Are All Losses Created Equal: A Neural Collapse Perspective

Jinxin Zhou
Ohio State University
zhou.3820@osu.edu

Chong You
Google Research
cyou@google.com

Xiao Li
University of Michigan
xlxiao@umich.edu

Kangning Liu
New York University
kl3141@nyu.edu

Sheng Liu
New York University
shengliu@nyu.edu

Qing Qu
University of Michigan
qingqu@umich.edu

Zhihui Zhu*
Ohio State University
zhu.3440@osu.edu

Abstract

While cross entropy (CE) is the most commonly used loss function to train deep neural networks for classification tasks, many alternative losses have been developed to obtain better empirical performance. Among them, which one is the best to use is still a mystery, because there seem to be multiple factors affecting the answer, such as properties of the dataset, the choice of network architecture, and so on. This paper studies the choice of loss function by examining the last-layer features of deep networks, drawing inspiration from a recent line work showing that the global optimal solution of CE and mean-square-error (MSE) losses exhibits a *Neural Collapse* ($\mathcal{NC}$) phenomenon. That is, for sufficiently large networks trained until convergence, (i) all features of the same class collapse to the corresponding class mean and (ii) the means associated with different classes are in a configuration where their pairwise distances are all equal and maximized. We extend such results and show through global solution and landscape analyses that a broad family of loss functions including commonly used label smoothing (LS) and focal loss (FL) exhibits $\mathcal{NC}$. Hence, all relevant losses (i.e., CE, LS, FL, MSE) produce equivalent features on training data. In particular, based on the *unconstrained feature model* assumption, we provide either the global landscape analysis for LS loss or the local landscape analysis for FL loss and show that the (only!) global minimizers are $\mathcal{NC}$ solutions, while all other critical points are strict saddles whose Hessian exhibit negative curvature directions either in the global scope for LS loss or in the local scope for FL loss near the optimal solution. The experiments further show that $\mathcal{NC}$ features obtained from all relevant losses (i.e., CE, LS, FL, MSE) lead to largely identical performance on test data as well, provided that the network is sufficiently large and trained until convergence. The source code is available at https://github.com/jinxinzhou/nc_loss.

1 Introduction

Loss function is an indispensable component in the training of deep neural networks (DNNs). While cross-entropy (CE) loss is one of the most popular choices for classification tasks, studies over the past few years have suggested many improved versions of CE that bring better empirical performance. Some notable examples include label smoothing (LS) [1] where one-hot label is replaced by a smoothed label, focal loss (FL) [2] which puts more emphasis on hard misclassified samples and

*Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

reduces the relative loss on the already well-classified samples, and so on. Aside from CE and its variants, the mean squared error (MSE) loss which was typically used for regression tasks is recently demonstrated to have a competitive performance when compared to CE for classification tasks [3].

Despite the existence of many loss functions there is however a lack of consensus as to which one is the best to use, and the answer seems to depend on multiple factors such as properties of the dataset, choice of network architecture, and so on [4]. In this work, we aim to understand the effect of loss function in classification tasks from the perspective of characterizing the last-layer features and classifier of a DNN trained under different losses. Our study is motivated by a sequence of recent work that identify an intriguing *Neural Collapse* ($\mathcal{NC}$) phenomenon in trained networks, which refers to the following properties of the last-layer features and classifier:

(i) ***Variability Collapse*:** all features of the same class collapse to the corresponding class mean.

(ii) ***Convergence to Simplex ETF*:** the means associated with different classes are in a Simplex Equiangular Tight Frame (ETF) configuration where their pairwise distances are all equal and maximized.

(iii) ***Convergence to Self-duality*:** the class means are ideally aligned with the last-layer linear classifiers.

(iv) ***Simple Decision Rule*:** the last-layer classifier is equivalent to a Nearest Class-Center decision rule.

This $\mathcal{NC}$ phenomena is first discovered by Papyan et al. [5, 6] under canonical classification problems trained with the CE loss. Following with the CE loss, Han et al. [7] recently reported that DNNs trained with MSE loss for classification problems also exhibit similar $\mathcal{NC}$ phenomena. These results imply that deep networks are essentially learning maximally separable features between classes, and a max-margin classifier in the last layer upon these learned features. The intriguing empirical observation motivated a surge of theoretical investigation [7–22], mostly under a simplified *unconstrained feature model* [10] or *layer-peeled model* [12] that treats the last-layer features of each samples before the final classifier as free optimization variables. Under the simplified unconstrained feature model, it has been proved that the $\mathcal{NC}$ solution is the only global optimal solution for the CE and MSE losses which are also proved to have benign global landscape, explaining why the global $\mathcal{NC}$ solution can be obtained.

Contributions. While previous work provide thorough analysis for $\mathcal{NC}$ under CE and MSE losses, the theoretical analysis beyond CE and MSE losses is still limited, and their work only focus on one specific loss without a general format. In this paper, we consider a broad family of loss functions that includes CE and some other popular loss functions such as LS and FL as special cases. Under the *unconstrained feature model*, we theoretically demonstrate in Section 3 that the $\mathcal{NC}$ solution is the only global optimal solution to the family of loss functions. Moreover, we provide a global landscape analysis, showing that the LS loss function is a strict saddle function and FL loss function is a local strict saddle function [23–25]. A (local) strict saddle function is a function for which every critical point is either a global solution or a strict saddle point with negative curvature (locally). Hence, our result suggests that any optimizer can escape strict saddle points and converge to the global solution responding to $\mathcal{NC}$ for LS and FL. As far as we know, this paper is the first work that conducts global optimal solution and benign optimization landscape analysis beyond the scope of CE and MSE losses.

Our theoretical results explained above have important implications for understanding the role of loss function in training DNNs for classification tasks. Because all losses lead to $\mathcal{NC}$ solutions, their corresponding features are equivalent up to a rotation of the feature space. In other words, our analysis provides a theoretical justification for the following claim:

> *All losses (i.e., CE, LS, FL, MSE) lead to largely identical features on* ***training*** *data by large DNNs and sufficiently many iterations.*

We also provide an experimental verification of this claim through experiments in Section 4.1.

While $\mathcal{NC}$ reveals that all losses are equivalent at training time, it does not have a direct implication for the features associated with test data as well as the generalization performance [26]. In particular, a recent work [27] shows empirically that $\mathcal{NC}$ does not occur for the features associated with test data. Nonetheless, we show through empirical evidence that for large DNNs, $\mathcal{NC}$ on training

data well predicts the test performance. In particular, our empirical study in Section 4.2 shows the following:

*All losses (CE, LS, FL, MSE) lead to largely identical performance on **test** data by large DNNsand sufficiently many iterations.*

Our conclusion that all losses are created equal appears to go against existing evidence on the advantages of some losses over the others. Here we emphasize that our conclusion has an important premise, namely the neural network has sufficient approximation power and the training is performed for sufficiently many iterations. Hence, our conclusion implies that the better performance with particular choices of loss functions comes as a result that the training does not produce a globally optimal (i.e., $\mathcal{NC}$) solution. In such cases different losses lead to different solutions on the training data, and correspondingly different performance on test data. Such an understanding may provide important practical guidance on what loss to choose in different cases (e.g., different model sizes and different training time budgets), as well as for the design of new and better losses in the future. We note that our conclusion is based on natural accuracy, rather than model transferability or robustness, which is worth additional efforts to exploit and is left as future work.

2 The Problem Setup

A typical deep neural network $\Psi(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^K$ consists of a multi-layer nonlinear compositional feature mapping $\Phi(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^d$ and a linear classifier $(\boldsymbol{W}, \boldsymbol{b})$, which can be generally expressed as

$$\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}\Phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{b}, \tag{1}$$

where we use $\boldsymbol{\theta}$ to represent the network parameters in the feature mapping and $\boldsymbol{W} \in \mathbb{R}^{K\times d}$ and $\boldsymbol{b} \in \mathbb{R}^K$ to represent the linear classifier's weight and bias, respectively. Therefore, *all* the network parameters are the set of $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}\}$. For the input $\boldsymbol{x}$, the output of the feature mapping $\Phi_{\boldsymbol{\theta}}(\boldsymbol{x})$ is usually termed as the *representation* or *feature* learned from the network.

With an appropriate loss function, the parameters $\boldsymbol{\Theta}$ of the whole network are optimized to learn the underlying relation from the input sample $\boldsymbol{x}$ to their corresponding target $\boldsymbol{y}$ so that the output of the network $\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x})$ approximates the corresponding target, i.e. $\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \approx \boldsymbol{y}$ in term of the expectation over a distribution $\mathcal{D}$ of input-output data pairs $(\boldsymbol{x}, \boldsymbol{y})$. While it is hard to get access to the ground-truth distribution $\mathcal{D}$ in most cases, one can approximate the distribution $\mathcal{D}$ through sampling enough data pairs i.i.d. from $\mathcal{D}$. In this paper, we study the multi-class balanced classification tasks with K class and n samples per class, where we use the one-hot vector $\boldsymbol{y}_k \in \mathbb{R}^K$ with unity only in k-th entry $(1 \le k \le K)$ to denote the label of the i-th sample $\boldsymbol{x}_{k,i} \in \mathbb{R}^D$ in the k-th class. We then learn the parameters $\boldsymbol{\Theta}$ via minimizing the following empirical risk over the total $N = nK$ training samples

$$\min_{\boldsymbol{\Theta}} \; \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n} \mathcal{L}\left(\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i}), \boldsymbol{y}_k\right) \;+\; \frac{\lambda}{2}\left\|\boldsymbol{\Theta}\right\|_F^2, \tag{2}$$

where $\lambda > 0$ is the regularization parameter (a.k.a., the weight decay parameter[2]) and $\mathcal{L}\left(\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i}), \boldsymbol{y}_k\right)$ is a predefined loss function that appropriately measures the difference between the output $\Psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i})$ and the target $\boldsymbol{y}_k$. Some common loss functions used for training deep neural networks will be specified in the next section.

2.1 Commonly Used Training Losses

In this subsection, we first present four common loss functions for classification task. To simplify the notation, let $\boldsymbol{z} = \boldsymbol{W}\Phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{b}$ denote the network's output ("logit") vector for the input $\boldsymbol{x}$. Assume $\boldsymbol{z}$ belongs to the k-th class. Also let $\boldsymbol{y}_k^{\text{smooth}} = (1-\alpha)\boldsymbol{y}_k + \frac{\alpha}{K}\boldsymbol{1}_K$ denote the smoothed targets of k-th class, where $0 \le \alpha < 1$ and $\boldsymbol{1}_K \in \mathbb{R}^K$ is a vector with all entries equal to one. We will use z_ℓ, $y_{k,\ell}$ and y_ℓ^{smooth} to denote the ℓ-th entry of $\boldsymbol{z}$, $\boldsymbol{y}_k$ and $\boldsymbol{y}_k^{\text{smooth}}$, respectively, where $y_{k,k}^{\text{smooth}} = 1 - \frac{K-1}{K}\alpha$ and $y_{k,\ell}^{\text{smooth}} = \frac{\alpha}{K}$ for $k \ne \ell$.

[2]Without weight decay, the features and classifiers will tend to blow up for CE and many other losses.

Cross entropy (CE) is perhaps the most common loss for multi-class classification in deep learning. It measures the distance between the target distribution $\boldsymbol{y}_k$ and the network output distribution obtained by applying the softmax function on $\boldsymbol{z}$, resulting in the following expression

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{z}, \boldsymbol{y}_k) \;=\; -\log\left(\frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}\right). \tag{3}$$

Focal loss (FL) [2] is first proposed to deal with the extreme foreground-background class imbalance in dense object detection, which adaptively focuses less on the well-classified samples. Recent work [28, 29] reports that focal loss also improves calibration and automatically forms curriculum learning in multi-class classification setting. Letting $\gamma \geq 0$ denote the tunable focusing parameter, the focal loss can be expressed as:

$$\mathcal{L}_{\mathrm{FL}}(\boldsymbol{z}, \boldsymbol{y}_k) \;=\; -\left(1-\frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}\right)^{\gamma}\log\left(\frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}\right). \tag{4}$$

Label smoothing (LS) [1] replaces the hard targets in CE with smoothed targets $\boldsymbol{y}_k^{\text{smooth}}$ obtained from mixing the original targets $\boldsymbol{y}_k$ with a uniform distribution over all entries $\mathbf{1}_K$. Experiments in [30, 31] find that classification models trained with label smoothing have better calibration and generalization. Denoting by $0 \leq \alpha \leq 1$ the tunable smoothing parameter, the label smoothing loss function can be formulated as:

$$\mathcal{L}_{\mathrm{LS}}(\boldsymbol{z}, \boldsymbol{y}_k) \;=\; -\sum_{\ell=1}^{K} y_{k,\ell}^{\text{smooth}}\log\left(\frac{\exp(z_\ell)}{\sum_{j=1}^{K}\exp(z_j)}\right). \tag{5}$$

When $\alpha = 0$, the above label smoothing loss reduces to the CE loss.

Mean square error (MSE) is often used for regression but not classification task. The recent work [3] shows that classification networks trained with MSE loss achieve on par performance compared to those trained with the CE loss. Throughout our paper, we use the rescaled MSE version [3]:

$$\mathcal{L}_{\mathrm{MSE}}(\boldsymbol{z}, \boldsymbol{y}_k) \;=\; \kappa(z_k-\beta)^2 + \sum_{\ell\neq k}^{K} z_\ell^2, \tag{6}$$

where $\kappa > 0$ and $\beta > 0$ are hyperparameters.

2.2 Problem Formulation Based on Unconstrained Feature Models

Because of the interaction between a large number of nonlinear layers in the feature mapping $\Phi_{\boldsymbol{\theta}}$, it is tremendously challenging to analyze the optimization of deep neural networks. To simplify the difficulty of deep neural network analysis, a series of recent works of theoretically studying $\mathcal{NC}$ phenomenon use a so-called *unconstrained feature model* (or *layer-peeled model* in [12]) which treats the last-layer features as *free* optimization variables $\boldsymbol{h} = \Phi(\boldsymbol{x}) \in \mathbb{R}^d$. The reason behind the *unconstrained feature model* is that modern highly overparameterized deep networks are able to approximate any continuous functions [32–35] and the characterization of $\mathcal{NC}$ are only related with the last layer features. We adopt the same approach and study the effects of different training losses on the last-layer representations of the network under the unconstrained feature model. For convenient, let us denote

$$\begin{aligned}
\boldsymbol{W} &:= \begin{bmatrix}\boldsymbol{w}^1 & \boldsymbol{w}^2 & \cdots & \boldsymbol{w}^K\end{bmatrix}^\top \in \mathbb{R}^{K\times d},\\
\boldsymbol{H} &:= \begin{bmatrix}\boldsymbol{H}_1 & \boldsymbol{H}_2 & \cdots & \boldsymbol{H}_n\end{bmatrix} \in \mathbb{R}^{d\times N}, \;\text{and}\\
\boldsymbol{Y} &:= \begin{bmatrix}\boldsymbol{Y}_1 & \boldsymbol{Y}_2 & \cdots & \boldsymbol{Y}_K\end{bmatrix} \in \mathbb{R}^{K\times N},
\end{aligned}$$

where $\boldsymbol{w}^k$ is the k-th row vector of $\boldsymbol{W}$, all the features in the k-th class are denoted as $\boldsymbol{H}_i := [\boldsymbol{h}_{1,i} \;\; \cdots \;\; \boldsymbol{h}_{K,i}] \in \mathbb{R}^{d\times K}$ and $\boldsymbol{h}_{k,i}$ is the feature of the i-th sample in the k-th class, and $\boldsymbol{Y}_k := [\boldsymbol{y}_k \;\; \cdots \;\; \boldsymbol{y}_k] \in \mathbb{R}^{K\times n}$ for all $k = 1, 2, \cdots, K$ and $i = 1, 2, \cdots, n$. Based on the unconstrained feature model, we consider a slight variant of (2), given by

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} f(\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}) := \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}\left(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b}, \boldsymbol{y}_k\right) + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2, \tag{7}$$

where $\lambda_{\boldsymbol{W}}, \lambda_{\boldsymbol{H}}, \lambda_{\boldsymbol{b}} > 0$ are the penalty parameters for $\boldsymbol{W}$, $\boldsymbol{H}$, and $\boldsymbol{b}$, respectively.

By viewing the last-layer feature $\boldsymbol{H}$ as a free optimization variable, the simplified objective function (7) consider the weight decay about $\boldsymbol{W}$ and $\boldsymbol{H}$, which is slightly different from practice that the weight decay is imposed on all the network parameters $\boldsymbol{\Theta}$ as shown in (2). Nonetheless, the underlying rationale is that the weight decay on $\boldsymbol{\Theta}$ *implicitly* penalizes the energy of the features (i.e., $\|\boldsymbol{H}\|_F$) [16].

As $\mathcal{NC}$ phenomena for the learned features and classifiers is first discovered for neural networks trained with the CE loss [5], the CE loss has been mostly studied through the above simplified unconstrained feature model [8,9,11,12,16] to understand the $\mathcal{NC}$ phenomena. The work [7,10,14, 22] also studied the MSE loss, but the analysis there shows the solutions of the learned features and classifiers depend crucially on the bias term, while for CE loss with or without the bias term have no effect on the learned features and classifiers under the unconstrained feature model. The other losses such as focal loss and label smoothing have been less studied, though they are widely employed in practice to obtain better performance. This will be the subject of next section.

3 Understanding Loss Functions Through Unconstrained Features Model

In this section, we study the effect of different loss functions through the unconstrained features model. We will first present a contrastive property for general loss function $\mathcal{L}_{\mathrm{GL}}$ in Definition 1. We will then study the global optimality conditions in terms of the learned features and classifiers as well as geometric properties for (7) with such a general loss function $\mathcal{L}_{\mathrm{GL}}$.

3.1 A Contrastive Property for the Loss Functions

In this paper, we aim to provide a unified analysis for different loss functions. Towards that goal, we first present some common properties behind the CE, FL and FL to motivate the discussion. Taking CE as an example, we can lower bound it by

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{z}, \boldsymbol{y}_k) \geq \log\left(1 + (K-1)\exp\left(\frac{\sum_{j\neq k}(z_j - z_k)}{K-1}\right)\right) = \phi_{\mathrm{CE}}\left(\sum_{j\neq k}(z_j - z_k)\right) \tag{8}$$

where $\phi_{\mathrm{CE}}(t) = \log\left(1 + (K-1)\exp\left(\frac{t}{K-1}\right)\right)$, and the inequality achieves equality when $z_j = z_{j'}$ for all $j, j' \neq k$. This requirement is reasonable because the commonly used losses treat all the outputs except for the k-th output $\boldsymbol{z}$ identically. Since ϕ_{CE} is an increasing function, minimizing the CE loss $\mathcal{L}_{\mathrm{CE}}(\boldsymbol{z}, \boldsymbol{y}_k)$ is equivalent to maximizing $(K-1)z_k - \sum_{j\neq K} z_j$, which contrasts the k-th output z_k simultaneously to all the other outputs z_j for all $j \neq k$. Thus, we call (8) as a *contrastive property*. Maximizing $(K-1)z_k - \sum_{j\neq K} z_j$ would lead to a positive (and relatively large) z_k and negative (and relatively small) z_j. In particular, within the unit sphere $\|\boldsymbol{z}\|_2 = 1$, $(K-1)z_k - \sum_{j\neq K} z_j$ achieves its maximizer when $z_k = \sqrt{\frac{K-1}{K}}$ and $z_j = -\sqrt{\frac{1}{K(K-1)}}$ for all $j \neq k$, which satisfies the requirement $z_j = z_{j'}$ for all $j, j' \neq k$. Thus, $z_k = \sqrt{\frac{K-1}{K}}$ and $z_j = -\sqrt{\frac{1}{K(K-1)}}$ is also the global minimizer for ϕ_{CE} within the unit sphere $\|\boldsymbol{z}\|_2 = 1$. As the global minimizer is unique for each class, it encourages intra-class compactness. On the other hand, the minimizers to different classes are maximally distant, promoting inter-class separability.

Motivated by the above discussion, we now introduce the following properties for a general loss function $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$.

Definition 1 (Contrastive property)**.** *We say a loss function $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ satisfies the contrastive property if there exists a function ϕ such that $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ can be lower bounded by*

$$\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k) \geq \phi\left(\sum_{j\neq k}(z_j - z_k)\right) \tag{9}$$

where the equality holds only when $z_j = z'_j$ for all $j, j' \neq k$. Moreover, $\phi(t)$ satisfies

$$t^* = \arg\min_t \phi(t) + c|t| \text{ is unique for any } c > 0, \text{ and } t^* \leq 0. \tag{10}$$

In the appendix, we show that CE, FL and LS all satisfy this property. The motivation for (9) follows from the above discussion. In particular, (9) achieves equality when all the outputs except for the k-th one are identical, which holds for common loss functions since those outputs are treated identically. In (10), c is a constant related with the weight decay penalty parameters. By (9), we can find the global minimizer for $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ by minimizing the right hand side since the equality in (9) is achievable. Thus, the requirement of a unique minimizer (10) ensures a unique minimizer for the regularized $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$. This condition can be easily satisfied. For example, $\phi_{\mathrm{CE}}(t)$ defined in (8) for the CE loss is an increasing and strictly convex function and thus has unique minimizer for $\phi_{\mathrm{CE}}(t) + c|t|$. Along the same line, we require a negative minimizer $t^\star$ to ensure that the minimizer for the regularized $\mathcal{L}_{\mathrm{GL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ has k-th entry being its largest entry, which is required to ensure correct prediction since the largest entry predicts the class membership. Therefore, such a condition is generally satisfied by the common losses. For example, $\phi_{\mathrm{CE}}(t)$ is an increasing function and thus must have a non-positive minimizer for $\phi_{\mathrm{CE}}(t) + c|t|$. Finally, we note that the MSE loss is not included since it has different form than others and thus the analysis will be different. But as mentioned above, the MSE loss has been studied in [7, 10, 14, 22].

3.2 Landscape Analysis for the Unconstrained Features Model

We now study the global optimality conditions in terms of the learned features and classifiers as well as geometric properties for the training problem (7) with the general loss function $\mathcal{L}_{\mathrm{GL}}$ satisfying the above contrastive property.

Theorem 1 (Global Optimality Condition)**.** *Assume that the number of classes K is smaller than the feature dimension d, i.e., $K < d$, and the dataset is balanced for each class, $n = n_1 = \cdots = n_K$. Then any global minimizer $(\boldsymbol{W}^\star, \boldsymbol{H}^\star, \boldsymbol{b}^\star)$ of f in (7) with a loss function $\mathcal{L}_{\mathrm{GL}}$ satisfying the contrastive property in Definition 1 has following properties:*

$$\|\boldsymbol{w}^\star\|_2 = \left\|\boldsymbol{w}^{\star 1}\right\|_2 = \left\|\boldsymbol{w}^{\star 2}\right\|_2 = \cdots = \left\|\boldsymbol{w}^{\star K}\right\|_2, \quad \text{and} \quad \boldsymbol{b}^\star = b^\star \mathbf{1},$$

$$\boldsymbol{h}^\star_{k,i} = \sqrt{\frac{\lambda_{\boldsymbol{W}}}{\lambda_{\boldsymbol{H}} n}} \boldsymbol{w}^{\star k}, \quad \forall\, k \in [K],\ i \in [n], \quad \text{and} \quad \overline{\boldsymbol{h}}^\star_i := \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{h}^\star_{j,i} = \mathbf{0}, \quad \forall\, i \in [n],$$

where either $b^\star = 0$ or $\lambda_{\boldsymbol{b}} = 0$, and the matrix $\boldsymbol{W}^{\star\top}$ is in the form of K-simplex ETF structure (see appendix for the formal definition) in the sense that

$$\boldsymbol{W}^{\star\top} \boldsymbol{W}^\star = \|\boldsymbol{w}^\star\|_2^2 \frac{K}{K-1} \left(\boldsymbol{I}_K - \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^\top \right).$$

Its proof is given in Appendix C. At a high level, we lower bound the general loss function based on the contrastive property (9) and then check the equality conditions hold for the lower bounds. While similar strategy has been used for CE loss [12, 13, 16], our proof is different from previous work in terms of dealing with the nuclear norm and checking the structures of the minimizer per sample and class, enabling the global optimality analysis for general loss functions. Theorem 1 implies that for all the loss functions (e.g., CE, LS, and FL) satisfying the contrastive property, they share similar global solutions with $\mathcal{NC}$ property in the learned features and classifiers.

While Theorem 1 shows that $\mathcal{NC}$ features and classifiers are the only global minimizers to (7), it is not obvious whether local search algorithms (such as gradient descent) can efficiently find these benign global solutions. The reason is that the training problem (7) is nonconvex due to the bilinear form between $\boldsymbol{W}$ and $\boldsymbol{H}$. To address this challenge, we use the recent advances on the geometric analysis for nonconvex optimization [23–25, 36] to guarantee that the global solutions of (7) can be efficiently achieved by iterative algorithms. Towards that goal, we first present the following general results concerning the global landscape for (7).

Theorem 2 (Benign Landscape)**.** *Assume that the feature dimension d is larger than the number of classes K, i.e., $d > K$. Also assume $\mathcal{L}(\boldsymbol{z}, \boldsymbol{y})$ is a convex function in terms of $\boldsymbol{z}$. Then the objective function f in (7) is a strict saddle function with no spurious local minimum. That is, any of its critical point is either a global minimizer, or it is a strict saddle point whose Hessian has a strictly negative eigenvalue.*

This result is similar to [16, Theorem 3.2] which studies the particular CE loss. Though the result in [16] is about the CE loss, we checked its proof and it only uses convexity and smoothness and thus the result can be applied more generally for any smooth convex loss function $\mathcal{L}(\cdot, \boldsymbol{y})$. So we omit the proof of Theorem 2. We note that the geometric analysis is also closed related to nonconvex low-rank matrix problems [37–43] with the Burer-Moneirto factorization approach [44] if one views $\boldsymbol{W}$ and $\boldsymbol{H}$ as two factors of a matrix $\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{H}$. We refer to [16] for more discussions about the connections and differences.

We now exploit Theorem 2 for the label smoothing and focal loss. In the supplementary material, we show that LS is a convex function. Thus, the following result establishes global optimization landscape for the training problem (7) with such a loss.

Corollary 1 (Benign Landscape with LS)**.** *Assume that the feature dimension d is larger than the number of classes K, i.e., $d > K$. Then the objective function f in (7) with LS loss $\mathcal{L}_{\mathrm{LS}}$ is a strict saddle function with no spurious local minimum.*

Unlike LS, focal loss $\mathcal{L}_{\mathrm{FL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ is convex only in a local region rather than the entire space. For example, we can show that $\mathcal{L}_{\mathrm{FL}}(\boldsymbol{z}, \boldsymbol{y}_k)$ is convex within the region $\Omega = \{\boldsymbol{z} \in \mathbb{R}^K : \exp(z_k)/\sum_{j=1}^K \exp(z_j) \geq 0.21\}$. The set Ω contains a relative large region including the global minimizer which has the value $\frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)}$ approaching 1. Thus, we obtain a benign local landscape for the training problem (7) with FL.

Corollary 2 (Benign Landscape with FL)**.** *Assume that the feature dimension d is larger than the number of classes K, i.e., $d > K$. Then the objective function f in (7) with FL loss $\mathcal{L}_{\mathrm{FL}}$ has a benign local landscape: f is a strict saddle function with no spurious local minimum within the region $\{(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}) : \boldsymbol{W}\boldsymbol{h}_{k,i} + \boldsymbol{b} \in \Omega,\ 1 \leq k \leq K, 1 \leq i \leq n\}$.*

While Corollary 2 only provides a local benign landscape for the FL, we observe from experiments that gradient descent with random initialization always converges to a global solution with $\mathcal{NC}$ properties for (7). So we expect the training problem in (7) with FL loss has benign landscape in a much larger region. One direction is to show $\mathcal{L}_{\mathrm{FL}}(\cdot, \boldsymbol{y}_k)$ is locally convex in a much larger region Ω, but we leave the thorough investigation to future work. Noting that the CE and MSE losses are also convex, these results imply that (stochastic) gradient descent with random initialization [23, 36] almost surely finds the global solutions of the training problem in (7) with different training losses. This together with Theorem 1 implies that for different losses, gradient descent will always learn similar features and classifiers—those that exhibit the $\mathcal{NC}$ phenomenon.

4 Experiments

We conduct experiments with practical network architectures on standard image classification datasets to study the effect of different loss functions. First, Section 4.1 provides results to show that the $\mathcal{NC}$ phenomena are not restricted to networks trained via the CE and MSE losses. Rather, there is a family of loss functions, and for the purpose of illustration we pick FL and LS as two prominent special cases, that exhibit the same $\mathcal{NC}$ phenomena. Such results verify our theoretical results in Section 3. To demonstrate the implication of $\mathcal{NC}$ for test performance, we present experimental results in Section 4.2 with a varying number of training iterations and a varying width of networks, showing that *all* losses with $\mathcal{NC}$ global optimality have similar performance on the test dataset when the network is sufficiently large and trained long enough.

Before presenting the experiment results, we first introduce our experimental setup, including datasets, network architectures, training procedure, and metrics for measuring $\mathcal{NC}$.

Setup of Loss Function, Network Architecture, Dataset, and Training We focus on the CE, FL, LS and MSE loss functions for which we use $\gamma = 3$ for FL, $\alpha = 0.1$ for LS, and $\kappa = 1$ and $\beta = 15$ for MSE, except otherwise specificed. We train a WideResNet50 network [45] on CIFAR10 and CIFAR100 datasets [46] and a WideResNet18 network on miniImageNet [47] with various widths and number of iterations for image classification using these four different losses.[3] To examine the

[3] Similar results are expected on other architectures and dataset as $\mathcal{NC}$ is observed across a range of architectures and dataset in [5].

effect of model size, we experiment with four versions of WideResNet, denoted as WideResNet-X, where $X \in \{0.25, 0.5, 1, 2\}$ is a multiplier on the width of its corresponding standard WideResNet. Due to the page limit, we put all results on CIFAR100 and miniImageNet in the Appendix. We use standard preprocessing such that images are normalized (channel-wise) by their mean and standard deviation, as well as standard data augmentation. For optimization, we use SGD with momentum 0.9 and an initial learning rate 0.1 decayed by a factor of 0.1 at $\frac{3}{7}$ and $\frac{5}{7}$ of the total number of iterations. Following [28], the norm of gradient is clipped at 2 which can improve performance for all losses. For CIFAR10 and miniImageNet, the weight decay is set to 5×10^4 for all configurations with all losses. For CIFAR100, the weight decay is fine-tuned to achieve best accuracy for every configuration and loss.

Three $\mathcal{NC}$ Metrics $\mathcal{NC}_1$-$\mathcal{NC}_3$ during Network Training We use the same three metrics $\mathcal{NC}_1$-$\mathcal{NC}_3$ for the last-layer features and classifier as in [5,16,22] to measure the first three $\mathcal{NC}$ properties in Section 1. Before we describe these three metrics, let us denote the global mean $\boldsymbol{h}_G$ and k-th class mean $\overline{\boldsymbol{h}}_k$ of last-layer features $\{\boldsymbol{h}_{k,i}\}$ as

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

Within-class variability collapse is measured by $\mathcal{NC}_1$ which depicts the relative magnitude of the within-class covariance $\boldsymbol{\Sigma}_W = \frac{1}{nK}\sum_{k=1}^{K}\sum_{i=1}^{n}\left(\boldsymbol{h}_{k,i}-\overline{\boldsymbol{h}}_k\right)\left(\boldsymbol{h}_{k,i}-\overline{\boldsymbol{h}}_k\right)^\top \in \mathbb{R}^{d\times d}$ w.r.t. the between-class covariance $\boldsymbol{\Sigma}_B = \frac{1}{K}\sum_{k=1}^{K}\left(\overline{\boldsymbol{h}}_k-\boldsymbol{h}_G\right)\left(\overline{\boldsymbol{h}}_k-\boldsymbol{h}_G\right)^\top \in \mathbb{R}^{d\times d}$ of the last-layer features as following:

$$\mathcal{NC}_1 = \frac{1}{K}\operatorname{trace}\left(\boldsymbol{\Sigma}_W\boldsymbol{\Sigma}_B^{\dagger}\right),$$

where $\boldsymbol{\Sigma}_B^{\dagger}$ is the pseudo inverse of $\boldsymbol{\Sigma}_B$.

Convergence to simplex ETF is measured by $\mathcal{NC}_2$ which reflects the ℓ_2 distance between the normalized simplex ETF and the normalized $\boldsymbol{W}\boldsymbol{W}^\top$ as following:

$$\mathcal{NC}_2 \;:=\; \left\| \frac{\boldsymbol{W}\boldsymbol{W}^\top}{\|\boldsymbol{W}\boldsymbol{W}^\top\|_F} \;-\; \frac{1}{\sqrt{K-1}}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)\right\|_F,$$

where $\boldsymbol{W} \in \mathbb{R}^{K\times d}$ is the weight matrix of learned classifier.

Convergence to self-duality is measured by $\mathcal{NC}_3$ which calculates the ℓ_2 distance between the normalized simplex ETF and the normalized $\boldsymbol{W}\overline{\boldsymbol{H}}$ as following:

$$\mathcal{NC}_3 \;:=\; \left\| \frac{\boldsymbol{W}\overline{\boldsymbol{H}}}{\|\boldsymbol{W}\overline{\boldsymbol{H}}\|_F} \;-\; \frac{1}{\sqrt{K-1}}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)\right\|_F.$$

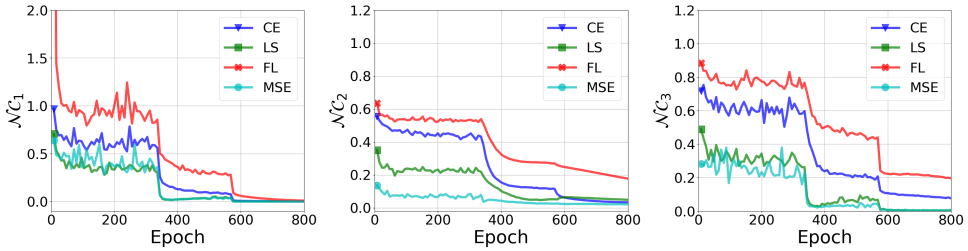
where $\overline{\boldsymbol{H}} = \begin{bmatrix}\overline{\boldsymbol{h}}_1 - \boldsymbol{h}_G & \cdots & \overline{\boldsymbol{h}}_K - \boldsymbol{h}_G\end{bmatrix} \in \mathbb{R}^{d\times K}$ is the centered class-mean matrix.

4.1 Prevalence of $\mathcal{NC}$ Across Varying Training Losses

We show that all loss functions lead to $\mathcal{NC}$ solutions during the terminal phase of training. The results on CIFAR10 using WideResNet50-2 and different loss functions is provided in Figure 1. We consistently observe that all three $\mathcal{NC}$ metrics across different losses converge to a small value as training progresses. This supports our theoretical results in Section 3 that the last-layer features learned under different losses are always maximally linearly separable and perfectly aligned with the linear classifier, and the features and the weight of linear classifier learned by different losses are almost equivalent up to a rotation and a scale of the feature space. The evolution of three $\mathcal{NC}$ metrics across different losses on CIFAR100 is in Appendix A.2.

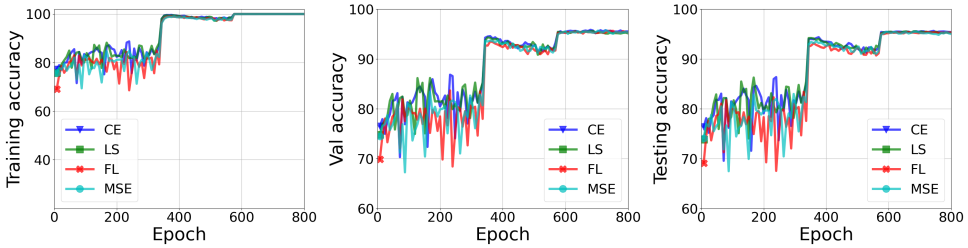
4.2 All Losses Lead to Largely Identical Performance

We show that all loss functions have largely identical performance once the training procedure converges to the $\mathcal{NC}$ global optimality. In Figure 2, we plot the evolution of the training accuracy,



(a) $\mathcal{NC}_1$ (CIFAR10) (b) $\mathcal{NC}_2$ (CIFAR10) (c) $\mathcal{NC}_3$ (CIFAR10)

Figure 1: **The evolution of $\mathcal{NC}$ metrics across different loss functions.** We train the WideResNet50-2 on CIFAR10 dataset for 800 epochs using different loss functions. From left to right: $\mathcal{NC}_1$ (variability collapse), $\mathcal{NC}_2$ (convergence to simplex ETF) and $\mathcal{NC}_3$ (convergence to self-duality).



(a) Train Acc (CIFAR10) (b) Val Acc (CIFAR10) (c) Test Acc (CIFAR10)

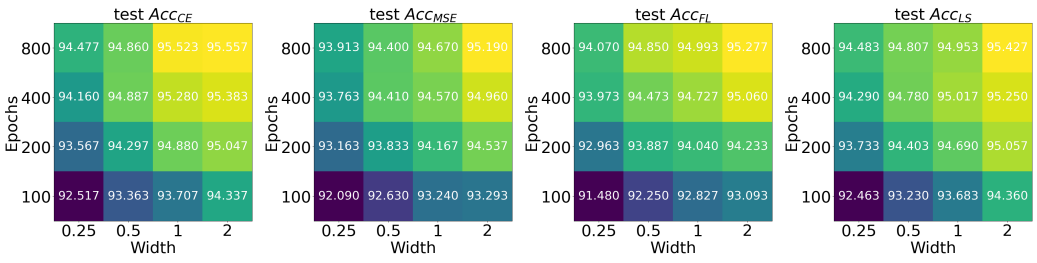
Figure 2: **The evolution of performance across different loss functions.** We train the WideResNet50-2 on CIFAR10 dataset for 800 epochs using different loss functions. From left to right: training accuracy, validation accuracy and test accuracy.

validation accuracy and test accuracy as training progresses, where all losses are optimized on the same WideResNet50-2 architecture and CIFAR10 for 800 epochs. To reduce the randomness, we average the results from 3 different random seeds per width-iteration configuration, and the test accuracy is reported based on the model with best accuracy on validation set, where we organize the validation set by holding out 10 percent data from the training set. The results consistently show that for all cases the training accuracy converges to one hundred percent (reaching to terminal phase), and the validation accuracy and test accuracy are largely the same, as long as the network is trained longer enough and converges to the $\mathcal{NC}$ global solution.

While previous work advocates the advantage of some losses over other others, our experiments show that when conditions between dataset and model allow for SGD to find an $\mathcal{NC}$ solution, all losses we tested produced indistinguishable results. In Figure 3, we plot the average test accuracy of different losses under different pairs of width and iterations. We consistently observe three phenomenon. First, with a fixed number of iterations, increasing the width of network improves the test accuracy for all losses. This is because the wider networks (more over-parameterized) are more powerful to fit the underlying mapping from input data to the targets. Second, with a fixed width of network, increasing the number of iterations improves the test accuracy for all losses. This is because the longer optimization leads the last-layer features and the linear classifer closer to the $\mathcal{NC}$ global solutions. Finally, while there are some unignorable difference between different losses in some width-iteration configurations, the results consistently show that all losses lead to largely identical performance when the network is sufficiently large and trained long enough to achieve a global $\mathcal{NC}$ solution (e.g. width=2 and epochs=800).

5 Conclusion

In this work we provided a theoretical study to extend the scope of $\mathcal{NC}$, a curious phenomenon associated with last-layer features and classifier weight of a classification network, from networks trained with particular losses (i.e., CE and MSE) to those trained via a broad family of loss functions including the popular LS and FL as special cases. Our theory not only establishes $\mathcal{NC}$ as the only global solutions, but also shows a benign optimization landscape that explains why $\mathcal{NC}$ solutions



(a) Test (CE) (b) Test (MSE) (c) Test (FL) (d) Test (LS)

Figure 3: **Illustration of test accuracy across different iterations-width settings.** The figure depicts the test accuracy of various iteration-width configurations for different loss functions on CIFAR10.

are easy to obtain in practice. Such results readily suggest that all relevant losses (i.e., CE, MSE, LS, and FL) produce entirely equivalent features on the training data. Although $\mathcal{NC}$ is an optimization phenomenon pertaining to training data only, we found through experiments that all relevant losses (i.e., CE, MSE, LS, and FL) lead to very similar test performance as well. Such a result may come as a surprise to the common belief that some losses are intrinsically better than the others, and clarify some mystery on how different losses affect the performance.

The family of loss functions considered in this paper by no means is inclusive of all possible loss functions that lead to $\mathcal{NC}$. There are many other popular loss functions, such as center loss [48], large-margin softmax (L-Softmax) loss [49] and many of its variants [50–52], which are all designed with the intuition of encouraging intra-class compactness and inter-class separability between learned features. In addition, many generalized versions of the cross-entropy loss such as those for robust learning under label noise [53–55] and long-tail distribution [56, 57] may have similar property as the vanilla cross-entropy loss. We conjecture that many of them provably produce $\mathcal{NC}$ solutions under unconstrained feature models, while leave a formal justification to future work. Beyond losses for classification task, $\mathcal{NC}$ may also arise with popular losses used in metric learning [58, 59] evidenced by recent study [60]. This means that the observations from this paper, namely all losses lead to largely the same test performance, may apply for all such losses as well.

Loss functions that do not lead to $\mathcal{NC}$. While the study in this paper covers many of the most commonly used loss functions for classification tasks, we note that there are alternative choices in the literature which do not induce $\mathcal{NC}$ features. Many of such losses such as [60–63] are particularly designed to discourage variability collapse and learn diverse features, which are shown to benefit model transferability [64] and robustness.

Acknowledgment

JZ and ZZ acknowledge support from NSF grants CCF-2240708 and CCF-2241298. XL and QQ acknowledge support from U-M START & PODS grants, NSF CAREER CCF 2143904, NSF CCF 2212066, NSF CCF 2212326, and ONR N00014-22-1-2529 grants. SL acknowledges support from NSF NRT-HDR Award 1922658 and Alzheimer's Association grant AARG- NTF-21-848627.

References

[1] 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*, pages 2818–2826, 2016.

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

[3] Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

[4] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25:49–59, 2016.

[5] 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.

[6] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020.

[7] 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.

[8] Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *arXiv preprint arXiv:2012.08465*, 2020.

[9] E Weinan and Stephan Wojtowytsch. On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers. *arXiv preprint arXiv:2012.05420*, 2020.

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

[11] Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pages 3821–3830. PMLR, 2021.

[12] Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Layer-peeled model: Toward understanding well-trained deep neural networks. *arXiv e-prints*, pages arXiv–2101, 2021.

[13] Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. An unconstrained layer-peeled perspective on neural collapse. *arXiv preprint arXiv:2110.02796*, 2021.

[14] Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. *arXiv preprint arXiv:2202.08087*, 2022.

[15] Tolga Ergen and Mert Pilanci. Revealing the structure of deep neural networks via convex duality. In *International Conference on Machine Learning*, pages 3004–3014. PMLR, 2021.

[16] 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, 2021.

[17] Akshay Rangamani and Andrzej Banburski-Fahey. Neural collapse in deep homogeneous classifiers and the role of weight decay. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4243–4247. IEEE, 2022.

[18] 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.

[19] Tomaso Poggio and Qianli Liao. Implicit dynamic regularization in deep networks. Technical report, Center for Brains, Minds and Machines (CBMM), 2020.

[20] Akshay Rangamani, Mengjia Xu, Andrzej Banburski, Qianli Liao, and Tomaso Poggio. Dynamics and neural collapse in deep classifiers trained with the square loss. *Technical report, Center for Brains, Minds and Machines (CBMM)*, 2021.

[21] Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Neural collapse in deep homogeneous claaifiers and the role of weight decay. In *International Conference on Machine Learning*, pages 3821–3830. PMLR, 2021.

[22] 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. *arXiv preprint arXiv:2203.01238*, 2022.

[23] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*, pages 797–842, 2015.

[24] Ju Sun, Qing Qu, and John Wright. When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*, 2015.

[25] Yuqian Zhang, Qing Qu, and John Wright. From symmetry to geometry: Tractable nonconvex problems. *arXiv preprint arXiv:2007.06753*, 2020.

[26] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[27] Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384*, 2022.

[28] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

[29] Leslie N Smith. Cyclical focal loss. *arXiv preprint arXiv:2202.08978*, 2022.

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

[31] Blair Chen, Liu Ziyin, Zihao Wang, and Paul Pu Liang. An investigation of how label smoothing affects generalization. *arXiv preprint arXiv:2010.12648*, 2020.

[32] G Cybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

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

[34] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: a view from the width. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6232–6240, 2017.

[35] 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.

[36] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on learning theory*, pages 1246–1257. PMLR, 2016.

[37] Benjamin D Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

[38] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *arXiv preprint arXiv:1605.07272*, 2016.

[39] Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3880–3888, 2016.

[40] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[41] Qiuwei Li, Zhihui Zhu, and Gongguo Tang. The non-convex geometry of low-rank matrix optimization. *Information and Inference: A Journal of the IMA*, 8(1):51–96, 2019.

[42] Xingguo Li, Junwei Lu, Raman Arora, Jarvis Haupt, Han Liu, Zhaoran Wang, and Tuo Zhao. Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization. *IEEE Transactions on Information Theory*, 65(6):3489–3514, 2019.

[43] Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

[44] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[45] 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*, pages 770–778, 2016.

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

[47] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[48] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[49] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[50] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[51] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[52] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[53] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[54] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[55] Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *International Conference on Learning Representations*, 2021.

[56] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

[57] Wittawat Jitkrittum, Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Elm: Embedding and logit margins for long-tail learning. *arXiv preprint arXiv:2204.13208*, 2022.

[58] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

[59] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[60] Elad Levi, Tete Xiao, Xiaolong Wang, and Trevor Darrell. Reducing class collapse in metric learning with easy positive sampling. *arXiv preprint arXiv:2006.05162*, 2020.

[61] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE international conference on computer vision*, pages 4595–4603, 2017.

[62] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019.

[63] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *arXiv preprint arXiv:2006.08558*, 2020.

[64] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems*, 34, 2021.

[65] Jan JM Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36(2):177–195, 1980.

[66] N Jakovčević Stor, I Slapničar, and Jesse L Barlow. Forward stable eigenvalue decomposition of rank-one modifications of diagonal matrices. *Linear Algebra and its Applications*, 487:301–315, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In the abstract, introduction, main results, and conclusion, we explicitly stat that our results are about unconstrained feature model.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This paper mainly focuses on understanding the neural collapse phenomena observed in practical neural networks. Based on this understanding, we propose to fix the last layer classifier as a Simplex ETF. So no potential negative societal impact is expected of this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We explicitly mention the assumptions in Theorem 1 and Theorem 2.
   (b) Did you include complete proofs of all theoretical results? [Yes] We include all the proofs in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 4
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All the results are obtained by averaging the resluts from 3 different random seeds. See Section 4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite them in Appendix A.
   (b) Did you mention the license of the assets? [Yes] This is mentioned in Appendix A.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] This is discussed in Appendix A.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The CIFAR datasets do not contain personally identifiable information or offensive content
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Our research does not involve with participants and screenshots
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Our research does not include any potential participant risks, with links to Institutional Review Board (IRB) approvals
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Our work does not require participants.