t-vMF Similarity For Regularizing Intra-Class Feature Distribution

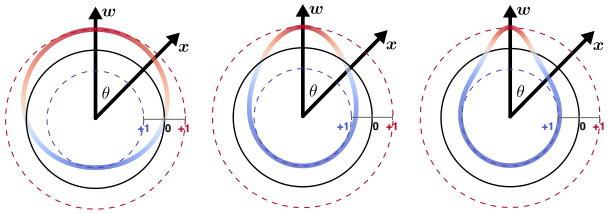
Takumi Kobayashi
National Institute of Advanced Industrial Science and Technology, Japan
takumi.kobayashi@aist.go.jp

Abstract

Deep convolutional neural networks (CNNs) leverage large-scale training dataset to produce remarkable performance on various image classification tasks. It, however, is difficult to effectively train the CNNs on some realistic learning situations such as regarding class imbalance, small-scale and label noises. Regularizing CNNs works well on learning with such deteriorated training datasets by mitigating overfitting issues. In this work, we propose a method to effectively impose regularization on feature representation learning. By focusing on the angle between a feature and a classifier which is embedded in cosine similarity at the classification layer, we formulate a novel similarity beyond the cosine based on von Mises-Fisher distribution of directional statistics. In contrast to the cosine similarity, our similarity is compact while having heavy tail, which contributes to regularizing intra-class feature distribution to improve generalization performance. Through the experiments on some realistic learning situations such as of imbalance, small-scale and noisy labels, we demonstrate the effectiveness of the proposed method for training CNNs, in comparison to the other regularization methods. Codes are available at https://github.com/tk1980/tvMF.

1. Introduction

Deep convolutional neural networks (CNNs) are fundamental methods to produce promising performance on various computer vision tasks including visual recognition [16, 27]. A large amount of parameters in CNNs are effectively optimized in an end-to-end manner on a large-scale dataset which contains plenty of image samples with detailed annotation; in other words, high-performance CNNs demand such a *healthy* dataset of large-scale and clean-labeled samples. For example, ImageNet [10], a standard benchmark dataset for image classification, is composed of a large number of training samples, each of which is assigned one of 1000 class labels, and those samples are uniformly distributed across classes without severe bias toward specific class categories. Such a data-hunger nature hinders CNNs from being applied to various real-world tasks. Due to the laborious procedure of collecting and annotating data, real-world tasks are frequently equipped with rather *deteriorated* training datasets which are subject to such as class imbalance, small-scale and label noises. The CNNs trained on those poor datasets degrade performance, e.g., due to overfitting.



(a) $\cos\theta$ (b) t-vMF ($\kappa = 4$) (c) t-vMF ($\kappa = 16$)

Figure 1. t-vMF similarity (7) compared to cosine similarity $\cos\theta$. The proposed t-vMF produces compact-support similarity function around the classifier weight $\boldsymbol{w}$ with the parameter κ to control the compactness. It orients features $\boldsymbol{x}$ toward $\boldsymbol{w}$ as a *implicit* regularization to enhance compact *intra-class* distribution. Colored line indicates similarity values in $[-1, +1]$ over the angle θ.

The bottleneck of CNNs could be alleviated by reducing their parameter size from the architectural viewpoint [19, 51] and data-augmentation techniques would contribute to virtually enlarge the training data by means of injecting perturbation into real image samples [12, 50]. On the other hand, as a rather general approach, some *regularizations* can be effectively introduced to CNNs for improving generalization performance [40, 46, 20, 31, 11].

A crucial feature representation is found in the neuron activations produced by the penultimate layer which are fed into the final classifier. Thus, regularization on those features contributes to enhancing feature representation learning even on the deteriorated datasets where training samples are too poorly collected to well model the intrinsic feature distribution. In the literature of deep learning, there are some regularization techniques for feature representation such as center loss [46] to reduce within-class variance and DropOut [40, 29] to inject stochastic perturbation. It is also possible to regularize features at a classifi-

cation layer through end-to-end learning. A representative approach would be large-margin loss [31, 30, 45, 11] by embedding large-margin criterion into a softmax cross-entropy loss. The large-margin criterion renders the classifier of high generalization performance [43] as well as favorable feature representation through the end-to-end learning. The large-margin methods modify logits of ground-truth class based on a *cosine similarity* between an input feature vector and the classifier weight at the classification.

In this work, we focus on the cosine similarity, a fundamental metric in the classifier, to impose regularization on features for improving performance especially on deteriorated training datasets. The cosine similarity is built on the angle between two vectors which is geometrically depicted on a unit hyper-sphere, and thus we leverage von Mises-Fisher (vMF) distribution [35], one of directional statistical models, to propose a novel similarity beyond the cosine similarity. The proposed similarity is a compact-support function over angles which enables us to *implicitly* regularize intra-class feature distribution (Fig. 1). While the method can be related to the regularization loss [46] and the large-margin methods [31, 11] which touch cosine similarity, the proposed method exhibits clear difference from those prior works in the following points: (1) the proposed similarity regulates features without introducing additional regularization loss, and (2) it is equally applied to all the classes without paying special attention to the ground-truth class. (3) It is also noteworthy that the proposed similarity can simply substitute the cosine similarity in a computationally efficient form implemented by only *one-line* code.

1.1. Related works

Regularization. We briefly review the regularization methods according to the simple neuron model, $z = \boldsymbol{w}^\top \boldsymbol{x}$ where the output z is computed by the inner product of the input feature $\boldsymbol{x}$ and (filter) weight $\boldsymbol{w}$.

CNN filter weights are usually subject to L_2-norm regularization, called weight decay [28]. This regularization is extended into Weight-Normalization [38] which leads to cosine similarity in conjunction with normalizing features [1].

DropOut [40] is a representative method to introduce stochastic perturbation into input features for regularizing CNNs; the effect of DropOut at the last classification layer is analyzed in [29]. Perturbation is also injected even to input images in the framework of data augmentation [12, 50] for classification and in denoising auto-encoder [44]. Feature distributions are regularized more directly by adding regularization loss such as center loss [46] and classifier loss [20] for improving within-class variance. The proposed method also works on improving intra-class distribution and it embeds regularization into logits (outputs) without modifying loss nor adding the regularization loss term.

Regularization on the output is mainly found in large-margin methods [31, 30, 45, 11] at the last classification layer leading to loss. The classification output z is characterized by cosine similarity between the input feature $\boldsymbol{x}$ and the classifier weight $\boldsymbol{w}$, and then the output for the ground-truth class is degraded based on the cosine similarity for inducing larger margin in classification. While the proposed method also modifies the cosine similarity, there is clear difference between them. The proposed method fairly treats all the classes without any bias toward the ground-truth class and simply replaces cosine similarity without annotation (label) information. Thereby, our method addresses regularization for intra-class distribution, while the large-margin methods focus on discrimination among classes; thus, the two approaches would be complementary.

Cosine Similarity. The cosine similarity has been applied in the framework of pair-wise matching such as for image retrieval [3] and the metric learning that learns lower dimensional feature representation [47]; a pair of images is generally processed through Siamese network to compute cosine similarity as a matching score [5]. The cosine similarity is also found in the classification of normalized features which contributes to favorable feature representation learning [30, 45, 11, 36, 18, 52, 15]. The proposed method is formulated to replace the similarity so that it could be applicable to various models. In the other research lines, the cosine similarity is embedded into CNNs such as for loss function [4] instead of (softmax) cross-entropy loss and for normalization [34] to replace Batch-/Layer-Norm [24, 1].

von Mises-Fisher Distribution. By regarding the cosine similarity as a metric on a unit hyper-sphere, we can naturally derive von Mises-Fisher (vMF) distribution [35] to statistically model samples of unit norm. The vMF is applied in machine learning community [39], such as text mining [2], user-behavior analysis [37] and clustering [14]. It is also employed in the literature of deep learning such as in semantic segmentation [21] and losses [52, 15]. The methods [52, 15] leverage the vMF model to formulate a loss based on cosine similarity. In contrast, we consider the vMF model in the process of producing logits to which the cosine similarity has so far been applied; we simply apply the classification loss of the normalized classifier [36, 18] which is almost the same as the vMF-based losses [52, 15].

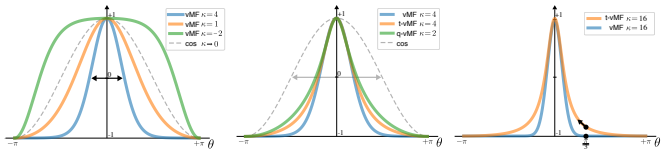
2. vMF-based Similarity Beyond Cosine

The linear classifier in CNNs is formulated as an inner product between a classifier weight $\boldsymbol{w}$ and a feature vector $\boldsymbol{x}$ produced by the penultimate layer, as follows[1]:

$$z_c = \boldsymbol{w}_c^\top \boldsymbol{x} = \|\boldsymbol{w}_c\| \|\boldsymbol{x}\| \cos\theta = \mathtt{s}_c(\boldsymbol{x}) \cos\theta, \qquad (1)$$

where z_c is a logit for the c-th class and the norms of $\boldsymbol{w}_c$ and $\boldsymbol{x}$ are reduced into a scaling factor $\mathtt{s}_c(\boldsymbol{x})$ which could be a

[1] We can simply remove a bias term while keeping performance.



(a) vMF (b) Comparison (c) Gradient

Figure 2. Proposed vMF-based similarities.

trainable parameter as discussed in Sec. 2.4. From a geometrical viewpoint, the classifier (1) is fundamentally characterized by the angle θ and it applies cosine function $\cos$ to measure a similarity based on θ; in this paper, it is referred to as a (similarity) *measuring function*. The cosine measuring function, however, has broad support region, which accordingly permits features to be distributed with larger within-class variance; as shown in Figs. 1a&2b, samples of $\theta \in (-\frac{\pi}{2}, +\frac{\pi}{2})$ exhibit positive similarity.

The features of larger within-class variance are known to degrade generalization performance as classically mentioned in discriminant analysis [13]. Empirically, the larger and imbalanced variances are observed in the learning on the deteriorated dataset to lower performance [26, 49]. On the other hand, small variance indicates that CNN produces effective feature representation consistent within a class. Compact intra-class feature distribution is equivalent to extracting class-intrinsic features shared among within-class samples; it thereby enhances generalization performance by mitigating overfitting. Thus, we formulate a method to induce compact intra-class distribution through improving the *support* region of cosine measuring function (Fig. 1). To this end, we employ the directional statistical model, von Mises-Fisher distribution [35], to deal with the angle θ.

2.1. Similarity by von Mises-Fisher Model

The angle θ between $\boldsymbol{w}_c$ and $\boldsymbol{x}$ in (1) is a core metric on a unit hyper-sphere. Samples on the sphere can be statistically modeled by von Mises-Fisher (vMF) distribution [35, 2] which is formulated as

$$\mathtt{p}(\tilde{\boldsymbol{x}}; \tilde{\boldsymbol{w}}, \kappa) = C_\kappa \exp(\kappa \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}}) = C_\kappa \exp(\kappa \cos\theta), \quad (2)$$

where $\tilde{\boldsymbol{x}}$ is a d-dimensional unit vector ($\|\tilde{\boldsymbol{x}}\| = 1$), $\tilde{\boldsymbol{w}}$ is a unit vector orienting the center of the distribution, κ is a parameter to control the concentration of the distribution to the vector $\tilde{\boldsymbol{w}}$, and C_κ is a normalization constant.

The vMF model (2) renders similarity between $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{w}}$ in a probabilistic sense, and the form (2) is rewritten by using a profile function $\mathtt{f}_e(d;\kappa) = \exp(-\frac{1}{2}\kappa d^2)$ into

$$\mathtt{p}(\tilde{\boldsymbol{x}}; \tilde{\boldsymbol{w}}, \kappa) = C_\kappa \exp(\kappa - \frac{1}{2}\kappa \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\|^2) = C'_\kappa f_e(\|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\|; \kappa). \quad (3)$$

The vMF *similarity* is essentially characterized by $f_e(\|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\|; \kappa)$ and thus we formally define the vMF similarity to substitute for $\cos\theta$ by

$$\phi_e(\cos\theta;\kappa) = 2\frac{\mathtt{f}_e(\|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\|;\kappa) - \mathtt{f}_e(2;\kappa)}{\mathtt{f}_e(0;\kappa) - \mathtt{f}_e(2;\kappa)} - 1 \quad (4)$$

$$= 2\frac{\exp(\kappa\cos\theta) - \exp(-\kappa)}{\exp(\kappa) - \exp(-\kappa)} - 1 \in [-1, 1], \quad (5)$$

where we rescale $f_e(\|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\|;\kappa)$ on $\|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}\| \in [0, 2]$ so that it is compatible with $\cos\theta \in [-1, +1]$. While the parameter κ controls concentration in the original vMF model (2) by $\kappa > 0$, the vMF measuring function (5) accepts various κ even including negative values; $\kappa \in (-\infty, 0) \cup (0, +\infty)$.

As shown in Fig. 2a, by controlling the parameter κ, the vMF similarity (5) exhibits distinctive properties in comparison to the cosine similarity as follows. (1) Larger $\kappa > 0$ induces compact similarity function, sensitively measuring similarity around $\theta = 0$. (2) $\kappa \to 0$ reconstructs the original cosine similarity, $\cos\theta$. (3) Smaller $\kappa < 0$ enlarges the support region of the measuring function beyond cosine. From the classification perspective, the first property is effective for improving intra-class compactness as a regularization. Namely, the vMF similarity ϕ_e with $\kappa > 0$ would reduce the within-class variance by orienting features $\boldsymbol{x}$ toward the classifier $\boldsymbol{w}_c$ to gain substantial similarity. It is noteworthy that the similarity fairly works on all classes without special treatment for the ground-truth class in contrast to the large-margin methods [31, 30, 45, 11] and is directly embedded in logits without additional regularization loss [46, 20]. We will discuss the case of $\kappa < 0$ in Secs. 2.5&3.4.

2.2. t-vMF Similarity

Though the vMF measuring function (5) renders compact similarity, the function contrarily has *light tail* where the similarity score is rapidly approaching -1 even by a bit larger angle θ as shown in Fig. 2bc. Such a *too compact* measuring function might hamper training CNNs since samples on the light tail hardly enjoy back-propagation due to vanishing gradient (Fig. 2c). This bottleneck is derived from the exponential profile function $\mathtt{f}_e(d;\kappa) = \exp(-\frac{1}{2}\kappa d^2)$. Similar discussion can be found in the other literature of t-SNE [42] which considers to match pointwise probability distributions for embedding samples in the lower-dimensional space. In that framework, the shape of probabilistic density function is required to be compact while having a heavy tail for well capturing the discriminative metrics in the original feature space, which is connected with our situation to design similarities.

Thus, we follow the approach of t-SNE [42] that extends SNE [17] by introducing heavy-tailed student-t distribution as an alternative to Gaussian. Considering that the vMF similarity (5) is built upon the exponential profile function $\mathtt{f}_e$, it can be modified by replacing $\mathtt{f}_e$ with the student-t

profile $\mathtt{f}_t(d;\kappa) = \frac{1}{1+\frac{1}{2}\kappa d^2}$ to formulate *t-vMF similarity* by

$$\phi_t(\cos\theta;\kappa) = 2\frac{\mathtt{f}_t(\|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{w}}\|;\kappa)-\mathtt{f}_t(2;\kappa)}{\mathtt{f}_t(0;\kappa)-\mathtt{f}_t(2;\kappa)} - 1 \tag{6}$$

$$= 2\frac{\frac{1}{1+\kappa(1-\cos\theta)} - \frac{1}{1+2\kappa}}{1-\frac{1}{1+2\kappa}} - 1 = \frac{1+\cos\theta}{1+\kappa(1-\cos\theta)} - 1, \tag{7}$$

where $\kappa \in (-\frac{1}{2}, +\infty)$ since $\frac{1}{2}\kappa d^2 > -1$ in $\mathtt{f}_t$ on $0 \le d^2 = (2-2\cos\theta) \le 4$. As shown in Fig. 2b, while the t-vMF similarity (7) is close to the vMF one (5) around $\theta = 0$, it additionally exhibits the following favorable properties. (1) The t-vMF measuring function is heavy-tailed in comparison to vMF (Fig. 2c) so that training CNNs stably proceeds even by larger κ. (2) The similarity (7) can be computed only by simple operation (one-line code) as shown in Algorithm 1 unlike the vMF (5) which depends on an exponential function. (3) $\kappa = 0$ exactly reconstructs the original cosine similarity, $\phi_t(\theta;\kappa=0) = \cos\theta$ without any careful treatment about practical computation.

Algorithm 1 Pseudocode of t-vMF similarity

```
# w: classifier weight vector
# x: input feature vector
# k: kappa parameter
def tvMFsimilarity(w, x, k):
    # Cosine similarity
    # linear: compute inner product
    # normalize: normalize by L2-norm
    cosine = linear(normalize(x),normalize(w))
    # One-line code for t-vMF (7)
    phi = (1+cosine)/(1+k*(1-cosine))-1
    return phi
```

2.3. q-vMF Similarity

These two similarities (5,7) can be viewed in a unified way by means of q-exponential function [41], $\mathtt{f}_q(d;\kappa) = [1-(1-q)\frac{1}{2}\kappa d^2]^{\frac{1}{1-q}}$. The q-exponential function contains the exponential and student-t functions by $q \to 1$ and $q = 2$, respectively. Thus, we can define the q-vMF similarity as

$$\phi_q(\cos\theta;\kappa) = 2\frac{\mathtt{f}_q(\|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{w}}\|;\kappa)-\mathtt{f}_q(2;\kappa)}{\mathtt{f}_q(0;\kappa)-\mathtt{f}_q(2;\kappa)} - 1 \tag{8}$$

$$= 2\frac{[1-(1-q)\kappa(1-\cos\theta)]^{\frac{1}{1-q}} - [1-2(1-q)\kappa]^{\frac{1}{1-q}}}{1-[1-2(1-q)\kappa]^{\frac{1}{1-q}}} - 1, \tag{9}$$

where $\kappa \in (-\frac{1}{2(q-1)}, +\infty)$. In particular, due to the above-mentioned property of the q-exponential function, $q \to 1$ leads to $\phi_q \to \phi_e$ (5) and $q = 2$ produces t-vMF $\phi_e = \phi_t$ (7). Though the computation (9) is more complicated than t-vMF (7), the measuring function is further flexibly controlled by q in addition to κ; the q-vMF of larger q constructs the heavier-tailed similarity beyond t-vMF (Fig. 2b).

2.4. Classifier

The vMF-based similarity is embedded into the following pseudo inner-product in stead of the cosine similarity:

$$\langle \boldsymbol{x}, \boldsymbol{w} \rangle_\phi = \|\boldsymbol{x}\|\|\boldsymbol{w}\|\phi\Big(\frac{\boldsymbol{w}^\top \boldsymbol{x}}{\|\boldsymbol{x}\|\|\boldsymbol{w}\|};\kappa\Big), \tag{10}$$

where $\phi(\cdot;\kappa)$ indicates one of the vMF-based similarities (5,7,9) parameterized by κ (and q for q-vMF).

In the experiments (Sec. 3), we employ a *normalized* classification via L_2-normalization of feature vectors and classifier weights to formulate the cross-entropy loss of

$$\mathtt{l}(\boldsymbol{x}, y) = -\log \frac{\exp\big(s\big\langle \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, \frac{\boldsymbol{w}_y}{\|\boldsymbol{w}_y\|}\big\rangle_\phi\big)}{\sum_{c=1}^{C}\exp\big(s\big\langle \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, \frac{\boldsymbol{w}_c}{\|\boldsymbol{w}_c\|}\big\rangle_\phi\big)} \tag{11}$$

$$= -\log \frac{\exp\big\{s\phi\big(\frac{\boldsymbol{w}_y^\top \boldsymbol{x}}{\|\boldsymbol{w}_y\|\|\boldsymbol{x}\|};\kappa\big)\big\}}{\sum_{c=1}^{C}\exp\big\{s\phi\big(\frac{\boldsymbol{w}_c^\top \boldsymbol{x}}{\|\boldsymbol{w}_c\|\|\boldsymbol{x}\|};\kappa\big)\big\}}, \tag{12}$$

where $\boldsymbol{x}$ and y are a feature vector produced by the penultimate layer and its ground-truth class label, respectively, and we introduce the *trainable* scaling factor s[2] for softmax [36, 18]; s is optimized in an end-to-end manner. The scaling parameter s compensates the norms of $\boldsymbol{w}_c$ and $\boldsymbol{x}$ via $s \approx \|\boldsymbol{w}_c\|\|\boldsymbol{x}\|$ in (1) on the assumption that the classifier weights $\boldsymbol{w}_c$ and the sample features $\boldsymbol{x}$ have consistent norm magnitudes across classes and samples, respectively. Those norm magnitudes are vulnerable to deterioration of the training dataset, such as regarding class imbalance, and thus normalized representation in (12) would be effective for learning on the deteriorated datasets [26, 49]. The normalized classifier (12) also renders favorable feature representation and is applied to various tasks [45, 11, 36, 18, 52, 15]. In the case of $\kappa = 0$, (12) is reduced to the softmax loss based on cosine similarity which is also referred to as vMF loss in [52, 15]; from this viewpoinnt, the proposed vMF similarities (5,7,9) to produce logits in (12) are clearly different from the vMF-based losses [52, 15].

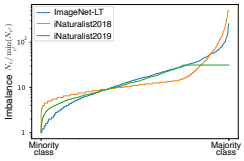
2.5. Discussion

In the end-to-end learning framework, the vMF-based measuring functions (5,7,9) produce compact support similarities (Fig. 2) to reduce within-class variance for effective feature representation; feature $\boldsymbol{x}$ is forced to be within the compact support around the classifier $\boldsymbol{w}_y$ for providing sufficient logit value to minimize the loss (12). It is noteworthy that such regularization is *implicitly* embedded in the proposed similarity without introducing a regularization term [46, 20] into a loss function. The regularization of the proposed similarities contributes to extracting class-specific

[2]We further re-parameterize it by $s = \log(1+\exp(s'))+1 > 1$ with $s' \in \mathbb{R}$ especially for stable training in large-margin methods.

Table 1. Datasets used in the experiments. *Imbalance* is defined by $N_c / \min_{c'}[N_{c'}]$ where N_c is the number of sample at the c-th class.

	ImageNet-LT [32]	iNat2018 [22]	iNat2019 [23]	ImageNet-S/N	ImageNet-SS
# of class	1000	8142	1010	1000	1000
# of samples	115846	437513	265213	115000	50000
Max. *Imbalance*	256	500	31.25	1	1



features shared among intra-class samples for improving generalization performance. The regularization would be effective for training on some poor datasets where the standard approach fails to learn effective features.

The above-mentioned compactness of measuring function is endowed by *positive* parameter value of $\kappa > 0$ in the three types of vMF-based similarities which are distinguished in terms of heaviness at tails (Fig. 2b). The vMF similarity (5) contains light tail on which samples would be less effectively optimized (Fig. 2c). The t-vMF (7) improves it by incorporating the student-t form in a manner similar to t-SNE [42] toward heavy-tail similarity. Those two models are unified by means of the q-exponential into the q-vMF (9) and it can provide further heavier-tailed similarity through tuning the additional parameter q. These similarities are empirically evaluated in Sec. 3.1.1.

We have discussed the effect of $\kappa > 0$ to improve intra-class feature distribution. On the other hand, the model with $\kappa < 0$ have diffrent impact on training CNNs. As shown in Fig. 2a, $\kappa < 0$ enlarges the support angle region in contrast to $\kappa > 0$. Through the competitive learning among classes $\{\tilde{w}_c\}_{c=1}^{C}$ in the softmax loss (12), the similarity of large support leads to enhancing *inter-class* discrimination due to the heavy overlap among similarities of different classes. In other words, $\kappa < 0$ reduces the classifier margins to enhance discriminativity in a similar way to large-margin approach (Fig. 4). Thus, the vMF-based similarities of $\kappa < 0$ would work on *large-scale balanced* datasets which prefer the inter-class discriminativity for improving performance than the regularization of intra-class compactness, since intra-class characteristics could be well modeled by plenty of samples even without regularization. Such effect can be empirically validated in Sec. 3.4.

The t-vMF similarity is slightly connected to the (generalized) student-t kernel [48]. The proposed t-vMF (7), however, results in a clearly different form than the kernel function of student-t and it is favorably parameterized by κ that is interpretable from the viewpoint of similarity compactness; it naturally unifies the cosine similarity as a special case of $\kappa = 0$. As to a kernel function, the Arc-kernel is also proposed in [8] based on the angle θ. It, however, is formulated in a computationally inefficient form while being inferior to ours in terms of compactness and tail.

3. Experimental Results

We apply the proposed method to training CNNs on three types of deteriorated training datasets regarding *imbalanced* classes, *small-scale* and *noisy* labels. It is generally difficult to effectively train deep CNNs in an end-to-end manner on those datasets. The proposed vMF-based similarities naturally impose regularization on the intra-class feature distribution through the softmax cross-entropy loss (12) for improving generalization performance.

Training procedure. We follow the training protocol of [25] by applying SGD optimizer with momentum 0.9, weight decay 10^{-4}, mini-batch size 256 and cosine-learning rate scheduling [33] (initial rate 0.2) over 90 training epochs; during training, the standard data augmentation [16] is applied to input images. The classification performance is measured by top-1 and top-5 error rates (%) through single center-crop evaluation protocol [27].

3.1. Learning on Imbalanced Dataset

While the standard benchmark datasets, such as ImageNet [10], are well balanced in terms of training samples per class category, real-world categories are occasionally distributed by long-tailed distribution, producing *imbalanced* numbers of training samples across classes, as shown in Tab. 1. The CNNs trained on such an imbalanced dataset are accordingly biased toward majority classes while paying less attention to the minorities.

In [25], simple two-stage learning is proposed for the imbalanced learning; a CNN is first trained in the standard way via uniformly sampling training images (mini-batches) and then only the classifier is further finetuned by *balanced* sampling across classes while freezing the feature extractor of the CNN. We follow this simple approach by applying the proposed similarities to the loss (12) of the first stage to optimize feature representation. For fair comparison, at the second stage of finetuning, the simple cosine similarity ($\kappa = 0$) is used in the softmax loss (12) for all the methods that we used in this experiment; at the second stage, the classifier is trained over 30 epochs while keeping the other optimization parameters shown above. Thus, we can evaluate how robust feature representation a method learns against imbalanced datasets by introducing regularization.

3.1.1 Ablation study

We first analyze the proposed methods by training ResNet-10 [16] on ImageNet-LT dataset [32] (Tab. 1).

Types of vMF. In Sec. 2, we proposed three types of vMF-based measuring functions (5,7,9) which are distinguished in terms of tail heaviness (Fig. 2b). To fairly compare the

Table 2. Performance comparison among vMF-based similarities with various κ on ImageNet-LT. We report top-1 error rate (%) with top-5.

κ	0 (cos)	2	4	8	16	32	64	128	256
vMF (5)	61.32 38.44	60.25 37.05	59.16 35.90	58.11 34.30	75.84 53.58	96.85 90.79	96.95 90.94	100.0 100.0	100.0 100.0
t-vMF (7)	61.32 38.44	60.40 37.01	59.17 35.98	58.18 34.47	57.30 32.92	56.49 31.97	56.31 31.78	57.22 32.03	58.66 33.32
q-vMF (9)	61.32 38.44	60.61 37.45	59.96 36.53	59.15 35.65	59.05 35.17	58.35 34.29	58.46 34.28	58.12 33.74	58.01 33.57

Table 3. Comparison to the other measuring function derived from the large-margin and kernel methods [31, 11, 8] on ImageNet-LT.

	$\cos(k\theta)$ [31]			$\cos(\theta+m)$ [11]			Arc-kernel [8]	
param.	$k=2$	4	8	$m=\pi/8$	$\pi/4$	$\pi/2$	$n=1$	$n=2$
Err.	59.67 36.05	61.02 38.00	61.54 38.45	60.93 37.67	60.81 37.45	57.83 34.14	61.63 38.32	60.80 37.75

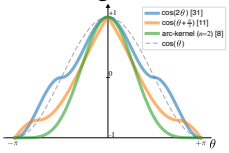
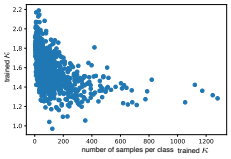


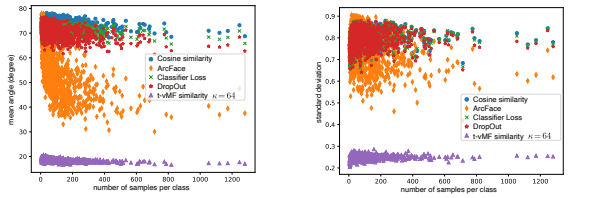
Table 4. Trainable κ in t-vMF (7) on ImageNet-LT.

(a) Two-types of parameterization

	single κ	class-wise $\{\kappa_c\}_{c=1}^{C}$
Err.	61.23 38.21	60.39 37.02
κ	0.37	1.58±0.19

(b) Plot of $\{\kappa_c\}_{c=1}^{1000}$





(a) Angle $\bar{\theta}_c$ to classifier $\boldsymbol{w}_c$ (b) Within-class standard deviation

Figure 3. Statistics of learned features.

tail, we control the parameter κ for respective models so that they exhibit similar compactness around $\theta = 0$; the same κ is applied to both vMF (5) and t-vMF (7), while q-vMF (9) applies half of κ and larger q to provide heavier tail; details are described in the supplementary material.

Performance results are shown in Tab. 2 and we can find the following. (1) Performance is improved by (moderately) larger κ which produces compact shape around $\theta = 0$. (2) The vMF similarity (5) works only with $\kappa < 16$ while degrading performance by $\kappa \geq 16$. The larger κ induces the lighter tail of the measuring function ϕ_e, thereby making it hard to proceed back-propagation on the samples outside of the support of the function (Fig. 2c) as discussed in Sec. 2.2. (3) On the other hand, q-vMF (9) that provides heavier tail contributes to stable learning, though being inferior to t-vMF (7). The heavier tail slightly harms compactness of the measuring function around $\theta = 0$ which is a key characteristic to regularize feature representation. Thus, we can conjecture that the t-vMF similarity (7) is favorable in terms of compactness and heavy tail, stably producing better performance at the larger κ without deteriorating the training of CNN even at very large $\kappa = 256$.

Cosine-based measuring function. The form of cosine similarity has been also discussed mainly in the framework of large-margin methods [31, 30, 45, 11]; the cosine similarity of the ground-truth class is degraded by modifying the form of the similarity. So modified cosine similarity is also applicable to ϕ in (12) for comparison to the proposed t-vMF similarity (7) in our framework toward compact intra-class representation. It should be noted that in this case the cosine similarities are fairly modified on all classes without taking special care of the ground-truth class; the large-margin methods themselves are tested in Sec. 3.1.2.

Tab. 3 shows the performance results by the arc-kernel [8] and the two forms of modified cosine similarities; multiplicative [31, 30] and additive [11] ones. As in Tab. 2, performance is improved by the measuring functions which exhibit compactness around $\theta = 0$. They, however, are less compact in comparison to t-vMF. Arc-kernel [8] has similar shape to ours but it is less compact and light-tailed similarity compared to t-vMF. Therefore, they are inferior to ours. It is also noteworthy that the t-vMF is more computationally efficient (Algorithm 1) than those comparison methods. The comparison result to these methods highlights effectiveness of the proposed t-vMF model for regularization.

Trainable κ. The parameter κ in the t-vMF similarity (7) is pre-fixed as shown in Tab. 2. According to the end-to-end learning principle, it is also possible to *optimize* κ as in the other CNN parameters. There are two conceivable ways of parameterization for κ. One is to introduce *single* trainable κ shared across all the classes, while the other way is to assign κ_c to respective classes and optimize $\{\kappa_c\}_{c=1}^{C}$. The parameter κ in t-vMF (7) is optimized over $\kappa \in [0, +\infty)$ based on the discussion in Sec. 2.5 by applying SGD in the same way as the other CNN parameters.

The performance results and the optimized κ values are also shown in Tab. 4a. Training *single* κ might be impeded by the high imbalance across the majority and minority classes, and thereby κ results in close to 0, pushing the t-vMF similarity toward the ordinary cosine similarity. On the other hand, the class-wise parameterization mitigates the imbalance in training κ_c, which is seperately assigned to each class c, to slightly improves performance. Nonetheless, the performances of the trainable t-vMF models are inferior to that of pre-fixed larger κ (Tab. 2). Train-

Table 5. Performance comparison (error rates %) on various datasets.

	(a) Imbalanced			(b) Small-scale		(c) Noisy
Dataset	ImageNet-LT [32]	iNat2018 [22]	iNat2019 [23]	ImageNet-S	ImageNet-SS	ImageNet-N
CNN	ResNet-10	ResNet-50	ResNet-50	ResNet-10	ResNet-10	ResNet-10
Softmax	61.32 38.44	35.95 17.28	27.23 7.95	55.53 31.58	70.52 48.47	82.34 67.61
L-Softmax [31]	60.27 37.13	35.32 16.77	26.70 7.89	53.41 29.60	65.83 41.74	77.42 58.87
ArcFace [11]	59.46 35.29	33.56 14.73	26.83 8.28	53.95 29.68	65.18 40.69	73.17 48.40
Center Loss [46]	60.82 37.79	35.17 16.94	27.53 7.82	55.11 31.24	70.03 47.72	81.80 66.17
Classifier Loss [20]	60.96 37.81	35.49 16.85	26.93 7.89	55.36 31.55	70.21 48.05	82.19 66.59
Virtual Softmax [7]	61.72 35.23	43.83 20.17	30.36 8.78	60.85 33.30	70.90 43.93	72.40 47.72
DropOut [40]	59.17 35.68	32.20 14.53	26.34 7.46	52.69 28.21	66.41 42.78	75.72 55.56
t-vMF (7) ($\kappa=$ 4)	59.17 35.98	31.57 13.56	25.22 6.70	53.58 29.36	67.32 43.82	77.28 58.53
t-vMF (7) ($\kappa=16$)	57.30 32.92	**28.92** 11.75	25.64 6.53	**52.06** 27.54	**64.77** 40.67	71.46 49.19
t-vMF (7) ($\kappa=64$)	**56.31** 31.78	29.69 11.90	**25.08** 7.10	52.51 28.09	65.73 40.86	**69.19** 45.66

Table 6. Detailed performance on imbalanced learning. We follow [25] to split classes into *Many*, *Medium* and *Few* categories.

	ImageNet-LT			iNaturalist2018		
	Many	Medium	Few	Many	Medium	Few
Softmax	47.16	66.10	84.29	28.15	34.67	39.59
L-Softmax	46.23	64.85	83.66	28.82	33.70	39.06
ArcFace	46.06	63.96	81.28	27.51	32.25	36.78
CenterLoss	46.85	65.28	84.41	28.46	33.47	39.05
ClassifierLoss	47.40	65.02	84.76	28.35	34.00	39.23
VirtualSoftmax	50.25	65.58	80.35	35.11	43.34	46.71
DropOut	45.90	63.20	82.22	**25.06**	30.74	35.92
t-vMF ($\kappa = 4$)	45.23	63.41	83.43	25.53	30.20	34.85
t-vMF ($\kappa = 16$)	**43.92**	61.20	81.16	25.85	**27.93**	31.08
t-vMF ($\kappa = 64$)	44.83	**59.36**	**77.74**	28.35	29.33	**30.46**

ing κ proceeds in cooperation with feature representation learning, and thus the trained κ reflects the characteristics of feature distribution derived from imbalanced data distribution; actually, we can see in Tab. 4b a slight trend that the minor classes receives larger κ while the majority ones are assigned smaller κ, which reflects the small variance in the minority classes and the large variance in the majority classes (Fig. 3). From the regularization viewpoint, however, κ should be assigned in a resistant manner against the imbalanced distribution. Thus, to remove the statistics derived from the imbalanced data, the parameter κ is *prefixed* by larger value.

Feature distribution. Fig. 3 shows the statistics of learned features in comparison to those by cosine similarity. In Fig. 3a, the mean angle $\bar{\theta}_c = E_{i|y_i=c}[\arccos(\tilde{\boldsymbol{w}}_c^\top \tilde{\boldsymbol{x}}_i)]$ between the classifier weight $\boldsymbol{w}_c$ and features $\boldsymbol{x}$ is shown for respective classes which are characterized by the number of samples per class. The standard cosine similarity provides larger angles due to the large support of measuring function (Fig. 2); actually, they are $70 \sim 80$ degrees. On the other hand, the t-vMF of $\kappa = 64$ contributes to orienting the features toward the classifier in virtue of the compact support measuring function ϕ_t (7). Accordingly, the within-class variance is reduced as shown in Fig. 3b which depicts the within-class standard deviation across classes.

3.1.2 Comparison to other methods

The proposed method is then compared with the other methods; we apply the regularization methods which are categorized into three groups, large-margin methods [31, 11], additional regularization losses [46, 20] and the others including DropOut [40] and virtual softmax loss [7]. These comparison methods are incorporated into the two-stage learning scheme [25] as in ours by modifying the loss (12) based on cosine similarity ($\kappa = 0$); the large-margin losses [31, 11] and virtual softmax [7] substitute for the softmax loss, the regularization loss [46, 20] is added to the softmax loss, and the DropOut is applied to feature $\boldsymbol{x}$. The hyper-parameters in those methods are determined so as to produce the best performance, for fair comparison; the set of hyper-parameters in those methods are detailed in the supplementary material. These methods are evaluated on ImageNet-LT [32], iNaturalist2018 [22] and iNaturalist2019 [23] (Tab. 1).

Performance results in Tab. 5a demonstrate that by introducing regularization into the feature representation, the performance on imbalanced classification is favorably improved. In particular, the comparison to classifier loss [20] highlits the effectiveness of our method. The classifier loss [20] is proposed to reduce the deviation around the classifier $\boldsymbol{w}_c$ in the framework of center loss [46] as

$$\mathtt{l}_{clsloss}(\boldsymbol{x}, y) = \mathtt{l}(\boldsymbol{x}, y) + \lambda \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{w}}_y\|_2^2, \qquad (13)$$

where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{w}}_y$ are normalized feature and classifier weight, respectively, and λ is a regularization parameter. The method is closely related to ours which also reduces such a deviation by *compact* measuring function ϕ_t (7) with $\kappa > 0$. As shown in Tab. 5, the proposed t-vMF is superior to the other regularization methods including the classifier loss [20]. This result validates our approach to implicitly embed the regularization into the similarity, especially compared to the additional regularization loss [20]. The performance by t-vMF is competitive to the reported ones; for

Table 7. Performance results of t-vMF on ImageNet dataset [10] by ResNet-50 [16].

κ	-0.45	-0.3	-0.15	0 (cos)	2	4	8	16	32	ArcFace [11]
Err.	22.73 6.49	22.62 6.46	22.81 6.67	23.05 6.58	22.90 6.57	22.99 6.76	23.56 6.80	23.64 6.95	23.78 7.12	23.28 7.33

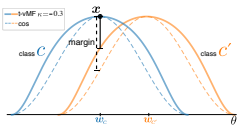


Figure 4. t-vMF of $\kappa < 0$.

ImageNet-LT, 64.4 [32], 58.4 [26], 58.2 [25], and for iNaturalist2018, 38.88 [9], 32.00 [6], 34.1 (30.5 by 200-epoch training) [25]; t-vMF also produces 28.46 by 200 epochs.

The t-vMF similarity of larger κ highly regularizes features and thus effectively contributes to performance improvement in severely imbalanced learning of ImageNet-LT and iNaturalist2018; the detailed performance comparison is shown in Tab. 6. On the other hand, the rather smaller κ which imposes weak regularization also works for iNaturalist2019 of marginal imbalance (Tab. 1). Thus, the proposed method copes with various degrees of imbalance through κ.

3.2. Learning on Small-Scale Dataset

We then evaluate the proposed method on small-scale training dataset. Two small-scale datasets are constructed by sampling sub-set of ImageNet [10]; ImageNet-S is built so as to be the same-scale as ImageNet-LT [32], and ImageNet-SS is defined as *smaller*-scale by further halving ImageNet-S, as shown in Tab. 1.

The small-scale issue could also be tackled such as by data augmentation techniques [12, 50]. In this experiment, we address the issue by regularizing the feature representation to improve generalization performance in the same way as Sec. 3.1. Those two approaches are complementary to each other and thus their combination could work; our future work includes to explore the practical combination.

The performance results are shown in Tab. 5b. Similarly to the imbalanced learning in Tab. 5a, the regularization methods also improve performance on the small-scale learning; especially, the t-vMF of $\kappa = 16$ performs well in comparison to the others.

3.3. Learning on Noisy Annotation

The proposed method is tested on the deteriorated situation where annotations of samples are less correct, *i.e.*, noisy labels. Annotating image samples in detail is such a laborious process that wrong labels could be frequently injected into the training samples in real-world situations. Detecting the wrong label is the classification process itself and thus it is hard to eliminate those label noise in advance. Incorrectly labeled samples confuse CNNs thereby disturbing the training. We here evaluate how much robustly the regularization methods learn CNNs against the noisy labels; note that image (content) quality is not degraded. For that purpose, we inject label noise into ImageNet-S (Tab. 1) to construct ImageNet-N by randomly switching the labels of samples other than ImageNet-SS into wrong ones; only the samples in the set of ImageNet-SS have correct labels.

The performance results are shown in Tab. 5c. By mixing the noisy samples with clean ones, the performance is degraded in comparison to those of ImageNet-SS in Tab. 5b which contains the same number of correct training samples as ImageNet-N. The proposed method enhances the consistency among intra-class samples through regularization imposed by larger κ to favorably improve performance; t-vMF of $\kappa = 64$ produces superior performance being only 3 point reduction from ImageNet-SS, in contrast to the others most of which degrade performance by about 10 points.

3.4. Learning on Healthy Dataset

We have so far discussed and analyzed the proposed method with $\kappa > 0$ on the deteriorated training datasets. Conversely, the method is here evaluated on *healthy* dataset with $\kappa < 0$. Tab. 7 shows the performance results on ImageNet dataset [10]. One can see that the normalized classifier in (12) effectively improves performance in comparison to the standard linear classifier (23.85 (top-1)/ 7.13 (top-5) reported in [16]). In contrast to the deteriorated situation, the larger $\kappa > 0$ slightly degrades performance since the regularization on intra-class would be less effective for plenty of samples which well model the intra-class structure. On the other hand, the t-vMF of smaller $\kappa < 0$ improves the performance of the original cosine similarity ($\kappa = 0$). As shown in Fig. 4 and Sec. 2.5, the smaller $\kappa < 0$ contributes to enhance *inter*-class discrimination among large number of samples; it is superior even to the large-margin method [11].

These results demonstrate the flexibility of the t-vMF similarity (7) such that it can cope with various types of training datasets from deteriorated to healthy one through the parameter κ. Our thorough analysis about the effect of κ would help to tune κ qualitatively based on the target learning situation or quantitatively such as via cross-validation.

4. Conclusion

We have proposed a novel similarity for improving intra-class feature representation. In contrast to the standard cosine similarity which has broad support region, the proposed method built on vMF model is formulated in a compact similarity function parameterized by κ. By further incorporating the student-t model, the method is equipped with compact support as well as heavy tail for effectively regularizing intra-class feature distribution. In the experiments on image classification using deteriorated training datasets, the proposed method improves performance of CNNs, being superior to the other regularization methods.

References

[1] Jimmy Le Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv,* 1607.06450, 2016. 2

[2] 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(12):1345–1382, 2005. 2, 3

[3] Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *WACV*, pages 638–647, 2019. 2

[4] Björn Barz and Joachim Denzler. Deep learning on small datasets without pre-training using cosine loss. In *WACV*, pages 1371–1380, 2020. 2

[5] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *NeurIPS*, 1994. 2

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 8

[7] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *NeurIPS*, page 1946–1956, 2018. 7

[8] Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *NeurIPS*, pages 342–350, 2009. 5, 6

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 8

[10] 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. 1, 5, 8

[11] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 1, 2, 3, 4, 6, 7, 8

[12] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv,* 1708.04552, 2017. 1, 2, 8

[13] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990. 3

[14] Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *ICML*, 2014. 2

[15] Md. Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv,* 1706.04264, 2017. 2, 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5, 8

[17] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *NeurIPS*, 2002. 3

[18] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: The marginal value of training the last weight layer. In *ICLR*, pages 5822–5830, 2018. 2, 4

[19] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv,* 1704.04861, 2017. 1

[20] Shaoli Huang and Dacheng Tao. All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning. *arXiv,* 1911.12476, 2019. 1, 2, 3, 4, 7

[21] Jyh-Jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D. Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *ICCV*, 2019. 2

[22] iNatrualist. The inaturalist 2018 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2018, 2018. 5, 7

[23] iNatrualist. The inaturalist 2019 competition dataset. https://www.kaggle.com/c/inaturalist-2019-fgvc6, 2019. 5, 7

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015. 2

[25] 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. 5, 7, 8

[26] BYUNGJU KIM and JUNMO KIM. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020. 3, 4, 8

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1, 5

[28] Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *NeurIPS*, pages 950–957, 1991. 2

[29] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *CVPR*, pages 2682–2690, 2019. 1, 2

[30] 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. 2, 3, 6

[31] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516, 2016. 1, 2, 3, 6, 7

[32] 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. 5, 7, 8

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[34] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *ICANN*, pages 382–391, 2018. 2

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

[36] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *CVPR*, pages 5822–5830, 2018. 2, 4

[37] Xiangju Qin, Pádraig Cunningham, and Michael Salter-Townshend. Online trans-dimensional von mises-fisher mixture models for user profiles. *Journal of Machine Learning Research*, 17(200):1–51, 2016. 2

[38] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, 2016. 2

[39] Suvrit Sra. Directional statistics in machine learning: a brief review. *arXiv,* 1605.00316, 2016. 2

[40] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 1, 2, 7

[41] Constantino Tsallis. What are the numbers that experiments provide? *Quimica Nova*, 17(468), 1994. 4

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 3, 5

[43] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998. 2

[44] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 2

[45] 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. 2, 3, 4, 6

[46] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016. 1, 2, 3, 4, 7

[47] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *WACV*, 2018. 2

[48] Rui Yang and Yukio Ohsawa. Applying the heavy-tailed kernel to the gaussian process regression for modeling point of sale data. In *ICANN*, 2017. 5

[49] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv,* 2001.01385, 2020. 3, 4

[50] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1, 2, 8

[51] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 1

[52] Xuefei Zhe, Shifeng Chen, and Hong Yan. Directional statistics-based deep metric learning for image classification and retrieval. *Pattern Recognition*, 93(9):113–123, 2019. 2, 4