Temperature in Cosine-based Softmax Loss

Takumi Kobayashi[†‡]
[†]National Institute of Advanced Industrial Science and Technology, Japan
[‡]University of Tsukuba, Japan
takumi.kobayashi@aist.go.jp

Abstract

While deep models are effectively trained based on a softmax cross-entropy loss, a cosine-based softmax loss also works for producing favorable feature embedding. In the cosine-based softmax, temperature plays a crucial role in properly scaling the logits of cosine similarities, though being manually tuned in ad-hoc ways as there is less prior knowledge about the temperature. In this paper, we address the challenging problem to adaptively estimate the temperature of cosine-based softmax in the framework of supervised image classification. By analyzing the cosine-based softmax representation from a geometrical viewpoint regarding features and classifiers, we construct a criterion in a least-square fashion which enables us to optimize the temperature at each sample via simple greedy search. Besides, our thorough analysis about temperature clarifies that feature embedding by the cosine-based softmax loss is endowed with diverse characteristics which are controllable by the temperature in an explainable way. The experimental results demonstrate that our optimized temperature contributes to determine a feasible range of temperature to control the feature characteristics and produces favorable performance on various image classification tasks.

1. Introduction

Deep neural networks are applied in various computer vision fields with great success [11]. While the network architecture has been steadily advancing, those models are effectively trained based on a softmax cross-entropy loss in most cases. In the softmax loss, relationships between input features and classifiers are exploited by means of inner-product, which is fundamentally characterized by the angle between those vectors, i.e., *cosine* similarity; feature embedding by the softmax loss exhibits *angular* discriminativity [25, 40]. Thus, a *cosine*-based softmax loss is an important variant of the softmax loss, paying much attention to the angle through L_2-normalization both of features and classifiers in the inner-product form.

The cosine-based softmax loss, dubbed as cos-softmax loss, is applied to lean feature embeddings [42, 45] such as for face images [9, 25, 37]. Lots of face classes (individuals) are effectively embedded on a hyper-sphere through the cos-softmax loss, frequently equipped with large-margin regularization, to attain discriminative and generalization power applicable to novel face images. For feature embedding, the cos-softmax loss is also applied in self-supervised learning, especially contrastive learning [5, 6, 12]; unlabeled samples are compared by means of cosine similarity in a way of instance discrimination. The feature representation learned by the cos-softmax loss is recently shown to exhibit favorable discriminativity for detecting out-of-distribution (OOD) samples based on feature norms [31].

While a cos-softmax loss contributes to effective feature embedding, it contains a critical issue regarding *temperature*. In the literature of deep learning using a standard softmax loss, softmax temperature is effectively utilized in knowledge distillation [13] and is also analyzed from a viewpoint of training dynamics [2]. For the cos-softmax loss, the temperature plays a more important role in scaling the cosine similarity bounded in $[-1, 1]$ for building meaningful loss function; either too low or too high temperature deteriorates the cos-softmax loss. Thus, in most cases, the temperature of cos-softmax is *manually* tuned so as to produce favorable performance in an empirical manner. Thus, it is a challenging problem to automatically determine proper temperature as it could be variable according to classification tasks and/or feature embedding space.

In this paper, we explore an approach to adaptively estimate the temperature of cos-softmax in a framework of supervised classification. For establishing criteria to optimize the temperature, we analyze the cos-softmax from two viewpoints of probabilistic and geometrical formulation. While the probabilistic aspect of cos-softmax is derived from von Mises-Fisher distribution [28] on a hyper-sphere, we also propose a geometrical viewpoint to connect the cos-softmax with projection onto a convex hull spanned by the classifier vectors, inspired by sparse-representation classification [41]. The method enables us to optimize tem-

perature by means of simple greedy search at each sample without introducing extra training procedure other than back-propagation for a backbone model. Besides, on the basis of the optimized temperature, we thoroughly analyze effect of cos-softmax temperature on feature embedding from various perspectives including generalization. The analysis clarifies that the embedded features exhibit diverse characteristics which are controllable by the temperature in an explainable/interpretable way.

Our contributions are summarized as follows:

- We propose a *least-square* approach to adaptively estimate temperature in a cos-softmax loss by exploiting geometrical relationships among features and classifiers in the supervised learning framework.
- Through the analysis regarding cos-softmax temperature, we clarify the diverse characteristics of feature embedding learned by cos-softmax losses with various temperatures. The optimized temperature by our method works as a lower bound of feasible temperature range to control the feature characteristics in an explainable way.
- In the experiments, we demonstrate that the method is applicable to provide effective temperature across diverse numbers of classes $C = 10 \sim 93431$ and on various tasks such as OOD detection and imbalanced classification.

1.1. Related works

Cosine-based softmax loss. While a standard softmax loss is widely applied to train deep models, a cos-softmax loss is also useful in the literature of metric learning to exploit intrinsic *angular* characteristics of feature representation. In the feature embedding, supervised learning equipped with classifiers is effectively applied to build discriminative feature representation [45] such as in a scenario of face recognition [9, 25, 36, 37]. The metric learning can also be formulated by exploiting relationships among samples and cos-softmax is applicable to pair-wise comparison of samples [42]. In recent years, the cos-softmax loss based on pair-wise samples is employed in contrastive learning [5, 6, 12] to learn feature embedding in a self-supervised way. On the other hand, in [31], L_2-norm of the feature representation learned by a cos-softmax loss is shown to exhibit favorable performance for detecting out-of-distribution (OOD) samples. In this work, we focus on a cos-softmax loss in the framework of supervised classification, which is versatile across various tasks, to analyze characteristics of feature embedding through a lens of softmax temperature.

Sotmax temperature. There are some works to analyze the softmax temperature. In a standard softmax loss, a role of temperature is analyzed by [2] from a viewpoint of training dynamics, presenting performance improvement in some cases. While the temperature is usually set to 1 in a standard softmax loss, it is crucial to tune the temperature in the cos-softmax loss since meaningful loss functions are build by properly scaling the logits (cosine similarities) bounded in $[-1, 1]$; in most cases, the temperature is manually tuned maybe in a trial-and-error manner. In the contrastive learning framework, effect of the cos-softmax temperature on encoding pair-wise sample relationships is analyzed in [22, 35] such as through a lens of uniformity [38] which is a specific viewpoint to self-supervised learning; though, the temperature needs to be manually tuned. In visual-textural contrastive learning, a trainable temperature is analyzed from a viewpoint of equilibrium in the loss [33]. In contrast to those works of contrastive learning, we analyze cos-softmax temperature in the supervised classification scenario, and formulate approaches to adaptively estimate the temperature.

The works [1, 17] provide approaches to calibrate logits by tuning softmax temperature in a *post-hoc* manner *after training* to enhance confidence of classification. In contrast, our method optimizes a temperature in an *in-vivo* manner *during* training toward effective model learning. As the confidence is slightly related to our analysis in Section 3.2.1, our method would have potential synergy with the post-hoc calibration, though it is beyond our scope in this paper.

2. Method

This paper focuses on a softmax cross-entropy loss based on *cosine* similarities between a feature vector and classifier weight vectors. A neural network ϕ_Θ equipped with parameters Θ extracts a d-dimensional feature vector $\boldsymbol{x} = \phi_\Theta(\mathcal{I}) \in \mathbb{R}^d$ from an input image $\mathcal{I}$. For image classification, the feature vector $\boldsymbol{x}$ is classified by using linear classifier weights $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C] \in \mathbb{R}^{d\times C}$ into C-dimensional logits $\boldsymbol{z} \in \mathbb{R}^C$, which are responsible for respective C classes, in a form of cosine similarity as

$$z_c = \tilde{\boldsymbol{x}}^\top \tilde{\boldsymbol{w}}_c, \text{ where } \tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} \text{ and } \tilde{\boldsymbol{w}}_c = \frac{\boldsymbol{w}_c}{\|\boldsymbol{w}_c\|_2}. \quad (1)$$

To build a loss function for training both Θ and $\boldsymbol{W}$, softmax cross-entropy is applied to the logits $\boldsymbol{z}$ by

$$\ell(\boldsymbol{z}, y) = -\log \frac{\exp(z_y/\tau)}{\sum_{c=1}^C \exp(z_c/\tau)} = -\log \frac{\exp(\kappa z_y)}{\sum_{c=1}^C \exp(\kappa z_c)}, \quad (2)$$

where y indicates a class label assigned to the input $\mathcal{I}$ and τ is a softmax temperature parameter; for ease of discussion, we use its reciprocal $\kappa = 1/\tau \in \mathbb{R}_+$ in this paper. The temperature affects the cos-softmax loss (2) [22, 35], thus requiring careful tuning for effective training; it is usually determined in a *manually* ad-hoc way [25, 37, 45] since an optimal temperature would be dependent on tasks and feature distributions, characteristics of which are hard to know in advance. Thus, we tackle the challenging problem to automatically determine temperature, i.e., κ in (2), by considering the two approaches; probabilistic ane geometric ones.

2.1. Probabilistic approach

The parameter κ is connected to bandwidth of von Mises-Fisher (vMF) distribution [28], a probability model on a hyper-sphere. We leverage kernel density estimation [3] to probabilistically model a feature $\tilde{\boldsymbol{x}}$ on the basis of classifier weights $\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C$ as follows;

$$p(\boldsymbol{x};\kappa) = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{Z_{\kappa,d}}\exp\left(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_c\right), \tag{3}$$

where $Z_{\kappa,d} = \frac{(2\pi)^{d/2}I_{d/2-1}(\kappa)}{\kappa^{d/2-1}}$ is a normalization constant involving bandwidth κ, dimensionality d and a modified Bessel function $I_{d/2-1}$ [28].

2.1.1. Rule of thumb

Through a theoretical approximation, the *rule-of-thumb* [3] in the vMF-KDE framework provides bandwidth of

$$\kappa^* = \frac{\bar{r}(d-\bar{r}^2)}{1-\bar{r}^2} \text{ where } \bar{r} = \left\|\frac{1}{C}\textstyle\sum_c\tilde{\boldsymbol{w}}_c\right\|_2. \tag{4}$$

As it assumes distribution of directional *samples*, there may be a gap between the characteristics of samples and classifiers $\tilde{\boldsymbol{w}}$ which are trainable to distinguish features. Besides, (4) is dependent on dimensionality d, possibly increasing κ for deep models that produce high-dimensional features.

2.1.2. Maximum likelihood

Based on the KDE framework (3), we can directly estimate κ by maximizing the log-likelihood $\log p(\tilde{\boldsymbol{x}};\kappa)$. It, however, is known that the normalization constant $Z_{\kappa,d}$ is hard to deal with in such a high dimensional space. Thus, we approximate it by the upper bound $\bar{Z}_{\kappa,d}$ [14] to formulate the following optimization problem;

$$\mathcal{E}_\kappa^{ML} = -\log\frac{1}{C}\sum_{c}^{C}\frac{1}{\bar{Z}_{\kappa,d}}\exp\left(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_c\right) \tag{5}$$

$$\kappa^* = \arg\min_\kappa \mathcal{E}_\kappa^{ML}. \tag{6}$$

The high dimensionality d in deep models hinders computation of the normalization constant, though it is slightly mitigated by the approximated $\bar{Z}_{\kappa,d}$. We will empirically evaluate this maximum-likelihood (ML) approach in Section 3.1.

2.2. Geometric approach

As discussed above, it might be less feasible to model feature vectors of high dimensionality in a probabilistic manner. To cope with the issue, we resort to geometric representation of feature $\tilde{\boldsymbol{x}}$ by using classifier weights $\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C$ in a softmax-based framework.

Similarly to sparse-representation classification [41], we can describe $\tilde{\boldsymbol{x}}$ on the basis of classifiers $\tilde{\boldsymbol{w}}_c$ as

$$\tilde{\boldsymbol{x}} \approx \sum_{c=1}^{C}\alpha_c\tilde{\boldsymbol{w}}_c,\ s.t.\ \boldsymbol{\alpha}\in\Omega\triangleq\{\boldsymbol{\alpha}\,|\textstyle\sum_c\alpha_c=1,\ \alpha_c\geq 0\,\forall c\}, \tag{7}$$

which leads to an optimization w.r.t the coefficients $\boldsymbol{\alpha}$ as

$$\min_{\boldsymbol{\alpha}\in\Omega}\frac{1}{2}\|\tilde{\boldsymbol{x}}-\textstyle\sum_c\alpha_c\tilde{\boldsymbol{w}}_c\|_2^2 \Rightarrow \min_{\boldsymbol{\alpha}\in\Omega}\boldsymbol{\alpha}^\top\tilde{\boldsymbol{W}}^\top\tilde{\boldsymbol{W}}\boldsymbol{\alpha}-\textstyle\sum_c\alpha_c z_c. \tag{8}$$

Geometrically speaking, it computes the projection of $\tilde{\boldsymbol{x}}$ onto a convex hull spanned by $\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C$ which is connected to a classifier subspace [19].

On the other hand, the softmax representation (2) can be derived from the optimization problem of

$$\left\{\frac{\exp(\kappa z_c)}{\sum_k\exp(\kappa z_k)}\right\}_{c=1}^{C} = \arg\min_{\boldsymbol{\alpha}\in\Omega}\frac{1}{\kappa}\textstyle\sum_c\alpha_c\log\alpha_c-\sum_c\alpha_c z_c, \tag{9}$$

the detail of which is shown in a supplementary material. The formulation (9) is analogous to (8); they maximize correlation to $\boldsymbol{z}$ at the second term while minimizing the regularization regarding $\boldsymbol{\alpha}$ at the first term. Thus, the optimizer in (9), i.e., softmax, is closely related to the optimizer in (8).

Based on the above analysis, we optimize κ by minimizing the projection error (8) in a least-square manner. To that end, the softmax substitutes for the coefficients as $\alpha_c = \frac{\exp(\kappa z_c)}{\sum_k\exp(\kappa z_k)}$ in the least-square (LS) formulation (8) to provide the following optimization problem;

$$\mathcal{E}_\kappa^{LS} = \frac{1}{2}\left\|\tilde{\boldsymbol{x}}-\sum_{c=1}^{C}\frac{\exp(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_c)}{\sum_k\exp(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_k)}\tilde{\boldsymbol{w}}_c\right\|_2^2, \tag{10}$$

$$\kappa^* = \arg\min_\kappa \mathcal{E}_\kappa^{LS}. \tag{11}$$

2.2.1. Discussion

Back-propagation. A convex combination of classifier weights $\tilde{\boldsymbol{w}}_c$ with softmax coefficients, fundamental representation in the LS (10), is also found in back-propagation. The softmax loss (2) has gradients w.r.t $\tilde{\boldsymbol{x}}$ as

$$-\frac{\partial\ell}{\partial\tilde{\boldsymbol{x}}} \propto \tilde{\boldsymbol{w}}_y-\sum_{c=1}^{C}\frac{\exp(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_c)}{\sum_k\exp(\kappa\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}_k)}\tilde{\boldsymbol{w}}_c, \tag{12}$$

in which the second term corresponds to the convex combination in (10). The loss gradient (12) updates the feature $\tilde{\boldsymbol{x}}$ by direction from the combination vector to the ground-truth classifier $\tilde{\boldsymbol{w}}_y$. As the combination vector gets close to $\tilde{\boldsymbol{x}}$ through the optimization (11), the feature representation $\tilde{\boldsymbol{x}}$ is effectively updated toward the target classifier $\tilde{\boldsymbol{w}}_y$ since the updating direction approaches the orientation of

$\tilde{\boldsymbol{w}}_y - \tilde{\boldsymbol{x}}$. Thereby, the LS-optimized κ^* in (11) contributes to reducing intra-class variance around $\tilde{\boldsymbol{w}}_y$.

In addition, we also mention that too small κ degrades the back-propagation (12). By $\kappa \to 0$, it is reduced to

$$-\frac{\partial \ell}{\partial \tilde{\boldsymbol{x}}} \propto \tilde{\boldsymbol{w}}_y - \frac{1}{C}\sum_{c=1}^{C} \tilde{\boldsymbol{w}}_c, \tag{13}$$

which excludes dependency on $\tilde{\boldsymbol{x}}$, thus failing to properly train the feature representation.

Characteristics of κ. The projection (8) generally provides sparse coefficients of $\boldsymbol{\alpha}$. If the feature $\tilde{\boldsymbol{x}}$ is far away from the classifier weights $\{\tilde{\boldsymbol{w}}_c\}_c$, the logits are fairly small, $z_c \ll 1 \, \forall c$. In that case, κ^* would be larger to produce the sparse coefficients induced in (8) from the small logits. In other words, κ^* would roughly reflect the distance between $\tilde{\boldsymbol{x}}$ and a convex hull spanned by $\tilde{\boldsymbol{w}}_c$; this is detailed in a supplementary material.

2.3. Optimization

The optimal κ is given at *each* sample and then fed into the loss (2) to trigger back-propagation for training the model Θ and classifier $\boldsymbol{W}$; κ is not subject to end-to-end learning.

While the rule-of-thumb (Section 2.1.1) provides a closed-form optimizer (4), the other two approaches based on maximum likelihood (Section 2.1.2) and least squares (Section 2.2) require seeking the optimizer in (6, 11). It is noteworthy that a simple line search works by computing the (easy-to-compute) criteria (5, 10) over candidates of κ;

$$\kappa^* = \arg\min_{\kappa \in \mathcal{K}} \mathcal{E}_\kappa, \tag{14}$$

where $\mathcal{K}$ indicates a *predefined candidate set* for κ; we construct $\mathcal{K}$ by equally-spaced 20 values ($|\mathcal{K}| = 20$) in log scale on $[e^{-2}, e^{5}]$, as detailed in a supplementary material.

3. Results

We apply the cosine-based softmax loss (2) to train deep models while optimizing κ (Section 2) on various scenarios of image classification; the training protocols are detailed in a supplementary material.

3.1. Optimization for κ

We first compare the three types of optimization approaches proposed in Section 2 by training ResNet-50 [11] ($d = 2048$) on ImageNet [8] dataset ($C = 1000$). For reference, *trainable* κ is also applied by end-to-end learning an auxiliary parameter $\kappa' \in \mathbb{R}$ such that $\kappa = \exp(\kappa') \in \mathbb{R}_+$.

The performance results are shown in Table 1 reporting optimized κ^* by the four methods after training; as κ^* is obtained in a sample-wise manner, we report an averaged κ^* across samples. The optimized κ^* at each training epoch is also shown in Figure 1.

Table 1. Classification accuracy (%) with optimized κ^* on ImageNet by ResNet-50.

Method	κ^*	Acc.
Rule (4)	65525	3.36
ML (6)	0.14	0.10
LS (11)	12.83	77.27
Trainable	120.8	75.51

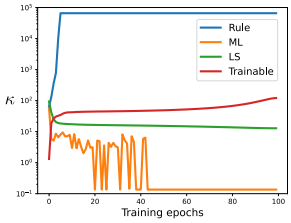


Figure 1. Optimized κ at each training epoch.

Rule-based approach (4) rapidly increases κ as the classifiers $\tilde{\boldsymbol{w}}_c$ are trained toward $\bar{r} = \|\frac{1}{C}\sum_C \tilde{\boldsymbol{w}}_c\|_2 \to 1$ which is attributed to relatively large κ induced by high dimensionality $d = 2048$. The issues regarding the high dimensionality and trainability of classifier weights significantly impedes the *rule* in Section 2.1.1 which is originally derived from probabilistic distribution of samples [3].

ML approach (6) produces too small κ^* on this ImageNet task, indicating uniform feature distribution in the KDE model (3). Due to the smaller κ, it is hard to properly update the features as analyzed in Section 2.2.1, degrading discriminativity, and thus uniformity of feature distribution distribution is not improved while retaining small κ^*. Besides, as discussed in Section 2.1.2, the normalization constant $\bar{Z}_{\kappa,d}$ is not so well defined in the high-dimensional feature space, which exacerbates estimation of κ.

LS approach (11) produces moderate κ, exhibiting favorable performance, in contrast to the above-mentioned two approaches that fail to estimate κ. This proposed method exploits geometrical projection from $\tilde{\boldsymbol{x}}$ onto a convex hull of classifiers to optimize κ with high robustness against feature dimensionality[1], thereby stably producing favorable κ^*. At the early stage of training, the features are distributed away from the classifiers, which causes larger κ^* as discussed in Section 2.2.1. Then, as the training proceeds, the features are so close to the classifier vectors that κ^* becomes smaller.

Trainable approach is inferior to the LS approach since it renders too large κ at the later training epochs. When most samples are correctly classified by sufficiently training a model, the trainable κ keeps increasing so that the softmax loss is (trivially) reduced by the enlarged logits.

These analyses show that the proposed least-square (LS) approach (Section 2.2) effectively optimizes the parameter κ, a reciprocal of temperature as $\tau^* = 1/\kappa^*$.

3.2. Analysis of κ

Next, we thoroughly analyze κ on standard image classification tasks using benchmark datasets of Cifar-10/100 [21], Food-101 [4] and ImageNet [8].

While our LS approach automatically estimates κ^*, we

[1]The projection is intrinsically performed in a lower-dimensional subspace spanned by $\tilde{\boldsymbol{x}}$ and $\{\tilde{\boldsymbol{w}}_c\}_{c=1}^{C}$.

prefix κ to various values which are constant throughout training. Table 2 shows performance results of all κ's, also reporting the performances of a standard softmax loss without normalization for reference. The proposed LS method effectively optimizes κ in an adaptive manner to respective tasks and models. On the same model ϕ_θ, the optimized κ^* are diverse; one can see different κ^* respectively on Cifar-10/100 using ResNet-34 and on ImageNet/Food-101 using ResNet-50. It is noteworthy that our method adaptively estimates κ across various numbers of classes; generally, larger κ^* is obtained on larger number of classes, as summarized in a supplementary material. Interestingly, the performances seem to be maximized roughly in $\kappa \sim [\kappa^*, 2\kappa^*]$, implying that κ^* works as a *lower bound* of feasible range in which κ produces competitive performance.

Then, we delve deeper into the learnt feature representation on various κ. For that purpose, we first clarify the distinctive characteristics of loss functions equipped with different κ. To ease discussion, we presume that non-target logits $z_c, \forall c \neq y$ are close to zeros, i.e., the angle $\angle(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}}_c) \approx 90°$, after sufficient number of training epochs. Accordingly, the softmax loss (2) is reduced to

$$\ell_\kappa(z_y) = -\kappa z_y + \log[\exp(\kappa z_y) + (C-1)], \qquad (15)$$

where the number of classes is denoted by C. As depicted in Figure 2, the function with smaller κ outputs higher loss scores at larger z_y to enforce the logit z_y to be close to 1, i.e., $\angle(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}}_y) \to 0°$, by keeping update for feature representation. On the other hand, the function with larger κ rapidly decreases loss, vanishing at smaller z_y ($\angle(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}}_y) \gg 0$); it endows larger z_y with little update in a similar way to Focal Loss [24]. Therefore, these loss functions with smaller and larger κ would lead to distinct characteristics of feature representation, especially regarding intra-class distribution around the classifier $\tilde{\boldsymbol{w}}_y$; that is, smaller/larger κ could be associated with smaller/larger feature variance.

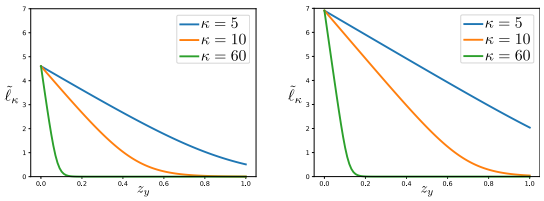
Then, we empirically analyze feature distribution in Figure 3 by applying t-SNE [34] to ResNet-50 features trained on ImageNet. It shows that intra-class distribution are broadened by changing $\kappa = 5$ to $\kappa = 60$. While exhibiting compact intra-class structure, $\kappa = 5$ collapses discrimination among classes since such a too small κ fails to learn discriminative features (Section 2.2.1), as shown by the low performance score in Table 2. Roughly speaking, $\kappa \sim \kappa^*$ contributes to both compact intra-class distribution and performance improvement. At the larger $\kappa = 60$, features are distributed with large intra-class variance, making the class boundaries less clear to degrade performance (Table 2).

We also quantitatively analyze the feature diversity in Table 3 which shows three types of measures: averaged intra-class angle $E_{(\tilde{\boldsymbol{x}},y)}\angle(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}}_y)$, averaged angle to classifier subspace $E_{\tilde{\boldsymbol{x}}}\angle(\tilde{\boldsymbol{x}}, \mathtt{span}(\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C))$ and rank of feature distribution. Smaller κ obviously contributes to intra-class compactness. As analyzed in Section 2.2.1, the LS approach effectively works for compact feature representation through back-propagation, and thus the optimized κ^* is regarded as a lower bound of κ that produces favorable compactness as well as competitive classification performance. Increasing κ induces not only larger intra-class variance but also higher deviation from the classifier subspace $\mathtt{span}(\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C)$, resulting in higher rank than $C = 1000$. From a viewpoint of *linear* C-class discrimination, only the feature representation in the classifier subspace, of which rank is at most C, matters to classification. Actually, in the standard softmax loss, features are contained in the classifier subspace, producing less deviation, and thus exhibit 784 rank which is close to $C = 1000$, though intra-class variance is relatively large. On the other hand, feature distribution produced by larger κ is deviated from the classifier subspace while rendering higher rank. Thus, the larger κ enables the model to extract features not so tailored for the target classification.

These analyses clarify that the parameter κ controls feature variance on the basis of κ^* (lower bound) in an *explainable* way; namely, two distinct characteristics of intra-class compactness and feature diversity are embedded by smaller and larger κ, respectively. We then explore the utility of those feature representations in the following sub-sections.

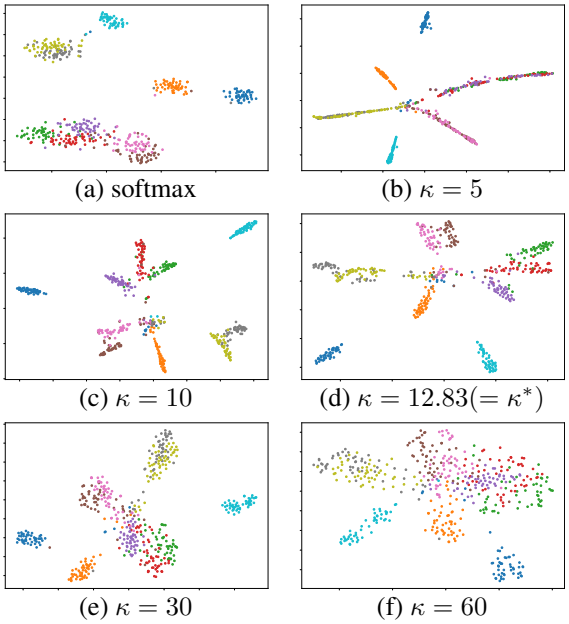
Table 2. Image classification accuracies (%).

Dataset	Cifar-10	Cifar-100	Food-101	ImageNet
C	10	100	101	1000
Model	ResNet34	ResNet34	ResNet50	ResNet50
d	512	512	2048	2048
softmax	95.56	79.07	83.03	77.12
LS (11)	95.49	78.26	83.27	77.27
(κ^*)	(5.39)	(11.17)	(8.30)	(12.83)
Fixed $\kappa = 1$	86.19	23.07	31.01	17.48
$\kappa = 5$	95.39	78.86	83.07	68.95
$\kappa = 10$	95.57	78.99	83.34	77.09
$\kappa = 20$	95.64	78.42	83.24	77.76
$\kappa = 30$	95.67	78.06	82.86	77.66
$\kappa = 40$	95.85	78.70	82.45	77.52
$\kappa = 50$	95.57	78.29	82.35	77.30
$\kappa = 60$	95.61	78.24	82.22	76.88
$\kappa = 2\kappa^*$	95.54	78.57	83.44	77.81



(a) $C = 100$ (b) $C = 1000$

Figure 2. Simplified loss functions (15) focusing on z_y.



(a) softmax (b) $\kappa = 5$ (c) $\kappa = 10$ (d) $\kappa = 12.83 (= \kappa^*)$ (e) $\kappa = 30$ (f) $\kappa = 60$

Figure 3. Feature distribution of the first 10 classes in ImageNet, which is visualized by applying t-SNE to ResNet-50 features.

Table 3. Feature diversity of ResNet-50 on ImageNet evaluation set ($C = 1000$). Angles are measured by degree ($^\circ$).

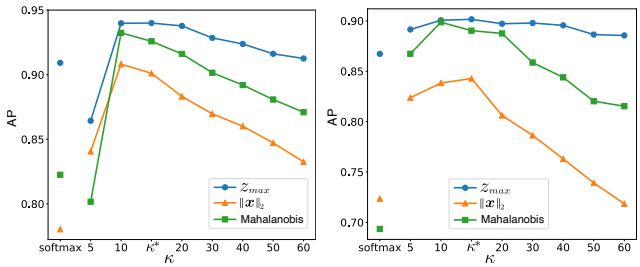
	$\angle(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{w}}_y)$	$\angle(\tilde{\boldsymbol{x}}, \mathtt{span}(\{\tilde{\boldsymbol{w}}_c\}_{c=1}^C))$	rank
softmax	65.19	7.54	887
$\kappa = 5$	21.65	0.72	526
10	31.03	6.59	1052
κ^*	37.05	14.14	1597
20	52.74	24.68	1806
30	63.05	31.06	1913
40	68.71	32.91	1922
50	72.16	33.88	1912
60	74.31	34.61	1907

3.2.1. Smaller κ

The small $\kappa \sim \kappa^*$ given by the LS approach leads to intra-class compact feature representation, which also contributes to enhancing fidelity to the target (C-class) classification. This characteristics can be leveraged to learn models of high *confidence*. In other words, the model learned by smaller κ is capable of detecting less confident samples such as miss-classified (MISS)[2] and out-of-distribution (OOD) samples. Deep models are expected to provide a classification result with a confidence score indicating how much confident the classification output is; they should assign high confidence to samples that are correctly classified and lower confidence to the other samples such as of MISS and OOD.

For empirical evaluation, ResNet-50 is trained on ImageNet (Table 2) and then is applied to detect MISS samples in ImageNet evaluation set as well as OOD samples drawn from Food-101 [4] dataset, compared to the correctly-classified samples on ImageNet evaluation set where we exclude a few classes related to food objects. In [31], the feature norm $\|\boldsymbol{x}\|_2$ is exploited as a confidence score to distinguish OOD samples from in-distribution ones, and it is favorably combined with a measure based on Mahalanobis distance [23]. We also applied a classical measure of maximum logit $z_{max} = \max_c z_c$ to this task. Figure 4 shows performance results by computing average precision of correctly-classified samples in comparison to MISS and OOD ones based on those three types of confidence scores.

[2]MISS samples are the ones that are in-distribution but classified into wrong classes.



(a) MISS detection (b) OOD detection

Figure 4. Average precision for detecting (a) MISS and (b) OOD samples by using three types of confidence measures.

Except for $\kappa = 5$ which impedes learning as discussed above, the smaller κ improves detection scores on all the measures, outperforming the results of a standard softmax loss. Even a classical maximum logit z_{max} works well in comparison to the others, reaching saturated (maximum) scores at $\kappa = \kappa^* (= 12.83)$. It should be noted that the small κ is less harmful in terms of the classification performance as shown in Table 2. This result demonstrates that the smaller κ contributing to compact intra-class feature representation helps to learn models of high confidence.

3.2.2. Larger κ

On the other hand, the larger κ leads to diverse feature representation beyond a classifier subspace. The diversity of features could contribute to encoding various image characteristics in a general way, not specific to the target task.

To explore the generality of the diverse features produced by the larger κ, we apply ResNet-50 pre-trained on ImageNet to the other downstream tasks via transfer learning. Table 4 shows performances of transfer leaning to various classification tasks in the two scenarios: (1) *linear probe* to freeze the pre-trained model as a fixed feature extractor followed by training only the linear classifier, and (2) *fine-tuning* to train both the pretrained model and the linear classifier in an end-to-end fashion.

Specifically, classification performance on linear probe (Table 4a) is significantly improved by enlarging κ; the features learnt such as by $\kappa = 60$ outperform those pre-trained by a standard softmax loss with a large margin. The result

Table 4. Classification performances (%) by transferring ResNet50 pretrained on ImageNet to various tasks.

	Dataset	CUB-200 [39]	Food-101 [4]	Car-196 [20]	Aircraft-100 [27]	SUN-397 [43]	DTD [7]	Flower-102 [30]
(a) Linear probe:	softmax	69.05	67.08	46.42	42.57	58.49	72.02	86.39
	Fixed $\kappa = 10$	47.13	48.23	23.44	22.29	44.69	59.15	55.47
	20	62.89	65.54	43.12	38.70	56.26	69.52	77.35
	30	70.33	69.36	53.59	46.77	58.85	72.71	85.40
	40	73.80	71.88	56.96	52.51	60.28	74.20	88.79
	50	75.27	73.58	60.44	55.99	61.09	72.66	90.05
	60	75.63	74.04	61.86	57.16	61.14	74.10	92.19
(b) Fine-tuning:	softmax	80.17	86.12	86.08	77.49	61.71	74.20	94.34
	Fixed $\kappa = 10$	79.76	86.28	85.32	76.71	62.22	73.56	90.57
	20	80.64	86.67	86.39	77.10	63.28	75.64	93.45
	30	81.68	86.79	86.60	78.84	63.53	76.81	94.64
	40	81.33	86.86	86.85	78.30	63.85	76.01	95.30
	50	81.59	87.10	86.81	77.76	63.95	76.44	95.71
	60	81.18	86.83	86.66	78.12	63.80	76.49	96.36

shows that the larger κ embeds mechanisms of general feature extraction into the model so as to produce effective feature representation generalizable toward various tasks, even without fine-tuning process. Thus, the deviation from the classifier subspace $\texttt{span}(\{\tilde{\boldsymbol{w}}_c\}_{c=1}^{C})$ shown in Table 3 implies effective features which are not so contributive to the target classification but applicable to characterize various objects on the other tasks. In the scenario of fine-tuning (Table 4b), the larger κ also exhibits favorable performance. Based on the empirical evaluation, we can conjecture that the larger κ is useful for (pre-)training *generalizable* model which works well in transfer learning.

3.2.3. Summary

In summary, the smaller $\kappa \sim \kappa^*$ estimated by our LS approach works for enhancing fidelity to the target classification task by embedding high confidence to the learnt models, while the larger κ contributes to improving generalization performance with high transferability across various tasks. Thus, based on the analyses about these smaller and larger κ, the middle κ such as $2\kappa^*$ is supposed to exhibit *moderate* specialization to the in-distribution samples with *moderate* generalization, which is favorable for classifying in-distribution *test* samples as shown in Table 2.

We have clarified that the cosine-based softmax loss is capable of controlling the characteristics of learnt models by using the softmax temperature (κ) in an *explainable* way based on the optimized κ^*; we can use κ^* to train the model of high confidence while applying larger $\kappa > 2\kappa^*$ to pretrain a generalizable model. In that sense, it enhances the interpretability of the learnt model.

3.3. "Cosify"

As shown above, the *cosine*-based softmax loss controls characteristics of models via κ. Then, we have the following question; *is it possible to embed such mechanism to the model pretrained on a standard softmax loss?*

To answer the question, we simply apply the cosine-based softmax loss in a fine-tuning manner to re-train the ResNet-50 that is pretrained on ImageNet by a standard softmax loss. For analyzing contribution of network depth, we finetune only a subset of blocks in the ResNet-50 which is composed of four convolution blocks. In the re-training, we apply two extreme $\kappa \in \{10, 60\}$ which induce specialization and generalization, respectively, as analyzed above.

The performance results are shown in Table 5 evaluating the specialization and generalization of the re-trained ("*cosified*") model in similar manners to Figure 4 and Table 4. The model is well *cosified* by $\kappa = 10$, exhibiting favorable performance regarding model confidence, while impeding generalization performance as in the model trained from scratch by the cosine softmax loss with $\kappa = 10$. On the other hand, it seems to be hard to *cosify* the pretrained model toward $\kappa = 60$ as the cosified model slightly falls behind the scratch model of $\kappa = 60$ in terms of generalization performance of linear probe on CUB200. As shown in Table 3, the feature representation of the pre-trained model is contained in the classifier subspace, thus being rather biased to that of the smaller κ, which makes it easier to *cosify* the model toward $\kappa = 10$. Nonetheless, the cosified model by $\kappa = 60$ improves generalization performance over the original pre-trained one especially in the scenario of fine-tuning. Besides, it is noteworthy that the *cosification* of a pretrained model can be performed even by retraining only the last block of ResNet-50 on both cases of $\kappa \in \{10, 60\}$; there is no big performance difference among fine-tuned blocks.

3.4. Other classification tasks

The LS approach (11) estimates κ effectively on the other classification tasks as follows.

Table 5. Performance of "cosified" ResNet50 which is originally pre-trained on a standard softmax loss. These scores are measured in the same ways as in Table 2, Figure 4 and Table 4.

finetune blocks 1	2	3	4	ImageNet Acc.	Specialization AP for MISS Z_{max}	Specialization AP for MISS $\|\boldsymbol{x}\|_2$	Specialization AP for OOD Z_{max}	Specialization AP for OOD $\|\boldsymbol{x}\|_2$	Generalization CUB200 linear	Generalization CUB200 finetune
softmax (orig.)				77.12	0.9092	0.7804	0.8673	0.7236	69.05	80.17
$\kappa = 10$										
from scratch				77.09	0.9398	0.9082	0.9006	0.8383	47.13	79.76
✓	✓	✓	✓	76.50	0.9421	0.9073	0.9039	0.8497	51.28	79.43
-	✓	✓	✓	76.68	0.9413	0.9078	0.9020	0.8466	50.79	80.21
-	-	✓	✓	76.42	0.9416	0.9019	0.9002	0.8507	51.04	79.62
-	-	-	✓	76.47	0.9420	0.9094	0.9019	0.8564	49.91	79.53
$\kappa = 60$										
from scratch				76.88	0.9125	0.8326	0.8856	0.7185	75.63	81.18
✓	✓	✓	✓	77.09	0.9154	0.8583	0.8678	0.7679	71.52	82.13
-	✓	✓	✓	76.85	0.9154	0.8584	0.8655	0.7673	71.49	82.59
-	-	✓	✓	76.86	0.9144	0.8587	0.8664	0.7734	71.54	82.66
-	-	-	✓	76.71	0.9144	0.8601	0.8650	0.7841	70.88	82.59

3.4.1. Face recognition

Face images are successfully embedded into the angular-based feature representation [9, 36, 37] which is learned by the cosine-based softmax loss with large-margin regularization. We train ResNet34 on MS1M-RetinaFace dataset [10] by using CosFace loss [37] to which the LS approach (11) and various κ are applied as in Table 2; the LS method based on softmax projection (10) is applicable to the CosFace Loss which touches the target logit z_y for enhancing large-margin classification. Table 6 reports verification performance (%) on the datasets of LFW [15], CFP-FP [32] and AgeDB-30 [29] by following the protocol in [9]. Our LS approach produces favorable κ^*, an effective lower bound of κ; the middle $\kappa = 2\kappa^*$ produces competitive performance at in-domain (face) evaluation as discussed in Section 3.2.3. This experimental result demonstrates that the LS method works well even on large-scale classes ($C = 93431$) and the large-margin loss of CosFace [37].

3.4.2. Imbalanced classification

As analyzed in Section 3.2.1, the smaller κ endows the model with regularization to enhance intra-class compactness of feature representation. The regularization could be helpful for improving performance on long-tailed class distributions which induce imbalanced intra-class feature variances across head (majority) and tail (minority) classes as shown in [44]. We apply the cosine-based softmax loss to train ResNet-50 in the framework of long-tailed learning [18] on datasets of ImageNet-LT [26], Places-LT [26] and iNaturalist2018 [16]. Table 7 shows performance results by various fixed κ and LS-optimized κ^*. In this task, the smaller κ close to our lower bound κ^* produces better performance by properly regularizing intra-class feature distributions. Particularly, in iNaturalist2018, too small $\kappa = 10$ significantly degrades performance while our $\kappa^* = 18.54$ works well; $\kappa = 10$ is away from $\kappa^* = 18.54$ in the iNaturalist2018, though it is effectively applied to the other datasets where $\kappa^* \sim 10$. In ImageNet-LT derived from ImageNet, the LS approach produces $\kappa^* = 12.76$ which is almost the same value as $\kappa^* = 12.83$ on ImageNet (Table 2), demonstrating the robustness of the LS-based estimation.

Table 6. Face verification performance (%) of ResNet-34 trained by CosFace [37] loss.

Dataset	MS1M-RetinaFace ($C = 93431$)		
Model	ResNet34 ($d = 512$)		
Eval. dataset	Age30	CFP/FP	LFW
LS ($\kappa^* = 16.56$)	97.88	98.54	99.80
Fixed $\kappa = 10$	97.00	97.99	99.72
20	97.75	98.50	99.70
30	98.08	98.64	99.73
40	98.15	98.70	99.73
50	98.10	98.61	99.77
60	97.98	98.43	99.72
$2\kappa^*$	98.27	98.60	99.73

Table 7. Classification accuracy (%) on long-tailed datasets.

Dataset	ImageNet-LT	iNat2018	Places-LT
C	1000	8142	365
Model	ResNet50	ResNet50	ResNet50
d	2048	2048	2048
softmax	48.51	66.76	27.19
LS (κ^*)	49.16 (12.76)	67.75 (18.54)	28.00 (11.8)
Fixed $\kappa = 10$	49.00	57.36	27.19
20	49.06	67.21	25.86
30	48.14	67.87	24.80
40	47.26	66.88	24.42
50	46.33	65.87	23.79
60	45.61	64.94	23.72
$2\kappa^*$	48.52	67.16	25.69

4. Conclusion

We have proposed a novel method to adaptively estimate temperature in a cos-softmax loss. The method is formulated in a least-square manner by exploiting geometrical relationships between features and classifiers. In addition, we thoroughly analyze temperature to clarify that feature embedding exhibits diverse characteristics which are controllable by the temperature in an explainable way. The experimental results demonstrate that the optimized temperature works as lower bound of feasible temperature range while providing favorable performance such as on OOD detection and imbalanced classification.

References

[1] Sergio A.Balanya, Juan Maronas, and Daniel Ramos. Adaptive temperature scaling for robust calibration of deep neural networks. *Neural Computing and Applications*, 36(14): 8073–8095, 2024. 2

[2] Atish Agarwala, Jeffrey Pennington, Yann Dauphin, and Sam Schoenholz. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv,* 2010.07344, 2020. 1, 2

[3] Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005. 3, 4

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014. 4, 6, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv,* 2003.04297, 2020. 1, 2

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4

[9] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 1, 2, 8

[10] Jiankang Deng, Jia Guo, Debing Zhang, Yafeng Deng, Xiangju Lu, and Song Shi. Lightweight face recognition challenge. In *ICCVW*, 2019. 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 4

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross-Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[13] Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*, 2014. 1

[14] Kurt Hornik and Bettina Grün. movMF: An r package for fitting mixtures of von mises-fisher distribution. *Journal of Statistical Software*, 58(10):1–31, 2014. 3

[15] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 8

[16] iNatrualist. The inaturalist 2018 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2018, 2018. 8

[17] Tom Joy, Francesco Pinto, Ser-Nam Lim, Philip H. S. Torr, and Puneet K. Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In *AAAI*, pages 14919–14926, 2023. 2

[18] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 8

[19] Takumi Kobayashi. Direct-sum approach to integrate losses via classifier subspace. In *BMVC*, 2024. 3

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Workshop on 3D Representation and Recognition*, 2013. 7

[21] Alex Krizhevsky and Geoffrey E. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 4

[22] Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. In *ICLR*, 2023. 2

[23] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018. 6

[24] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[25] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017. 1, 2

[26] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019. 8

[27] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013. 7

[28] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics (2nd edition)*. John Wiley and Sons Ltd., 2000. 1, 3

[29] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected in-the-wild age database. In *CVPRW*, 2017. 8

[30] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 7

[31] SoonCheol Noh, DongEon Jeong, and Jee-Hyong Lee. Simple and effective out-of-distribution detection via cosine-based softmax loss. In *ICCV*, 2023. 1, 2, 6

[32] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M. Patel, Rama Chellappa, and David W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 8

[33] Zhun Sun and Chao Li. Analyzing the impact of learnable softmax temperature in contrastive visual-textual alignment systems: Benefits, drawbacks, and alternative approaches. *TMLR*, 2024. 2

[34] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 5

[35] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021. 2

[36] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: l_2 hypersphere embedding for face verification. In *ACM MM*, 2017. 2, 8

[37] 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 *CVPR*, pages 5265–5274, 2018. 1, 2, 8

[38] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 2

[39] Peter Welinder, Steve Branson, Takashi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 7

[40] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 1

[41] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE TPAMI*, 31(2):210–227, 2009. 1, 3

[42] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. Improving generalization via scalable neighborhood component analysis. In *ECCV*, 2018. 1, 2

[43] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[44] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, pages 5704–5713, 2019. 8

[45] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2019. 1, 2