.17	+5.53
	R50	CE	71.60	44.20	70.32	55.17	74.95	48.79	78.58	67.79
		BCE	71.75	64.07	71.87	62.82	75.25	68.84	78.47	72.68
		Δ	+0.15	+19.9	+1.55	+7.65	+0.30	+20.1	−0.11	+4.89
	D121	CE	60.79	32.93	57.23	39.82	63.65	38.76	68.99	57.15
		BCE	61.10	53.47	58.35	47.68	63.56	57.28	69.40	63.52
		Δ	+0.21	+20.5	+1.12	+7.85	−0.09	+18.5	+0.41	+6.37

Table 2 shows the classification results of the three models ("R18","R50", and "D121" respectively stand for ResNet18, ResNet50, and DenseNet121) on the test set of CIFAR10 and CIFAR100. From the table, one can find that, BCE is better than CE in term of accuracy ($\mathcal{A}$) in most cases, and it performs consistently and significantly superior to CE in term of uniform accuracy ($\mathcal{A}_{\text{Uni}}$). Taking CIFAR10 for example, among the twelve pairs of models trained by CE and BCE, BCE slightly reduced the accuracy of two pairs of models, while the gain of uniform accuracy introduced by BCE is 0.82% at least for the all models. For CIFAR100, the gain of BCE in uniform accuracy could be more than 20%, and the classification accuracy of BCE is still higher than that of CE in most cases. These results illustrate that BCE can usually achieve better classification results than CE, which is likely resulted from its enhancement in compactness and distinctiveness among sample features.

Furthermore, similar to BCE, the better data augmentation techniques and optimizer can simultaneously improve the classification results of models. For example, Mixup, CutMix, and AdamW increase $\mathcal{A}$ and $\mathcal{A}_{\text{Uni}}$ from 92.82% and 85.20% to 95.72% and 94.34%, respectively, for ResNet18 trained on CIFAR10. In addition, the higher performance of BCE than CE with only DA1 implies that the superiority of BCE is not resulted from the alignment with Mixup and CutMix, which is not consistent with the statements about BCE by Wightman et al. (2021).

As the uniform accuracy simultaneously reflect the intra-class compactness and inter-class distinctiveness, the higher uniform accuracy $\mathcal{A}_{\text{Uni}}$ of BCE-trained models implies their better feature properties. Table 3 presents the compactness ($\mathcal{E}_{\text{com}}$) and distinctiveness ($\mathcal{E}_{\text{dis}}$) of the trained models. One can clearly observe that, in most cases, BCE improves the compactness and distinctiveness of the sample features extracted by the models compared to CE, which is consistent with our expectations and provides a solid and reasonable explanation for the higher performance of BCE in tasks that require feature comparison, such as facial recognition and verification (Wen et al., 2022; Zhou et al., 2023).

Table 3: The feature properties on the test set of CIFAR10 and CIFAR100. The feature compactness ($\mathcal{E}_{\text{com}}$) and distinctiveness ($\mathcal{E}_{\text{dis}}$) of BCE-trained models are usually better than that of CE-trained models. See supplementary for the definitions of $\mathcal{E}_{\text{com}}$ and $\mathcal{E}_{\text{dis}}$.

$\mathcal{D}$	$\mathcal{M}$	Loss	SGD				AdamW			
			DA1		DA1+DA2		DA1		DA1+DA2	
			$\mathcal{E}_{\text{com}}$	$\mathcal{E}_{\text{dis}}$	$\mathcal{E}_{\text{com}}$	$\mathcal{E}_{\text{dis}}$	$\mathcal{E}_{\text{com}}$	$\mathcal{E}_{\text{dis}}$	$\mathcal{E}_{\text{com}}$	$\mathcal{E}_{\text{dis}}$
CIFAR10	R18	CE	0.8541	0.2553	0.8148	0.2088	0.8546	0.2694	0.8929	0.3307
		BCE	**0.9056**	**0.3049**	**0.8438**	**0.2387**	**0.9140**	**0.3254**	**0.9178**	**0.3669**
	R50	CE	0.8351	0.1782	0.8564	0.2027	0.8547	0.2332	**0.9529**	**0.3782**
		BCE	**0.8990**	**0.2322**	**0.8693**	**0.2161**	**0.8912**	**0.2720**	0.9168	0.3569
	D121	CE	0.7874	0.3123	0.7672	0.2805	0.7463	0.3070	0.8201	**0.3194**
		BCE	**0.8458**	**0.3319**	**0.8089**	**0.2973**	**0.8302**	**0.3371**	**0.8371**	0.3190
CIFAR100	R18	CE	0.7234	**0.2699**	0.7127	0.2575	0.6923	0.2895	0.7140	**0.3073**
		BCE	**0.7331**	0.2624	**0.7289**	**0.2688**	**0.7265**	**0.2930**	**0.7422**	0.2906
	R50	CE	0.7084	0.2002	0.7101	0.1866	0.6886	0.2581	0.7229	**0.3646**
		BCE	**0.7326**	**0.2196**	**0.7400**	**0.2184**	**0.7517**	**0.2783**	**0.7631**	0.3254
	D121	CE	0.7120	**0.3097**	0.7280	**0.3171**	0.6472	0.2981	0.6998	**0.3403**
		BCE	**0.7324**	0.2947	**0.7363**	0.3049	**0.7091**	**0.3008**	**0.7262**	0.3259

Fig. 4 visually shows the feature distributions on the testing data of CIFAR10 for ResNet18 trained by CE (left) and BCE (right) with "DA1+DA2" and AdamW. One can find that, for CE-trained model, its feature distributions for categories 3 and 5 (i.e., "cat" and "dog") overlap with each other, and the sample features of the third category exhibit clear internal dispersion; in contrast, the features of BCE-trained ResNet18 for these categories are distributed in more compact areas and have significant gaps between them, implying better feature compactness and distinctiveness.

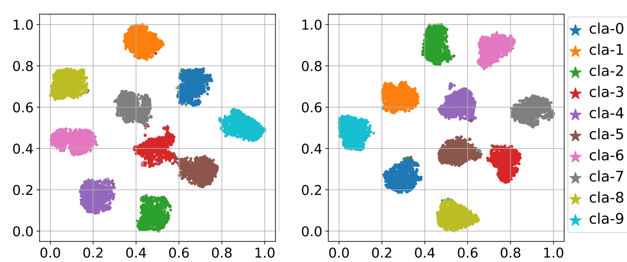


Figure 4: Distributions of features extracted by ResNet18 trained on CIFAR10 using CE (left) and BCE (right) in t-SNE.

5 CONCLUSIONS

This paper compares CE and BCE losses in deep feature learning. Both the losses can maximize the intra-class compactness and inter-class distinctiveness among sample features, i.e., leading to neural collapse when reaching their minimums. In the training, CE implicitly enhances the feature properties by correctly classifying samples one-by-one. In contrast, BCE can adjust the positive and negative decision scores across samples, and, in this process, its classifier bias plays a substantial and consistent role, making it explicitly enhance the intra-class compactness and inter-class distinctiveness of features. Therefore, BCE can usually achieve better classification performance.

Limitations. The decision scores measure the inner product/similarity of sample features to each classifier vector, which reflect the feature properties. However, it does not directly calculate the measurements among samples, nor can it be used to directly measure the compactness and distinctiveness of sample features. In the future, we will analyze the CE and BCE losses used for feature contrastive learning and compare the feature properties brought by them.

References

Mouin Ben Ammar, Nacim Belkhir, Sebastian Popescu, Antoine Manzanera, and Gianni Franchi. NECO: Neural collapse based out-of-distribution detection. In *International Conference on Learning Representations*, 2024.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020.

Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9589–9600, 2023.

Sanghyuk Chun. Improved probabilistic image-text representations. In *The Twelfth International Conference on Learning Representations*, 2024.

Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43):e2103091118, 2021.

Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. In *International Conference on Learning Representations*, 2023.

Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised contrastive learning. In *International Conference on Machine Learning*, pp. 3821–3830. PMLR, 2021.

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. SegNeXt: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:1140–1156, 2022.

X. Y. Han, Vardan Papyan, and David L. Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. In *International Conference on Learning Representations*, 2022.

Zhiwei Hao, Jianyuan Guo, Kai Han, Han Hu, Chang Xu, and Yunhe Wang. Revisit the power of vanilla knowledge distillation: from small scale to large scale. *Advances in Neural Information Processing Systems*, 36, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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

Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. TransReID: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15013–15022, 2021.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Hoyong Kim and Kangil Kim. Fixed non-negative orthogonal classifier: Inducing zero-mean neural collapse with feature dimension separation. In *International Conference on Learning Representations*, 2024.

Takumi Kobayashi. Two-way multi-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7476–7485, 2023.

Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *Transactions on Machine Learning Research*, 2023.

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

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

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pp. 507–516. PMLR, 2016.

Jianfeng Lu and Stefan Steinerberger. Neural collapse under cross-entropy loss. *Applied and Computational Harmonic Analysis*, 59:224–241, 2022.

Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *Transactions on Machine Learning Research*, 2023.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. In *International Conference on Machine Learning*, pp. 21478–21505. PMLR, 2022.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *European conference on computer vision*, pp. 516–533. Springer, 2022.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. NormFace: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.

Haoqi Wang, Zhizhong Li, and Wayne Zhang. Get the best of both worlds: Improving accuracy and transferability by grassmann class representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22478–22487, 2023.

Yining Wang, Junjie Sun, Chenyue Wang, Mi Zhang, and Min Yang. Navigate beyond shortcuts: Debiased learning through the lens of neural collapse. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12322–12331, 2024.

Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. In *International Conference on Learning Representations*, 2022.

Ross Wightman, Hugo Touvron, and Herve Jegou. ResNet strikes back: An improved training procedure in timm. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021.

Zhengzhuo Xu, Ruikang Liu, Shuo Yang, Zenghao Chai, and Chun Yuan. Learning imbalanced data with vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15793–15803, 2023.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in neural information processing systems*, 35:37991–38002, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Jiancan Zhou, Xi Jia, Qiufu Li, Linlin Shen, and Jinming Duan. Uniface: Unified cross-entropy loss for deep face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20730–20739, 2023.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. *Advances in Neural Information Processing Systems*, 35:31697–31710, 2022.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

BCE vs. CE in Deep Feature Learning

Supplementary Material

S-1 Neural collapse and feature property

S-1.1 Neural collapse

The neural collapse was first found by Papyan et al. (2020), which refers to four properties about the sample features $\{\boldsymbol{h}_i^{(k)}\}$ and the classifier vectors $\{\boldsymbol{w}_k\}$ at the terminal phase of training (Han et al., 2022), as list in Sec. 2.2. These four properties can be formulized as follows.

- **NC1**, within-class variability collapse, $\boldsymbol{\Sigma}_B^\dagger \boldsymbol{\Sigma}_W \to \mathbf{0}$, where

$$\boldsymbol{\Sigma}_B = \frac{1}{K}\sum_{k=1}^{K}\big(\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}\big)\big(\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}\big)^T \tag{22}$$

$$\boldsymbol{\Sigma}_W = \frac{1}{\sum_k n_k}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\big(\boldsymbol{h}_i^{(k)} - \bar{\boldsymbol{h}}^{(k)}\big)\big(\boldsymbol{h}_i^{(k)} - \bar{\boldsymbol{h}}^{(k)}\big)^T \tag{23}$$

$$\bar{\boldsymbol{h}}^{(k)} = \frac{1}{n_k}\sum_{i=1}^{n_k}\boldsymbol{h}_i^{(k)}, \tag{24}$$

$$\bar{\boldsymbol{h}} = \frac{1}{\sum_k n_k}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\boldsymbol{h}_i^{(k)} \tag{25}$$

  and † denotes the Mooer-Penrose pseudo-inverse;

- **NC2**, convergence to simplex equiangular tight frame,

$$\big\|\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}\big\|_2 - \big\|\bar{\boldsymbol{h}}^{(k')} - \bar{\boldsymbol{h}}\big\|_2 \to 0, \tag{26}$$

$$\frac{\big\langle \bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}},\ \bar{\boldsymbol{h}}^{(k')} - \bar{\boldsymbol{h}}\big\rangle}{\big\|\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}\big\|_2\big\|\bar{\boldsymbol{h}}^{(k')} - \bar{\boldsymbol{h}}\big\|_2} \to \begin{cases} 1, & k = k', \\ -\frac{1}{K-1}, & k \neq k'; \end{cases} \tag{27}$$

- **NC3**, convergence to self-duality,

$$\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2} - \frac{\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}}{\big\|\bar{\boldsymbol{h}}^{(k)} - \bar{\boldsymbol{h}}\big\|_2} \to 0; \tag{28}$$

- **NC4**, simplification to nearest class center,

$$\arg\max_j \{\boldsymbol{w}_j \boldsymbol{h} - b_j\}_{j=1}^{K} \to \arg\min_j \{\|\boldsymbol{h} - \bar{\boldsymbol{h}}^{(j)}\|_2\}_{j=1}^{K}. \tag{29}$$

In Sec. 4, we applied three metrics, $\mathcal{NC}_1, \mathcal{NC}_2, \mathcal{NC}_3$, to measure the above properties, similar to that defined in (Zhu et al., 2021; Zhou et al., 2022):

$$\mathcal{NC}_1 := \frac{1}{K}\mathrm{trace}\big(\boldsymbol{\Sigma}_W \boldsymbol{\Sigma}_B^\dagger\big), \tag{30}$$

$$\mathcal{NC}_2 := \Big\|\frac{\tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^T}{\|\tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^T\|_F} - \frac{1}{\sqrt{K-1}}\big(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\big)\Big\|_F, \tag{31}$$

$$\mathcal{NC}_3 := \Big\|\frac{\boldsymbol{W}\tilde{\boldsymbol{H}}}{\|\boldsymbol{W}\tilde{\boldsymbol{H}}\|_F} - \frac{1}{\sqrt{K-1}}\big(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\big)\Big\|_F, \tag{32}$$

where

$$\tilde{\boldsymbol{W}} = [\boldsymbol{w}_1 - \bar{\boldsymbol{w}}, \boldsymbol{w}_2 - \bar{\boldsymbol{w}}, \cdots, \boldsymbol{w}_K - \bar{\boldsymbol{w}}]^T \in \mathbb{R}^{K\times d}, \tag{33}$$

$$\tilde{\boldsymbol{H}} = [\bar{\boldsymbol{h}}^{(1)} - \bar{\boldsymbol{h}}, \bar{\boldsymbol{h}}^{(2)} - \bar{\boldsymbol{h}}, \cdots, \bar{\boldsymbol{h}}^{(K)} - \bar{\boldsymbol{h}}] \in \mathbb{R}^{d\times K}, \tag{34}$$

$$\bar{\boldsymbol{w}} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{w}_k. \tag{35}$$

When defining $\mathcal{NC}_2$, Zhu et al. (2021) and Zhou et al. (2022) did not subtract the classifier vectors with their mean, i.e., the original $\mathcal{NC}_2$ is defined as $\left\|\frac{\boldsymbol{W}\boldsymbol{W}^T}{\|\boldsymbol{W}\boldsymbol{W}^T\|_F}-\frac{1}{\sqrt{K-1}}\left(\boldsymbol{I}_K-\frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\right)\right\|_F$, with $\boldsymbol{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_K]^T\in\mathbb{R}^{K\times d}$.

As mentioned by Zhu et al. (2021) and Zhou et al. (2022), due to the "ReLU" activation functions before the FC classifiers in the deep models, the feature mean $\tilde{\boldsymbol{h}}_i=\frac{1}{K}\sum_{k=1}^K\boldsymbol{h}_i^{(k)}$ will be non-negative, which conflicts with $\tilde{\boldsymbol{h}}_i=\mathbf{0}$ required by Theorems 1 and 2. Then, the average features/class centers of K categories do not directly form an ETF, while the globally-centered average features form ETF, i.e., NC2 properties described by Eqs. (26) and (27). As the proof of Theorems 1 and 2, in the neural collapse, the features of each category will be parallel to its classifier vector, i.e., $\boldsymbol{h}_i^{(k)}=\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k$ in Eqs (126,127). Therefore, the classifier vectors $\{\boldsymbol{w}_k\}$ should also subtract their global mean before form an ETF. In other words, the third NC property should be

$$\textbf{NC3'}:\quad \frac{\boldsymbol{w}_k-\bar{\boldsymbol{w}}}{\left\|\boldsymbol{w}_k-\bar{\boldsymbol{w}}\right\|_2}-\frac{\bar{\boldsymbol{h}}^{(k)}-\bar{\boldsymbol{h}}}{\left\|\bar{\boldsymbol{h}}^{(k)}-\bar{\boldsymbol{h}}\right\|_2}\to 0. \tag{36}$$

As our analysis, when a model falling to the neural collapse, its classification accuracy $\mathcal{A}$ and uniform accuracy $\mathcal{A}_{\text{Uni}}$ must be 100% on the training dataset.

S-1.2 Feature property

In the experiments, we applied four metrics to compare the performance of CE and BCE, i.e., classification accuracy $\mathcal{A}$, uniform accuracy $\mathcal{A}_{\text{Uni}}$, feature compactness $\mathcal{E}_{\text{com}}$, and distinctiveness $\mathcal{E}_{\text{dis}}$. These metrics on the training data will be maximized when the model, classifier, and loss in the neural collapse.

In a classification task, suppose a dataset $\mathcal{D}=\bigcup_{k=1}^K\mathcal{D}_k=\bigcup_{k=1}^K\bigcup_{i=1}^{n_k}\left\{\boldsymbol{X}_i^{(k)}\right\}$ from K categories, where $\boldsymbol{X}_i^{(k)}$ denotes the ith sample from the category k. For the sample $\boldsymbol{X}_i^{(k)}$ in $\mathcal{D}$, a model $\mathcal{M}$ converts it into its feature $\boldsymbol{h}_i^{(k)}=\mathcal{M}(\boldsymbol{X}_i^{(k)})\in\mathbb{R}^d$, where d is the length of the feature vector. A linear, full connection (FC) classifier $\mathcal{C}=\left\{(\boldsymbol{w}_k,b_k)\right\}_{k=1}^K$ transform the feature into K decision scores $\left\{\boldsymbol{w}_j\boldsymbol{h}_i^{(k)}-b_j\right\}_{j=1}^K$. Then, the sample can be correctly classified if

$$\boldsymbol{w}_k\boldsymbol{h}_i^{(k)}-b_k=\max\left\{\boldsymbol{w}_j\boldsymbol{h}_i^{(k)}-b_j\right\}_{j=1}^K, \tag{37}$$

which is equivalent to

$$k=\arg\max_{\ell}\{\boldsymbol{w}_\ell^T\boldsymbol{h}^{(k)}-b_\ell\}. \tag{38}$$

The the commonly used **classification accuracy** can be defined as

$$\mathcal{A}(\mathcal{M},\mathcal{C})=\frac{|\mathcal{D}(\mathcal{M},\mathcal{C})|}{|\mathcal{D}|}\times 100\%, \tag{39}$$

where

$$\mathcal{D}(\mathcal{M},\mathcal{C})=\bigcup_{k=1}^K\left\{\boldsymbol{X}^{(k)}:k=\arg\max_{\ell}\{\boldsymbol{w}_\ell^T\boldsymbol{h}^{(k)}-b_\ell\},\boldsymbol{X}^{(k)}\in\mathcal{D}_k,\boldsymbol{h}^{(k)}=\mathcal{M}(\boldsymbol{X}^{(k)})\right\}, \tag{40}$$

consisting of the all samples correctly classified by $\mathcal{M}$ and $\mathcal{C}$ in $\mathcal{D}$.

Eq. (37) implies a dynamic threshold $t_{\boldsymbol{X}}$ separating the positive and negative decision scores. Inspired by Eq. (17), we define uniform accuracy by using a unified threshold. Firstly, for given dataset $\mathcal{D}$ and model $\mathcal{M}$, classifier $\mathcal{C}$ with a fixed threshold t, we denote a subset of $\mathcal{D}$ as

$$\mathcal{D}(\mathcal{M},\mathcal{C};t)=\bigcup_{k=1}^K\left\{\boldsymbol{X}^{(k)}\in\mathcal{D}_k:\boldsymbol{w}_k\boldsymbol{h}^{(k)}-b_k>t\geq\max\left\{\boldsymbol{w}_j^T\boldsymbol{h}^{(k)}-b_j\right\}_{\substack{j=1\\j\neq k}}^K,\boldsymbol{h}^{(k)}=\mathcal{M}(\boldsymbol{X}^{(k)})\right\} \tag{41}$$

which is the biggest subset of $\mathcal{D}$ uniformly classified by $\mathcal{M}$ and $\mathcal{C}$ with t. Then the ratio

$$\mathcal{A}_{\text{Uni}}(\mathcal{M}, \mathcal{C}; t) = \frac{|\mathcal{D}(\mathcal{M}, \mathcal{C}; t)|}{|\mathcal{D}|} \times 100\%, \tag{42}$$

is the corresponding uniform accuracy, and the maximum ratio with varying thresholds, i.e.,

$$\mathcal{A}_{\text{Uni}}(\mathcal{M}, \mathcal{C}) = \max_{t \in \mathbb{R}} \mathcal{A}_{\text{Uni}}(\mathcal{M}, \mathcal{C}; t), \tag{43}$$

is defined as the final **uniform accuracy**.

In practice, to calculate the uniform accuracy $\mathcal{A}_{\text{Uni}}$, the sets of positive and negative decision scores for the all samples

$$\mathcal{S}_{\text{pos}} = \bigcup_{k=1}^{K} \left\{ \boldsymbol{w}_k \boldsymbol{h}_i^{(k)} - b_k : i = 1, 2, \cdots, n_k \right\}, \tag{44}$$

$$\mathcal{S}_{\text{neg}} = \bigcup_{k=1}^{K} \bigcup_{\substack{j=1 \\ j \neq k}}^{K} \left\{ \boldsymbol{w}_j \boldsymbol{h}_i^{(k)} - b_j : i = 1, 2, \cdots, n_k \right\} \tag{45}$$

are first computed, and denote

$$s_{\text{pos-min}} = \min(\mathcal{S}_{\text{pos}}) \qquad \text{and} \qquad s_{\text{neg-max}} = \max(\mathcal{S}_{\text{pos}}). \tag{46}$$

If $s_{\text{pos-min}} \geq s_{\text{neg-max}}$, the classification accuracy $\mathcal{A}$ and the uniform one $\mathcal{A}_{\text{Uni}}$ must be 100%, otherwise, $N = 200$ thresholds $\{t_i\}_{i=1}^{N}$ are evenly taken from the interval $[s_{\text{pos-min}}, s_{\text{neg-max}}]$, and $N = 200$ uniform accuracy $\mathcal{A}_{\text{Uni}}(\mathcal{M}, \mathcal{C}; t_i)$ are figured out, while the best one $\max \left\{ A_{\text{Uni}}(\mathcal{M}, \mathcal{C}; t_i) \right\}_{i=1}^{N}$ is chosen as the final uniform accuracy $\mathcal{A}_{\text{Uni}}$. In this calculation, the final results will be slightly different when different numbers (N) of thresholds are taken in the score interval.

By Eqs. (17), a model with higher uniform accuracy, it would lead to more samples from category $k, \forall k \in [K]$, whose inner products (positive similarities/decision scores without bias) with the classifier vector $\boldsymbol{w}_k$ are greater than $b_k + t$, implying higher intra-class compactness in each category, and requires more samples whose inner products (negative similarities/decision scores without bias) with the classifier vectors of other categories are less than $b_j + t$, implying higher inter-class distinctiveness among all categories. For the intra-class **compactness** $\mathcal{E}_{\text{com}}$ and inter-class **distinctiveness** $\mathcal{E}_{\text{dis}}$ among sample features, we define them as

$$\mathcal{E}_{\text{com}} = \frac{1}{2} \left[\frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{n_k^2} \sum_{i=1}^{n_k} \sum_{i'=1}^{n_k} \frac{\langle \boldsymbol{h}_i^{(k)} - \bar{\boldsymbol{h}}, \boldsymbol{h}_{i'}^{(k)} - \bar{\boldsymbol{h}} \rangle}{\|\boldsymbol{h}_i^{(k)} - \bar{\boldsymbol{h}}\| \|\boldsymbol{h}_{i'}^{(k)} - \bar{\boldsymbol{h}}\|} \right) + 1 \right], \tag{47}$$

$$\mathcal{E}_{\text{dis}} = \frac{1}{2} \left[1 - \frac{1}{K(K-1)} \sum_{k=1}^{K} \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \left(\frac{1}{n_k} \frac{1}{n_{k'}} \sum_{i=1}^{n_k} \sum_{i'=1}^{n_{k'}} \frac{\langle \boldsymbol{h}_i^{(k)}, \boldsymbol{h}_{i'}^{(k')} \rangle}{\|\boldsymbol{h}_i^{(k)}\| \|\boldsymbol{h}_{i'}^{(k')}\|} \right) \right], \tag{48}$$

where $\bar{\boldsymbol{h}} = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \boldsymbol{h}_i^{(k)}$ is the global feature center.

Due to the neural collapse, the compactness $\mathcal{E}_{\text{com}}$ might be higher than $\frac{1}{2} - \frac{1}{2(K-1)}$, and the distinctiveness $\mathcal{E}_{\text{dis}}$ might be lower than $\frac{1}{2} + \frac{1}{2(K-1)}$, for the model $\mathcal{M}$ and classifier $\mathcal{C}$ which have been well trained on the dataset $\mathcal{D}$.

S-2 Experimental settings and results

S-2.1 Experimental settings

Table S-4: Experimental settings in our experiments.

		Neural collapse		Classification			
		setting-1	setting-2	setting-3	setting-4	setting-5	setting-6
Hyper-parameter	epochs	100	100	100	100	100	100
	optimizer	SGD	AdamW	SGD	AdamW	SGD	AdamW
	batch size	128	128	128	128	128	128
	learning rate	0.01	0.001	0.01	0.001	0.01	0.001
	learning rate decay	step	cosine	step	cosine	step	cosine
	weight decay λ	✗	✗	5×10^{-4}	0.05	5×10^{-4}	0.05
	weight decay $\lambda_{\boldsymbol{W}}$	5×10^{-4}	5×10^{-4}	✗	✗	✗	✗
	weight decay $\lambda_{\boldsymbol{H}}$	5×10^{-4}	5×10^{-4}	✗	✗	✗	✗
	weight decay $\lambda_{\boldsymbol{b}}$	5×10^{-4}	5×10^{-4}	✗	✗	✗	✗
	warmup epochs	0	0	0	0	0	0
Data Aug.	random cropping	✗	✗	✓	✓	✓	✓
	horizontal flipping	✗	✗	0.5	0.5	0.5	0.5
	label smoothing	✗	✗	✗	✗	0.1	0.1
	mixup alpha	✗	✗	✗	✗	0.8	0.8
	cutmix alpha	✗	✗	✗	✗	1.0	1.0
	mixup prob.	✗	✗	✗	✗	0.8	0.8
	normalization	mean = $[0.4914, 0.4822, 0.4465]$, std = $[0.2023, 0.1994, 0.2010]$					

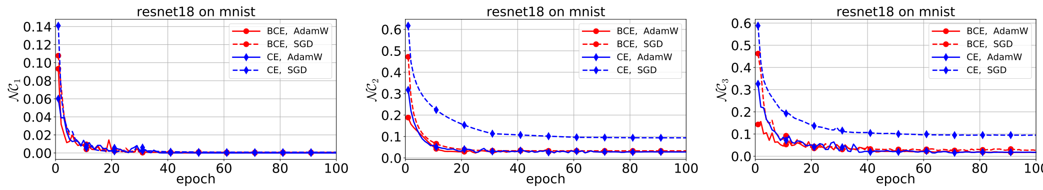
In Sec. 4, we train ResNet18, ResNet50, and DenseNet121 on MNIST, CIFAR10, and CIFAR100, respectively. Table S-4 shows the experimental settings. In default, we train the models using setting-1 and setting-2 in the experiments of neural collapse, and apply setting-3, setting-4, setting-5, and setting-6 in the experiments of classification.

Table S-5: The numerical results of the three models trained on MNIST, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

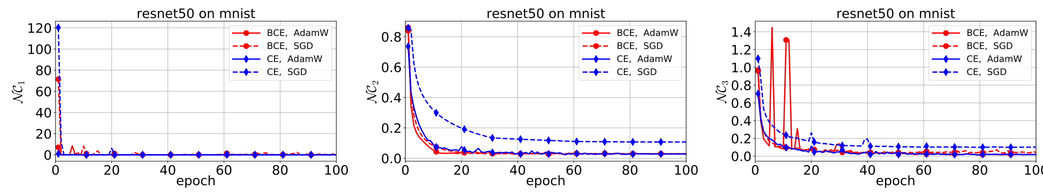
		MNIST			
		SGD		AdamW	
		CE	BCE	CE	BCE
ResNet18	$\hat{\rho}$	219.0960	407.1362	212.2180	357.9696
	s_{pos}	5.6439 ± 0.1437	6.4008 ± 0.1236	5.6331 ± 0.0120	7.7460 ± 0.0113
	s_{neg}	-0.6302 ± 0.2073	-3.4987 ± 0.1137	-0.6259 ± 0.0127	-2.1233 ± 0.0291
	$\hat{b}$	-0.0074 ± 0.0852	2.2170 ± 0.0308	0.0001 ± 0.0328	3.5134 ± 0.0337
	$\alpha(\hat{b})$	—	-0.0268	—	-0.0086
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$100.00/100.00$	$100.00/100.00$	$100.00/100.00$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$99.43/99.31$	$99.59/99.52$	$99.62/99.57$	$99.65/99.61$
ResNet50	$\hat{\rho}$	217.7276	396.7711	212.2304	357.2365
	s_{pos}	5.6383 ± 0.6400	6.5393 ± 1.6509	5.6389 ± 0.0380	7.7706 ± 0.0573
	s_{neg}	-0.6271 ± 0.5978	-3.2512 ± 1.9658	-0.6266 ± 0.0220	-2.1029 ± 0.0429
	$\hat{b}$	0.0039 ± 0.0733	2.4674 ± 0.0492	0.0001 ± 0.0328	3.5322 ± 0.0329
	$\alpha(\hat{b})$	—	-0.0217	—	-0.0084
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$99.68/99.64$	$99.79/99.76$	$100.00/100.00$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$98.98/98.79$	$99.01/98.88$	$99.60/99.57$	$99.53/99.52$
DenseNet121	$\hat{\rho}$	224.1426	414.7491	212.2337	355.5479
	s_{pos}	5.5774 ± 0.1217	6.1977 ± 0.0987	5.6318 ± 0.1132	7.8030 ± 0.0377
	s_{neg}	-0.6193 ± 0.1221	-3.6421 ± 0.1048	-0.6258 ± 0.3427	-2.0508 ± 0.0314
	$\hat{b}$	0.0010 ± 0.0570	2.0705 ± 0.0264	0.0002 ± 0.0324	3.5767 ± 0.0344
	$\alpha(\hat{b})$	—	-0.0302	—	-0.0081
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$100.00/99.99$	$100.00/100.00$	$99.63/99.62$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$99.45/99.40$	$99.54/99.52$	$99.29/99.22$	$99.64/99.60$

S-2.2 Experimental results of neural collapse

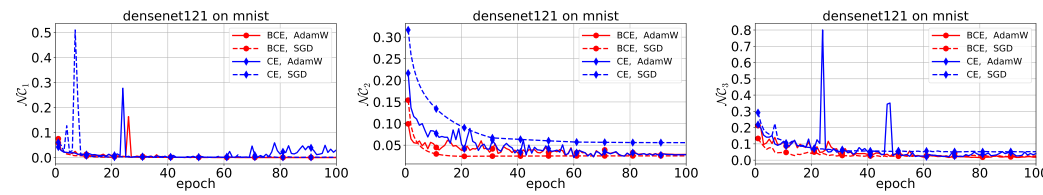
In this section, we show the experimental results of neural collapse. Most of these results are calculated on the training data of the three datasets.



(a) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet18 on MNIST

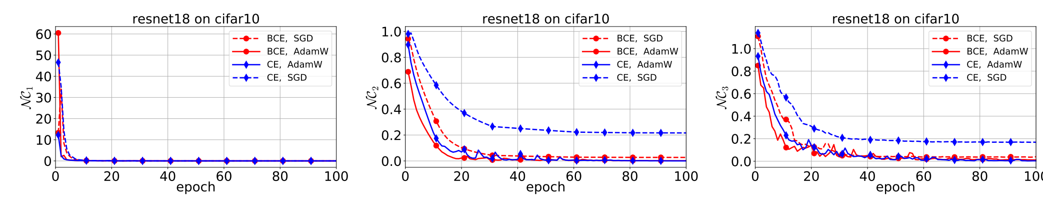


(b) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet50 on MNIST

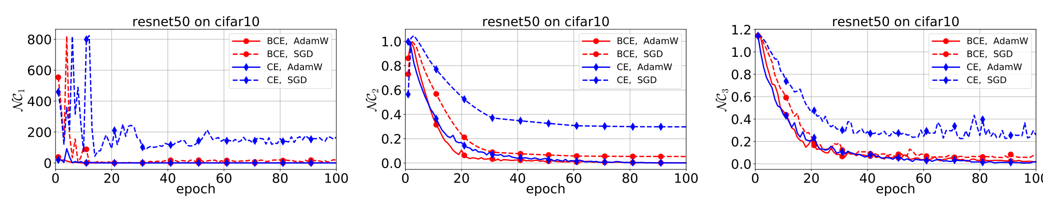


(c) $\mathcal{NC}_1$, $\mathcal{NC}_1$, and $\mathcal{NC}_1$ of DenseNet121 on MNIST

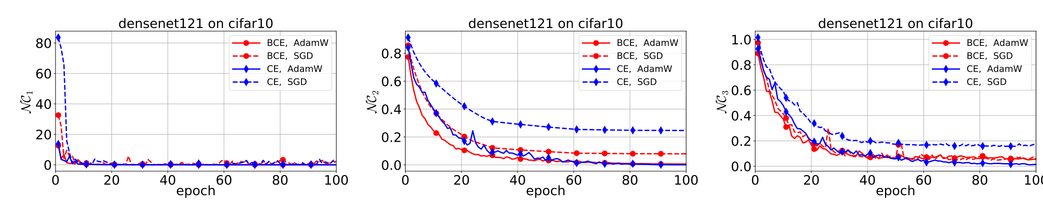
Figure S-5: The evolution of the three NC metrics in the training of ResNet18 (top), ResNet50 (middle), DenseNet121 (bottom) on MNIST with CE and BCE using SGD and AdamW, respectively, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.



(a) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet18 on CIFAR10



(b) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet50 on CIFAR10

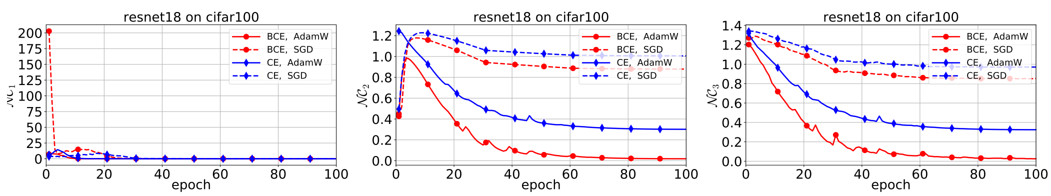


(c) $\mathcal{NC}_1$, $\mathcal{NC}_1$, and $\mathcal{NC}_1$ of DenseNet121 on CIFAR10

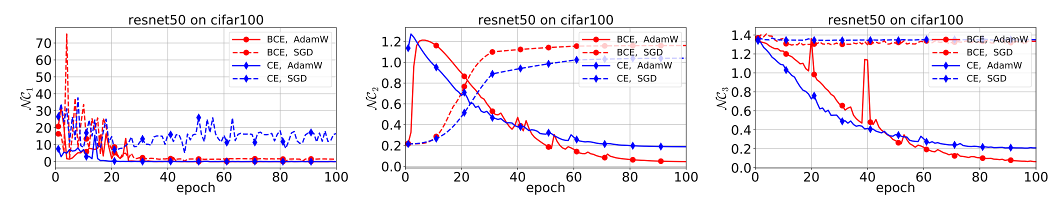
Figure S-6: The evolution of the three NC metrics in the training of ResNet18 (top), ResNet50 (middle), DenseNet121 (bottom) on CIFAR10 with CE and BCE using SGD and AdamW, respectively, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

Table S-6: The numerical results of the three models trained on CIFAR10, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

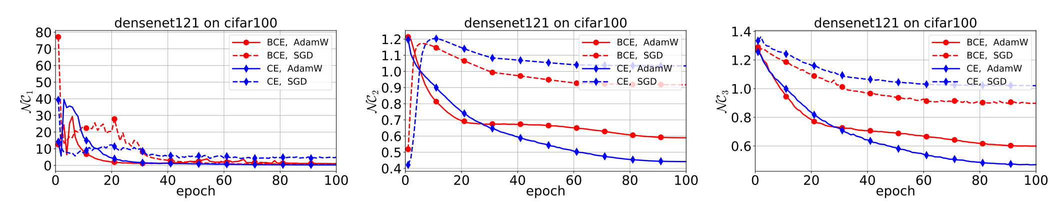
		CIFAR10			
		SGD		AdamW	
		CE	BCE	CE	BCE
ResNet18	$\hat{\rho}$	221.7685	395.3918	212.4173	366.6813
	s_{pos}	5.7103 ± 0.2252	6.5627 ± 0.2042	5.6393 ± 0.0568	7.5025 ± 0.0549
	s_{neg}	-0.6386 ± 0.3574	-3.4557 ± 0.1939	-0.6265 ± 0.0066	-2.3582 ± 0.0225
	$\hat{b}$	-0.0085 ± 0.0430	2.2618 ± 0.0678	-0.0001 ± 0.0038	3.2905 ± 0.0080
	$\alpha(\hat{b})$	—	-0.0266	—	-0.0105
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$99.99/99.98$	$100.00/100.00$	$100.00/100.00$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$79.22/75.71$	$81.19/78.78$	$86.66/84.72$	$86.58/85.07$
ResNet50	$\hat{\rho}$	220.8594	382.4440	212.3374	369.2447
	s_{pos}	5.7365 ± 8.2056	6.5614 ± 4.3923	5.6386 ± 0.1062	7.4351 ± 0.2787
	s_{neg}	-0.6439 ± 14.1340	-3.5695 ± 7.0134	-0.6266 ± 0.0150	-2.4493 ± 0.2165
	$\hat{b}$	0.0045 ± 0.1430	2.4002 ± 0.1496	-0.0000 ± 0.0053	3.2051 ± 0.0309
	$\alpha(\hat{b})$	—	-0.0242	—	-0.0114
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$99.61/99.52$	$99.65/99.32$	$99.99/99.99$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$76.28/73.08$	$78.41/76.35$	$85.73/84.33$	$85.76/84.98$
DenseNet121	$\hat{\rho}$	225.0609	392.8198	212.7966	360.5613
	s_{pos}	5.7225 ± 1.7228	6.2376 ± 0.8437	5.6150 ± 0.2851	7.6743 ± 0.1239
	s_{neg}	-0.6348 ± 0.8664	-3.6171 ± 1.6284	-0.6240 ± 0.0330	-2.1715 ± 0.0604
	$\hat{b}$	0.0012 ± 0.0364	2.0875 ± 0.1229	0.0003 ± 0.0061	3.4612 ± 0.0203
	$\alpha(\hat{b})$	—	-0.0318	—	-0.0090
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	$99.40/99.03$	$99.72/99.62$	$99.87/99.86$	$100.00/100.00$
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	$77.30/74.41$	$79.16/77.95$	$81.54/80.15$	$82.34/81.70$



(a) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet18 on CIFAR100



(b) $\mathcal{NC}_1$, $\mathcal{NC}_2$, and $\mathcal{NC}_3$ of ResNet50 on CIFAR100



(c) $\mathcal{NC}_1$, $\mathcal{NC}_1$, and $\mathcal{NC}_1$ of DenseNet121 on CIFAR100

Figure S-7: The evolution of the three NC metrics in the training of ResNet18 (top), ResNet50 (middle), DenseNet121 (bottom) on CIFAR100 with CE and BCE using SGD and AdamW, respectively, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

Table S-7: The numerical results of the three models trained on CIFAR100, with $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

		CIFAR100			
		SGD		AdamW	
		CE	BCE	CE	BCE
ResNet18	$\hat{\rho}$	954.3918	1732.6035	846.4734	1708.9231
	s_{pos}	8.3613 ± 0.4316	3.5152 ± 0.2392	7.5183 ± 0.0997	4.0202 ± 0.0696
	s_{neg}	-0.0848 ± 1.3897	-6.5934 ± 1.2718	-0.0754 ± 0.2580	-5.6834 ± 0.0438
	$\hat{b}$	0.0004 ± 0.2356	0.8407 ± 0.0678	0.0005 ± 0.0097	1.1317 ± 0.0007
	$\alpha(\hat{b})$	—	-0.2672	—	-0.2147
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	99.95/99.81	99.98/99.97	99.98/99.96	99.98/99.97
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	34.61/17.99	42.06/30.61	56.58/47.29	60.48/43.04
ResNet50	$\hat{\rho}$	36.2794	289.5987	838.0098	1710.3754
	s_{pos}	0.5404 ± 9.8551	-4.6656 ± 16.0695	7.3906 ± 0.3560	3.9356 ± 1.5798
	s_{neg}	0.6182 ± 11.4828	-6.2663 ± 29.8421	-0.0745 ± 0.1935	-5.7441 ± 1.1971
	$\hat{b}$	0.0006 ± 0.0592	0.3210 ± 0.0241	0.0005 ± 0.0073	1.1239 ± 0.0044
	$\alpha(\hat{b})$	—	-0.4090	—	-0.2160
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	2.52/0.05	7.67/0.44	99.83/99.76	99.77/99.62
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	2.48/0.06	7.16/0.39	55.51/50.77	53.55/49.18
DenseNet121	$\hat{\rho}$	894.4895	1597.8596	900.5263	1761.0126
	s_{pos}	8.4473 ± 0.8321	3.0569 ± 1.6496	8.1030 ± 0.4805	4.0875 ± 0.2246
	s_{neg}	-0.0842 ± 1.6340	-6.6552 ± 2.6035	-0.0800 ± 0.4365	-5.8613 ± 0.7152
	$\hat{b}$	-0.0012 ± 0.2239	0.8313 ± 0.0983	0.0016 ± 0.0948	1.1306 ± 0.0145
	$\alpha(\hat{b})$	—	-0.2714	—	-0.2141
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for training	99.15/94.38	99.38/99.23	99.80/99.78	99.98/99.97
	$\mathcal{A}/\mathcal{A}_{\text{Uni}}$ for testing	37.48/24.20	39.93/35.19	50.31/37.87	52.41/49.81

NC metrics, the final classifier bias, and the final decision scores. Figs. S-5 - S-7 shows the evolution of the three NC metrics in the training of ResNet18, ResNet50, DenseNet121 on MNIST, CIFAR10, and CIFAR100 with CE and BCE. In the training on MNIST and CIFAR10, the NC metrics of both CE and BCE approach zero at the terminal phase of training, and that of BCE decrease faster than that of CE at the first 20 epochs. In the training on CIFAR100, which is a more challenging dataset than MNIST and CIFAR10, the NC metrics of models trained by SGD do not decrease to zero, while that of models trained by AdamW approach zero, and the NC metrics of BCE decrease faster than that of CE in most cases. Table S-5 - S-7 present the numerical results of the final models at the 100th epoch. In these tables, $\hat{\rho} = \|\hat{\boldsymbol{W}}\|_F^2$, where $\hat{\boldsymbol{W}} = [\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, \cdots, \hat{\boldsymbol{w}}_K]^T \in \mathbb{R}^{K \times d}$ is the final trained classifier weight; "s_{pos}" rows list the mean and standard deviations of the final positive decision scores without biases, i.e.,

$$\text{Mean}(s_{\text{pos}}) = \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \hat{\boldsymbol{w}}_k \boldsymbol{h}_i^{(k)}, \tag{49}$$

$$\text{Std}(s_{\text{pos}}) = \sqrt{\sum_{k=1}^{K} \sum_{i=1}^{n} \frac{\left(\hat{\boldsymbol{w}}_k \boldsymbol{h}_i^{(k)} - \text{Mean}(s_{\text{pos}})\right)^2}{nK}}, \tag{50}$$

"s_{neg}" rows list that of the final negative decision scores without biases, i.e.,

$$\text{Mean}(s_{\text{neg}}) = \frac{1}{nK(K-1)} \sum_{k=1}^{K} \sum_{\substack{j=1 \\ j \neq k}}^{K} \sum_{i=1}^{n} \hat{\boldsymbol{w}}_j \boldsymbol{h}_i^{(k)}, \tag{51}$$

$$\text{Std}(s_{\text{neg}}) = \sqrt{\sum_{k=1}^{K} \sum_{\substack{j=1 \\ j \neq k}}^{K} \sum_{i=1}^{n} \frac{\left(\hat{\boldsymbol{w}}_j \boldsymbol{h}_i^{(k)} - \text{Mean}(s_{\text{neg}})\right)^2}{nK(K-1)}}, \tag{52}$$

and "$\hat{b}$" rows list that of the final classifier bias $\hat{\boldsymbol{b}} = [\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_K]^T \in \mathbb{R}^K$, i.e.,

$$\text{Mean}(\hat{b}) = \frac{1}{K} \sum_{k=1}^{K} \hat{b}_k, \tag{53}$$

$$\mathrm{Std}(\hat{b}) = \sqrt{\frac{\sum_{k=1}^{K}\left(\hat{b}_k - \mathrm{Mean}(\hat{b})\right)^2}{K}}. \tag{54}$$

"$\alpha(\hat{b})$" rows list the value of function $\alpha(b)$ at point $\mathrm{Mean}(\hat{b})$, where

$$\alpha(b) = -\frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} + \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b\right)\right)} + \lambda_{\boldsymbol{b}} b, \tag{55}$$

is the function at the RHS of Eq. (12).

Besides the classification accuracy $\mathcal{A}$ and uniform accuracy $\mathcal{A}_{\text{Uni}}$ of the final models on the training data, Tables S-5, S-6, and S-7 have also presented that on the testing data.

Table S-8: The numerical results of ResNet18 trained on MNIST with fixed weight decay $\lambda_{\boldsymbol{b}}$ for the classifier bias.

Loss	Opt.	$\bar{b}$	$\hat{\rho}$	s_{pos}	s_{neg}	$\hat{b}$	$\alpha(\hat{b})$
CE	SGD	0	218.9428	5.6648 ± 0.1673	-0.6323 ± 0.2360	-0.0179 ± 0.1228	—
		1	218.8023	5.6337 ± 0.1473	-0.6290 ± 0.2097	0.9821 ± 0.1149	—
		2	218.3450	5.6456 ± 0.1556	-0.6318 ± 0.2213	1.9821 ± 0.1122	—
		3	218.3319	5.6399 ± 0.1521	-0.6295 ± 0.2132	2.9821 ± 0.1163	—
		4	219.2994	5.6628 ± 0.1600	-0.6321 ± 0.2281	3.9820 ± 0.1307	—
		5	219.5797	5.6611 ± 0.1780	-0.6329 ± 0.2411	4.9820 ± 0.1279	—
		6	220.0522	5.6458 ± 0.1598	-0.6301 ± 0.2245	5.9820 ± 0.1312	—
		8	219.4256	5.6410 ± 0.1608	-0.6311 ± 0.2284	7.9821 ± 0.1194	—
		10	219.2911	5.6411 ± 0.1601	-0.6300 ± 0.2152	9.9821 ± 0.1250	—
	AdamW	0	212.2146	5.6360 ± 0.0250	-0.6262 ± 0.0189	-0.0180 ± 0.0486	—
		1	212.2138	5.6355 ± 0.0353	-0.6262 ± 0.0194	0.9828 ± 0.0493	—
		2	212.2151	5.6336 ± 0.0258	-0.6260 ± 0.0189	1.9821 ± 0.0487	—
		3	212.2152	5.6336 ± 0.0264	-0.6260 ± 0.0189	2.9825 ± 0.0486	—
		4	212.2161	5.6307 ± 0.0274	-0.6257 ± 0.0191	3.9823 ± 0.0491	—
		5	212.2143	5.6308 ± 0.0264	-0.6257 ± 0.0189	4.9809 ± 0.0486	—
		6	212.2143	5.6323 ± 0.0264	-0.6258 ± 0.0189	5.9822 ± 0.0486	—
		8	212.2163	5.6347 ± 0.0262	-0.6261 ± 0.0189	7.9812 ± 0.0486	—
		10	212.2151	5.6340 ± 0.0263	-0.6260 ± 0.0189	9.9829 ± 0.0486	—
BCE	SGD	0	393.2500	7.1748 ± 0.1277	-2.8219 ± 0.1379	3.0789 ± 0.0489	-0.0120
		1	374.9337	7.7515 ± 0.1578	-2.2877 ± 0.1468	3.6658 ± 0.0709	-0.0070
		2	362.5949	8.1822 ± 0.1525	-1.9121 ± 0.1604	4.1078 ± 0.1053	-0.0045
		3	355.2978	8.5608 ± 0.1634	-1.6192 ± 0.1568	4.4557 ± 0.0981	-0.0030
		4	354.6479	8.8711 ± 0.1473	-1.3347 ± 0.1725	4.7949 ± 0.1094	-0.0019
		5	355.9634	9.2305 ± 0.1503	-1.0452 ± 0.1960	5.1493 ± 0.1192	-0.0009
		6	361.1938	9.5688 ± 0.1355	-0.7519 ± 0.1688	5.5084 ± 0.0869	-0.0002
		8	385.6802	10.3761 ± 0.1400	-0.0997 ± 0.2436	6.3418 ± 0.0989	0.0007
		10	426.3013	11.5173 ± 0.1430	0.7786 ± 0.3075	7.4858 ± 0.1021	0.0010
	AdamW	0	350.4272	9.3081 ± 0.0352	-1.0348 ± 0.0321	5.2388 ± 0.0609	-0.0006
		1	350.4283	9.3015 ± 0.0345	-1.0340 ± 0.0321	5.2389 ± 0.0609	-0.0006
		2	350.4292	9.3029 ± 0.0357	-1.0342 ± 0.0321	5.2388 ± 0.0609	-0.0006
		3	350.4275	9.3028 ± 0.0364	-1.0342 ± 0.0321	5.2388 ± 0.0609	-0.0006
		4	350.4248	9.3039 ± 0.0362	-1.0343 ± 0.0320	5.2388 ± 0.0609	-0.0006
		5	350.4250	9.3100 ± 0.0358	-1.0350 ± 0.0320	5.2388 ± 0.0608	-0.0006
		6	350.4302	9.3063 ± 0.0345	-1.0346 ± 0.0321	5.2388 ± 0.0608	-0.0006
		8	350.4304	9.3094 ± 0.0356	-1.0349 ± 0.0321	5.2389 ± 0.0609	-0.0006
		10	350.4330	9.3109 ± 0.0369	-1.0351 ± 0.0321	5.2388 ± 0.0609	-0.0006

The failures in the experiments of neural collapse. According to the above figures and tables, one can find the models trained with SGD are easily to fail in the experiments of neural collapse, including the ResNet50 trained on MNIST, ResNet50 and DenseNet121 trained on CIFAR10, and the three models trained CIFAR100. The standard deviations of positive/negative decision scores produced by these models are usually larger than 0.5. These failed models in the neural collapse can be roughly classified into two types:

- The two ResNet50 trained on CIFAR100 with SGD. They are completely failed models. The standard deviations of the decision scores are very high, even more than 20, and, for the

BCE-trained model, the means of the positive and negative decision scores are relatively close, while for the CE-trained model, the mean of positive scores is even less than that of negative ones, indicating that most of the samples were not correctly classified. The classification accuracy $\mathcal{A}$ on the training dataset are only 2.52% and 7.67% with CE and BCE.

- The other failed models trained with SGD, including the ResNet50 trained on MNIST and CIFAR10, DenseNet121 trained on CIFAR10, ResNet18 and DenseNet121 trained on CIFAR100. These models have achieved almost 100% classification accuracy and uniform accuracy on the training dataset. However, according to the standard deviations of decision scores and the NC metrics, we conclude that they do not reach the state of neural collapse.

These failures in the experiments of neural collapse reveal more relationships among classification and neural collapse. In the training, zero classification error appears before zero uniform classification error, which appears before the neural collapse, or, in contrary, the model reaching the neural collapse has the uniform accuracy of 100%, and the model with the uniform accuracy of 100% has also the accuracy 100% on the classification. Both the reverses are not true.

Table S-9: The numerical results of ResNet18 trained on MNIST with varying weight decay $\lambda_{\boldsymbol{b}}$ for the classifier bias.

Loss	Opt.	$\lambda_{\boldsymbol{b}}$	$\hat{\rho}$	s_{pos}	s_{neg}	$\hat{b}$	$\alpha(\hat{b})$
CE	SGD	5×10^{-1}	218.6677	5.6511 ± 0.1144	-0.6304 ± 0.1854	-0.0000 ± 0.0002	—
		5×10^{-2}	218.6658	5.6662 ± 0.1176	-0.6321 ± 0.2031	-0.0000 ± 0.0017	—
		5×10^{-3}	218.5622	5.6427 ± 0.1076	-0.6296 ± 0.1917	0.0013 ± 0.0156	—
		5×10^{-4}	219.4882	5.6527 ± 0.1287	-0.6322 ± 0.2352	4.0998 ± 0.0796	—
		5×10^{-5}	219.0555	5.6526 ± 0.1407	-0.6310 ± 0.2192	9.1337 ± 0.1038	—
		5×10^{-6}	219.2227	5.6426 ± 0.1507	-0.6307 ± 0.2209	9.8940 ± 0.1111	—
	AdamW	5×10^{-1}	212.2359	5.6329 ± 0.0340	-0.6259 ± 0.0037	-0.0000 ± 0.0001	—
		5×10^{-2}	212.2369	5.6372 ± 0.0335	-0.6264 ± 0.0037	0.0000 ± 0.0010	—
		5×10^{-3}	212.2328	5.6382 ± 0.0186	-0.6265 ± 0.0038	0.0000 ± 0.0083	—
		5×10^{-4}	212.2152	5.6339 ± 0.0257	-0.6260 ± 0.0128	0.0010 ± 0.0324	—
		5×10^{-5}	212.2158	5.6316 ± 0.0221	-0.6257 ± 0.0174	3.4803 ± 0.0448	—
		5×10^{-6}	212.2147	5.6330 ± 0.0256	-0.6259 ± 0.0186	8.9169 ± 0.0480	—
BCE	SGD	5×10^{-1}	472.0906	4.2473 ± 0.1306	-5.6495 ± 0.1260	0.0036 ± 0.0000	-0.1683
		5×10^{-2}	471.6918	4.2916 ± 0.1134	-5.5975 ± 0.1029	0.0362 ± 0.0003	-0.1640
		5×10^{-3}	452.0422	4.6706 ± 0.1199	-5.1987 ± 0.0936	0.4031 ± 0.0037	-0.1269
		5×10^{-4}	358.9137	9.0244 ± 0.1190	-0.7897 ± 0.1281	4.8403 ± 0.0604	-0.0018
		5×10^{-5}	414.4364	11.0715 ± 0.1306	0.5388 ± 0.2787	7.0401 ± 0.0959	0.0008
		5×10^{-6}	424.8451	11.4847 ± 0.1327	0.7536 ± 0.3067	7.4372 ± 0.0973	0.0010
	AdamW	5×10^{-1}	483.3321	4.2399 ± 0.0308	-5.6315 ± 0.0215	0.0036 ± 0.0000	-0.1636
		5×10^{-2}	482.1844	4.2698 ± 0.0306	-5.5977 ± 0.0213	0.0358 ± 0.0003	-0.1598
		5×10^{-3}	470.6640	4.5928 ± 0.0281	-5.2753 ± 0.0201	0.3577 ± 0.0033	-0.1256
		5×10^{-4}	356.5036	7.7870 ± 0.0130	-2.0822 ± 0.0285	3.5514 ± 0.0330	-0.0083
		5×10^{-5}	347.1199	9.0726 ± 0.0303	-1.1593 ± 0.0304	4.9903 ± 0.0537	-0.0012
		5×10^{-6}	350.0225	9.2915 ± 0.0372	-1.0489 ± 0.0319	5.2119 ± 0.0599	-0.0006

The bias decay parameter $\lambda_{\boldsymbol{b}}$. In Sec. 4, we conducted experiments with fixed $\lambda_{\boldsymbol{b}} = 0$ and varying $\lambda_{\boldsymbol{b}} = 0.5, 0.05, 5\times10^{-3}, 5\times10^{-4}, 5\times10^{-5}, 5\times10^{-6}$ to further compare CE and BCE in neural collapse. Fig. 3 have visually shown the results, and we here present the numerical results in Tables S-8 and S-9. In our experiments, the classifier weight $\boldsymbol{W}$ and bias $\boldsymbol{b}$ are initialized using "kaiming uniform", i.e., He initialization (He et al., 2015). The initialized classifier bias is with zero-mean, i.e., $\frac{1}{K}\sum_{k=1}^{K} b_k \approx 0$, and we add them with $0, 1, 2, 3, 4, 5, 6, 8, 10$, respectively, to adjust their average value in the experiments with fixed $\lambda_{\boldsymbol{b}}$.

The batch size. In the proof of Theorem 1 and 2, it applied the feature matrix $\boldsymbol{H}$ including the features of all samples, to explore the the lower bounds for the CE and BCE losses, i.e.,

$$\boldsymbol{H} = \left[h_1^{(1)}, h_2^{(1)}, \cdots, h_n^{(1)}, h_1^{(2)}, h_2^{(2)}, \cdots, h_n^{(2)}, \cdots, h_1^{(K)}, h_2^{(K)}, \cdots, h_n^{(K)}\right]. \tag{56}$$

However, batch algorithm was applied in the practical training of deep models, and the batch size would affect the experimental numerical results. To verify this conclusion, a group of experiments were conducted with varying batch size. We trained ResNet18 on MNIST using SGD and AdamW using setting-1 and setting-2, while the initial learning rates were adjusted according to the batch size, $0.01 \times \frac{\text{batch size}}{128}$ for SGD and $0.001 \times \frac{\text{batch size}}{128}$ for AdamW. Fig. S-8 visually shows the distributions of the final classifier bias and the positive/negative decision scores, and Table S-10 lists the final numerical results. From these results, one can find that the bias results still conform to our analysis when batch size ≤ 1024, i.e., the classifier bias converges to zero in the training with CE loss and $\lambda_{\boldsymbol{b}} > 0$, and the clssifier bias separates the positive and negative decision scores in the training with BCE loss.

Table S-10: The numerical results of ResNet18 trained on MNIST with varying batch size and $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

Loss	Opt.	batch size	$\hat{\rho}$	s_{pos}	s_{neg}	$\hat{b}$	$\alpha(\hat{b})$
CE	SGD	16	100.9731	6.7176 ± 0.3270	-0.7538 ± 0.1950	-0.0074 ± 0.0523	—
		32	130.1404	6.3375 ± 0.2425	-0.7110 ± 0.1709	-0.0074 ± 0.0478	—
		64	168.6290	6.0159 ± 0.1562	-0.6737 ± 0.2052	-0.0074 ± 0.0547	—
		128	219.0960	5.6439 ± 0.1437	-0.6302 ± 0.2073	-0.0074 ± 0.0852	—
		256	285.6314	5.3200 ± 0.1586	-0.5936 ± 0.2070	-0.0074 ± 0.1259	—
		512	379.3403	4.9776 ± 0.2735	-0.5535 ± 0.2921	-0.0073 ± 0.2526	—
		1024	522.5523	4.6562 ± 1.3926	-0.5173 ± 0.8343	-0.0073 ± 1.0641	—
		2048	473.7898	3.5759 ± 2.6771	-0.3972 ± 2.0373	-0.0072 ± 1.8399	—
	AdamW	16	87.6451	6.5511 ± 0.0110	-0.7279 ± 0.0089	0.0003 ± 0.0211	—
		32	118.0328	6.2558 ± 0.0101	-0.6951 ± 0.0104	0.0003 ± 0.0253	—
		64	158.4980	5.9506 ± 0.0106	-0.6612 ± 0.0117	0.0002 ± 0.0293	—
		128	212.2180	5.6331 ± 0.0120	-0.6259 ± 0.0127	0.0001 ± 0.0328	—
		256	282.9370	5.3168 ± 0.0148	-0.5908 ± 0.0133	0.0000 ± 0.0357	—
		512	375.4274	4.9968 ± 0.0209	-0.5552 ± 0.0140	-0.0001 ± 0.0380	—
		1024	496.6912	4.6627 ± 0.0631	-0.5199 ± 0.0238	-0.0190 ± 0.0472	—
		2048	668.3063	4.3236 ± 0.3703	-0.4906 ± 0.2909	-0.0153 ± 0.2964	—
BCE	SGD	16	199.6890	6.1841 ± 0.3002	-5.9379 ± 0.2665	0.7828 ± 0.0223	-0.0660
		32	255.9898	6.1508 ± 0.2184	-5.2761 ± 0.1932	1.1506 ± 0.0214	-0.0546
		64	324.7408	6.2846 ± 0.1600	-4.4319 ± 0.1295	1.6456 ± 0.0254	-0.0399
		128	407.1362	6.4008 ± 0.1236	-3.4987 ± 0.1137	2.2170 ± 0.0308	-0.0268
		256	501.1286	6.6422 ± 0.1347	-2.5493 ± 0.1501	2.8605 ± 0.0740	-0.0167
		512	631.7796	6.6413 ± 0.2725	-1.9155 ± 0.2544	3.2338 ± 0.1859	-0.0127
		1024	816.6544	6.3274 ± 0.4653	-1.5393 ± 0.4515	3.3466 ± 0.3554	-0.0119
		2048	351.9647	1.7449 ± 2.4487	-0.5243 ± 1.6982	2.6332 ± 1.5391	0.0077
	AdamW	16	189.2794	6.5169 ± 0.0240	-5.3841 ± 0.0215	1.2651 ± 0.0119	-0.0457
		32	242.1592	6.7110 ± 0.0169	-4.5302 ± 0.0202	1.7885 ± 0.0167	-0.0322
		64	300.8807	7.1079 ± 0.0118	-3.4518 ± 0.0229	2.5261 ± 0.0234	-0.0188
		128	357.9696	7.7460 ± 0.0113	-2.1233 ± 0.0291	3.5134 ± 0.0337	-0.0086
		256	455.2137	7.6247 ± 0.0112	-1.6013 ± 0.0256	3.8010 ± 0.0325	-0.0068
		512	590.9918	7.2831 ± 0.0271	-1.3210 ± 0.0270	3.8500 ± 0.0375	-0.0064
		1024	790.8874	6.6204 ± 0.1011	-1.3148 ± 0.1126	3.5830 ± 0.0899	-0.0089
		2048	1019.6438	5.9625 ± 0.2969	-1.2607 ± 0.2750	3.3303 ± 0.2111	-0.0122

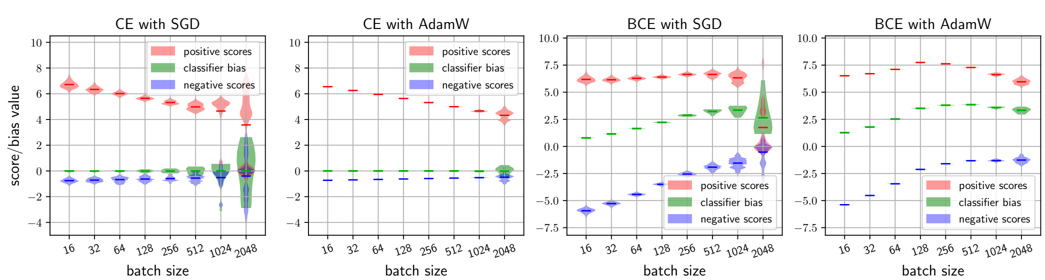


Figure S-8: The distributions of the final classifier bias and positive/negative decision scores for ResNet18 trained on MNIST with different batch sizes, while $\lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{H}} = \lambda_{\boldsymbol{b}} = 5 \times 10^{-4}$.

The decision score results are very different from that in the experiments with fixed batch size. For examples, in the training with CE loss and fixed batch size $= 128$, the positive and negative decision scores converge to about 5.64 and -0.63, respectively, and the value of $\hat{\rho} = \|\hat{\boldsymbol{W}}\|_F^2$ converge to about 219 and 212 in the training by SGD and AdamW, respectively, as shown in Tables S-8 and S-9. In contrast, these values varies as the batch size in the experiments with varying batch sizes.

In addition, the positive/negative decision scores did not converge to the theoretical values in Eq. (13) in our experiments; we believe it is resulted from the difference between the batch algorithm and the proof of Theorems. We roughly replaced n with $\frac{\text{batch size}}{K}$ in computing $\alpha(\hat{b})$.

S-2.3 EXPERIMENTAL RESULTS OF CLASSIFICATION

In the experiments of classification in Sec. 4.2, we train the models for 100 epochs. In each training, the model with best classification accuracy $\mathcal{A}$ is chosen as the final model, which was used to compute the uniform accuracy $\mathcal{A}_{\text{Uni}}$ presented in Table 2. In Table S-11 and S-12, we list their numerical results on the training and test dataset of CIFAR10 and CIFAR100. In these experiments, though the classification accuracy $\mathcal{A}$ of some models on the training datasets have reached 100%, neural collapse has not caused during the training. An obvious evidence is that both positive and negative decision scores have not converged, with large standard deviations, whether on the training set or testing set. The small standard deviations of the final classification bias might be more resulted from their initialization.

From Tables S-11 and S-12, one can find that, the gaps between the means of positive and negative decision scores of BCE-trained models are usually larger than that of CE-trained models, while in some cases, the standard deviations of the positive/negative decision scores of BCE-trained models are higher than that of CE-trained models. However, without any modification, the standard deviations and the gap between the positive and negative means cannot be precisely used to evaluate the intra-class compactness and inter-class distinctiveness. The decision score is calculated by the norm of the classifier vector and the feature vector, with the angle between them. The diverse $\hat{\rho}$ of CE-trained and BCE-trained models indicates different norms of the classifier vectors.

In Fig. 4, the all features are first projected into 2-dimension space from d-dimension space, and $d = 1024$ for ResNet50, which are then translated and scaled into the region of $[0, 1] \times [0, 1]$. We finally plot these feature points on the 2D plane.

Table S-11: The numerical results of ResNet18, ResNet50, DenseNet121 trained on CIFAR10 for classification.

$\mathcal{M}$	Op.	DA	Loss	classifier		on training data				on testing data	
				$\hat{\rho}$	$\hat{b}$	s_{pos}	s_{neg}	$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$	s_{pos}	s_{neg}
ResNet18	SGD	1	CE	34.86	-0.01 ± 0.03	14.9 ± 3.54	-1.68 ± 2.64	99.98	97.55	13.9 ± 4.73	-1.56 ± 2.89
			BCE	52.33	2.89 ± 0.03	12.9 ± 2.75	-9.70 ± 2.67	100.00	99.99	11.3 ± 4.93	-9.25 ± 3.30
		1&2	CE	12.59	-0.01 ± 0.02	3.23 ± 0.38	-0.37 ± 0.62	98.02	95.99	3.09 ± 0.61	-0.35 ± 0.70
			BCE	19.66	2.84 ± 0.02	3.86 ± 0.45	-0.86 ± 0.66	98.71	98.01	3.66 ± 0.80	-0.84 ± 0.77
	AdamW	1	CE	85.52	-0.00 ± 0.01	12.3 ± 3.60	-12.4 ± 4.12	99.99	99.57	10.9 ± 5.52	-12.1 ± 4.58
			BCE	113.9	2.16 ± 0.02	16.3 ± 2.95	-20.0 ± 4.50	100.00	100.00	14.2 ± 6.37	-19.0 ± 5.75
		1&2	CE	36.26	-0.01 ± 0.01	2.54 ± 0.18	-1.13 ± 0.38	99.96	99.88	2.41 ± 0.52	-1.12 ± 0.50
			BCE	44.16	2.14 ± 0.01	3.57 ± 0.20	-1.74 ± 0.38	99.96	99.94	3.34 ± 0.81	-1.72 ± 0.56
ResNet50	SGD	1	CE	18.74	0.00 ± 0.03	17.4 ± 3.16	-2.00 ± 3.30	99.99	98.09	16.1 ± 4.56	-1.86 ± 3.73
			BCE	29.07	2.83 ± 0.04	13.7 ± 2.33	-12.4 ± 3.07	99.99	99.98	11.9 ± 5.08	-11.8 ± 3.83
		1&2	CE	8.18	0.00 ± 0.04	3.28 ± 0.35	-0.39 ± 0.56	98.25	96.65	3.14 ± 0.63	-0.37 ± 0.66
			BCE	13.86	2.65 ± 0.03	3.68 ± 0.45	-1.08 ± 0.61	98.79	98.24	3.47 ± 0.85	-1.06 ± 0.75
	AdamW	1	CE	143.9	0.01 ± 0.02	16.7 ± 5.64	-18.6 ± 6.76	100.00	98.95	14.9 ± 7.87	-18.2 ± 7.23
			BCE	153.4	2.20 ± 0.01	21.9 ± 6.82	-28.5 ± 9.09	99.97	99.96	19.4 ± 10.1	-27.2 ± 10.4
		1&2	CE	79.80	0.00 ± 0.01	2.44 ± 0.25	-1.16 ± 0.33	99.96	99.89	2.28 ± 0.57	-1.16 ± 0.44
			BCE	102.6	2.14 ± 0.00	3.35 ± 0.24	-1.58 ± 0.44	99.95	99.94	3.16 ± 0.72	-1.55 ± 0.56
DenseNet121	SGD	1	CE	48.02	0.00 ± 0.02	10.5 ± 2.37	-1.16 ± 2.18	99.30	93.29	9.57 ± 3.41	-1.06 ± 2.41
			BCE	64.94	2.93 ± 0.03	9.06 ± 1.76	-6.05 ± 1.74	99.45	99.24	7.75 ± 3.63	-5.71 ± 2.32
		1&2	CE	14.99	0.00 ± 0.02	2.89 ± 0.67	-0.32 ± 0.67	91.20	86.80	2.77 ± 0.80	-0.30 ± 0.71
			BCE	19.60	2.86 ± 0.02	3.69 ± 0.88	-0.65 ± 0.73	92.38	90.28	3.52 ± 1.08	-0.62 ± 0.80
	AdamW	1	CE	139.4	0.00 ± 0.01	10.2 ± 3.06	-10.6 ± 4.44	99.97	98.48	8.70 ± 5.00	-10.4 ± 4.86
			BCE	156.6	2.17 ± 0.01	13.1 ± 2.78	-15.1 ± 4.22	99.97	99.97	10.9 ± 5.93	-14.4 ± 5.13
		1&2	CE	39.93	0.00 ± 0.01	2.31 ± 0.28	-1.28 ± 0.48	98.83	98.10	2.14 ± 0.64	-1.26 ± 0.58
			BCE	40.53	2.18 ± 0.01	3.40 ± 0.42	-1.65 ± 0.52	98.81	98.51	3.13 ± 0.94	-1.60 ± 0.67

Table S-12: The numerical results of ResNet18, ResNet50, DenseNet121 trained on CIFAR100 for classification.

$\mathcal{M}$	Opt.	DA	Loss	classifier		on training data				on testing data	
				$\hat{\rho}$	$\hat{b}$	s_{pos}	s_{neg}	$\mathcal{A}$	$\mathcal{A}_{\text{Uni}}$	s_{pos}	s_{neg}
ResNet18	SGD	1	CE	317.6	0.00 ± 0.02	15.8 ± 3.15	-0.18 ± 3.04	99.79	76.32	13.0 ± 5.06	-0.15 ± 3.06
			BCE	408.8	2.89 ± 0.02	9.39 ± 2.87	-10.0 ± 2.94	99.94	99.69	5.28 ± 5.95	-9.64 ± 3.06
		1&2	CE	138.8	0.00 ± 0.02	5.82 ± 1.33	-0.07 ± 0.98	88.26	73.67	5.09 ± 1.67	-0.06 ± 1.00
			BCE	163.7	2.89 ± 0.01	3.42 ± 1.44	-3.22 ± 0.99	88.56	80.23	2.64 ± 1.90	-3.19 ± 1.02
	AdamW	1	CE	1007.	0.00 ± 0.02	12.5 ± 4.18	-13.4 ± 5.16	99.98	92.02	7.47 ± 7.81	-13.1 ± 5.19
			BCE	1372.	2.14 ± 0.02	15.3 ± 4.85	-21.2 ± 6.34	99.98	99.97	7.05 ± 10.2	-19.7 ± 6.47
		1&2	CE	476.9	0.00 ± 0.02	4.49 ± 0.82	-2.04 ± 0.99	99.25	95.86	3.15 ± 1.77	-2.14 ± 1.09
			BCE	576.1	2.18 ± 0.02	3.67 ± 0.80	-4.13 ± 0.84	99.18	98.25	2.22 ± 1.84	-4.01 ± 0.96
ResNet50	SGD	1	CE	258.8	0.00 ± 0.01	17.7 ± 3.12	-0.19 ± 3.56	99.90	79.70	14.5 ± 5.17	-0.16 ± 3.58
			BCE	328.3	2.87 ± 0.01	10.0 ± 2.82	-11.6 ± 3.40	99.86	99.62	5.37 ± 6.20	-10.9 ± 3.46
		1&2	CE	102.7	0.00 ± 0.01	5.97 ± 1.41	-0.07 ± 1.07	87.46	72.45	5.29 ± 1.71	-0.06 ± 1.05
			BCE	118.4	2.86 ± 0.01	3.59 ± 1.48	-3.33 ± 0.98	89.17	81.80	2.75 ± 1.96	-3.29 ± 1.02
	AdamW	1	CE	2157.	0.00 ± 0.01	13.9 ± 5.49	-19.4 ± 7.18	99.98	87.42	8.29 ± 9.45	-19.2 ± 7.20
			BCE	2863.	2.15 ± 0.02	17.6 ± 4.92	-25.7 ± 7.34	99.98	99.97	8.49 ± 11.2	-23.5 ± 7.67
		1&2	CE	1334.	0.00 ± 0.02	4.42 ± 0.67	-1.96 ± 0.87	99.69	97.86	3.06 ± 1.90	-2.25 ± 1.07
			BCE	1440.	2.18 ± 0.02	3.81 ± 0.82	-4.27 ± 0.80	99.67	99.22	2.28 ± 1.96	-4.21 ± 0.93
DenseNet121	SGD	1	CE	337.9	-0.00 ± 0.02	12.9 ± 3.33	-0.12 ± 2.83	92.46	56.75	10.5 ± 4.75	-0.10 ± 2.82
			BCE	383.8	2.95 ± 0.02	6.03 ± 2.64	-7.83 ± 2.81	92.85	87.12	3.36 ± 4.28	-7.34 ± 2.91
		1&2	CE	143.2	-0.00 ± 0.02	4.62 ± 1.67	-0.04 ± 1.01	67.23	47.68	4.26 ± 1.84	-0.04 ± 1.01
			BCE	161.5	2.90 ± 0.01	2.14 ± 1.68	-2.95 ± 1.04	68.15	56.43	1.74 ± 1.85	-2.93 ± 1.05
	AdamW	1	CE	1090.	-0.00 ± 0.01	9.39 ± 3.69	-12.3 ± 4.98	99.89	83.67	4.74 ± 6.83	-12.1 ± 4.99
			BCE	1146.	2.17 ± 0.01	9.78 ± 2.72	-16.0 ± 4.77	99.86	99.55	3.66 ± 7.20	-14.6 ± 5.03
		1&2	CE	430.2	-0.00 ± 0.01	3.82 ± 1.13	-2.00 ± 1.00	91.18	80.57	2.85 ± 1.83	-2.07 ± 1.06
			BCE	474.5	2.20 ± 0.01	2.70 ± 1.16	-3.85 ± 0.89	90.66	85.83	1.79 ± 1.83	-3.82 ± 0.97

S-3 PROOF OF THEOREM 2

Zhou et al. (2022) have proved that the loss satisfying contrastive property can cause neural collapse. CE loss, focal loss, and label smoothing loss satisfy this property, while BCE does not, and we proof that BCE can also result in the neural collapse in this paper.

Definition S-1 *(Contrastive property (Zhou et al., 2022)). A loss function $\mathcal{L}(\boldsymbol{z})$ satisfies the contrastive property if there exists a function ϕ such that $\mathcal{L}(\boldsymbol{z})$ can be lower bounded by*

$$\mathcal{L}(\boldsymbol{z}) \geq \phi\Big(\sum_{\substack{j=1 \\ j\neq k}}^{K} \left(z_j - z_k\right)\Big) \tag{57}$$

where the equality holds only when $z_j = z_\ell$ for $\forall j, \ell \neq k$, and the function $\phi(x)$ satisfies

$$x^\star = \arg\min_x \phi(x) + c|x| \tag{58}$$

is unique for $\forall c > 0$, and $x^\star \leq 0$. ∎

Theorem S-3 *(Zhou et al., 2022) Assume that the feature dimension d is larger than the category number K, i.e., $d \geq K-1$, and $\mathcal{L}$ is satisfying the contrastive property. Then any global minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ of $f(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ defined using $\mathcal{L}$ with Eq. (3) obeys the following properties,*

$$\|\boldsymbol{w}^\star\| = \|\boldsymbol{w}_1^\star\| = \|\boldsymbol{w}_2^\star\| = \cdots = \|\boldsymbol{w}_K^\star\|, \tag{59}$$

$$\boldsymbol{h}_i^{(k)\star} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k^\star,\ \forall\, k \in [K],\ i \in [n], \tag{60}$$

$$\tilde{\boldsymbol{h}}_i^\star := \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{h}_i^{(k)\star} = \boldsymbol{0}, \forall\, i \in [n], \tag{61}$$

$$\boldsymbol{b}^\star = b^\star \boldsymbol{1}_K, \tag{62}$$

where either $b^\star = 0$ or $\lambda_{\boldsymbol{b}} = 0$. The matrix $\boldsymbol{W}^{\star T}$ forms a K-simplex ETF in the sense that

$$\frac{1}{\|\boldsymbol{w}^\star\|_2^2}\boldsymbol{W}^{\star T}\boldsymbol{W}^\star = \frac{K}{K-1}\Big(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^T\Big), \tag{63}$$

where $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$ denotes the identity matrix, $\boldsymbol{1}_K \in \mathbb{R}^K$ denotes the all ones vector. ∎

Theorem S-4 *Assume that the feature dimension d is larger than the number of classes K, i.e., $d \geq K-1$. Then any global minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ of*

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) := g_{\text{bce}}(\boldsymbol{W}\boldsymbol{H} - \boldsymbol{b}\boldsymbol{1}^T) + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2 \tag{64}$$

with

$$g_{\text{bce}}(\boldsymbol{W}\boldsymbol{H} - \boldsymbol{b}\boldsymbol{1}^T) := \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n} \mathcal{L}_{\text{bce}}(\boldsymbol{W}\boldsymbol{h}_i^{(k)} - \boldsymbol{b}, \boldsymbol{y}_k), \tag{65}$$

obeys the following

$$\|\boldsymbol{w}^\star\| = \|\boldsymbol{w}_1^\star\| = \|\boldsymbol{w}_2^\star\| = \cdots = \|\boldsymbol{w}_K^\star\|,\ \textit{and}\ \ \boldsymbol{b}^\star = b^\star\boldsymbol{1}, \tag{66}$$

$$\boldsymbol{h}_i^{(k)\star} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k^\star,\ \forall\, k \in [K],\ i \in [n],\ \textit{and}\ \ \tilde{\boldsymbol{h}}_i^\star := \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{h}_i^{(k)\star} = \boldsymbol{0}, \forall\, i \in [n], \tag{67}$$

and the matrix $\frac{1}{\|\boldsymbol{w}^\star\|_2}\boldsymbol{W}^{\star T}$ forms a K-simplex ETF in the sense that

$$\frac{1}{\|\boldsymbol{w}^\star\|_2^2}\boldsymbol{W}^{\star T}\boldsymbol{W}^\star = \frac{K}{K-1}\Big(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^T\Big), \tag{68}$$

where $b^\star$ is the solution of equation

$$\lambda_{\boldsymbol{b}} b = \frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)}. \tag{69}$$

Proof *According to Lemma 1, any critical point $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of $f(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ satisfies*

$$\boldsymbol{W}^T\boldsymbol{W} = \frac{\lambda_{\boldsymbol{H}}}{\lambda_{\boldsymbol{W}}}\boldsymbol{H}^T\boldsymbol{H}. \tag{70}$$

Let $\rho = \|\boldsymbol{W}\|_F^2$ for any critical point $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$. Then, according to Lemma 3, for any $c_1, c_2 \geq 0$,

$$\begin{aligned} &f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) \\ \geq\ &\left[\lambda_{\boldsymbol{W}} - \left(\frac{2K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\sqrt{\frac{n\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}}\right]\rho - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C \end{aligned} \tag{71}$$

where

$$C = \frac{c_1}{1+c_1}\log\left(\frac{1+c_1}{c_1}\right) + \frac{\log(1+c_1)}{1+c_1} + \frac{K-1}{1+c_2}\left[c_2\log\left(\frac{1+c_2}{c_2}\right) + \log(1+c_2)\right]. \tag{72}$$

According to Lemma 4, the inequality (71) achieves its equality when

$$\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = \cdots = \|\boldsymbol{w}_K\|, \text{ and } \boldsymbol{b} = b^\star\mathbf{1}, \tag{73}$$

$$\boldsymbol{h}_i^{(k)} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k,\ \forall\, k \in [K],\ i \in [n], \text{ and } \tilde{\boldsymbol{h}}_i = \frac{1}{K}\sum_{k=1}^K \boldsymbol{h}_i^{(k)} = \mathbf{0}, \forall\, i \in [n], \tag{74}$$

$$\boldsymbol{W}\boldsymbol{W}^T = \frac{\rho}{K-1}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\right), \tag{75}$$

$$c_1 = \exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b^\star\right), \text{ and } c_2 = \exp\left(b^\star + \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right), \tag{76}$$

where $b^\star$ is the solution of equation

$$\lambda_{\boldsymbol{b}} b = \frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)}. \tag{77}$$

According to Lemma 5, the equation (77) in terms of b has only one solution $b^\star$.

Given $\lambda_{\boldsymbol{W}}, \lambda_{\boldsymbol{H}}, \lambda_{\boldsymbol{b}} > 0$, $f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ is convex function, which achieves its minimum with finite $\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}$. Therefore, the right side of inequality (71) is a consistent when $\lambda_{\boldsymbol{W}}, \lambda_{\boldsymbol{H}}, \lambda_{\boldsymbol{b}}$ are fixed and Eqs. (73, 74, 75, 76) hold, which finishes the proof. ∎

Lemma 1 *Any critical point $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of Eq. (64) obeys*

$$\boldsymbol{W}^T\boldsymbol{W} = \frac{\lambda_{\boldsymbol{H}}}{\lambda_{\boldsymbol{W}}}\boldsymbol{H}\boldsymbol{H}^T, \text{ and } \|\boldsymbol{W}\|_F^2 = \frac{\lambda_{\boldsymbol{H}}}{\lambda_{\boldsymbol{W}}}\|\boldsymbol{H}\|_F^2. \tag{78}$$

Proof *See Lemma D.2 in reference (Zhu et al., 2021).* ∎

Lemma 2 *For any $\boldsymbol{h}_i^{(k)}$ with $c_1, c_2 > 0$, the BCE loss is lower bounded by*

$$\mathcal{L}_{\text{bce}}(\boldsymbol{W}\boldsymbol{h}_i^{(k)}, \boldsymbol{y}_k) \geq \frac{1}{1+c_1}\Big(-\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} + b_k\Big) + \frac{1}{1+c_2}\sum_{\substack{j=1\\ j\neq k}}^K\Big(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j\Big) + C, \tag{79}$$

where

$$C = \frac{c_1}{1+c_1}\log\left(\frac{1+c_1}{c_1}\right) + \frac{\log\left(1+c_1\right)}{1+c_1} + \frac{K-1}{1+c_2}\left[c_2\log\left(\frac{1+c_2}{c_2}\right) + \log\left(1+c_2\right)\right]. \tag{80}$$

The inequality becomes an equality when

$$\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j = \boldsymbol{w}_\ell^T \boldsymbol{h}_i^{(k)} - b_\ell, \ \forall\, j, \ell \neq k, \tag{81}$$

and

$$c_1 = \exp\Big(\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} - b_k\Big), \tag{82}$$

$$c_2 = \exp\Big(b_j - \boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)}\Big), \ \ j \neq k. \tag{83}$$

Proof *By the concavity of the* $\log(1+\mathrm{e}^x)$*, we have,*

$$\sum_{k=1}^{K} \log\big(1+\exp(x_k)\big) \geq K \log\left(1+\exp\Big(\frac{\sum_{k=1}^K x_k}{K}\Big)\right), \ \ \forall x_k \in \mathbb{R}. \tag{84}$$

Then,

$$\begin{aligned}
&\mathcal{L}_{\text{bce}}(\boldsymbol{W}\boldsymbol{h}_i^{(k)} + \boldsymbol{b}, \boldsymbol{y}_k) && (85)\\
=\ & \log\big(1+\exp(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k)\big) + \sum_{\substack{j=1\\ j\neq k}}^{K} \log\big(1+\exp(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j)\big) && (86)\\
\geq\ & \log\big(1+\exp(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k)\big) + \big(K-1\big)\log\left[1+\exp\left(\frac{\sum_{\substack{j=1\\ j\neq k}}^{K}\big(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\big)}{K-1}\right)\right] && (87)\\
=\ & \log\left(\frac{c_1}{1+c_1}\frac{1+c_1}{c_1} + \frac{1+c_1}{1+c_1}\exp\Big(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k\Big)\right)\\
& + \big(K-1\big)\log\left[\frac{c_2}{1+c_2}\frac{1+c_2}{c_2} + \frac{1+c_2}{1+c_2}\exp\left(\frac{\sum_{\substack{j=1\\ j\neq k}}^{K}\big(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\big)}{K-1}\right)\right] && (88)\\
\geq\ & \frac{c_1}{1+c_1}\log\left(\frac{1+c_1}{c_1}\right) + \frac{1}{1+c_1}\log\left(\big(1+c_1\big)\exp\Big(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k\Big)\right)\\
& + \big(K-1\big)\left\{\frac{c_2}{1+c_2}\log\left(\frac{1+c_2}{c_2}\right) + \frac{1}{1+c_2}\log\left[\big(1+c_2\big)\exp\left(\frac{\sum_{\substack{j=1\\ j\neq k}}^{K}\big(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\big)}{K-1}\right)\right]\right\} && (89)\\
=\ & \frac{1}{1+c_1}\Big(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k\Big) + \frac{1}{1+c_2}\sum_{\substack{j=1\\ j\neq k}}^{K}\Big(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\Big)\\
& + \underbrace{\frac{c_1}{1+c_1}\log\left(\frac{1+c_1}{c_1}\right) + \frac{\log\big(1+c_1\big)}{1+c_1} + \frac{K-1}{1+c_2}\left[c_2\log\left(\frac{1+c_2}{c_2}\right) + \log\big(1+c_2\big)\right]}_{C}. && (90)
\end{aligned}$$

The first inequality is derived from the concavity of $\log(1+\mathrm{e}^x)$*, i.e., Eq. (84), which achieves the equality if and only if*

$$\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j = \boldsymbol{w}_\ell^T \boldsymbol{h}_i^{(k)} - b_\ell, \ \forall\, j, \ell \neq k \in [K]. \tag{91}$$

The second inequality is derived from the concavity of $\log(x)$*,*

$$\log\big(tx_1 + (1-t)x_2\big) \geq t\log(x_1) + (1-t)\log(x_2), \ \ \forall x_1, x_2 \in \mathbb{R} \ \ \textit{and} \ \ t \in [0,1], \tag{92}$$

which achieves its equality if and only if $x_1 = x_2$*, or* $t = 0$*, or* $t = 1$*. Then, the second inequality holds for any* $c_1, c_2 \geq 0$*, and it becomes an equality if and only if*

$$\frac{1+c_1}{c_1} = (1+c_1)\exp\Big(-\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} + b_k\Big) \ \ \textit{or} \ \ c_1 = 0 \ \ \textit{or} \ \ c_1 = +\infty, \ \ \textit{and} \tag{93}$$

$$\frac{1+c_2}{c_2} = (1+c_2)\exp\left(\frac{\sum_{\substack{j=1\\ j\neq k}}^{K}\big(\boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)} - b_j\big)}{K-1}\right) \ \ \textit{or} \ \ c_1 = 0 \ \ \textit{or} \ \ c_1 = +\infty. \tag{94}$$

It is trivial when $c_1 = 0$ or $c_1 = +\infty$ or $c_2 = 0$ or $c_2 = +\infty$. Then, we get

$$c_1 = \exp\left(\boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} - b_k\right), \tag{95}$$

$$c_2 = \exp\left(\frac{\sum_{\substack{j=1\\j\neq k}}^K \left(b_j - \boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)}\right)}{K-1}\right) \overset{(91)}{=} \exp\left(b_j - \boldsymbol{w}_j^T \boldsymbol{h}_i^{(k)}\right), \quad j \neq k, \tag{96}$$

which are desired. ∎

Lemma 3 *Let*

$$\boldsymbol{W} = \left[\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K\right]^T \in \mathbb{R}^{K\times d}, \tag{97}$$

$$\boldsymbol{H} = \left[h_1^{(1)}, \cdots, h_n^{(1)}, \cdots, h_1^{(K)}, \cdots, h_n^{(K)}\right] \in \mathbb{R}^{d\times N} \tag{98}$$

with $N = nK$. Then, for any critical point $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of Eq. (64) and any $c_1, c_2 \geq 0$, we have

$$f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) \geq \left[\lambda_{\boldsymbol{W}} - \left(\frac{1}{N(1+c_2)} + \frac{1}{N(1+c_1)}\right)\sqrt{\frac{n\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}}\right]\rho - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C \tag{99}$$

with $C = \frac{c_1}{1+c_1}\log\left(\frac{1+c_1}{c_1}\right) + \frac{\log(1+c_1)}{1+c_1} + \frac{K-1}{1+c_2}\left[c_2 \log\left(\frac{1+c_2}{c_2}\right) + \log(1+c_2)\right]$.

Proof *According to Lemma 1, Eq. (79) holds for any $c_1, c_2 > 0$ and any $\boldsymbol{h}_i^{(k)}$ with $k \in [K],\ i \in [n]$. We take the same c_1 and c_2 for all $\boldsymbol{h}_i^{(k)}$, then*

$$(1+c_1)(1+c_2)\left[g_{\text{bce}}(\boldsymbol{W}\boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}^T) - C\right] \tag{100}$$

$$= (1+c_1)(1+c_2)\left[\frac{1}{N}\sum_{k=1}^K\sum_{i=1}^n \mathcal{L}_{\text{bce}}(\boldsymbol{W}\boldsymbol{h}_i^{(k)} + \boldsymbol{b}, \boldsymbol{y}_k) - C\right] \tag{101}$$

$$\geq \frac{1}{N}\sum_{k=1}^K\sum_{i=1}^n\left[(1+c_2)\left(-\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} + b_k\right) + (1+c_1)\sum_{\substack{j=1\\j\neq k}}^K\left(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j\right)\right] \tag{102}$$

$$= \frac{1+c_1}{N}\sum_{k=1}^K\sum_{i=1}^n\sum_{\substack{j=1\\j\neq k}}^K\left(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j\right) - \frac{1+c_2}{N}\sum_{k=1}^K\sum_{i=1}^n\left(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\right) \tag{103}$$

$$= \frac{1+c_1}{N}\sum_{k=1}^K\sum_{i=1}^n\left(\sum_{j=1}^K\left(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j\right) - \boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} + b_k\right) - \frac{1+c_2}{N}\sum_{k=1}^K\sum_{i=1}^n\left(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\right) \tag{104}$$

$$= \frac{1+c_1}{N}\sum_{k=1}^K\sum_{i=1}^n\sum_{j=1}^K\left(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j - \boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} + b_k\right) + \frac{1+c_1}{N}\sum_{k=1}^K\sum_{i=1}^n\sum_{\substack{j=1\\j\neq k}}^K\left(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\right)$$
$$- \frac{1+c_2}{N}\sum_{k=1}^K\sum_{i=1}^n\left(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\right) \tag{105}$$

$$= \frac{1+c_1}{N}\left[\sum_{k=1}^K\sum_{i=1}^n\sum_{j=1}^K\left(\boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)} - b_j\right) - \sum_{k=1}^K\sum_{i=1}^n\sum_{j=1}^K\left(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\right)\right]$$
$$+ \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N}\right)\sum_{k=1}^K\sum_{i=1}^n \boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N}\right)\sum_{k=1}^K\sum_{i=1}^n b_k \tag{106}$$

$$= \frac{1+c_1}{N}\sum_{i=1}^n\left[\sum_{k=1}^K\left(\sum_{j=1}^K \boldsymbol{w}_k^T\boldsymbol{h}_i^{(j)} - K\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)}\right)\underbrace{-\sum_{k=1}^K\sum_{j=1}^K b_j + \sum_{k=1}^K\sum_{j=1}^K b_k}_{0}\right]$$

$$+ \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \sum_{k=1}^{K} \sum_{i=1}^{n} \boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k \tag{107}$$

$$= \frac{1+c_1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \boldsymbol{w}_k^T \left(\tilde{\boldsymbol{h}}_i - \boldsymbol{h}_i^{(k)} \right) + \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \sum_{k=1}^{K} \sum_{i=1}^{n} \boldsymbol{w}_k^T \boldsymbol{h}_i^{(k)} - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k \tag{108}$$

where $\tilde{\boldsymbol{h}}_i = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{h}_i^{(k)}$.

According to the AM-GM inequality, we have

$$\boldsymbol{u}^T \boldsymbol{v} \geq -\frac{c}{2} \|\boldsymbol{u}\|_2^2 - \frac{1}{2c} \|\boldsymbol{v}\|_2^2, \ \forall\, \boldsymbol{u},\, \boldsymbol{v} \in \mathbb{R}^d, \ \forall\, c \geq 0. \tag{109}$$

Then,

$$\begin{aligned}
& (1+c_1)(1+c_2)\left[g_{\text{bce}}(\boldsymbol{W}\boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}^T) - C\right] \\
\geq\ & -\frac{1+c_1}{n} \left(\frac{c_3}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_3} \sum_{i=1}^{n} \sum_{k=1}^{K} \left\| \tilde{\boldsymbol{h}}_i - \boldsymbol{h}_i^{(k)} \right\|_2^2 \right) \\
& - \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \left(\frac{c_4}{2} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_4} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{h}_i^{(k)}\|_2^2 \right) \\
& - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k
\end{aligned} \tag{110}$$

$$\begin{aligned}
=\ & -\frac{1+c_1}{n} \left[\frac{c_3}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_3} \sum_{i=1}^{n} \left(\sum_{k=1}^{K} \left\| \boldsymbol{h}_i^{(k)} \right\|_2^2 - K \|\tilde{\boldsymbol{h}}_i\|_2^2 \right) \right] \\
& - \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \left(\frac{c_4}{2} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_4} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{h}_i^{(k)}\|_2^2 \right) \\
& - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k
\end{aligned} \tag{111}$$

$$\begin{aligned}
=\ & -\frac{1+c_1}{n} \left(\frac{c_3}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_3} \sum_{i=1}^{n} \sum_{k=1}^{K} \left\| \boldsymbol{h}_i^{(k)} \right\|_2^2 \right) \\
& - \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \left(\frac{c_4}{2} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{w}_k\|_2^2 + \frac{1}{2c_4} \sum_{k=1}^{K} \sum_{i=1}^{n} \|\boldsymbol{h}_i^{(k)}\|_2^2 \right) \\
& - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k + \frac{1+c_1}{2nc_3} \sum_{i=1}^{n} K \|\tilde{\boldsymbol{h}}_i\|_2^2
\end{aligned} \tag{112}$$

$$\begin{aligned}
=\ & -\frac{1+c_1}{n} \left(\frac{nc_3}{2} \|\boldsymbol{W}\|_F^2 + \frac{1}{2c_3} \|\boldsymbol{H}\|_F^2 \right) \\
& - \left(\frac{1+c_1}{N}(K-1) - \frac{1+c_2}{N} \right) \left(\frac{nc_4}{2} \|\boldsymbol{W}\|_F^2 + \frac{1}{2c_4} \|\boldsymbol{H}\|_F^2 \right) \\
& - \left(\frac{1+c_1}{K}(K-1) - \frac{1+c_2}{K} \right) \sum_{k=1}^{K} b_k + \frac{1+c_1}{2nc_3} \sum_{i=1}^{n} K \|\tilde{\boldsymbol{h}}_i\|_2^2
\end{aligned} \tag{113}$$

and the inequality becomes an equality if and only if

$$c_3 \boldsymbol{w}_k = \boldsymbol{h}_i^{(k)} - \tilde{\boldsymbol{h}}_i, \ \ \forall\, k \in [K],\ i \in [n], \ \ \textit{and} \tag{114}$$

$$c_4 \boldsymbol{w}_k = -\boldsymbol{h}_i^{(k)}, \qquad \forall\, k \in [K],\ i \in [n], \tag{115}$$

which can be achieved only when $\tilde{\boldsymbol{h}}_i = \mathbf{0}$.

Let $\rho = \|\boldsymbol{W}\|_F^2$. Then, by using Lemma 1, we have $\|\boldsymbol{H}\|_F^2 = \frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\rho$, and

$$\begin{aligned}
& f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) \\
=\ & g_{\text{bce}}(\boldsymbol{W}\boldsymbol{H} + \boldsymbol{b}\mathbf{1}^T) + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2 && (116) \\
\geq\ & -\frac{1}{n(1+c_2)}\left(\frac{nc_3}{2}\|\boldsymbol{W}\|_F^2 + \frac{1}{2c_3}\left\|\boldsymbol{H}\right\|_F^2\right) \\
& - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2}\|\boldsymbol{W}\|_F^2 + \frac{1}{2c_4}\left\|\boldsymbol{H}\right\|_F^2\right) \\
& - \left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)\sum_{k=1}^K b_k + \frac{1}{2nc_3(1+c_2)}\sum_{i=1}^n K\left\|\tilde{\boldsymbol{h}}_i\right\|_2^2 + C \\
& + \frac{\lambda_{\boldsymbol{W}}}{2}\rho + \frac{\lambda_{\boldsymbol{H}}}{2}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\rho + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2 && (117) \\
=\ & -\frac{1}{n(1+c_2)}\left(\frac{nc_3}{2}\rho + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\rho\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2}\rho + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\rho\right) \\
& - \left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)\sum_{k=1}^K b_k + \frac{1}{2nc_3(1+c_2)}\sum_{i=1}^n K\left\|\tilde{\boldsymbol{h}}_i\right\|_2^2 + C + \lambda_{\boldsymbol{W}}\rho + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2 && (118) \\
=\ & \left[\lambda_{\boldsymbol{W}} - \frac{1}{n(1+c_2)}\left(\frac{nc_3}{2} + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2} + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2 - \left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)\sum_{k=1}^K b_k + \frac{1}{2nc_3(1+c_2)}\sum_{i=1}^n K\left\|\tilde{\boldsymbol{h}}_i\right\|_2^2 + C && (119) \\
=\ & \left[\lambda_{\boldsymbol{W}} - \frac{1}{n(1+c_2)}\left(\frac{nc_3}{2} + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2} + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& + \frac{\lambda_{\boldsymbol{b}}}{2}\sum_{k=1}^K\left[b_k - \frac{1}{\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)\right]^2 - \frac{1}{2\lambda_{\boldsymbol{b}}}\sum_{k=1}^K\left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)^2 \\
& + \frac{1}{2nc_3(1+c_2)}\sum_{i=1}^n K\left\|\tilde{\boldsymbol{h}}_i\right\|_2^2 + C && (120) \\
\geq\ & \left[\lambda_{\boldsymbol{W}} - \frac{1}{n(1+c_2)}\left(\frac{nc_3}{2} + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2} + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& + \frac{\lambda_{\boldsymbol{b}}}{2}\sum_{k=1}^K\left[b_k - \frac{1}{\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right)\right]^2 - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C && (121) \\
\geq\ & \left[\lambda_{\boldsymbol{W}} - \frac{1}{n(1+c_2)}\left(\frac{nc_3}{2} + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2} + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C, && (122)
\end{aligned}$$

where the inequality (121) achieves its equality if and only if

$$\tilde{\boldsymbol{h}}_i = \mathbf{0}, \ \ \forall i \in [n], \tag{123}$$

and the inequality (122) becomes an equality whenever either

$$\lambda_{\boldsymbol{b}} = 0 \;\; or \;\; b_k = \frac{1}{\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right), \;\; \forall k \in [K]. \tag{124}$$

Due to $\lambda_{\boldsymbol{b}} > 0$ and c_1, c_2 are same for any $k \in [K]$, therefore

$$b_k = b_j, \;\; \forall k, j \in [K]. \tag{125}$$

Based on Eqs. (114) and (123), we have

$$c_3 \boldsymbol{w}_k = \boldsymbol{h}_i^{(k)} \Rightarrow c_3^2 = \frac{\sum_{i=1}^n \sum_{k=1}^K \|\boldsymbol{h}_i^{(k)}\|_2^2}{\sum_{i=1}^n \sum_{k=1}^K \|\boldsymbol{w}_k\|_2^2} = \frac{\|\boldsymbol{H}\|_F^2}{n\|\boldsymbol{W}\|_F^2} = \frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}} \Rightarrow c_3 = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}; \tag{126}$$

similarly, from Eq. (115), we get

$$c_4 \boldsymbol{w}_k = -\boldsymbol{h}_i^{(k)} \Rightarrow c_4^2 = \frac{\sum_{i=1}^n \sum_{k=1}^K \|\boldsymbol{h}_i^{(k)}\|_2^2}{\sum_{i=1}^n \sum_{k=1}^K \|\boldsymbol{w}_k\|_2^2} = \frac{\|\boldsymbol{H}\|_F^2}{n\|\boldsymbol{W}\|_F^2} = \frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}} \Rightarrow c_4 = -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}. \tag{127}$$

Plugging them into Eq. (119), we get

$$\begin{aligned}
& f_{\text{bce}}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) \\
\geq \; & \left[\lambda_{\boldsymbol{W}} - \frac{1}{n(1+c_2)}\left(\frac{nc_3}{2} + \frac{1}{2c_3}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right) - \left(\frac{K-1}{N(1+c_2)} - \frac{1}{N(1+c_1)}\right)\left(\frac{nc_4}{2} + \frac{1}{2c_4}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C & (128) \\
= \; & \left[\lambda_{\boldsymbol{W}} - \left(\frac{1}{n(1+c_2)} - \frac{K-1}{N(1+c_2)} + \frac{1}{N(1+c_1)}\right)\left(\frac{n}{2}\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}} + \frac{1}{2}\sqrt{\frac{n\lambda_{\boldsymbol{H}}}{\lambda_{\boldsymbol{W}}}}\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}\right)\right]\rho \\
& - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C & (129) \\
= \; & \left[\lambda_{\boldsymbol{W}} - \left(\frac{1}{n(1+c_2)} - \frac{K-1}{N(1+c_2)} + \frac{1}{N(1+c_1)}\right)\sqrt{\frac{n\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}}\right]\rho \\
& - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C & (130) \\
= \; & \left[\lambda_{\boldsymbol{W}} - \left(\frac{1}{N(1+c_2)} + \frac{1}{N(1+c_1)}\right)\sqrt{\frac{n\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}}}}\right]\rho - \frac{1}{2K\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{1+c_2} - \frac{1}{1+c_1}\right)^2 + C & (131)
\end{aligned}$$

which is desired. ∎

Lemma 4 *Under the same assumptions of Lemma 3, the lower bound in Eq. (99) is achieved for any critical point $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of Eq. (64) if and only if the following hold*

$$\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = \cdots = \|\boldsymbol{w}_K\|, \;\; and \;\; \boldsymbol{b} = b^\star \boldsymbol{1}, \tag{132}$$

$$\boldsymbol{h}_i^{(k)} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k, \; \forall\, k \in [K], \; i \in [n], \;\; and \;\; \tilde{\boldsymbol{h}}_i = \frac{1}{K}\sum_{k=1}^K \boldsymbol{h}_i^{(k)} = \boldsymbol{0}, \forall\, i \in [n], \tag{133}$$

$$\boldsymbol{W}\boldsymbol{W}^T = \frac{\rho}{K-1}\left(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^T\right), \tag{134}$$

$$c_1 = \exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b^\star\right), \;\; and \;\; c_2 = \exp\left(b^\star + \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right), \tag{135}$$

where $b^\star$ is the solution of equation

$$\lambda_{\boldsymbol{b}} b = \left[\frac{K-1}{K\left(1 + \exp\left(b + \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1 + \exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b\right)\right)}\right]. \tag{136}$$

Proof *With the proof of Lemma 3, to achieve the lower bound, it needs at least Eqs. (114), (115), and (123) to hold, i.e.,*

$$\tilde{\boldsymbol{h}}_i = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{h}_i^{(k)} = \mathbf{0},\ \forall\, i\in[n],\ \ \textit{and}\ \ \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k = \boldsymbol{h}_i^{(k)},\ \forall\, k\in[K],\ i\in[n], \tag{137}$$

and further implies

$$\sum_{k=1}^{K}\boldsymbol{w}_k = \sqrt{\frac{n\lambda_{\boldsymbol{H}}}{\lambda_{\boldsymbol{W}}}}\sum_{k=1}^{K}\boldsymbol{h}_i^{(k)} = \mathbf{0}. \tag{138}$$

Then,

$$c_1 = \exp\Big(\boldsymbol{w}_k^T\boldsymbol{h}_i^{(k)} - b_k\Big) = \exp\Big(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\|\boldsymbol{w}_k\|_2^2 - b_k\Big),\ \ \forall k\in[K], \tag{139}$$

$$c_2 = \exp\Big(b_j - \boldsymbol{w}_j^T\boldsymbol{h}_i^{(k)}\Big) = \exp\Big(b_j - \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k^T\boldsymbol{w}_j\Big),\ \ \forall j\neq k\in[K], \tag{140}$$

Since that c_1, c_2 are chosen to be the same for any $j\neq k\in[K]$, therefore,

$$\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\|\boldsymbol{w}_k\|_2^2 - b_k = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\|\boldsymbol{w}_j\|_2^2 - b_j,\ \ \forall k,j\in[K], \tag{141}$$

$$\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k^T\boldsymbol{w}_j - b_j = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\boldsymbol{w}_k^T\boldsymbol{w}_\ell - b_\ell,\ \ \forall j\neq \ell\in[K], \forall k\in[K], \tag{142}$$

With the proof of Lemma 2, to achieve the lower bound, it needs at least Eqs. (91) to hold, then,

$$\begin{aligned}
&\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\|\boldsymbol{w}_k\|_2^2 - b_k \\
\overset{(138)}{=}\quad & -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\sum_{\substack{j=1\\ j\neq k}}\boldsymbol{w}_j^T\boldsymbol{w}_k - b_k && (143)\\
\overset{(142)}{=}\quad & -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\sum_{\substack{\ell=1\\ \ell\neq k}}^{K}\boldsymbol{w}_k^T\boldsymbol{w}_\ell - b_k + \sum_{\substack{j=1\\ j\neq k}}(b_\ell - b_j) && (144)\\
=\quad & -(K-1)\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\underbrace{\boldsymbol{w}_k^T\boldsymbol{w}_\ell}_{\ell\neq k} - 2b_k + (K-1)b_\ell - K\bar{b} && (145)\\
\overset{(141,142)}{\Longrightarrow}\quad & -2b_k + (K-1)b_\ell - K\bar{b} = -2b_\ell + (K-1)b_j - K\bar{b} && (146)\\
\Longleftrightarrow\quad & b_k = b_\ell,\ \ \forall \ell\neq k\in[K], && (147)
\end{aligned}$$

which is conforming to Eq. (125) when $\lambda_{\boldsymbol{b}} > 0$. Then, combining with Eqs. (141) and (138),

$$\|\boldsymbol{w}_k\|_2^2 = \|\boldsymbol{w}_j\|_2^2 = \frac{\|\boldsymbol{W}\|_F^2}{K} = \frac{\rho}{K},\ \ \forall k,j\in[K], \tag{148}$$

$$\|\boldsymbol{w}_k\|_2^2 = -(K-1)\sum_{\substack{j=1\\ j\neq k}}^{K}\boldsymbol{w}_k^T\boldsymbol{w}_j \Rightarrow \boldsymbol{w}_k^T\boldsymbol{w}_j = -\frac{1}{K-1}\frac{\rho}{K},\ \ \forall j\neq k\in[K]. \tag{149}$$

Therefore,

$$\boldsymbol{W}\boldsymbol{W}^T = \frac{\rho}{K-1}\Big(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\Big). \tag{150}$$

Plugging (148) and (149) into (139) and (140)

$$c_1 = \exp\Big(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} - b\Big), \tag{151}$$

$$c_2 = \exp\Big(b + \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\Big), \tag{152}$$

where $b = b_k = b_j$. When $\lambda_{\boldsymbol{b}} > 0$, substitute Eqs. (151) and (152) into (124), we have

$$b = \frac{1}{\lambda_{\boldsymbol{b}}}\left(\frac{K-1}{K(1+c_2)} - \frac{1}{K(1+c_1)}\right) \tag{153}$$

$$= \frac{1}{\lambda_{\boldsymbol{b}}}\left[\frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)}\right]. \tag{154}$$

When $\lambda_{\boldsymbol{b}} = 0$, substitute Eq. (137) into

$$\frac{\partial f_{\text{bce}}}{\partial b_k} = \frac{1}{nK}\left(n - \sum_{j=1}^{K}\sum_{i=1}^{n}\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k\boldsymbol{h}_i^{(j)}+b_k}}\right) = 0,\ \ \forall k \in [K], \tag{155}$$

we have

$$0 = \frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)}, \tag{156}$$

by combining with Eqs. (148) and (149). ∎

Lemma 5 *The equation*

$$\lambda_{\boldsymbol{b}}\, b = \frac{K-1}{K\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)} - \frac{1}{K\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)} \tag{157}$$

has only one solution.

Proof *A number $b^\star$ is a solution of equation (157) if and only if it is a solution of*

$$\overbrace{\lambda_{\boldsymbol{b}}Kb + \frac{1}{1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)}}^{\beta_1(b)} = \overbrace{\frac{K-1}{1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)}}^{\beta_2(b)}. \tag{158}$$

When $\lambda_{\boldsymbol{b}} > 0$,

$$\beta_1(b) \to -\infty,\ \ \beta_2(b) \to K-1 \quad \textit{as}\ \ b \to -\infty \tag{159}$$

$$\beta_1(b) \to +\infty,\ \ \beta_2(b) \to 0 \quad \textit{as}\ \ b \to +\infty, \tag{160}$$

and if $\lambda_{\boldsymbol{b}} = 0$,

$$\beta_1(b) = 0,\ \ \beta_2(b) \to K-1 \quad \textit{as}\ \ b \to -\infty \tag{161}$$

$$\beta_1(b) \to +\infty,\ \ \beta_2(b) \to 0 \quad \textit{as}\ \ b \to +\infty. \tag{162}$$

Therefore, the curves of $\beta_1(b)$ and $\beta_2(b)$ must intersect at least once in the plane, i.e., the equations (157) and (158) have solutions.

In addition,

$$\frac{\mathrm{d}\beta_1(b)}{\mathrm{d}b} = \lambda_{\boldsymbol{b}}K + \frac{\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)}{\left(1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}-b\right)\right)^2} > 0, \tag{163}$$

$$\frac{\mathrm{d}\beta_2(b)}{\mathrm{d}b} = -\frac{(K-1)\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)}{\left(1+\exp\left(b+\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}\right)\right)^2} < 0, \tag{164}$$

i.e., $\beta_1(b)$ is strictly increasing, while $\beta_2(b)$ is strictly decreasing. Therefore, they can intersect at only one point. ∎

Lemma 6 *When the class number $K > 2$ and*

$$\lambda_{\boldsymbol{b}}\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1}+\frac{1}{2(K-1)} > \frac{1}{1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1}\right)}, \tag{165}$$

the final critical bias $b^\star$ could uniformly separate the all positive decision scores

$$\left\{\boldsymbol{w}_k^{\star T}\boldsymbol{h}_i^{(k)\star} : k \in [K], i \in [n]\right\} \tag{166}$$

and the all negative decision scores

$$\left\{\boldsymbol{w}_j^{\star T}\boldsymbol{h}_i^{(k)\star} : k, j \in [K], i \in [n], k \neq j\right\}, \tag{167}$$

where

$$\boldsymbol{W}^\star = \left[\boldsymbol{w}_1^\star, \boldsymbol{w}_2^\star, \cdots, \boldsymbol{w}_K^\star\right]^T \tag{168}$$

$$\boldsymbol{H}^\star = \left[\boldsymbol{h}_1^{(1)\star}, \cdots, \boldsymbol{h}_n^{(1)\star}, \cdots, \boldsymbol{h}_1^{(K)\star}, \cdots, \boldsymbol{h}_n^{(K)\star}\right] \tag{169}$$

$$\boldsymbol{b}^\star = (b^\star, b^\star, \cdots, b^\star)^T = b^\star\mathbf{1}_K \tag{170}$$

form the critical point of function $f(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ in Eq. (64).

Proof *According to Lemma 4, for the critical point $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$, we have*

$$\boldsymbol{w}_k^{\star T}\boldsymbol{h}_i^{(k)\star} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}, \quad \forall k \in [K], i \in [n] \tag{171}$$

$$\boldsymbol{w}_j^{\star T}\boldsymbol{h}_i^{(k)\star} = -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}, \quad \forall k, j \in [K], i \in [n], k \neq j. \tag{172}$$

Let $b_{\text{neg}} = -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)}, b_{\text{pos}} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K}$. Then, the critical $b^\star$ separating the all positive and negative score if and only if

$$b_{\text{neg}} = -\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K(K-1)} < b^\star < \sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K} = b_{\text{pos}} \tag{173}$$

which, according to the proof of Lemma 5, is equivalent to

$$\beta_1(b_{\text{neg}}) < \beta_2(b_{\text{neg}}) \text{ and } \beta_1(b_{\text{pos}}) > \beta_2(b_{\text{pos}}). \tag{174}$$

Due to

$$\beta_1(b_{\text{neg}}) < \beta_2(b_{\text{neg}}) \Leftrightarrow -\lambda_{\boldsymbol{b}}\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1} + \frac{1}{1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1}\right)} < \frac{K-1}{2}$$

$$\Leftarrow \frac{1}{1+\mathrm{e}^0} < \frac{K-1}{2} \Leftarrow 2 < K \tag{175}$$

$$\beta_1(b_{\text{pos}}) > \beta_2(b_{\text{pos}}) \Leftrightarrow \lambda_{\boldsymbol{b}}\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\rho + \frac{1}{2} > \frac{K-1}{1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1}\right)} \tag{176}$$

$$\Leftrightarrow \lambda_{\boldsymbol{b}}\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1} + \frac{1}{2(K-1)} > \frac{1}{1+\exp\left(\sqrt{\frac{\lambda_{\boldsymbol{W}}}{n\lambda_{\boldsymbol{H}}}}\frac{\rho}{K-1}\right)}, \tag{177}$$

it completes the proof. ∎

S-4 More discussion about decision scores in the training

In the training, the decision scores are updated along the negative direction of their gradients during the back propagation stage, i.e.,

$$\boldsymbol{w}_k \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_k \boldsymbol{h}^{(k)} - \eta \frac{\partial f_\mu(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial\big(\boldsymbol{w}_k \boldsymbol{h}^{(k)}\big)}, \ \forall k \in [K], \tag{178}$$

$$\boldsymbol{w}_j \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_j \boldsymbol{h}^{(k)} - \eta \frac{\partial f_\mu(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})}{\partial\big(\boldsymbol{w}_j \boldsymbol{h}^{(k)}\big)}, \ \forall j \neq k \in [K], \tag{179}$$

where η is the learning rate, and $\mu \in \{\mathrm{ce}, \mathrm{bce}\}$.

In the training with CE, the updating formulas are

$$\boldsymbol{w}_k \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_k \boldsymbol{h}^{(k)} + \eta \Bigg(1 - \frac{\mathrm{e}^{\boldsymbol{w}_k \boldsymbol{h}^{(k)} - b_k}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}^{(k)} - b_\ell}}\Bigg), \tag{180}$$

$$\boldsymbol{w}_j \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_j \boldsymbol{h}^{(k)} - \eta \frac{\mathrm{e}^{\boldsymbol{w}_j \boldsymbol{h}^{(k)} - b_j}}{\sum_\ell \mathrm{e}^{\boldsymbol{w}_\ell \boldsymbol{h}^{(k)} - b_\ell}}. \tag{181}$$

Then, for the samples with diverse initial decision scores, it is difficult to update their decision scores to the similar level, if they own the similar predicted probabilities belong to each categories.

In the training with BCE, the updating formulas are

$$\boldsymbol{w}_k \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_k \boldsymbol{h}^{(k)} + \eta \Bigg(1 - \frac{1}{1 + \mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}^{(k)} + b_k}}\Bigg), \tag{182}$$

$$\boldsymbol{w}_j \boldsymbol{h}^{(k)} \leftarrow \boldsymbol{w}_j \boldsymbol{h}^{(k)} - \eta \frac{1}{1 + \mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}^{(k)} + b_j}}. \tag{183}$$

Then, for the sample with small positive decision score $\boldsymbol{w}_k \boldsymbol{h}^{(k)}$, its predicted probability $\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}^{(k)} + b_k}}$ to its category will be also small, and the score updating amplitude $\eta\Big(1 - \frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}^{(k)} + b_k}}\Big)$ will be large; in contrary, for the sample with large positive score $\boldsymbol{w}_k \boldsymbol{h}^{(k)}$, the probability $\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}^{(k)} + b_k}}$ will be also large, and the updating amplitude $\eta\Big(1 - \frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_k \boldsymbol{h}^{(k)} + b_k}}\Big)$ will be small. This property helps to update the all positive decision scores to be in uniform high level.

Similarly, for the sample with large negative decision score $\boldsymbol{w}_j \boldsymbol{h}^{(k)}$, its predicted probability $\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}^{(k)} + b_j}}$ to other category will be also large, so is the score updating amplitude $\eta \frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}^{(k)} + b_j}}$; in contrary, for the sample with small negative score $\boldsymbol{w}_j \boldsymbol{h}^{(k)}$, the probability $\frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}^{(k)} + b_k}}$ will be small, so is the updating amplitude $\eta\Big(1 - \frac{1}{1+\mathrm{e}^{-\boldsymbol{w}_j \boldsymbol{h}^{(k)} + b_k}}\Big)$. This property helps to update the all negative decision scores to be in uniform low level.