Cross Entropy versus Label Smoothing: A Neural Collapse Perspective

Anonymous authors
Paper under double-blind review

Abstract

Label smoothing loss is a widely adopted technique to mitigate overfitting in deep neural networks. This paper studies label smoothing from the perspective of Neural Collapse (NC), a powerful empirical and theoretical framework which characterizes model behavior during the terminal phase of training. We first show empirically that models trained with label smoothing converge faster to neural collapse solutions and attain a stronger level of neural collapse compared to those trained with cross-entropy loss. Furthermore, we show that at the same level of NC1, models under label smoothing loss exhibit intensified NC2. These findings provide valuable insights into the impact of label smoothing on model performance and calibration. Then, leveraging the unconstrained feature model, we derive closed-form solutions for the global minimizers under both label smoothing and cross-entropy losses. We show that models trained with label smoothing have a lower conditioning number and, therefore, theoretically converge faster. Our study, combining empirical evidence and theoretical results, not only provides nuanced insights into the differences between label smoothing and cross-entropy losses, but also serves as an example of how the powerful neural collapse framework can be used to improve our understanding of DNNs.

1 Introduction

The effectiveness of a deep neural network (DNN) hinges significantly on the choice of the loss function during training. While cross-entropy loss is one of the most popular choices for classification tasks, many alternatives with improved empirical performance have been proposed. Among these, label smoothing (Szegedy et al., 2016; Müller et al., 2019; Lukasik et al., 2020) has emerged as a common technique to enhance the performance of DNNs. Instead of supervising the model training with one-hot key labels, label smoothing introduces a soft target label by blending the hard target label with a uniform distribution over the labels. This procedure is generally understood as a means of regularisation for improving the model's generalizability.

In this paper, we examine the benefits of label smoothing from the perspective of neural collapse (Papyan et al., 2020), a new and powerful framework for obtaining an improved understanding of DNNs.

Background and Related Work

Deep neural networks (DNNs) typically consist of a non-linear feature extractor and a linear classification layer. Neural collapse, first observed in (Papyan et al., 2020), occurs in DNNs for classification tasks where the data is balanced and cross-entropy loss is employed. It characterizes the geometric properties of the features produced by the feature extractor and the weight vectors of the classifier during the terminal phase of training (TPT):

- NC1: The learned features of samples from the same class approach their respective class means.
- NC2: The collapsed features from different classes and the classification weight vectors approach the vertices of a simplex equiangular tight frame (ETF).
- NC3: Up to rescaling, the linear classifier weights approach the corresponding class means, giving a self-dual configuration.

Other than cross-entropy loss, neural collapse phenomena have been observed and studied under the mean-squared loss as well (Han et al., 2021; Poggio & Liao, 2020; Zhou et al., 2022a). Furthermore, research has extended neural collapse investigations to scenarios involving imbalanced data (Fang et al., 2021; Hong & Ling, 2023; Thrampoulidis et al., 2022; Dang et al., 2023; Hong & Ling, 2023) and cases with a large number of classes (Jiang et al., 2023).

In addition to being observed empirically, neural collapse has also been proven to arise mathematically. Beginning with Mixon et al. (2020), a series of studies have employed approximation models such as the Unconstrained Feature Models (UFMs) Mixon et al. (2020) or layer-peeled models (Fang et al., 2021) to provide theoretical evidence for the emergence of neural collapse. These optimization models simplify a DNN by treating the last-layer features as free variables to optimize over, which is justified due to the expressiveness of DNNs (Hornik, 1991; Cybenko, 1989; Lu et al., 2017; Shaham et al., 2018). Theoretical advancements in neural collapse not only enhance our understanding of DNNs but also inspire new techniques to improve their performance in diverse applications, such as imbalanced learning (Xie et al., 2023; Liu et al., 2023), transfer learning (Galanti et al., 2021; Li et al., 2022), and continual learning (Yu et al., 2022; Yang et al., 2023), etc.

Several studies have explored UFMs under various loss functions and regularization strategies (Wojtowytsch et al., 2020; Zhu et al., 2021; Dang et al., 2023; Lu & Steinerberger, 2022; Tirer & Bruna, 2022; Tirer et al., 2023; Yaras et al., 2022; Zhou et al., 2022b). Particularly, Zhou et al.(Zhou et al., 2022b) established that under the UFM model, the global minimizers for a wide range of loss functions, including cross-entropy loss and label smoothing loss, have the idealized neural collapse properties and that the UFM model has a benevolent landscape allowing the global minimizer to be effectively attained using iterative algorithms. However, their investigation does not provide insights into why label smoothing consistently outperforms cross-entropy loss, or why label smoothing converges faster during training. In this paper, leveraging the neural collapse framework, we conduct an in-depth investigation of training under these loss functions, aiming to explain the reasons behind the observed superiority of label smoothing loss over cross-entropy loss.

Our Contributions

Our study begins with a thorough empirical comparison between cross-entropy (CE) loss and label smoothing (LS) loss during the training process. Specifically, we carefully study how the last layer features and linear classifiers evolve during training. Our findings are as follows:

1. Compared with CE loss, models trained under LS loss exhibit accelerated convergence in terms of training error, testing error, and neural collapse metrics. Furthermore, they converge to a more pronounced level of NC1 and NC2.
2. Along with accelerated convergence, label smoothing loss maintains a distinct balance between NC1 and NC2. Notably, as compared with CE loss, LS loss results in a more pronounced level of NC2 when reaching a comparable level of NC1. We argue that this phenomenon originates from the implicit inductive bias introduced by LS loss, which equalizes the logits of all non-target classes and thus promotes the emergence of a simplex ETF structure in both the learned features and classification weight. We posit that the emphasis on NC2 in label smoothing (LS), which promotes maximally separable features between classes, enhances the model's generalization performance. Conversely, an excessive level of NC1 may cause the features to overly specialize in the training data, hindering the model's ability to generalize effectively.
3. Models trained with LS loss exhibit improved calibration Guo et al. (2017) by implicitly regularizing classification weights and features during training. However, if temperature scaling Guo et al. (2017) is applied as post-processing to counteract the regularization effect, label smoothing can lead to deteriorated model calibration. This is because an excessive level of NC1 under label smoothing can cause the model to be overconfident in its predictions, even when they are incorrect, thereby negatively impacting model calibration.

Of equal importance to the empirical results, we perform a mathematical analysis of the convergence properties of the UFM models under CE and LS losses. While Zhou et al. Zhou et al. (2022a) demonstrate that, for a broad class of loss functions, including CE and LS losses, the global minimizers exhibit neural collapse properties, the authors neither derive the exact form of these global minimizers nor thoroughly examine the landscape around them.

1. We first derive closed-form solutions for the global minimizers under both CE and LS loss functions, which explicitly depend on the smoothing hyperparameter δ.
2. Utilizing these closed-form solutions, we conduct a second-order theoretical analysis of the optimization landscape around their respective global optimizers. Within the UFM context, our mathematical analysis reveals that LS exhibits a more well-conditioned landscape around the global minimum, which facilitates the faster convergence observed in our empirical study.

This paper provides a significantly deeper understanding of why LS provides better convergence and performance than CE loss. Additionally, the paper illustrates how the powerful framework of neural collapse and its associated mathematical models can be employed to gain a more nuanced understanding of the "why" of DNNs. However, we acknowledge that theoretical evidence about how neural collapse impacts model generalization is lacking in this study. We hope that our work will inspire future research into the intricate interplay between neural collapse, convergence speed, and model generalizability.

2 Preliminaries

2.1 The Problem Setup

A deep neural network is comprised of two key components: a feature extractor and a linear classifier. The feature extractor $\phi_{\boldsymbol{\theta}}(\cdot)$ is a nonlinear mapping that maps the input $\boldsymbol{x}$ to the corresponding feature embedding $\boldsymbol{h} := \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^d$. Meanwhile, the linear classifier involves a weight matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K] \in \mathbb{R}^{d\times K}$ and a bias vector $\boldsymbol{b} \in \mathbb{R}^K$. Consequently, the architecture of a deep neural network is captured by the following equation:

$$f_\Theta(\boldsymbol{x}) := \boldsymbol{W}^\top \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{b}, \tag{1}$$

where $\Theta := \{\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}\}$ represents the set of all model parameters. In this work, we consider training a deep neural network using a balanced dataset denoted as $\{(\boldsymbol{x}_{ki}, \boldsymbol{y}_{ki})\}_{1\leq k\leq K, 1\leq i\leq n}$. This dataset consists of samples distributed across K distinct classes, with n samples allocated per class. Here, $\boldsymbol{x}_{ki}$ represents the i-th sample from the k-th class, and $\boldsymbol{y}_{ki}$ is a one-hot vector with unity solely in the k-th entry. Our objective is to learn the parameters Θ by minimizing the empirical risk over the total $N = nK$ training samples:

$$\min_{\Theta} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} l\left(f_\Theta(\boldsymbol{x}_{ki}), \boldsymbol{y}_{ki}\right) + \frac{\lambda}{2}\|\Theta\|_F^2, \tag{2}$$

where $l(\cdot,\cdot)$ denotes the chosen loss function, and $\lambda > 0$ is the regularization parameter (i.e., the weight decay parameter).

2.2 Training Losses

To simplify the notation, we use $\boldsymbol{z} = \boldsymbol{W}^\top \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{b}$ to represent the network's output logit vector for a given input $\boldsymbol{x}$, and $\boldsymbol{p} = \text{Softmax}(\boldsymbol{z})$ to denote the predicted distribution from the model. The cross-entropy between the target distribution $\boldsymbol{y}$ and the predicted distribution $\boldsymbol{p}$ is defined as $l_{CE}(\boldsymbol{y}, \boldsymbol{p}) = -\sum_k y_k \log(p_k)$, where $\boldsymbol{y}$ is a one-hot vector with a value of 1 in the dimension corresponding to the target class. In contrast, LS loss minimizes the cross-entropy between the smoothed soft label $\boldsymbol{y}^\delta$ and the predicted distribution $\boldsymbol{p}$, denoted as $l_{CE}(\boldsymbol{y}^\delta, \boldsymbol{p})$, where the soft label $\boldsymbol{y}^\delta = (1-\delta)\boldsymbol{y} + (\delta/K)\mathbf{1}_K$ combines the hard ground truth label $\boldsymbol{y}$ with a uniform distribution over the labels. Here $\mathbf{1}_K$ denotes the K-dimensional vector of all ones. The hyper-parameter δ determines the degree of smoothing.

For simplicity, we use the following formulation to represent both CE loss and LS loss:

$$l_{CE}(\boldsymbol{y}^\delta, \boldsymbol{p}) = -\sum_k y_k^\delta \log(p_k), \tag{3}$$

where $y_k^\delta = (1-\delta)y_k + \delta/K$. The provided loss corresponds to CE loss when $\delta = 0$, and for any other value of $\delta \in (0,1)$, it represents LS loss.

3 Empirical Analysis of Cross-Entropy and Label Smoothing Losses

This section conducts a comprehensive empirical comparison between cross-entropy (CE) loss and label smoothing (LS) loss from the perspective of neural collapse. In Section 3.1, we examine the convergence of the model to neural collapse solutions under both CE and LS losses. Our results demonstrate that while models under both loss functions converge to neural collapse, LS loss induces faster convergence and reaches a more pronounced level of neural collapse. Beyond convergence rates, Section 3.2 delves into the dynamics of convergence, revealing that LS loss introduces a bias toward solutions with a symmetric simplex ETF structure, thereby enforcing NC2. In Section 3.3, we attempt to understand the impact of label smoothing on model calibration from the perspective of neural collapse. In addition, we also investigate how the smoothing hyperparameter δ influences neural collapses during the model training in the appendix B.4.

Experiment Setup. We conducted experiments on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100, and STL-10 (Coates et al., 2011). For consistency with prior studies (Papyan et al., 2020; Zhu et al., 2021), we adopt ResNet-18 (He et al., 2016) as the backbone network for CIFAR-10 and ResNet-50 (He et al., 2016) as the backbone for CIFAR-100 and STL-10.

To focus on the behaviors associated with neural collapse and minimize the influence of other factors, we adopt standard preprocessing techniques without introducing any data augmentation for the training data. To comprehensively analyze the models' behavior during TPT, we extend the training period to 800 epochs. For all datasets, we employ a default batch size of 128, stochastic gradient descent with a momentum of 0.9, and a learning rate that is initialized at 0.05 and undergoes multi-step decay, decreasing by a factor of 0.1 at epochs 150 and 350. We use a default weight decay of 5×10^{-4}, except for the experiments in Section 3.3, where a weight decay value of 1×10^{-4} is used.

Metrics for Measuring NC. We assess neural collapse in the last-layer features and the classifiers using metrics based on the properties introduced in Section 1, with metrics similar to those presented in (Papyan et al., 2020). For convenience, we denote the global mean and class mean of the last-layer features as:

$$\boldsymbol{h}_G = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\boldsymbol{h}_{ki}, \quad \bar{\boldsymbol{h}}_k = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{h}_{ki}, (1 \le k \le K).$$

Within class variability (NC1) measures the relative magnitude of the within-class covariance matrix $\Sigma_W := \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}(\boldsymbol{h}_{ki} - \bar{\boldsymbol{h}}_k)(\boldsymbol{h}_{ki} - \bar{\boldsymbol{h}}_k)^\top$ compared to the between-class covariance matrix $\Sigma_B := \frac{1}{K}\sum_{k=1}^{K}(\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G)(\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G)^\top$ of the last-layer features. It is formulated as:

$$NC_1 = \frac{1}{K} trace\left(\Sigma_W \Sigma_B^\dagger\right), \tag{4}$$

where $\Sigma_B^\dagger$ denotes the pseudo inverse of Σ_B.

Distance to simplex ETF (NC2) quantifies the difference between the product of the classifier weight matrix and the centered class mean feature matrix, and a simplex ETF, defined as follows[1]:

$$NC_2 := \left\| \frac{\boldsymbol{W}^\top \overline{\boldsymbol{H}}}{\left\|\boldsymbol{W}^\top \overline{\boldsymbol{H}}\right\|_F} - \frac{1}{\sqrt{K-1}}\left(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^\top\right) \right\|_F, \tag{5}$$

where $\overline{\boldsymbol{H}} = [\bar{\boldsymbol{h}}_1 - \boldsymbol{h}_G, \cdots, \bar{\boldsymbol{h}}_K - \boldsymbol{h}_G] \in \mathbb{R}^{d\times K}$ represents centered class mean matrix.

Self-duality (NC3) measures the distance between the classifier weight matrix $\boldsymbol{W}$ and the centered class-means $\overline{\boldsymbol{H}}$:

$$NC_3 := \left\| \frac{\boldsymbol{W}}{\left\|\boldsymbol{W}\right\|_F} - \frac{\overline{\boldsymbol{H}}}{\left\|\overline{\boldsymbol{H}}\right\|_F} \right\|_F. \tag{6}$$

---

[1] When the bias $\boldsymbol{b}$ is an all-zero vector or a constant vector, $NC_2 = 0$ implies that the average logit matrix, defined as $\overline{\boldsymbol{Z}} = \boldsymbol{W}^\top \overline{\boldsymbol{H}} + \boldsymbol{b}\boldsymbol{1}_K^T$, satisfies $\overline{\boldsymbol{Z}} = a\left(\boldsymbol{I} - \frac{1}{K}\boldsymbol{1}_K\boldsymbol{1}_K^\top\right)$ for some constant a, i.e., $\overline{\boldsymbol{Z}}$ is a simplex ETF.

It is evident that when NC_2 in (5) and NC_3 in (6) both reach zero, the matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ form the same simplex ETF up to some scaling factor. Thus, our definitions of NC1-NC3 capture the same concepts as the definitions in (Papyan et al., 2020) and (Zhu et al., 2021). We say that neural collapse occurs if NC1, NC2, and NC3 collectively approach zero during the Terminal Phase of Training (TPT).

3.1 Terminal Phase Training under Label Smoothing and Cross-Entropy Loss

In this section, we investigate the distinct behaviors exhibited by models trained under cross-entropy (CE) and label smoothing (LS) losses during TPT. We conduct experiments on CIFAR10, CIFAR100, and STL10 datasets, with LS loss employing a default smoothing hyperparameter of $\delta = 0.05$. We visualize the training process in Figure 1.

The leftmost column in Figure 1 (with a detailed view provided in Figure 5 in the Appendix) illustrates the progression of training and testing errors throughout the training process. As expected, models trained under LS loss exhibit lower testing errors compared to those under CE loss for all datasets, showcasing an improvement in model generalizability with LS loss. Furthermore, models trained with LS loss exhibit faster convergence for both training and testing errors. Figure 1 additionally presents the three neural collapse metrics for models trained under both cross-entropy and label smoothing losses. While the values of NC3, representing the alignment of $\boldsymbol{W}$ and $\boldsymbol{H}$, remain consistently low and comparable under both loss functions, models with LS loss exhibit faster convergence in both NC1 and NC2 and eventually reach lower levels for both NC1 and NC2. We provide a theoretical explanation for these empirical findings in Section 4. Additionally, in Section 3.3, we establish a connection between NC1 and model calibration under both loss functions.

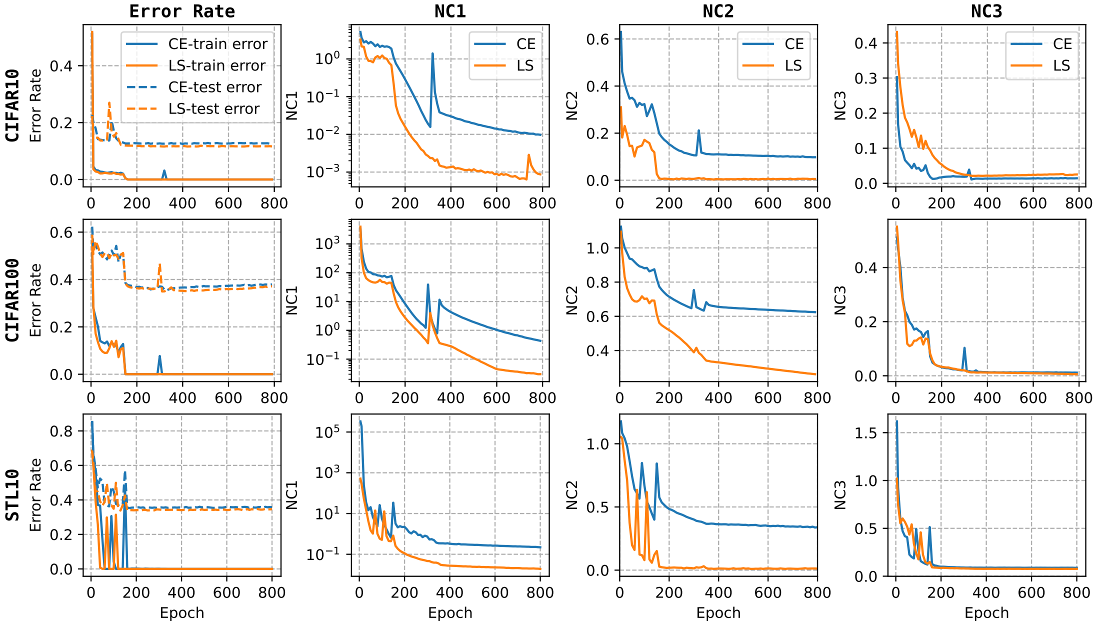

Figure 1: **Neural collapse with cross-entropy and label smoothing losses**. Columns from left to right represent the model's error rate (Train/Test), NC1, NC2, and NC3 under CE and LS losses.

3.2 Label Smoothing Induces Enhanced NC2

Along with the faster convergence to neural collapse under LS loss, we further observe that LS loss maintains a distinct balance between NC1 and NC2 throughout training. Specifically, we find that for the same level of NC1, LS consistently provides a more pronounced manifestation of NC2 compared to CE Loss. Figure 2 provides scatter plots of NC1 and NC2 under both loss functions, with the NC metrics recorded every 10 training

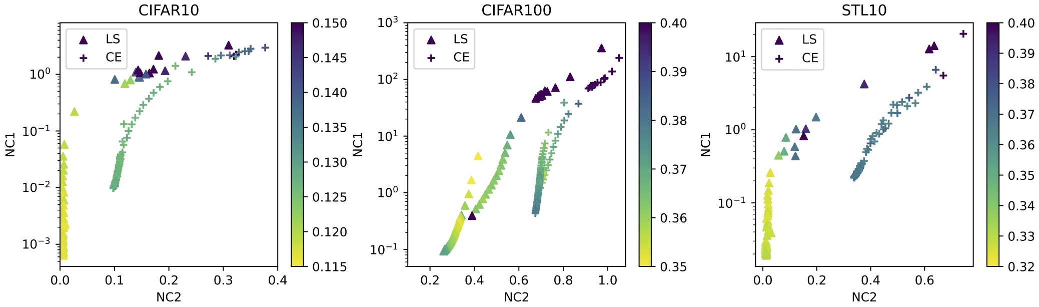

Figure 2: Scatter plots for **NC1 vs. NC2 under CE/LS losses** with colors indicating the testing error rate, for CIFAR10 (left), CIFAR100 (middle), and STL10 (right) datasets.

epochs. Across all three datasets, the data points under LS loss consistently position to the left of those for CE loss. This observation suggests that at equivalent levels of NC1, LS loss induces an intensified level of NC2.

To gain insight into this phenomenon, we now closely examine the formulations of cross-entropy and label smoothing loss. Given an input $\boldsymbol{x}$, the output logit and the predicted distribution are equal to $\boldsymbol{z} = f_\Theta(\boldsymbol{x})$ and $\boldsymbol{p} = \text{Softmax}(\boldsymbol{z})$, respectively. As per Equation 3, assuming the observation $\boldsymbol{x}$ is from the k-th class, CE loss is formulated as $l_{CE} = -\log(p_k)$. To minimize cross-entropy loss, the emphasis is solely on making the logit of the target class larger than the logits of non-target classes *without* constraining the logit variation among the non-target classes.

On the other hand, LS with smoothing hyperparameter δ is given by

$$l_{LS} = -\left(\left(1 - \frac{K-1}{K}\delta\right)\log(p_k) + \frac{\delta}{K}\sum_{l \neq k}\log(p_l)\right).$$

According to Jensen's inequality, we have

$$\frac{1}{K-1}\sum_{l \neq k}\log(p_l) \leq \log\left(\frac{1}{K-1}\sum_{l \neq k}p_l\right) = \log\left(1 - p_k\right),$$

with equality achieved only if $p_l = p_{l'}$ (for $l, l' \neq k$). This implies that LS loss reaches its minimum only if the predicted probabilities for non-target classes are all equal. *Thus, LS loss strengthens the inductive bias towards equalizing the logits of the non-target classes.* This property aligns with the definition of NC_2 in (5), explaining why label smoothing loss reinforces NC2. As demonstrated in Section B.1 in the appendix, for deep neural networks with L2 regularization, the convergence of NC_2 as defined in (5) to zero indicates that both the classification weight $\boldsymbol{W}$ and class mean features $\overline{\boldsymbol{H}}$ converge to a simplex ETF structure. The convergence of classifier weights and class mean features towards a simplex ETF promotes the development of maximally separable features and classifiers, a critical factor for enhancing overall model performance. We posit that the improvement in model generalizability observed with LS loss can be attributed to the prioritization of enhancing NC2 over NC1 during the training process. Nonetheless, this observation primarily serves as supporting evidence. Theoretical evidence and more rigorous validation of the benefits of NC2 remain topics for future research.

3.3 Label Smoothing and Model Calibration

DNN models often suffer from poor calibration, where the assigned probability values to class labels tend to overestimate the actual likelihood of correctness. This issue arises from the high capacity of DNN models, making them prone to overfitting the training data. Guo et al. (Guo et al., 2017) introduced the expected calibration error (ECE) as a metric for assessing model calibration. Additionally, they proposed temperature scaling as an effective post-processing technique for improving calibration, which involves dividing a network's logits by a scalar $T > 0$ before applying softmax, thereby softening (for $T > 1$) or sharpening (for $T < 1$) the predicted probability to make it more aligned with the true confidence levels.

Previous studies (Müller et al., 2019) have highlighted the effectiveness of label smoothing in improving model calibration. Conversely, our empirical study reveals intriguing findings: without temperature scaling, label smoothing (with properly tuned smoothing parameter δ) enhances model calibration compared to cross-entropy loss, *but when temperature scaling is applied as post-processing, label smoothing leads to inferior model calibration.* This inconsistency motivates us to investigate the underlying reason. In this section, we provide an explanation from the perspective of neural collapse.

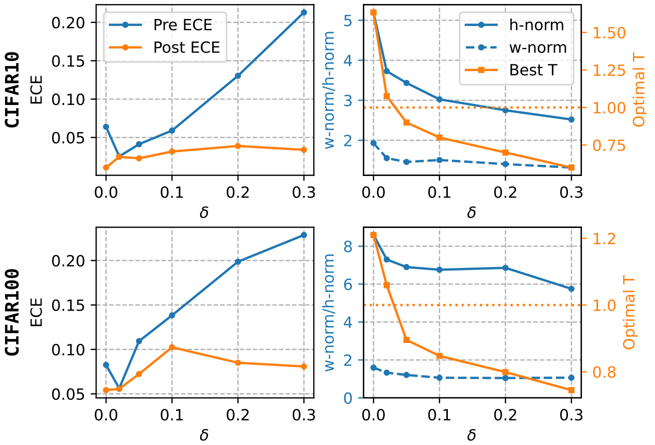

Figure 3: Plots of ECE Pre/Post temperature scaling (left column), alongside feature and classification vector norms and optical T (right column) for CIFAR10 and CIFAR100 datasets with varying values of δ.

Regularization effect of label smoothing on weight and feature norms. We trained ResNet18 on CIFAR10 and ResNet50 on CIFAR100 using label smoothing loss with various smoothing hyperparameters ranging from 0 to 0.3. Then, we evaluated models' testing ECE both before and after temperature scaling, as depicted in Figure 3. Our observations reveal that compared to cross-entropy loss ($\delta = 0$), LS loss with properly chosen δ lead to lower model calibration error (ECE), while larger values of δ may degrade model calibration. This is because tuning the smoothing hyperparameter δ has a similar effect to tuning temperature T in temperature scaling. Particularly, as shown empirically in the right column Figure 3 and theoretically in Theorem 4.1, increasing δ decreases both the feature and classification vector norms [2], thereby reducing the confidence level of model predictions. Consequently, the corresponding optimal temperature T decreases alongside the increase in δ. In fact, there exists a smoothing hyperparameter δ^* for which the optimal T equals 1, making temperature scaling post-processing unnecessary. Selecting $\delta > \delta^*$ results in an excessively low confidence level in the model's predictions, leading to a notable increase in test ECE.

However, if we apply temperature scaling to counteract the regularization effect of LS loss on feature and classification weight, models trained with LS loss demonstrate higher test ECE (Post ECE as shown in the left column of Figure 3) compared to models trained with CE loss. We further investigate how this phenomenon is attributed to the more pronounced level of NC1 under LS loss.

Excessive NC1 adversely impacts model calibration. To investigate the impact of NC1 on model calibration, we analyze how miscalibration occurs during training, using CIFAR10 data with CE and LS loss ($\delta = 0.05$) as examples. We specifically partition the test set into two subgroups: those correctly classified and those incorrectly classified by the model. In Figure 4, the left column illustrates the average CE loss and the average entropy of the model predictions for both subgroups of test samples. The middle section of the figure showcases the NC1 metric for the entire test dataset, as well as for each of the subgroups individually. Meanwhile, the right column of Figure 4 presents the training and testing misclassification errors, accompanied by the test set ECE.

In Figure 4, models trained with CE and LS losses show similar trends. For correctly classified test samples, both test loss and average entropy exhibit a consistent decreasing trend, indicating the model's increasing confidence in its correct predictions. Conversely, for incorrectly classified test samples, although the average loss starts to increase after 350 epochs, the overall downward trend of entropy suggests the model's growing confidence in its incorrect predictions. Moreover, the middle plot in Figure 4 indicates that NC1 for both

---

[2] The feature norm is computed as the average norm of each class mean feature, represented by $\sum_{k=1}^{K} \|\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G\|/K$. The classification vector norm is determined as the mean norm of the weight vector for each class given by $\sum_{k=1}^{K} \|\boldsymbol{w}_k\|/K$, where $\boldsymbol{w}_k \in \mathbb{R}^d$ represents the k-th column of the classifier weight $\boldsymbol{W}$.

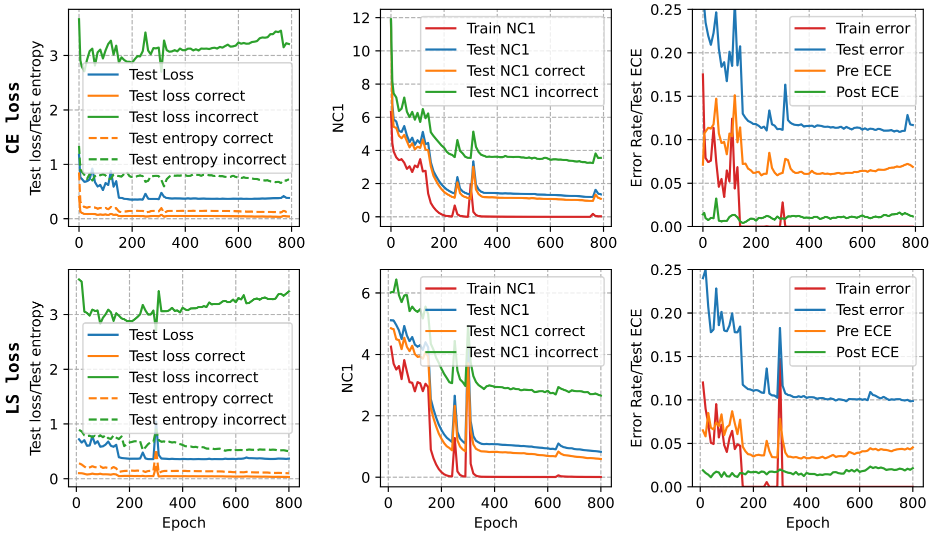

Figure 4: Leftmost plot: average error and entropy for correctly and incorrectly classified test samples. Middle plot: NC1 for the training set, testing set, and correctly/incorrectly classified testing samples. Right plot: model classification error rate and the test ECE before and after temperature scaling.

correctly and incorrectly classified test samples consistently decreases during training. For misclassified samples, a smaller NC1 indicates that their feature vectors align more closely with the mean of the incorrectly predicted class, making the model more confident about its incorrect predictions. [3] As a result, the entropy for misclassified test samples continues to decrease, and the test ECE starts to rise during TPT. And if we apply temperature scaling to counteract the regularization effect of LS loss on feature and classification vector norms, LS loss results in higher ECE due to its stronger NC1.

4 Theoretical Analysis

4.1 Unconstrained Feature Model

Analyzing deep neural networks poses significant challenges because of the non-linearities and complex interactions between layers in the feature mapping $\boldsymbol{h} := \phi_{\boldsymbol{\theta}}(\boldsymbol{x})$. The Unconstrained Feature Model (UFM) (Fang et al., 2021) simplifies DNN models by treating the features of the last layer as free optimization variables. This choice is motivated by the idea that over-parameterized DNN models are able to approximate any continuous functions (Hornik, 1991; Cybenko, 1989; Lu et al., 2017; Shaham et al., 2018).

Recall that the dimension of a feature vector $\boldsymbol{h}$ is denoted by d. Let $\boldsymbol{H} = \{\boldsymbol{h}_{ki}\}_{1\le k\le K, 1\le i\le n} \in \mathbb{R}^{d\times N}$ be the feature matrix of all training samples with $\boldsymbol{h}_{ki}$ denoting the $((k-1)n+i)$-th column of $\boldsymbol{H}$. Under the UFM, we investigate regularized empirical risk minimization, a variant of the formulation in Equation 2:

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n} l_{CE}(\boldsymbol{W}^\top \boldsymbol{h}_{ki} + \boldsymbol{b}, \boldsymbol{y}_k^\delta) + \frac{\lambda_W}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_H}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2, \tag{7}$$

where $\boldsymbol{y}_k^\delta = (1-\delta)\boldsymbol{e}_k + (\delta/K)\boldsymbol{1}_K$ is the soft label for class k with a smoothing parameter δ. Here $\boldsymbol{e}_k$ a one-hot vector with the k-th element equal to 1, and $\lambda_W, \lambda_H, \lambda_b > 0$ are the regularization parameters.

Let $\boldsymbol{Y} = [\boldsymbol{e}_1\boldsymbol{1}_n^\top, \cdots, \boldsymbol{e}_K\boldsymbol{1}_n^\top] \in \mathbb{R}^{K\times N}$ represent the matrix form of the (hard) ground truth labels. Consequently, the matrix form of the soft labels can be represented as $\boldsymbol{Y}^\delta = (1-\delta)\boldsymbol{Y} + \frac{\delta}{K}\boldsymbol{1}_K\boldsymbol{1}_N^\top$. The empirical

---

[3] For misclassified test samples, we assess the within-class variance relative to the incorrectly predicted class.

risk minimization in (7) has an equivalent matrix form:

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \mathcal{L}(\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}) := \frac{1}{N} l_{CE}(\boldsymbol{W}^\top \boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}_N^\top, \boldsymbol{Y}^\delta) + \frac{\lambda_W}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_H}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2, \tag{8}$$

where $l_{CE}(\boldsymbol{W}^\top \boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}_N^\top, \boldsymbol{Y}^\delta)$ computes the cross-entropy column-wise and takes the sum.

4.2 Theoretical Results

Within the UFM framework, Zhou et al. (Zhou et al., 2022b) demonstrate that, under a wide range of loss functions, including CE loss and LS loss, all the global minimizers satisfy neural collapses properties, and in particular $\boldsymbol{H}$ and $\boldsymbol{W}$ form aligned simplex ETFs. Moreover, they show that every critical point is either a global minimizer or a strict saddle point, which implies that these global minimizers can be effectively attained through iterative algorithms. However, their work does not provide the exact expression of the global minimizers, nor does it provide a landscape analysis (condition number) across different loss functions, which is critical for understanding the model's convergence behavior. In contrast, our work derives the exact solutions of both $\boldsymbol{W}$ and $\boldsymbol{H}$ for UFM models with CE and LS losses. Furthermore, based on these solutions, we employ conditioning number analysis to closely compare the optimization landscape of the models surrounding their respective global minimizers, and thereby provide an explanation for the accelerated convergence observed under LS loss. The proofs of our theorems can be found in the Appendix.

Let $\lambda_Z := \sqrt{\lambda_W \lambda_H}$ and let a^δ be defined by

(i) if $\sqrt{KN}\lambda_Z + \delta \geq 1$, then $a^\delta = 0$;

(ii) if $\sqrt{KN}\lambda_Z + \delta < 1$, then

$$a^\delta = \log\left(\frac{K}{\sqrt{KN}\lambda_Z + \delta} - K + 1\right).$$

Theorem 4.1. *(Global Optimizer). Assume that the feature dimension d is greater than or equal to the number of classes K, i.e., $d \geq K$, and the dataset is balanced. Then any global optimizer of $(\boldsymbol{W},\boldsymbol{H},\boldsymbol{b})$ of (8) satisfies the following condition:*

There is a semi-orthogonal matrix $\boldsymbol{P} \in \mathbb{R}^{d\times K}$ (i.e., $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}_K$) such that:

(i) The classification weight matrix $\boldsymbol{W}$ is given by

$$\boldsymbol{W} = \left(\frac{n\lambda_H}{\lambda_W}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K\boldsymbol{1}_K^\top/K\right). \tag{9}$$

(ii) The matrix of last-layer features $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \left(\frac{\lambda_W}{n\lambda_H}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K\boldsymbol{1}_K^\top/K\right)\boldsymbol{Y}. \tag{10}$$

(iii) The bias $\boldsymbol{b}$ is a zero vector, i.e. $\boldsymbol{b} = \boldsymbol{0}$.

The above theorem provides an explicit closed-form solution for the global minimizer $(\boldsymbol{W},\boldsymbol{H},\boldsymbol{b})$. Notably, our findings indicate that an increase in δ is associated with a decrease in the norm of $\boldsymbol{W}$ and $\boldsymbol{H}$. This observation aligns closely with the empirical results detailed in Section 3.3.

Our observations in Section 3 reveal that LS loss demonstrates accelerated convergence in terms of the NC metrics and training error. To better understand this phenomenon, we analyze the optimization landscape under the UFM. Note that condition number of the Hessian matrix, representing the ratio of the largest to the smallest eigenvalue, plays a crucial role in convergence rate analysis (Nocedal & Wright, 1999; Trefethen & Bau, 2022). In the vicinity of the local minimizer, a smaller condition number typically signifies a faster convergence rate. We now present our main theoretical result, which, to the best of our knowledge, is the first result providing insight into the convergence rate to the optimal solution under UFM models.

Theorem 4.2. *(Optimization Landscape). Assume the feature dimension d is greater than or equal to the number of classes K ($d \geq K$) and the dataset is balanced. In addition, assume the regularization parameters and the smoothing parameter satisfy $\sqrt{KN\lambda_W\lambda_H} + \delta < 1$. Then, at the global minimizer (as provided in Equation 9-10), the Hessian of the empirical loss with respect to $\boldsymbol{W}$ when $\boldsymbol{H}$ is fixed (also the Hessian w.r.t. $\boldsymbol{H}$ when $\boldsymbol{W}$ is fixed) has smaller condition number under label smoothing loss ($0 < \delta < 1 - \sqrt{KN\lambda_W\lambda_H}$) than under cross-entropy loss ($\delta = 0$).*

This theorem suggests that the LS loss results in a better-conditioned optimization landscape in the vicinity of the global minimizers, facilitating faster convergence behavior, as observed in Section 3.1. It is noteworthy that while the theorem primarily analyzes the local landscape around the global minimizer, experimental results demonstrate that gradient descent with random initialization indeed converges faster to the global solution of neural collapse under label smoothing loss.

5 Discussion

We conducted a comprehensive empirical comparison of CE loss and LS loss during the training process. We found that models trained under LS loss exhibit accelerated convergence in terms of training error and neural collapse metrics. Furthermore, they converge to a more pronounced level of NC1 and NC2. Along with the accelerated convergence, we found that label smoothing maintains a distinct balance between NC1 and NC2. We posit that the emphasis on NC2 in LS enhances the model's generalization performance. Additionally, we investigated the impact of LS on model calibration from the perspective of neural collapse, revealing that label smoothing has a regularization effect on the classifier weight and feature norms, and excessively small NC1 values may adversely affect model calibration.

We performed a mathematical analysis of the convergence properties of the UFM models under CE and LS losses. We first derived closed-form solutions for the global minimizers under both loss functions. Then we conducted a second-order theoretical analysis of the optimization landscape around their respective global optimizers, which reveals that LS exhibits a better-conditioned optimization landscape around the global minimum, which facilitates the faster convergence observed in our empirical study.

This paper provides a significantly deeper understanding of why LS excels in terms of convergence, performance, and model calibration compared to CE loss. Additionally, it illustrates how the powerful framework of neural collapse and its associated mathematical models can be employed to gain a more nuanced understanding of the "why" of DNNs. We expect that these results will inspire future research into the interplay between neural collapse, convergence speed, and model generalizability.

References

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Hien Dang, Tho Tran, Stanley Osher, Hung Tran-The, Nhat Ho, and Tan Nguyen. Neural collapse in deep linear networks: from balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023.

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.

Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. *arXiv preprint arXiv:2112.15121*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

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.

Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *arXiv preprint arXiv:2309.09725*, 2023.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Jiachen Jiang, Jinxin Zhou, Peng Wang, Qing Qu, Dustin Mixon, Chong You, and Zhihui Zhu. Generalized neural collapse for a large number of classes. *arXiv preprint arXiv:2310.05351*, 2023.

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

Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Principled and efficient transfer learning of deep models via neural collapse. *arXiv preprint arXiv:2212.12206*, 2022.

Xuantong Liu, Jianfeng Zhang, Tianyang Hu, He Cao, Yuan Yao, and Lujia Pan. Inducing neural collapse in deep long-tailed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 11534–11544. PMLR, 2023.

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

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pp. 6448–6458. PMLR, 2020.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *arXiv preprint arXiv:2011.11619*, 2020.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632, 2005.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Vardan Papyan, XY 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.

Tomaso Poggio and Qianli Liao. Explicit regularization and implicit bias in deep network classifiers trained with the square loss. *arXiv preprint arXiv:2101.00072*, 2020.

Uri Shaham, Alexander Cloninger, and Ronald R Coifman. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018.

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.

Christos Thrampoulidis, Ganesh Ramachandra Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *Advances in Neural Information Processing Systems*, 35:27225–27238, 2022.

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.

Tom Tirer, Haoxiang Huang, and Jonathan Niles-Weed. Perturbation analysis of neural collapse. In *International Conference on Machine Learning*, pp. 34301–34329. PMLR, 2023.

Lloyd N Trefethen and David Bau. *Numerical linear algebra*, volume 181. Siam, 2022.

Stephan Wojtowytsch et al. On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers. *arXiv preprint arXiv:2012.05420*, 2020.

Liang Xie, Yibo Yang, Deng Cai, and Xiaofei He. Neural collapse inspired attraction–repulsion-balanced loss for imbalanced learning. *Neurocomputing*, 527:60–70, 2023.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*, 2023.

Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. *Advances in neural information processing systems*, 35: 11547–11560, 2022.

Longhui Yu, Tianyang Hu, Lanqing Hong, Zhen Liu, Adrian Weller, and Weiyang Liu. Continual learning by modeling intra-class variation. *arXiv preprint arXiv:2210.05398*, 2022.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. In *International Conference on Machine Learning*, pp. 27179–27202. PMLR, 2022a.

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, 2022b.

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.

A Appendix

B Additional Experiments

B.1 Metrics for Measuring NC

We introduced the metrics for measuring neural collapse in Section 3. For convenience, we restate them here, providing additional details about the subtle differences between our definitions and those used in prior works (Zhu et al., 2021; Zhou et al., 2022b). We denote the global mean and classwise means of the last-layer features as:

$$\boldsymbol{h}_G = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\boldsymbol{h}_{ki}, \quad \bar{\boldsymbol{h}}_k = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{h}_{ki}, (1 \le k \le K),$$

where $\boldsymbol{h}_G$ is the global mean and $\bar{\boldsymbol{h}}_k$ represents the mean of class k. The metrics for neural collapse are then defined as follows:

(NC1) Within class variability collapse measures the relative magnitude of the within-class covariance $\Sigma_W := \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}(\boldsymbol{h}_{ki} - \bar{\boldsymbol{h}}_k)(\boldsymbol{h}_{ki} - \bar{\boldsymbol{h}}_k)^\top$ compared to the between-class covariance matrix $\Sigma_B := \frac{1}{K}\sum_{k=1}^{K}(\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G)(\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G)^\top$ of the last-layer features. It is formulated as:

$$NC_1 = \frac{1}{K} trace\left(\Sigma_W \Sigma_B^\dagger\right),$$

where $\Sigma_B^\dagger$ denotes the pseudo inverse of Σ_B.

(NC2) Convergence to simplex ETF quantifies the difference between the normalized classifier weight matrix and the centered class mean features in comparison to a normalized simplex ETF, defined as follows:

$$NC_2 := \left\| \frac{\boldsymbol{W}^\top \overline{\boldsymbol{H}}}{\|\boldsymbol{W}^\top \overline{\boldsymbol{H}}\|_F} - \frac{1}{\sqrt{K-1}} \left(\boldsymbol{I}_K - \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^\top \right) \right\|_F, \tag{11}$$

where $\overline{\boldsymbol{H}} = [\bar{\boldsymbol{h}}_1 - \boldsymbol{h}_G, \cdots, \bar{\boldsymbol{h}}_K - \boldsymbol{h}_G] \in \mathbb{R}^{d\times K}$ represents centered class mean matrix.

This definition differs slightly from those in (Zhu et al., 2021; Zhou et al., 2022b). However, it can be shown that the convergence of $NC2$ in Equation (11) to zero indicates the simultaneous convergence of both $\boldsymbol{W}$ and $\boldsymbol{H}$ towards the simplex ETF structure. Specifically, when the bias $\boldsymbol{b}$ is an all-zero vector or a constant vector, NC_2 as defined in (11) approaching zero indicates that the average logit matrix, formulated as $\overline{\boldsymbol{Z}} = \boldsymbol{W}^\top \overline{\boldsymbol{H}} + \boldsymbol{b}\mathbf{1}_K^T$, satisfies the condition $\overline{\boldsymbol{Z}} = a\left(\boldsymbol{I} - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)$ for some constant a. Remarkably, this matrix aligns with the simplex-encoding label (SEL) matrix introduced in (Thrampoulidis et al., 2022), up to a scaling factor a.

From Proposition C.4, we have

$$\min_{\boldsymbol{W}^\top \boldsymbol{H} = \boldsymbol{Z}} \frac{1}{2}\left(\lambda_W \|\boldsymbol{W}\|_F^2 + \lambda_H \|\boldsymbol{H}\|_F^2\right) \geq \|\overline{\boldsymbol{Z}}\|_*$$

where $\|\overline{\boldsymbol{Z}}\|_*$ represents the nuclear norm of $\overline{\boldsymbol{Z}}$ and the equality holds only when $\boldsymbol{H} = \overline{\boldsymbol{H}}\boldsymbol{Y}$ and $\overline{\boldsymbol{H}} = \sqrt{\lambda_W / n\lambda_H}\boldsymbol{W}$, indicating the self-duality of the class mean feature and the classification vector. Consequently, during the model training with an L2 penalty on both $\boldsymbol{W}$ and $\boldsymbol{H}$ (with the norm of $\boldsymbol{H}$ implicitly penalized by penalizing the model parameters Θ), the convergence of NC_2 as defined in (11) to zero indicates the simultaneous convergence of both $\boldsymbol{W}$ and $\boldsymbol{H}$ towards the simplex ETF structure.

(NC3) Convergence to self-duality measures the distance between the classifier weight matrix $\boldsymbol{W}$ and the centered class-means $\overline{\boldsymbol{H}}$:

$$NC_3 := \left\| \frac{\boldsymbol{W}}{\|\boldsymbol{W}\|_F} - \frac{\overline{\boldsymbol{H}}}{\|\overline{\boldsymbol{H}}\|_F} \right\|_F. \tag{12}$$

B.2 Experiment Setup

In this section, we provide with more details for reproducing the experiments presented in the paper. In particular, we emphasize that the datasets involved in the paper, namely CIFAR10, CIFAR100, and STL10, are all publicly available for academic usage: CIFAR10 and CIFAR100 datasets are made available under the MIT license. All experiments are conducted on a single RTX3090 GPU with 24GB memory. For consistency with prior studies (Papyan et al., 2020; Zhu et al., 2021), we adopt ResNet-18 (He et al., 2016) as the backbone network for CIFAR-10 and ResNet-50 (He et al., 2016) as the backbone for CIFAR-100 and STL-10.

To focus on behaviors associated with neural collapse and minimize the influence of other factors, we adopted standard preprocessing techniques without introducing any data augmentation for the training data. We extended the training period to 800 epochs to analyze the model's behavior during the terminal phase of training. For all datasets, we used a default batch size of 128, stochastic gradient descent with a momentum of 0.9, and a learning rate initialized at 0.05, which undergoes multi-step decay, decreasing by a factor of 0.1 at epochs 150 and 350. We considered two different scenarios for model regularization:

- **Model with Weight Decay**. For the default setting, we trained the models with weight decay regularization set at 5×10^{-4}. It's worth noting that the norm of $\boldsymbol{H}$ is implicitly penalized by the weight decay on the model parameters Θ.
- **Simulated UFM Model**. To simulate the unconstrained feature model, we turn off the weight decay and add an L2 penalty on the classifier parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ and the last-layer feature $\boldsymbol{H}$ to the cross-entropy (CE) or label smoothing (LS) loss. This setting was only used in Section B.6 to validate the theoretical results.

B.3 Neural Collapses under Cross-Entropy Loss and Label Smoothing Loss

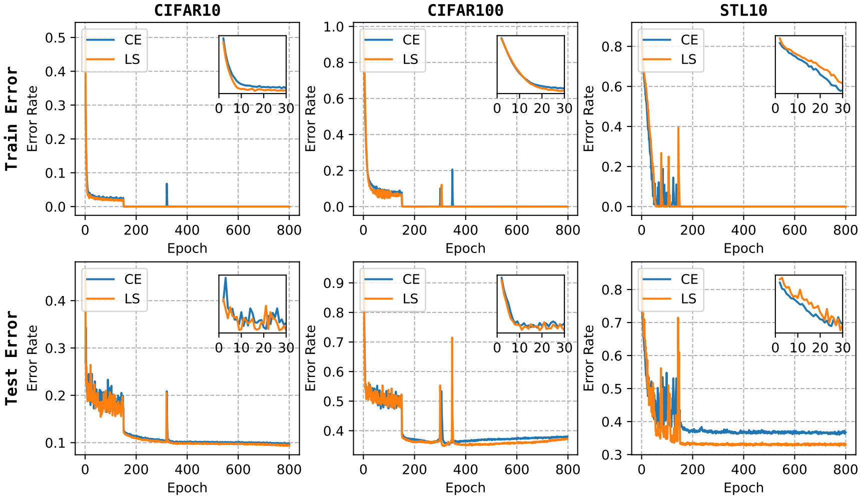

Figure 5: Comparison of training (top row) and testing (bottom row) error rates using cross-entropy (CE) and label smoothing (LS) losses.

This section presents additional visualizations supporting the conclusions drawn in the main paper. Figure 5 provides a detailed zoom-in plot illustrating the progression of training and testing errors throughout the model training process for the CIFAR-10, CIFAR-100, and STL-10 datasets, under both cross-entropy and label smoothing losses. Notably, the visualization suggests that models trained with label smoothing exhibit a more rapid convergence in both training and testing errors compared to their cross-entropy counterparts.

B.4 Impact of the Smoothing Hyperparameter

In Section 3.1 and 3.2, a default smoothing hyperparameter of $\delta = 0.05$ is utilized. This section explores how the choice of δ impacts the model's convergence to neural collapse and its generalizability. Specifically, we consider various values for δ within the interval $[0, 1)$.

Figure 6 presents the results of the experiments. The experiments yield several important insights. Firstly, the smoothing hyperparameter significantly influences the model's convergence to neural collapse. The curves of NC1 and NC2 as a function of δ (2nd column in Figure 6) reveal a U-shaped trend, with the levels of NC3 remaining relatively consistent across different values of δ. Exceptionally small and large values of δ result in reduced collapse for both NC1 and NC2.

Secondly, the average norms of the classifier vectors and the class mean features decrease as δ increases. This observation aligns with intuition: given features and classifier vectors that form a simplex ETF structure, decreasing their norms softens the output probabilities. For LS loss, a higher smoothing parameter δ corresponds to smoother target labels, and consequently, the features and classification vectors with lower norms can achieve close-to-zero label smoothing loss. This observation was also supported by the closed-form expressions for $\boldsymbol{W}$ and $\boldsymbol{H}$ provided in Theorem 4.1. Thirdly, the trend of the test error also exhibits a U-shape, underscoring the necessity of selecting an appropriate δ. When δ approaches 1, the nearly uniform smoothed labels do not provide effective training signals to update the parameters. Additionally, the norm of both classification vectors and the last-layer features decrease towards zero, causing the classifier to be dominated by noise, leading to rapid deterioration in model performance. Finally, we observe a surprising robustness in both neural collapse metrics and model performance for the choice of $\delta \in [0.02, 0.8]$. For classification tasks with a larger number of classes, such as CIFAR-100, opting for a relatively large δ is viable.

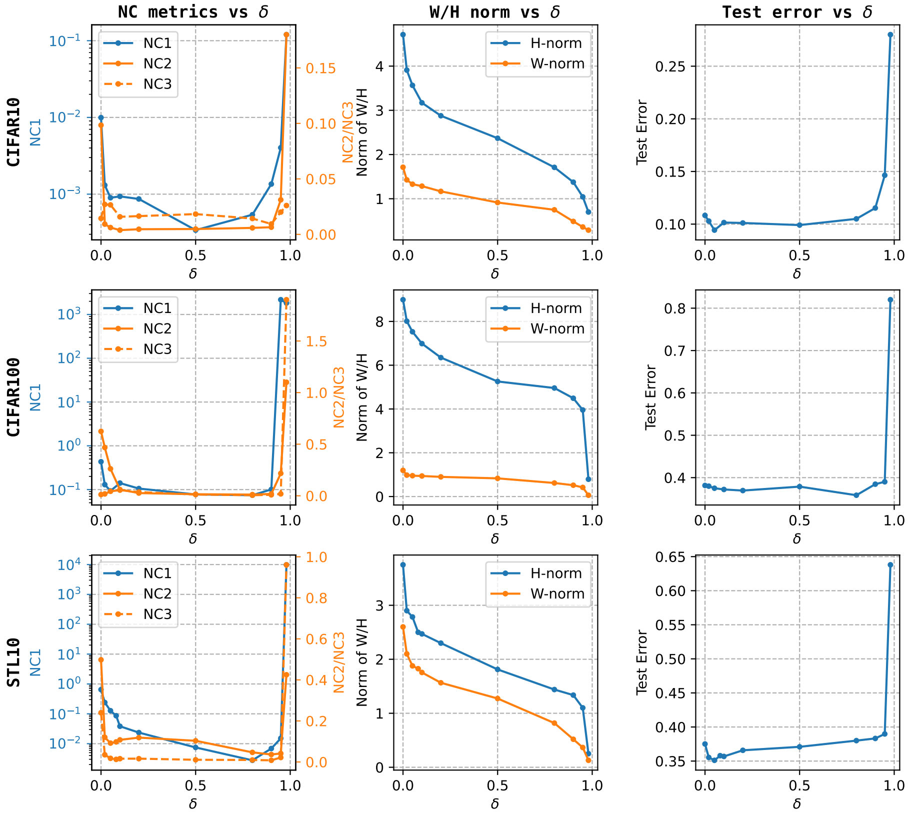

Figure 6: **The effect of the smoothing hyperparameter** δ. The columns from left to right visualize (a) NC metrics vs. δ, (b) Average norm of classification weight vectors and mean features vs. δ, and (c) Testing performance (test error rate) vs. δ.

B.5 Weight and Feature Regularization Improve Model Calibration

As mentioned in Section 3.3, label smoothing loss (with properly chosen δ) can improve model calibration. This improvement is due to the regularization effect of label smoothing on the classification vector norm and last-layer feature norm, which leads to predictions with lower confidence. To support this, Figure 7 presents a 20-bin reliability plot (Niculescu-Mizil & Caruana, 2005) for models trained under CE and LS losses for CIFAR-10, with a default smoothing hyperparameter of 0.05. Additionally, we include a plot showing the percentage of samples in each confidence bin. Clearly, LS loss exhibits less confidence in its predictions and better model calibration compared to CE loss without temperature scaling. In the right column of Figure 7, we employ temperature scaling to calibrate the model, with the hyperparameter T selected through cross-validation. Notably, after temperature scaling, models trained with CE loss achieve even better calibration than those trained with LS loss. This is because temperature scaling counteracts the regularization effect of label smoothing on the classification vector norm and feature norm. Consequently, LS loss can result in worse test expected calibration error (ECE) due to an excessive level of NC1.

B.6 Empirical Validation of Theoretical Results

In Theorem 4.1, we derive closed-form solutions for both the mean feature matrix $\overline{\boldsymbol{H}}$ and the classifier weight matrix $\boldsymbol{W}$, both exhibiting a simplex ETF structure. To validate these theoretical findings, we train ResNet18 models on CIFAR10 dataset with varying smoothing parameters δ. To ensure consistency with the UFM model, we adopted the simulated UFM model as introduced in Section B.2, where weight decay is disabled, and L2 regularization is applied to the last-layer features and classification parameters, ensuring

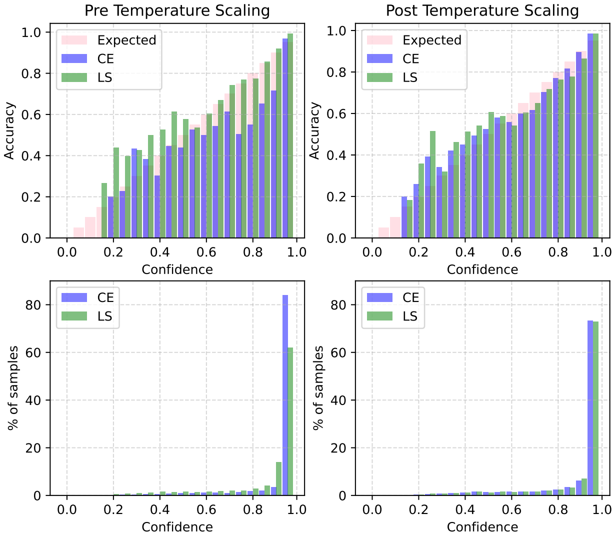

Figure 7: Reliability plots with 20 bins (top row) and the percentage of samples in each bin (bottom row) for models trained with cross-entropy (CE) and label smoothing (LS) losses, both pre- (left column) and post- (right column) temperature scaling. The results are based on the CIFAR-10 dataset.

exact regularization effects. We set the regularization parameters as follows: $\lambda_W = 10^{-3}$, $\lambda_H = 10^{-6}$, and $\lambda_b = 10^{-2}$. Under these settings, we compare the trained average feature norm $\overline{\|\boldsymbol{h}\|} = \sum_{k=1}^{K} \|\bar{\boldsymbol{h}}_k - \boldsymbol{h}_G\|/K$ and average classification vector norm $\overline{\|\boldsymbol{w}\|} = \sum_{k=1}^{K} \|\boldsymbol{w}_k\|/K$ with their theoretical counterparts derived from Theorem 4.1 which can be represented as

$$\overline{\|\boldsymbol{h}\|}^* = \sqrt{(K-1)a^\delta/K}\left(\frac{n\lambda_H}{\lambda_W}\right)^{-1/4}, \quad \overline{\|\boldsymbol{w}\|}^* = \sqrt{(K-1)a^\delta/K}\left(\frac{n\lambda_H}{\lambda_W}\right)^{1/4}.$$

Figure 8 illustrates the differences between the trained predictions $\overline{\|\boldsymbol{h}\|}$ and $\overline{\|\boldsymbol{w}\|}$ and their theoretical solutions for models trained under LS loss with different δ. The plot shows a clear decreasing trend in the relative error between the empirical and theoretical values for both the feature and classification vector norms, with differences reducing to below 0.01 across all cases. These results demonstrate that the empirical findings align with the theoretical predictions, thereby verifying Theorem 4.1.

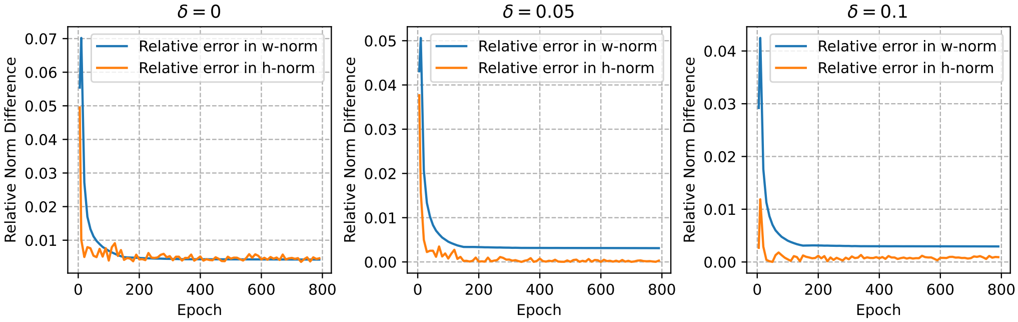

Figure 8: Relative error of classification vector norm $\overline{\|\boldsymbol{w}\|}$ and feature norm $\overline{\|\boldsymbol{h}\|}$ compared to their theoretical results over epochs for ResNet-18 on CIFAR10.

According to Theorem 4.2, models trained with label smoothing loss (satisfying $\sqrt{KN}\lambda_Z + \delta < 1$) converges faster than models trained with cross-entropy loss. In Figure 9, we train ResNet18 and ResNet50 on CIFAR10 and CIFAR100, respectively, with a default weight decay value of 5×10^{-4}. Comparing models trained with

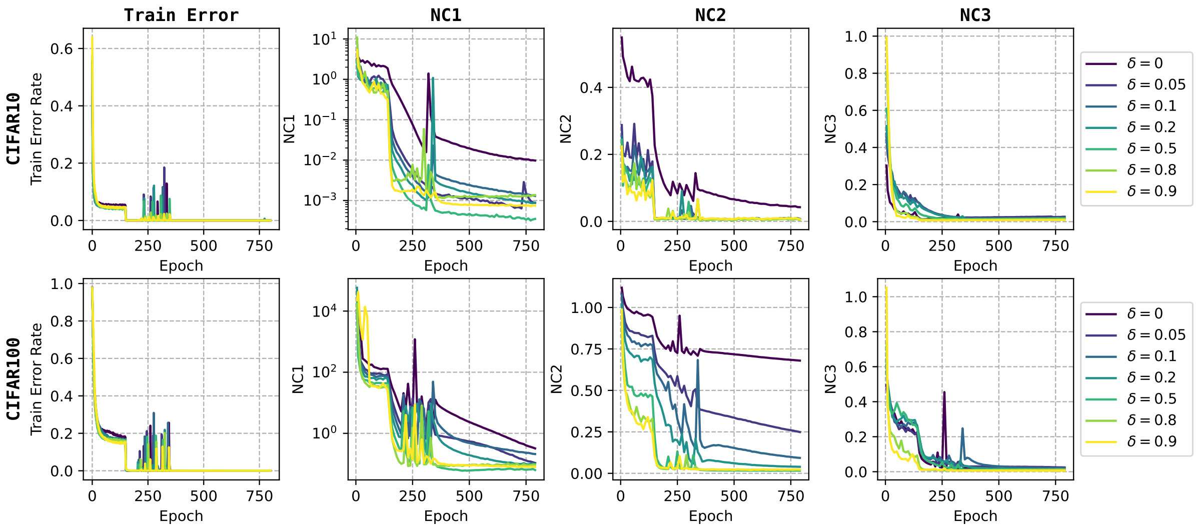

Figure 9: Comparison of model convergence under weight decay regularization. From left to right, the figures display the models' training error rate, NC1, NC2, and NC3 with varying smoothing hyperparameters δ.

different smoothing hyperparameters, we observe that for a wide range of δ values, models with larger δ converge faster in terms of both training error and NC metrics.

Furthermore, we investigate model convergence under the simulated UFM model, where weight decay is disabled, and L2 regularization is added. As illustrated in Figure 10, across a wide range of δ such that $\sqrt{KN}\lambda_Z + \delta < 1$, we observe faster convergence for larger δ values.

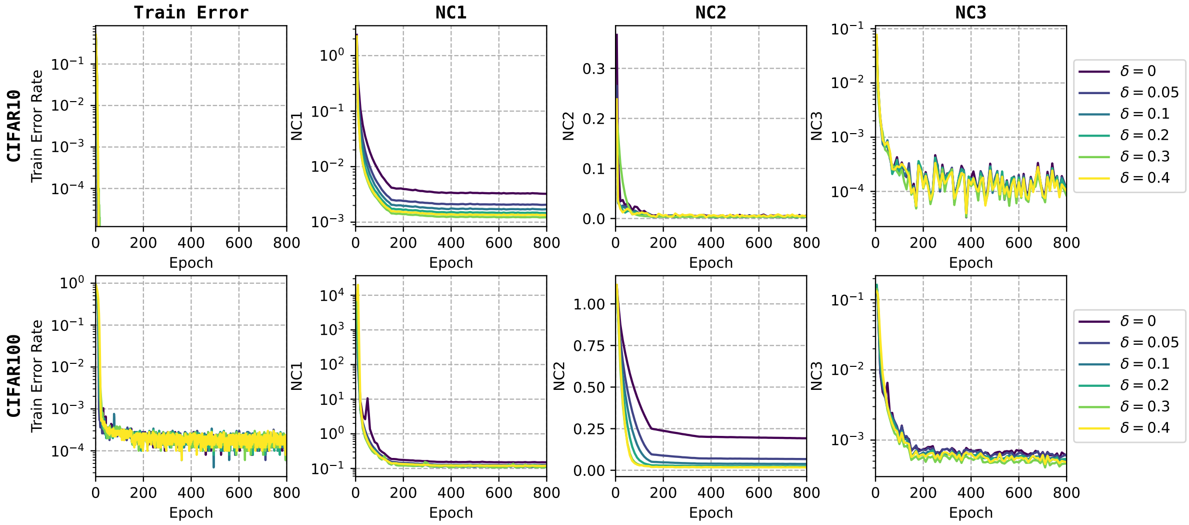

Figure 10: Comparison of model convergence under simulated UFM model. From left to right, the figures display the models' training error rate, NC1, NC2, and NC3 with varying smoothing hyperparameters δ.

C Proofs

The first key observation of the phenomenon of neural collapse, that is (NC1), refers to a type of collapse that involves the convergence of the feature of samples from the same class to a unique mean feature vector.

The second key observation of neural collapse, namely (NC2), involves these unique mean feature vectors (after recentering by their global mean) as they form an equiangular tight frame (ETF), i.e., they share the same pairwise angles and length. Before providing the theoretical poof, we first formally define the rank K canonical simplex ETF in the definition below.

C.1 Basics

Definition C.1. (K-simplex ETF) A K-simplex ETF is a collection of points in $\mathbb{R}^d$ specified by the columns of the matrix

$$\boldsymbol{M} = \sqrt{\frac{K}{K-1}}\boldsymbol{P}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right),$$

where $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$ is the identity matrix and $\mathbf{1}_K \in \mathbb{R}^K$ is the ones vector, and $\boldsymbol{P} \in \mathbb{R}^{d\times K}(d \geq K)$ is a partial-orthogonal matrix such that $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}_K$.

Note that the matrix $\boldsymbol{M}$ satisfies:

$$\boldsymbol{M}^\top\boldsymbol{M} = \frac{K}{K-1}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right).$$

Next, we prove a series of lemmas that will prove crucial upon establishing our main theorems.

Lemma C.2. *(Young's Inequality) Let p, q be positive numbers satisfying $\frac{1}{p} + \frac{1}{q} = 1$. Then for any $a, b \in \mathbb{R}$, we have*

$$|ab| \leq \frac{|a|^p}{p} + \frac{|b|^q}{q},$$

where the equality holds if and only if $|a|^p = |b|^q$. The case for $p = q = 2$ is just the AM-GM inequality which is $|ab| \leq \frac{1}{2}\left(a^2 + b^2\right)$, where the equality holds if and only if $|a| = |b|$

Lemma C.3. *For any fixed $\boldsymbol{Z} \in \mathbb{R}^{K\times N}$ and $\alpha > 0$, we have*

$$\min_{\boldsymbol{Z}=\boldsymbol{W}^\top\boldsymbol{H}} \frac{1}{2\sqrt{\alpha}}\left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right) = \|\boldsymbol{Z}\|_*. \tag{13}$$

Here $\|\boldsymbol{Z}\|_*$ denotes the nuclear norm of $\boldsymbol{Z}$:

$$\|\boldsymbol{Z}\|_* := \sum_k \sigma_k(\boldsymbol{Z}) = trace(\Sigma), \text{ with } \boldsymbol{Z} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top,$$

where $\{\sigma_k\}_{k=1}^{min(K,N)}$ denote the singular values of $\boldsymbol{Z}$, and $\boldsymbol{Z} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$ is the singular value decomposition (SVD) of $\boldsymbol{Z}$.

Proof. Let $\boldsymbol{Z} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{Z}$. From the fact that $\boldsymbol{U}\boldsymbol{U}^\top = \mathbf{I}$, $\boldsymbol{V}\boldsymbol{V}^\top = \mathbf{I}$, and $trace\left(\boldsymbol{A}^\top\boldsymbol{A}\right) = \|\boldsymbol{A}\|_F^2$, we have

$$\begin{aligned}\|\boldsymbol{Z}\|_* = trace(\Sigma) &= \frac{1}{2\sqrt{\alpha}}trace\left(\sqrt{\alpha}\boldsymbol{U}^\top\boldsymbol{U}\Sigma\right) + \frac{\sqrt{\alpha}}{2}trace\left(\frac{1}{\sqrt{\alpha}}\Sigma\boldsymbol{V}^\top\boldsymbol{V}\right)\\ &= \frac{1}{2\sqrt{\alpha}}\left(\left\|\alpha^{1/4}\boldsymbol{U}\Sigma^{1/2}\right\|_F^2 + \alpha\left\|\alpha^{-1/4}\Sigma^{1/2}\boldsymbol{V}^\top\right\|_F^2\right).\end{aligned}$$

This implies that there exists some $\boldsymbol{W} = \alpha^{1/4}\Sigma^{1/2}\boldsymbol{U}^\top$ and $\boldsymbol{H} = \alpha^{-1/4}\Sigma^{1/2}\boldsymbol{V}^\top$, such that $\|\boldsymbol{Z}\|_* = \frac{1}{2\sqrt{\alpha}}\left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right)$, which further indicates that

$$\|\boldsymbol{Z}\|_* \geq \min_{\boldsymbol{Z}=\boldsymbol{W}^\top\boldsymbol{H}} \frac{1}{2\sqrt{\alpha}}\left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right). \tag{14}$$

On the other hand, for any $\boldsymbol{W}^\top \boldsymbol{H} = \boldsymbol{Z}$, we have

$$\begin{aligned}\|\boldsymbol{Z}\|_* &= trace(\Sigma) = trace(\boldsymbol{U}^\top \boldsymbol{Z} V) = trace(\boldsymbol{U}^\top \boldsymbol{W}^\top \boldsymbol{H}\boldsymbol{V}) \\ &\leq \frac{1}{2\sqrt{\alpha}} \left\|\boldsymbol{U}^\top \boldsymbol{W}^\top\right\|_F^2 + \frac{\sqrt{\alpha}}{2} \|\boldsymbol{H}\boldsymbol{V}\|_F^2 = \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right),\end{aligned}$$

where the first inequality is guaranteed by Young's inequality in Lemma C.2, and equality only holds when $\boldsymbol{W}\boldsymbol{U} = \sqrt{\alpha}\boldsymbol{H}\boldsymbol{V}$. The last equality follows because $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}$ and $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}$. Therefore, we have

$$\|\boldsymbol{Z}\|_* \leq \min_{\boldsymbol{Z}=\boldsymbol{W}^\top\boldsymbol{H}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right). \tag{15}$$

Combining the results in (14) and (15), we complete the proof.

□

Proposition C.4. *Consider matrics $\boldsymbol{H} = [\boldsymbol{H}_1, \cdots, \boldsymbol{H}_n] \in \mathbb{R}^{d\times N}$ and $\boldsymbol{Z} = [\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n] \in \mathbb{R}^{K\times N}$, where $\boldsymbol{H}_i \in \mathbb{R}^{d\times K}$ and $\boldsymbol{Z}_i \in \mathbb{R}^{K\times K}$ with $N = nK$. Let $\overline{\boldsymbol{H}} = \frac{1}{n}\sum_i \boldsymbol{H}_i$ and $\overline{\boldsymbol{Z}} = \frac{1}{n}\sum_i \boldsymbol{Z}_i$. Then $\boldsymbol{Z} = \boldsymbol{W}^\top\boldsymbol{H}$ indicates $\overline{\boldsymbol{Z}} = \boldsymbol{W}^\top\overline{\boldsymbol{H}}$. If $\overline{\boldsymbol{Z}}$ is a symmetric matrix, then we have*

$$\min_{\boldsymbol{Z}=\boldsymbol{W}^\top\boldsymbol{H}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right) = \min_{\overline{\boldsymbol{Z}}=\boldsymbol{W}^\top\overline{\boldsymbol{H}}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha n\|\overline{\boldsymbol{H}}\|_F^2\right) = \|\overline{\boldsymbol{Z}}\|_*, \tag{16}$$

with the minimum is reached only if $\boldsymbol{H}_i = \overline{\boldsymbol{H}}$ (for $\forall i = 1, \cdots, n$) and $\boldsymbol{W} = \sqrt{\alpha n}\overline{\boldsymbol{H}}$.

Proof. From Lemma C.2, we have $\|\boldsymbol{H}\|_F^2 = \sum_i \|\boldsymbol{H}_i\|_F^2 \geq n\|\overline{\boldsymbol{H}}\|_F^2$, with the equality hold only when $\boldsymbol{H}_i = \boldsymbol{H}_j$ for any $i \neq j$. Consequently, this yields the following result:

$$\min_{\boldsymbol{Z}=\boldsymbol{W}^\top\boldsymbol{H}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha\|\boldsymbol{H}\|_F^2\right) = \min_{\overline{\boldsymbol{Z}}=\boldsymbol{W}^\top\overline{\boldsymbol{H}}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha n\|\overline{\boldsymbol{H}}\|_F^2\right).$$

Utilizing Lemma C.3, we further deduce:

$$\min_{\overline{\boldsymbol{Z}}=\boldsymbol{W}^\top\overline{\boldsymbol{H}}} \frac{1}{2\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha n\|\overline{\boldsymbol{H}}\|_F^2\right) = \|\overline{\boldsymbol{Z}}\|_*.$$

This minimum is achieved only when $\boldsymbol{W} = \sqrt{\alpha n}\overline{\boldsymbol{H}}$, thus completing the proof. □

D Proof of Theorem 4.1

D.1 Proof of Theorem 4.1

In this section, we present the proof of Theorem 4.1 in Section 4, which we restate as follows.

Theorem D.1. *(Global Optimizer). Assume that the feature dimension d is no less than the number of classes K, i.e., $d \geq K$, and the dataset is balanced. Then any global optimizer of $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of*

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) := \frac{1}{N} l_{CE}(\boldsymbol{W}^\top\boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}_N^\top, \boldsymbol{Y}^\delta) + \frac{\lambda_W}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_H}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2 \tag{17}$$

satisfies the following properties:

The classification weight matrix $\boldsymbol{W}$ is given by

$$\boldsymbol{W} = \left(\frac{\lambda_H n}{\lambda_W}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K\boldsymbol{1}_K^\top/K\right). \tag{18}$$

The matrix of last-layer feature $\boldsymbol{H}$ can be represented as

$$\boldsymbol{H} = \overline{\boldsymbol{H}}\boldsymbol{Y}, \quad \overline{\boldsymbol{H}} = \left(\frac{\lambda_W}{\lambda_H n}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K\boldsymbol{1}_K^\top/K\right). \tag{19}$$

The bias $\boldsymbol{b}$ *is a zero vector, i.e.* $\boldsymbol{b} = \boldsymbol{0}$.

Here, $\boldsymbol{P} \in \mathbb{R}^{d \times K} (d \geq K)$ *is a partial orthogonal matrix such that* $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}_K$ *and* a^δ *satisfies:*

- *(i) if* $\sqrt{KN}\lambda_Z + \delta \geq 1$*, then* $a^\delta = 0$*;*
- *(ii) if* $\sqrt{KN}\lambda_Z + \delta < 1$*, then*

$$a^\delta = \log\left(\frac{K}{\sqrt{KN}\lambda_Z + \delta} - K + 1)\right).$$

with $\lambda_Z = \sqrt{\lambda_W \lambda_H}$.

Proof. The main idea of proving Theorem 4.1 is first to connect the problem (17) to its corresponding convex counterpart. This allows us to derive the precise form of the global minimizer for the convex optimization problem. Subsequently, we can further characterize the specific structures of $\boldsymbol{W}$ and $\boldsymbol{H}$ based on the acquired global minimizer.

Connection of (17) to a Convex Problem. Let $\boldsymbol{Z} = \boldsymbol{W}^\top \boldsymbol{H} \in \mathbb{R}^{K \times N}$ represent the output logit matrix with $N = nK$ and $\alpha = \frac{\lambda_H}{\lambda_W}$. Utilizing Lemma C.3, we get:

$$\begin{aligned}\min_{\boldsymbol{W}^\top \boldsymbol{H} = \boldsymbol{Z}} \lambda_W \|\boldsymbol{W}\|_F^2 + \lambda_H \|\boldsymbol{H}\|_F^2 &= \sqrt{\lambda_W \lambda_H} \min_{\boldsymbol{W}^\top \boldsymbol{H} = \boldsymbol{Z}} \frac{1}{\sqrt{\alpha}} \left(\|\boldsymbol{W}\|_F^2 + \alpha \|\boldsymbol{H}\|_F^2\right) \\ &= 2\sqrt{\lambda_W \lambda_H} \|\boldsymbol{Z}\|_*,\end{aligned} \tag{20}$$

where $\|\boldsymbol{Z}\|_*$ represent nuclear norm of $\boldsymbol{Z}$. Additionally from Lemma C.3, the minimum is attained only when $\boldsymbol{H} = \overline{\boldsymbol{H}}\boldsymbol{Y}$ and $\boldsymbol{W} = \sqrt{\alpha n}\overline{\boldsymbol{H}} = \sqrt{\frac{\lambda_H}{\lambda_W n}}\overline{\boldsymbol{H}}$.

Let $\lambda_Z := \sqrt{\lambda_W \lambda_H}$, then Equation (17) becomes:

$$\min_{\boldsymbol{Z}, \boldsymbol{b}} \mathcal{L}(\boldsymbol{Z}, \boldsymbol{b}) := \frac{1}{N} l_{CE}(\boldsymbol{Z} + \boldsymbol{b}\boldsymbol{1}_N^\top, \boldsymbol{Y}^\delta) + \lambda_Z \|\boldsymbol{Z}\|_* + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2, \tag{21}$$

which is a convex optimization problem.

Characterizing the Optimal Solution of (17) based on the Convex Program (21): We first derive the exact form of the global minimizer for the convex optimization problem (21). Particularly, we first establish that the predicted logit vectors within each class collapse to their sample means, i.e., $\boldsymbol{Z} = \overline{\boldsymbol{Z}}\boldsymbol{Y}$ (Lemma D.3). Subsequently, we derive the closed-form solution of $\overline{\boldsymbol{Z}}$ in Lemma D.5.

Furthermore, from Lemma C.3, we establish that the minimum in (20) is attained only if $\boldsymbol{H} = \overline{\boldsymbol{H}}\boldsymbol{Y}$ and $\boldsymbol{W} = \sqrt{\frac{\lambda_H}{\lambda_W n}}\overline{\boldsymbol{H}}$. Since $\boldsymbol{Z} = \boldsymbol{W}^\top \boldsymbol{H}$, combining the above result with Lemma D.5 yields the global minimizer $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ of (17), satisfying:

$$\boldsymbol{W} = \left(\frac{\lambda_H n}{\lambda_W}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K \boldsymbol{1}_K^\top / K\right), \tag{22}$$

$$\boldsymbol{H} = \overline{\boldsymbol{H}}\boldsymbol{Y}, \quad \overline{\boldsymbol{H}} = \left(\frac{\lambda_W}{\lambda_H n}\right)^{1/4} \sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{1}_K \boldsymbol{1}_K^\top / K\right), \tag{23}$$

and

$$\boldsymbol{b} = \boldsymbol{0},$$

where $\boldsymbol{P} \in \mathbb{R}^{d \times K} (d \geq K)$ is a partial orthogonal matrix such that $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}_K$ and a^δ is defined per the specifications in Lemma D.5.

Hence, we conclude the proof.

□

D.2 Supporting Lemma

Lemma D.2. *(Optimality Condition) The first-order optimality condition of $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{b})$ in Equation (21) is*

$$N^{-1}\left(\boldsymbol{Y}^{\delta}-\boldsymbol{P}\right) \in \lambda_Z \partial\|\boldsymbol{Z}\|_*, \quad N^{-1}\left(\boldsymbol{Y}^{\delta}-\boldsymbol{P}\right)\mathbf{1}_N=\lambda_b \boldsymbol{b}, \tag{24}$$

where $\boldsymbol{P}$ is the prediction matrix defined as

$$\boldsymbol{P}=\{\boldsymbol{p}_{ki}\}_{1\le k\le K, 1\le i\le n} \in \mathbb{R}^{K\times N}, \quad \boldsymbol{p}_{ki}:=\frac{\exp(\boldsymbol{z}_{ki}+\boldsymbol{b})}{\langle \exp(\boldsymbol{z}_{ki}+\boldsymbol{b}), \mathbf{1}_K\rangle}, \tag{25}$$

$\boldsymbol{Y}^{\delta}$ is the matrix of the smoothed soft target defined as

$$\boldsymbol{Y}^{\delta}=(1-\delta)\boldsymbol{Y}+\frac{\delta}{K}\mathbf{1}_K\mathbf{1}_N^{\top}, \quad \boldsymbol{Y}=[\boldsymbol{e}_1\mathbf{1}_n^{\top}, \cdots, \boldsymbol{e}_K\mathbf{1}_n^{\top}] \in \mathbb{R}^{K\times N}, \tag{26}$$

and $\partial\|\boldsymbol{Z}\|_*$ represents the subdifferential of the nuclear norm of $\boldsymbol{Z}$.

Proof. Consider

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{b})=\frac{1}{N} l_{CE}(\boldsymbol{Z}+\boldsymbol{b}\mathbf{1}_N^{\top}, \boldsymbol{Y}^{\delta})+\lambda_Z\|\boldsymbol{Z}\|_*+\frac{\lambda_b}{2}\|\boldsymbol{b}\|^2$$

in (21). Define

$$\phi(\boldsymbol{Z}, \boldsymbol{b})=\frac{1}{N} l_{CE}(\boldsymbol{Z}+\boldsymbol{b}\mathbf{1}_N^{\top}, \boldsymbol{Y}^{\delta})=\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n} l_{CE}(\boldsymbol{z}_{ki}+\boldsymbol{b}, \boldsymbol{y}_k^{\delta}), \tag{27}$$

where $\boldsymbol{y}_k^{\delta}=(1-\delta)\boldsymbol{e}_k+\frac{\delta}{K}\mathbf{1}_K$ is the smoothed target. The gradient of ϕ is

$$\frac{\partial\phi}{\partial \boldsymbol{z}_{ki}}=\frac{1}{N}\left(\boldsymbol{p}_{ki}-\boldsymbol{y}_k^{\delta}\right), \quad \frac{\partial\phi}{\partial \boldsymbol{b}}=\frac{1}{N}\sum_{k,i}\left(\boldsymbol{p}_{ki}-\boldsymbol{y}_k^{\delta}\right),$$

whose matrix form is:

$$\frac{\partial\phi}{\partial \boldsymbol{Z}}=\frac{1}{N}\left(\boldsymbol{P}-\boldsymbol{Y}^{\delta}\right), \quad \frac{\partial\phi}{\partial \boldsymbol{b}}=\frac{1}{N}\left(\boldsymbol{P}-\boldsymbol{Y}^{\delta}\right)\mathbf{1}_N.$$

Hence the gradient (subgradient) of $\mathcal{L}$ is

$$\frac{\partial\mathcal{L}}{\partial \boldsymbol{Z}}=N^{-1}(\boldsymbol{P}-\boldsymbol{Y}^{\delta})+\lambda_Z\partial\|\boldsymbol{Z}\|_*, \quad \frac{\partial\mathcal{L}}{\partial \boldsymbol{b}}=N^{-1}(\boldsymbol{P}-\boldsymbol{Y}^{\delta})\mathbf{1}_N+\lambda_b\boldsymbol{b},$$

where $\partial\|\boldsymbol{Z}\|_*$ is the subdifferential of the nuclear norm at $\boldsymbol{Z}$. Thus, $(\boldsymbol{Z}, \boldsymbol{b})$ is a global minimizer of $\mathcal{L}$ if its gradient (subgradient) is equal to zero, i.e.

$$N^{-1}(\boldsymbol{Y}^{\delta}-\boldsymbol{P}) \in \lambda_Z\partial\|\boldsymbol{Z}\|_*, \quad N^{-1}(\boldsymbol{Y}^{\delta}-\boldsymbol{P})\mathbf{1}_N=\lambda_b\boldsymbol{b}.$$

□

Lemma D.3. *Assume that the number of classes K is less than the feature dimension d, i.e., $K \le d$, and the dataset is balanced. Then the prediction vectors, formulated as $\boldsymbol{z}_{ki}=f_{\Theta}(\boldsymbol{x}_{ki}), (1\le k\le K, 1\le i\le n)$ within each class collapse to their sample means $\bar{\boldsymbol{z}}_k$:*

$$\boldsymbol{z}_{ki}=\bar{\boldsymbol{z}}_k, \quad 1\le i\le n, \tag{28}$$

In other words, the prediction matrix $\boldsymbol{Z}$ can be written as the following factorized form:

$$\boldsymbol{Z}=\overline{\boldsymbol{Z}}\boldsymbol{Y} \in \mathbb{R}^{K\times N} \tag{29}$$

where

$$\overline{\boldsymbol{Z}}=[\bar{\boldsymbol{z}}_1, \ldots, \bar{\boldsymbol{z}}_K], \quad \boldsymbol{Y}=[\boldsymbol{e}_1\mathbf{1}_n^{\top}, \cdots, \boldsymbol{e}_K\mathbf{1}_n^{\top}].$$

Proof. The proof follows from the convexity of the loss function. Recall that the loss function in (21) was given by

$$\mathcal{L}(\boldsymbol{Z}, \boldsymbol{b}) := \phi(\boldsymbol{Z}, \boldsymbol{b}) + \lambda_Z \|\boldsymbol{Z}\|_* + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2,$$

where $\phi(\boldsymbol{Z}, \boldsymbol{b})$ is the Cross-Entropy (CE) or Label Smoothing (LS) loss (depending on the value of the smoothing parameter δ) as defined in (27). Recalling that $\{\boldsymbol{z}_{ik}\}_{i=1}^n$ belong to the same class and $\bar{\boldsymbol{z}}_k = n^{-1}\sum_{i=1}^n \boldsymbol{z}_{ki}$ is their respective sample mean. From Jensen's inequality, we have

$$\begin{aligned}
\phi(\boldsymbol{Z}, \boldsymbol{b}) &= \frac{1}{Kn}\sum_{k=1}^K\sum_{i=1}^n l_{CE}(\boldsymbol{z}_{ki} + b, \boldsymbol{y}_k^\delta) \\
&\geq \frac{1}{K}\sum_{k=1}^K l_{CE}\left(\frac{1}{n}\sum_{i=1}^n(\boldsymbol{z}_{ki} + b), \boldsymbol{y}_k^\delta\right) \\
&= \frac{1}{K}\sum_{k=1}^K l_{CE}(\bar{\boldsymbol{z}}_k + \boldsymbol{b}, \boldsymbol{y}_k^\delta)
\end{aligned}$$

where the inequality becomes equality only when $\boldsymbol{z}_{ki} = \bar{\boldsymbol{z}}_k$.

In the rest of the proof, we employ a permutation argument. Let $\boldsymbol{Z}_l = [\boldsymbol{z}_{1l}, \cdots, \boldsymbol{z}_{Kl}]$ for $1 \leq l \leq n$ and $\tilde{\boldsymbol{Z}} = [\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_n]$. Let Γ_i represent a distinct permutation of n. Consider $\tilde{\boldsymbol{Z}}_{\Gamma_i} = \tilde{\boldsymbol{Z}}\Pi_{\Gamma_i}$, where Π_{Γ_i} is a permutation matrix rearranging $\tilde{\boldsymbol{Z}}$ so that the elements $\boldsymbol{Z}_l$ are ordered according to Γ_i. Then

$$\|\tilde{\boldsymbol{Z}}_{\Gamma_i}\|_* = \|\tilde{\boldsymbol{Z}}\|_* = \|\boldsymbol{Z}\|_*.$$

Since $\|\cdot\|_*$ is a convex function, we deduce

$$\|\boldsymbol{Z}\|_* = \frac{1}{n!}\left(\sum_i \left\|\tilde{\boldsymbol{Z}}_{\Gamma_i}\right\|_*\right) \geq \left\|\frac{1}{n!}\sum_i \tilde{\boldsymbol{Z}}_{\Gamma_i}\right\|_*,$$

where the inequality becomes equality only when $\boldsymbol{Z}_l = \overline{\boldsymbol{Z}}(1 \leq l \leq n)$ or equivalently $\boldsymbol{z}_{ki} = \bar{\boldsymbol{z}}_k$. As a result, it holds that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{Z}, \boldsymbol{b}) &= \phi(\boldsymbol{Z}, \boldsymbol{b}) + \lambda_Z\|\boldsymbol{Z}\|_* + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2 \\
&\geq \frac{1}{K}\sum_{k=1}^K l_{CE}(\bar{z}_k + \boldsymbol{b}, \boldsymbol{y}_k^\delta) + \lambda_Z\|\overline{\boldsymbol{Z}}\|_* + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2 \\
&= \frac{1}{N} l_{CE}\left(\overline{\boldsymbol{Z}}Y + \boldsymbol{b}\boldsymbol{1}_N^\top, \boldsymbol{Y}^\delta\right) + \lambda_Z\|\overline{\boldsymbol{Z}}\|_* + \frac{\lambda_b}{2}\|\boldsymbol{b}\|^2.
\end{aligned}$$

The global optimality of $\mathcal{L}(\boldsymbol{Z}, \boldsymbol{b})$ implies that $\boldsymbol{Z} = \overline{\boldsymbol{Z}}Y$, i.e., $\boldsymbol{z}_{ki} = \bar{\boldsymbol{z}}_k$ for $1 \leq k \leq n$.

□

Lemma D.4. *Assume $\boldsymbol{z}_{ki} = \bar{\boldsymbol{z}}_k$, for $1 \leq k \leq K, 1 \leq i \leq n$ and $\langle \bar{\boldsymbol{z}}_k, \boldsymbol{1}_K\rangle = 0$, then it holds that*

$$\begin{aligned}
&N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right) = \lambda_Z\left(n^{-1/2}\left[\left(\overline{\boldsymbol{Z}}\,\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}} + \overline{\boldsymbol{R}}\right), \\
&\frac{n}{N}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right)\boldsymbol{1}_K = \lambda_b \boldsymbol{b},
\end{aligned}$$

where $\boldsymbol{J}_K \in \mathbb{R}^{K\times K}$ is the ones matrix, $\overline{\boldsymbol{Z}} = [\bar{\boldsymbol{z}}_1, \cdots, \bar{\boldsymbol{z}}_K] \in \mathbb{R}^{K\times K}$, and $\overline{\boldsymbol{R}}$ satisfies $\overline{\boldsymbol{R}}\,\overline{\boldsymbol{Z}}^\top = 0$, $\overline{\boldsymbol{Z}}^\top\overline{\boldsymbol{R}} = 0$, and $\|\overline{\boldsymbol{R}}\| \leq n^{-1/2}$.

In particular, if $\overline{\boldsymbol{Z}}$ is of rank $K-1$, then

$$N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right) = \frac{\lambda_Z}{\sqrt{n}}\left(\left[\left(\overline{\boldsymbol{Z}}\,\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}}\right).$$

Proof. Under the assumption, $\boldsymbol{z}_{ki} = \bar{\boldsymbol{z}}_k$, $1 \leq i \leq n$, and $\langle \bar{\boldsymbol{z}}_k, \mathbf{1}_K \rangle = 0$, we have $\boldsymbol{Z} = \overline{\boldsymbol{Z}}\boldsymbol{Y}$ and $\boldsymbol{P} = \overline{\boldsymbol{P}}\boldsymbol{Y}$, where $\overline{\boldsymbol{P}}$ is defined as

$$\overline{\boldsymbol{P}} = [\bar{\boldsymbol{p}}_1, \cdots, \bar{\boldsymbol{p}}_K]$$

with $\bar{\boldsymbol{p}}_k$ as the probability vector w.r.t. the $\bar{\boldsymbol{z}}_k + \boldsymbol{b}$. Then the optimality condition in (24) reduces to

$$N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right)\boldsymbol{Y} \in \lambda_Z \partial \|\boldsymbol{Z}\|_*. \tag{30}$$

Let $\boldsymbol{Z} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{Z}$, then we have:

$$\partial\|\boldsymbol{Z}\|_* = \{\boldsymbol{U}\boldsymbol{V}^\top + \boldsymbol{R} : \|\boldsymbol{R}\| \leq 1, \boldsymbol{U}^\top\boldsymbol{R} = 0, \boldsymbol{R}\boldsymbol{V} = 0\},$$

where

$$\boldsymbol{U}\boldsymbol{V}^\top = \left[\left(\boldsymbol{Z}\boldsymbol{Z}^\top\right)^\dagger\right]^{1/2}\boldsymbol{Z}.$$

Since $\boldsymbol{Z} = \overline{\boldsymbol{Z}}Y$ and $\boldsymbol{Y}\boldsymbol{Y}^\top = n\boldsymbol{I}_K$, we further get:

$$\boldsymbol{U}\boldsymbol{V}^\top = n^{-1/2}\left[\left(\overline{\boldsymbol{Z}}\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}}\boldsymbol{Y}.$$

Then (30) is equivalent to:

$$N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right)\boldsymbol{Y} = \lambda_Z\left(n^{-1/2}\left[\left(\overline{\boldsymbol{Z}}\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}}\boldsymbol{Y} + \boldsymbol{R}\right),$$

where $\boldsymbol{R}$ is in the form of

$$\boldsymbol{R} = \overline{\boldsymbol{R}}\boldsymbol{Y}, \quad \overline{\boldsymbol{R}} = [\bar{\boldsymbol{r}}_1, \cdots, \bar{\boldsymbol{r}}_K]$$

such that

$$\overline{\boldsymbol{R}}\boldsymbol{Y}\left(\overline{\boldsymbol{Z}}\boldsymbol{Y}\right)^\top = n\overline{\boldsymbol{R}}\overline{\boldsymbol{Z}}^\top = 0, \quad \overline{\boldsymbol{Z}}^\top\overline{\boldsymbol{R}} = 0, \quad \|n^{1/2}\overline{\boldsymbol{R}}\| \leq 1.$$

This further leads to

$$N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right) = \lambda_Z\left(n^{-1/2}\left[\left(\overline{\boldsymbol{Z}}\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}} + \overline{\boldsymbol{R}}\right)$$

where $\overline{\boldsymbol{R}}\overline{\boldsymbol{Z}}^\top = 0$, $\overline{\boldsymbol{Z}}^\top\overline{\boldsymbol{R}} = 0$ and $\|\overline{\boldsymbol{R}}\| \leq n^{-1/2}$.

For $\boldsymbol{b}$, it is easy to see that the optimality condition in equation 24 reduces to

$$\lambda_b\boldsymbol{b} = N^{-1}\left(\boldsymbol{Y}^\delta - \boldsymbol{P}\right)\mathbf{1}_N = \frac{n}{N}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right)\mathbf{1}_K.$$

Now, if $\overline{\boldsymbol{Z}}$ is of rank $K-1$, since $\mathbf{1}_K^\top\overline{\boldsymbol{Z}} = 0$, the columns of $\overline{\boldsymbol{R}}$ are parallel to $\mathbf{1}_K$. Moreover, $\overline{\boldsymbol{P}}$ is a positive left stochastic matrix with $\overline{\boldsymbol{P}}^\top\mathbf{1}_K = \mathbf{1}_K$, and therefore $\boldsymbol{I}_K - \overline{\boldsymbol{P}}$ is also of rank $K-1$. The left stochasticity of $\overline{\boldsymbol{P}}$ further deduces that $(\boldsymbol{I}_K - \overline{\boldsymbol{P}})^T\mathbf{1}_K = 0$. In other words, $\mathbf{1}_K$ is in the left null space of $\boldsymbol{I}_K - \overline{\boldsymbol{P}}$, which in turn implies $\overline{\boldsymbol{R}} = 0$. This leads to

$$N^{-1}\left((1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}}\right) = \frac{\lambda_Z}{\sqrt{n}}\left(\left[\left(\overline{\boldsymbol{Z}}\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}}\right).$$

□

Lemma D.5. *Assume that the number of classes K is less than or equal to the feature dimension d, i.e., $K \leq d$, and the dataset is balanced. Then the global minimizer $(\boldsymbol{Z}, \boldsymbol{b})$ of (21) satisfies the following properties:*

$$\boldsymbol{Z} = \overline{\boldsymbol{Z}}\boldsymbol{Y}, \quad \overline{\boldsymbol{Z}} = a^\delta(\boldsymbol{I}_K - \boldsymbol{J}_K/K), \quad \boldsymbol{b} = 0. \tag{31}$$

In Particular, we have

(i) if $\sqrt{KN}\lambda_Z + \delta \geq 1$*, then* $a^\delta = 0$*;*

(ii) if $\sqrt{KN}\lambda_Z + \delta < 1$*, then*

$$a^\delta = \log\left(\frac{K}{\sqrt{KN}\lambda_Z + \delta} - K + 1)\right)$$

Proof. From Lemma D.3, we have $\boldsymbol{Z} = \overline{\boldsymbol{Z}}Y$. Moreover, according to Lemma 8 in (Zhou et al., 2022b), there exists a constant a such that the global optimum of (21) satisfies:

$$\boldsymbol{Z} = a\left(\boldsymbol{I}_K - \mathbf{1}_K\mathbf{1}_K^\top/K\right)\boldsymbol{Y}, \quad \boldsymbol{b} = 0.$$

Equivalently, we have $\overline{\boldsymbol{Z}} = a\left(\boldsymbol{I}_K - \mathbf{1}_K\mathbf{1}_K^\top/K\right)$. Further, we define $\overline{\boldsymbol{P}} = [\bar{\boldsymbol{p}}_1, \cdots, \bar{\boldsymbol{p}}_K]$ with $\bar{\boldsymbol{p}}_k$ as the probability vector w.r.t. the logit $\boldsymbol{z}_k + \boldsymbol{b}$. Then,

$$\overline{\boldsymbol{P}} = \frac{\boldsymbol{J}_K + (e^a - 1)\boldsymbol{I}_K}{K - 1 + e^a}, \tag{32}$$

and

$$\begin{aligned}(1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}} &= (1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \frac{\boldsymbol{J}_K + (e^a - 1)\boldsymbol{I}_K}{K - 1 + e^a} \\ &= \left(\frac{K}{K-1+e^a} - \delta\right)\left(\boldsymbol{I}_K - \frac{1}{K}\boldsymbol{J}_K\right)\end{aligned}$$

Recall that the first-order optimality condition as expressed in Lemma D.2 indicates that:

$$\begin{aligned}(1-\delta)\boldsymbol{I}_K + \frac{\delta}{K}\boldsymbol{J}_K - \overline{\boldsymbol{P}} &= N\lambda_Z\left(n^{-1/2}\left[\left(\overline{\boldsymbol{Z}}\overline{\boldsymbol{Z}}^\top\right)^\dagger\right]^{1/2}\overline{\boldsymbol{Z}} + \overline{\boldsymbol{R}}\right) &(33)\\ &= N\lambda_Z\left(\operatorname{sign}(a)\sqrt{\frac{1}{n}}(\boldsymbol{I}_K - \boldsymbol{J}_K/K) + \overline{\boldsymbol{R}}\right). &(34)\end{aligned}$$

Suppose $a \neq 0$, then $\overline{\boldsymbol{Z}}$ has rank $K-1$ and we have $\overline{\boldsymbol{R}} = 0$. Thus the above optimality condition implies:

$$\left(\frac{K}{K-1+e^a} - \delta\right)(\boldsymbol{I}_K - \boldsymbol{J}_K/K) = \operatorname{sign}(a)\sqrt{KN}\lambda_Z(\boldsymbol{I}_K - \boldsymbol{J}_K/K),$$

which is equivalent to

$$\frac{K}{K-1+e^a} - \delta = \operatorname{sign}(a)\sqrt{NK}\lambda_Z.$$

For $a > 0$, the above equation has the following solution:

$$a = \log\left(\frac{K}{\sqrt{NK}\lambda_Z + \delta} - K + 1\right) \quad \text{if} \quad \sqrt{NK}\lambda_Z + \delta < 1.$$

If $\sqrt{NK}\lambda_Z + \delta \geq 1$, then we select $a = 0$, and noting that in that case $\overline{\boldsymbol{P}} = K^{-1}\boldsymbol{J}_K$, we have that the optimality condition in (33) implies that $\overline{\boldsymbol{R}}$ satisfies

$$(1-\delta)\boldsymbol{I}_K + \frac{\delta - 1}{K}\boldsymbol{J}_K = N\lambda_Z\overline{\boldsymbol{R}}.$$

We get

$$\overline{\boldsymbol{R}} = \frac{1}{N\lambda_Z}\left[(1-\delta)\boldsymbol{I}_K + \frac{\delta-1}{K}\boldsymbol{J}_K\right],$$

where $\|\overline{\boldsymbol{R}}\| \leq n^{-1/2}$ meets the requirement of the optimality condition.

□

E Proof of Theorem 4.2

In this section of the appendices, we prove Theorem 4.2.

Proof. To provide insight into the accelerated convergence of the model under label smoothing loss, we examine the Hessian matrix of the empirical loss function, defined as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \phi(\boldsymbol{W},\boldsymbol{H}) := \frac{1}{N} l_{CE}(\boldsymbol{W}^\top \boldsymbol{H}, \boldsymbol{Y}^\delta). \tag{35}$$

Under commonly used deep learning frameworks, the model parameters are updated iteratively, and it is common and often practical to analyze the Hessian matrix with respect to $\boldsymbol{H}$, $\boldsymbol{W}$ individually rather than considering the full Hessian. Particularly, we demonstrate that the Hessian of the empirical loss concerning $\boldsymbol{W}$ when $\boldsymbol{H}$ is fixed and the Hessian w.r.t. $\boldsymbol{H}$ when $\boldsymbol{W}$ is fixed are positive semi-definite at the global optimizer under both cross-entropy and label smoothing losses. Furthermore, we establish that the condition numbers of these Hessian matrices are notably lower under the LS loss in comparison to the CE loss.

Hessian matrix with respect to $\boldsymbol{Z}$. Let $\boldsymbol{Z} = \boldsymbol{W}^\top \boldsymbol{H} \in \mathbb{R}^{K\times N}$ represent the prediction logit matrix. In the proof of Lemma D.2, we obtained the first-order partial derivatives of the loss ϕ:

$$\frac{\partial \phi}{\partial \boldsymbol{z}_{ki}} = \frac{1}{N}\left(\boldsymbol{p}_{ki} - \boldsymbol{y}_k^\delta\right), \quad \frac{\partial \phi}{\partial \boldsymbol{b}} = \frac{1}{N}\sum_{k,i}\left(\boldsymbol{p}_{ki} - \boldsymbol{y}_k^\delta\right),$$

where $\boldsymbol{p}_{ki}$ is the prediction for the i-th sample that belongs to the k-th class and $\boldsymbol{y}_k^\delta$ is the soft label for class k with a smoothing parameter δ. These partial derivatives lead to the corresponding second-order partial derivatives of the loss ϕ:

$$\frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}^2} = \frac{1}{N}\left(\operatorname{diag}(\boldsymbol{p}_{ki}) - \boldsymbol{p}_{ki}\boldsymbol{p}_{ki}^\top\right), \quad \frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}\partial \boldsymbol{z}_{k'i'}} = 0, \forall (k,i) \neq (k',i'),$$

$$\frac{\partial^2 \phi}{\partial \boldsymbol{b}^2} = \frac{1}{N}\sum_{k,i}\left(\operatorname{diag}(\boldsymbol{p}_{ki}) - \boldsymbol{p}_{ki}\boldsymbol{p}_{ki}^\top\right), \quad \frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}\partial \boldsymbol{b}} = \frac{1}{N}\left(\operatorname{diag}(\boldsymbol{p}_{ki}) - \boldsymbol{p}_{ki}\boldsymbol{p}_{ki}^\top\right).$$

From Theorem 4.1, we have the global optimizer satisfies $\boldsymbol{z}_{ki}^2 = \bar{\boldsymbol{z}}_k$ and $\boldsymbol{p}_{ki}^2 = \bar{\boldsymbol{p}}_k, 1 \le k \le K, 1 \le i \le n$. Consequently, it follows that $\frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}^2} = \frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki'}^2} (\forall 1 \le i, i' \le n)$. To simply the notation, we denote

$$\boldsymbol{D}_k = \operatorname{diag}(\bar{\boldsymbol{p}}_k) - \bar{\boldsymbol{p}}_k \bar{\boldsymbol{p}}_k^\top. \tag{36}$$

Then we have $\frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}^2} = \boldsymbol{D}_k / N$.

The closed-form solution for $\overline{\boldsymbol{P}}$ in (32) yields the expression:

$$\bar{\boldsymbol{p}}_k = \frac{1}{K-1+e^{a^\delta}}\left((e^{a^\delta}-1)\boldsymbol{e}_k + \mathbf{1}_K\right). \tag{37}$$

To simplify the notation, we denote

$$p_t = e^{a^\delta}/(K-1+e^{a^\delta}), \quad p_n = 1/(K-1+e^{a^\delta}), \tag{38}$$

where p_t (p_n) is the predicted probability for the target (non-target) class at the global optimal solution. Under this notation, we have $p_t + (K-1)p_n = 1$ and $\bar{\boldsymbol{p}}_k = (p_t - p_n)\boldsymbol{e}_k + p_n \mathbf{1}_K$.

For any $1 \le k \le K$, $\boldsymbol{p}_k \boldsymbol{p}_k^\top$ is a positive matrix and the associated Laplacian $\boldsymbol{D}_k$ is positive-semidefinite with all eigenvalues non negative. The smallest eigenvalue of the Laplacian matrix $\boldsymbol{D}_k$ is $\sigma_1 = 0$ with the corresponding eigenvector $\boldsymbol{v}_1 = \mathbf{1}_K / \|\mathbf{1}_K\|$.

Define

$$\boldsymbol{v}_2 = (K\boldsymbol{e}_1 - \mathbf{1}_K)/\|K\boldsymbol{e}_1 - \mathbf{1}_K\|,$$

then we have

$$\begin{aligned}\boldsymbol{D}_k\boldsymbol{v}_2 &= \operatorname{diag}\left(\bar{\boldsymbol{p}}_k\right)\boldsymbol{v}_2 - \bar{\boldsymbol{p}}_k\bar{\boldsymbol{p}}_k^\top\boldsymbol{v}_2\\ &= Kp_tp_n\boldsymbol{v}_2,\end{aligned}$$

from which we get Kp_tp_n is an eigenvalue of $\boldsymbol{D}_k$ with the corresponding eigenvector $\boldsymbol{v}_2$.

Consider the outer-product matrix $\boldsymbol{p}_k\boldsymbol{p}_k^\top$, its largest eigenvalue is $\|\boldsymbol{p}_k\|^2$ with the eigenvector $\boldsymbol{p}_k/\|\boldsymbol{p}_k\|$. Its null space is of dimension $K-1$. Particularly, we can find a set of the basis vectors for $\text{Null}(\boldsymbol{p}_k\boldsymbol{p}_k^\top)$ as follows:

$$\{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_{K-1} : \text{span}(\boldsymbol{o}_1, \boldsymbol{p}_k) = \text{span}(\boldsymbol{e}_k, \boldsymbol{p}_k)\}, \tag{39}$$

which means the vectors $\boldsymbol{o}_1$ and $\boldsymbol{p}_k$ span the same 2D space as $\boldsymbol{e}_k$ and $\boldsymbol{p}_k$.

Note that

$$\text{span}(\boldsymbol{e}_k, \boldsymbol{p}_k) = \text{span}(\boldsymbol{v}_1, \boldsymbol{v}_2),$$

where $\boldsymbol{v}_1, \boldsymbol{v}_2$ are the eigenvector for $\boldsymbol{D}_k$.

Then we can show that $\boldsymbol{o}_2 \cdots, \boldsymbol{o}_{K-1}$ are eigenvectors of $\boldsymbol{D}_k$ with the corresponding eigenvalue as $\sigma_3 = p_n$. Particularly, with $\boldsymbol{o}_l$ orthonormal to both $\bar{\boldsymbol{p}}_k$ and $\boldsymbol{e}_k$, we have

$$\begin{aligned}\boldsymbol{D}_k\boldsymbol{o}_l &= \operatorname{diag}\left(\bar{\boldsymbol{p}}_k\right)\boldsymbol{o}_l - \bar{\boldsymbol{p}}_k\bar{\boldsymbol{p}}_k^\top\boldsymbol{o}_l\\ &= \operatorname{diag}\left(\bar{\boldsymbol{p}}_k\right)\boldsymbol{o}_l\\ &= p_n\boldsymbol{o}_l,\end{aligned}$$

for $\boldsymbol{o}_l (2 \le l \le K-1)$ defined in (39).

To summarize the unique eigenvalues of the matrix $\boldsymbol{D}$ are

$$\sigma_1 = 0, \quad \sigma_2 = p_n, \quad \sigma_3 = Kp_tp_n \tag{40}$$

with p_t and p_n defined in (38).

Hessian matrix with respect to $\boldsymbol{H}$. The gradient of ϕ with respect to $\boldsymbol{h}_{ki}$ is

$$\frac{\partial\phi}{\partial\boldsymbol{h}_{ki}} = \frac{\partial\boldsymbol{z}_{ki}}{\partial\boldsymbol{h}_{ki}}\frac{\partial\phi}{\partial\boldsymbol{z}_{ki}} = \boldsymbol{W}\frac{\partial\phi}{\partial\boldsymbol{z}_{ki}}.$$

Further, we can easily get the corresponding Hessian:

$$\frac{\partial^2\phi}{\partial\boldsymbol{h}_{ki}^2} = \boldsymbol{W}\frac{\partial^2\phi}{\partial\boldsymbol{z}_{ki}^2}\boldsymbol{W}^\top = \frac{1}{N}\boldsymbol{W}\boldsymbol{D}_k\boldsymbol{W}^\top, \quad \frac{\partial^2\phi}{\partial\boldsymbol{h}_{ki}\partial\boldsymbol{h}_{k'i'}} = 0, \tag{41}$$

where $\boldsymbol{D}_k$ is the Laplacian matrix as defined in (36).

From Theorem 4.1, we have

$$\boldsymbol{W} = \left(\frac{\lambda_H n}{\lambda_W}\right)^{1/4}\sqrt{a^\delta}\boldsymbol{P}\left(\boldsymbol{I}_K - \boldsymbol{J}_K/K\right).$$

For simplicity, we denote $a_W^\delta = (\lambda_H n/\lambda_W)^{1/4}\sqrt{a^\delta}$, then $\boldsymbol{W} = a_W^\delta\boldsymbol{P}(\boldsymbol{I}_K - \boldsymbol{J}_K/K)$. Under this notation, we have

$$\begin{aligned}\frac{1}{N}\boldsymbol{W}\boldsymbol{D}_k\boldsymbol{W}^\top &= \frac{(a_W^\delta)^2}{N}\boldsymbol{P}(\boldsymbol{I}_K - \boldsymbol{J}_K/K)\boldsymbol{D}_k(\boldsymbol{I}_K - \boldsymbol{J}_K/K)\boldsymbol{P}^\top \quad (42)\\ &= \frac{(a_W^\delta)^2}{N}\boldsymbol{P}\boldsymbol{D}_k\boldsymbol{P}^\top. \quad (43)\end{aligned}$$

Note that $\boldsymbol{P} \in \mathbb{R}^{d\times K}, (d > K)$ is a partial orthogonal matrix. Given the eigenvalues of $\boldsymbol{D}_k$ in (40), the eigenvalues for (42) are:

$$\lambda_1 = 0, \quad \lambda_2 = \frac{(a_W^\delta)^2}{N} p_n, \quad \lambda_3 = \frac{(a_W^\delta)^2}{N} K p_t p_n \tag{44}$$

with the corresponding multiplicities $m(\lambda_1) = 1 + d - K$, $m(\lambda_2) = K - 2$, $m(\lambda_3) = 1$.

From (47), the hessian of ϕ with respect to $\boldsymbol{h} = \mathrm{vec}(\boldsymbol{H})$ is a block diagonal matrix which can be expressed as

$$\frac{\partial^2 \phi}{\partial \boldsymbol{h}^2} = \frac{1}{N} \mathrm{blkdiag}(\boldsymbol{W}\boldsymbol{D}_1\boldsymbol{W}^\top, \cdots, \boldsymbol{W}\boldsymbol{D}_1\boldsymbol{W}^\top, \cdots, \boldsymbol{W}\boldsymbol{D}_K\boldsymbol{W}^\top, \cdots, \boldsymbol{W}\boldsymbol{D}_K\boldsymbol{W}^\top). \tag{45}$$

The unique eigenvalues of the Hessian matrix $\frac{\partial^2 \phi}{\partial \boldsymbol{h}^2}$ are the same as provided in (44). Given that the Hessian matrix contains a zero eigenvalue, our analysis centers on its condition number within the non-zero eigenvalue space. This is calculated as follows:

$$\kappa(\nabla^2_{\boldsymbol{h}} \phi) = \lambda_3/\lambda_2 = K p_t \tag{46}$$

Considering the formula for p_t given in (38), an increase in δ leads to a decrease in p_t, subsequently resulting in a smaller condition number.

Hessian matrix with respect to $\boldsymbol{W}$. The gradient of ϕ with respect to $\boldsymbol{w}_l (l = 1, \cdots, K)$ is

$$\frac{\partial \phi}{\partial \boldsymbol{w}_l} = \frac{\partial \boldsymbol{z}_{ki}}{\partial \boldsymbol{w}_l} \frac{\partial \phi}{\partial \boldsymbol{z}_{ki}}.$$

Further, we get the corresponding Hessian:

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial \boldsymbol{w}_l \partial \boldsymbol{w}_{l'}} &= \sum_{k,i} \frac{\partial \boldsymbol{z}_{ki}}{\partial \boldsymbol{w}_{l'}} \frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}^2} \left(\frac{\partial \boldsymbol{z}_{ki}}{\partial \boldsymbol{w}_l}\right)^\top \\
&= \sum_{k,i} \boldsymbol{h}_{ki} \boldsymbol{e}_{l'}^\top \frac{\partial^2 \phi}{\partial \boldsymbol{z}_{ki}^2} \boldsymbol{e}_l \boldsymbol{h}_{ki}^\top \\
&= \sum_{k,i} \boldsymbol{D}_k(l', l) \boldsymbol{h}_{ki} \boldsymbol{h}_{ki}^\top \\
&= n \sum_{k} \boldsymbol{D}_k(l', l) \bar{\boldsymbol{h}}_k \bar{\boldsymbol{h}}_k^\top
\end{aligned}$$

where $\boldsymbol{D}_k(l', l)$ is (l', l)-th element in $\boldsymbol{D}_k$ and $\bar{\boldsymbol{h}}_k$ is the k-th column vector in $\overline{\boldsymbol{H}}$ as defined in 19. From the defination of $\boldsymbol{D}_k$ in (36), we have

$$\frac{\partial^2 \phi}{\partial \boldsymbol{w}_l \partial \boldsymbol{w}_{l'}} = \begin{cases} n\overline{\boldsymbol{H}} \left[\mathrm{diag}(\bar{\boldsymbol{p}}_l) - \mathrm{diag}(\bar{\boldsymbol{p}}_l)^2\right] \overline{\boldsymbol{H}}^\top & \text{if } l = l' \\ n\overline{\boldsymbol{H}} \left[-\mathrm{diag}(\bar{\boldsymbol{p}}_l)\mathrm{diag}(\bar{\boldsymbol{p}}_{l'})\right] \overline{\boldsymbol{H}}^\top & \text{if } l \neq l' \end{cases}$$

where $\bar{\boldsymbol{p}}_l$ is the average prediction vector as defined in (37). Particularly, the l-th element of $\bar{\boldsymbol{p}}_l$ equals p_t and the others equal p_n.

From (19), we have

$$\overline{\boldsymbol{H}} = \left(\frac{\lambda_W}{\lambda_H n}\right)^{1/4} \sqrt{a^\delta} \boldsymbol{P} \left(\boldsymbol{I}_K - \boldsymbol{J}_K/K\right).$$

Hence, the Hessian of ϕ w.r.t. the $\boldsymbol{w} = \mathrm{vec}(\boldsymbol{W})$ can be written as:

$$\frac{\partial^2 \phi}{\partial \boldsymbol{w}^2} = n \left(\frac{\lambda_W}{\lambda_H n}\right)^{1/2} a^\delta \boldsymbol{D}_P \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{D}_\Pi \boldsymbol{D}_P^\top \tag{47}$$

where

$$\boldsymbol{D}_P = \mathrm{blkdiag}(\boldsymbol{P}, \cdots, \boldsymbol{P}) \in \mathbb{R}^{Kd\times K^2}, \tag{48}$$

$$\boldsymbol{D}_\Pi = \text{blkdiag}(\boldsymbol{\Pi}, \cdots, \boldsymbol{\Pi}) \in \mathbb{R}^{K^2 \times K^2}, \quad \boldsymbol{\Pi} = \boldsymbol{I}_K - \frac{\mathbf{1}_K \mathbf{1}_K^\top}{K} \tag{49}$$

and

$$\boldsymbol{S} = \begin{bmatrix} \Lambda_1 & & \\ & \ddots & \\ & & \Lambda_K \end{bmatrix} - \begin{bmatrix} \Lambda_1 \\ \vdots \\ \Lambda_K \end{bmatrix} \begin{bmatrix} \Lambda_1 & \cdots & \Lambda_K \end{bmatrix} \tag{50}$$

with $\Lambda_l = \text{diag}\left(\bar{\boldsymbol{p}}_l\right)$.

Since $\boldsymbol{P}$ is a partial orthogonal matrix, the condition number of (47) is the same as the condition number of the following matrix:

$$\boldsymbol{B} = \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{D}_\Pi. \tag{51}$$

Subsequently, we proceed to derive the eigenvalues for the above matrix $\boldsymbol{B}$.

First, considering an eigenvector $\boldsymbol{v}$ of $\boldsymbol{B}$, let Δ be a $K \times K$ matrix such that $\boldsymbol{v} = \text{vec}(\Delta)$. Considering any Δ satisfying $\Delta\boldsymbol{\Pi} = 0$ or $\Delta^\top \boldsymbol{\Pi} = 0$, for the corresponding $\boldsymbol{v} = \text{vec}(\Delta)$ we get $\boldsymbol{D}_\Pi \boldsymbol{v} = 0$ and $\boldsymbol{v}^\top \boldsymbol{D}_\Pi = 0$, wich further yields $\boldsymbol{B}\boldsymbol{v} = 0$. Since $\text{rank}(\boldsymbol{\Pi}) = K - 1$, it follows that $\lambda_1 = 0$ is an eigenvalue of $\boldsymbol{B}$ with multiplicity $2K - 1$.

On the other hand, consider any Δ that satisfies the conditions:

$$\text{diag}(\Delta) = 0, \quad \Delta \mathbf{1}_K = \Delta^\top \mathbf{1}_K = 0. \tag{52}$$

It is noteworthy that from $\Delta \mathbf{1}_K = \Delta^\top \mathbf{1}_K = 0$, the corresponding vector $\boldsymbol{v}$ satisfies $\boldsymbol{D}_\Pi \boldsymbol{v} = \boldsymbol{v}$. If Δ further satisfies $\text{diag}(\Delta) = 0$ then $\boldsymbol{S}\boldsymbol{v} = p_n \boldsymbol{v}$. Consequently, we deduce that

$$\boldsymbol{B}\boldsymbol{v} = \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{D}_\Pi \boldsymbol{v} = \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{v} = \boldsymbol{D}_\Pi p_n \boldsymbol{v} = p_n \boldsymbol{v}$$

We can conclude that p_n is an eigenvalue of $\boldsymbol{B}$ with multiplicity $K^2 - K - (2K - 1) = K^2 - 3K + 1$.

Next, let us consider the matrix $\Delta = \boldsymbol{\Pi}\text{diag}(\boldsymbol{u})\boldsymbol{\Pi}$, where $\boldsymbol{u}$ is a vector satisfying $\boldsymbol{u}^\top \mathbf{1}_K = 0$. The vectorized form of Δ can be denoted as

$$\boldsymbol{v} = \text{vec}(\Delta) = \begin{bmatrix} \boldsymbol{\Pi}^\top \text{diag}(\boldsymbol{u}) \boldsymbol{\Pi}_1 \\ \boldsymbol{\Pi}^\top \text{diag}(\boldsymbol{u}) \boldsymbol{\Pi}_2 \\ \vdots \\ \boldsymbol{\Pi}^\top \text{diag}(\boldsymbol{u}) \boldsymbol{\Pi}_K \end{bmatrix},$$

where $\boldsymbol{\Pi}_k$ represents the k-th column of the matrix $\boldsymbol{\Pi}$. Due to the properties $\boldsymbol{\Pi} = \boldsymbol{\Pi}^\top$ and $\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}$, it follows that $\boldsymbol{D}_\Pi \boldsymbol{v} = \boldsymbol{v}$. Additionally, given $\boldsymbol{u}^\top \mathbf{1}_K = 0$, we obtain $\boldsymbol{S}\boldsymbol{v} = \lambda_2 \boldsymbol{v}$ with $\lambda_2 = (1 - p_t + p_n) \cdot \frac{p_n + (K-1)p_t}{K}$. Consequently, we conclude that $\lambda_2 = (1 - p_t + p_n) \cdot \frac{p_n + (K-1)p_t}{K}$ is an eigenvalue of $\boldsymbol{B}$ with a multiplicity equal to the degrees of freedom of $\boldsymbol{u}$, which is $K - 1$.

Lastly, considering $\boldsymbol{v} = \text{vec}(\boldsymbol{\Pi})$, we observe that $\boldsymbol{D}_\Pi \boldsymbol{v} = \boldsymbol{v}$. Further, we have $\boldsymbol{S}\boldsymbol{v} = K p_t p_n \boldsymbol{v}$. Hence, we obtain

$$\boldsymbol{B}\boldsymbol{v} = \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{D}_\Pi \boldsymbol{v} = \boldsymbol{D}_\Pi \boldsymbol{S} \boldsymbol{v} = K p_t p_n \boldsymbol{v}.$$

Therefore, $\boldsymbol{v} = \text{vec}(\boldsymbol{\Pi})$ is an eigenvector of $\boldsymbol{B}$ with eigenvalue $\lambda_3 = K p_t p_n$.

In summary, we identify the distinct eigenvalues of the matrix $\boldsymbol{B}$ as defined in (51) as follows:

$$\lambda_0 = 0, \quad \lambda_1 = p_n, \quad \lambda_2 = (1 - p_t + p_n) \cdot \frac{p_n + (K-1)p_t}{K}, \quad \lambda_3 = K p_n p_t \tag{53}$$

with the corresponding multiplicities $2K - 1$, $K^2 - 3K + 1$, $K - 1$, and 1, respectively.

Since the matrix $\boldsymbol{B}$ has zero eigenvalues, we consider its condition number within the non-zero eigenvalue subspace, which is given by $\lambda_3 / \lambda_1 = K p_t$. Consequently, we straightforwardly determine the condition number of the Hessian $\frac{\partial^2 \phi}{\partial \boldsymbol{w}^2}$ as defined in (47) is

$$\kappa(\nabla^2_{\boldsymbol{w}} \phi) = K p_t. \tag{54}$$

From the formula of p_t given in (38), it is evident that increasing δ leads to a decrease of p_t, and consequently a reduction in the condition number.

By combining (46) and (54), we demonstrate that the Hessian matrices, $\nabla^2_{\boldsymbol{W}}\phi(\boldsymbol{W},\boldsymbol{H})$ and $\nabla^2_{\boldsymbol{H}}\phi(\boldsymbol{W},\boldsymbol{H})$, are positive semi-definite at the global optimizer. Moreover, the condition numbers of these Hessian matrices are notably lower under the label smoothing loss (with $0 < \delta < 1 - \sqrt{KN\lambda_W\lambda_H}$) compared to the cross-entropy loss (with $\delta = 0$). This observation suggests that the optimization landscape of the empirical loss function is better conditioned around its global minimizer under label smoothing loss, thus completing our proof.

$\square$