Supplementary Material:
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

A. Comparison of three types of vMF-based similarities

This section supplements the experimental result in Sec.3.1.1 and Tab.2 of the main manuscript by minutely comparing three types of proposed similarities which are formulated as follows.

$$\text{vMF: } \phi_e(\cos\theta;\kappa) = 2\frac{\exp(\kappa\cos\theta)-\exp(-\kappa)}{\exp(\kappa)-\exp(-\kappa)} - 1, \tag{A.1}$$

$$\text{t-vMF: } \phi_t(\cos\theta;\kappa) = \frac{1+\cos\theta}{1+\kappa(1-\cos\theta)} - 1, \tag{A.2}$$

$$\text{q-vMF: } \phi_q(\cos\theta;\kappa,q) = 2\frac{[1-(1-q)\kappa(1-\cos\theta)]^{\frac{1}{1-q}} - [1-2(1-q)\kappa]^{\frac{1}{1-q}}}{1-[1-2(1-q)\kappa]^{\frac{1}{1-q}}} - 1, \tag{A.3}$$

where κ is the concentration parameter and q is the parameter of q-exponential only for q-vMF. As discussed in Sec.2, those three methods are distinguished mainly in terms of *tail* of the similarity measuring functions. To clarify the relationship between performance and the tail, we show the performance comparison with various parameters in Tab. A as well as the shapes of similarity measuring functions in Fig. A.

B. Comparison to Arc-kernel

As shown in Tab.3, the arc-kernel [2] is similar to ours in terms of shape of similarity measuring function; the actual form of the arc-kernel is given by

$$\text{Arc-kernel: } \phi_a(\theta;n) \propto (-1)^n(\sin\theta)^{2n+1}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)^n\left(\frac{\pi-\theta}{\sin\theta}\right), \tag{B.1}$$

where $\propto$ indicates to apply the standardization into $[-1,+1]$, and the actual computation forms are given by

$$\phi_a(\theta;n=2) \propto \frac{3}{2}\sin(2\theta) + (\pi-\theta)\{2+\cos(2\theta)\}, \tag{B.2}$$

$$\phi_a(\theta;n=4) \propto 40\sin(2\theta) + \frac{25}{4}\sin(4\theta) + 3(\pi-\theta)\{18+16\cos(2\theta)+\cos(4\theta)\}, \tag{B.3}$$

$$\begin{aligned}\phi_a(\theta;n=8) \propto & 444528\sin(2\theta) + 234612\sin(4\theta) + 32112\sin(6\theta) + \frac{6849}{8}\sin(8\theta) \\ & + 315(\pi-\theta)\{2450+3136\cos(2\theta)+784\cos(4\theta)+64\cos(6\theta)+\cos(8\theta)\}.\end{aligned} \tag{B.4}$$

Compared to our t-vMF (A.2), however, the arc-kernel is less-flexibly formulated due to the discrete parameter of order n (B.1) and the similarity measuring function is less compact with light tail as shown in Fig. B. Besides, the computational forms of the arc-kernel (B.2-B.4) are complicated and inefficient in comparison to t-vMF (A.2). In term of the classification performance, it is inferior to ours as shown in Tab. B.

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

vMF: κ	0 (cos)	2	4	8	16	32	64	128	256
t-vMF: κ	0 (cos)	2	4	8	16	32	64	128	256
q-vMF: (κ, q)	0 (cos)	(1,4)	(2,8)	(4,12)	(8,16)	(16,32)	(32,32)	(64,32)	(128,32)
vMF (A.1)	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 (A.2)	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 (A.3)	61.32 38.44	60.61 37.45	59.96 36.53	59.15 35.65	59.05 35.17	60.62 36.32	60.89 36.36	60.51 35.96	60.86 36.14

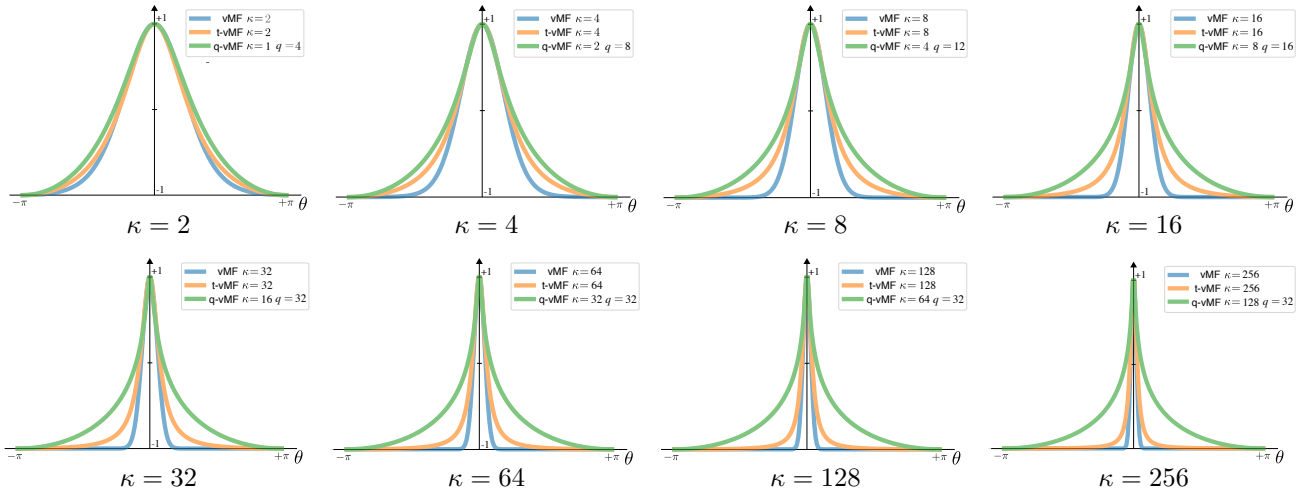

Figure A. Comparison of similarity measuring functions used in Tab. A. The parameters (κ, q) of q-vMF are tuned so as to produce the similar shape around $\theta = 0$ to vMF/t-vMF while having heavier tails.

Table B. Performance comparison to arc-kernel [2] on ImageNet-LT.

t-vMF: κ	2	4	8
arc-kernel: n	2	4	8
t-vMF (A.2)	60.40 37.01	59.17 35.98	58.18 34.47
arc-kernel [2]	60.80 37.75	60.05 37.10	58.93 35.54

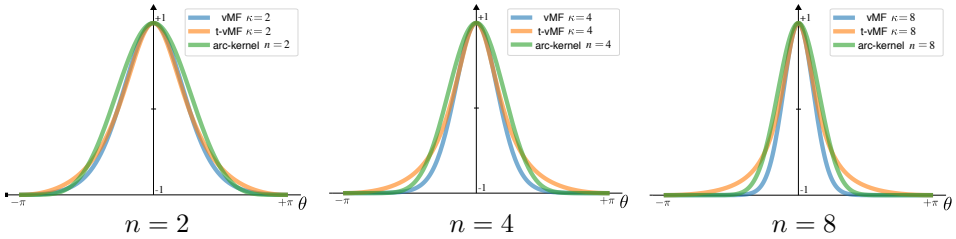

Figure B. Arc-kernels [2] with various orders n.

C. Hyper-parameters of comparison methods

We prefix the set of hyper-parameters of the comparison methods [7, 3, 9, 5, 1, 8] used in Tab.5 based on the respective papers and then report the best performance among them for fair comparison.

Large-margin methods. We apply large-margin softmax (L-softmax) [7], the representative large-margin method, as well as the sophisticated method of ArcFace [3]. L-softmax [7] equipped with $k = 2$ in $\cos(k\theta)$ is mixed up with the standard cosine similarity $\cos\theta$ by the convex mixing weight of $\frac{1}{1+1000\beta^t}$ during the training in order to gradually enhance the effect of large-margin; we set $\beta \in \{0.9995, 0.9999, 0.99999\}$. For ArcFace [3] which degrades the logit of ground-truth class by $\cos(\theta + m)$, we apply $m \in \{0.1, 0.2, 0.5\}$.

Regularization losses. The softmax cross-entropy loss can be accompanied by regularization losses of center loss [9] and

Table C. Performance results on the small-scale datasets. The lower rows report the performance of the combination methods with DropOut [8], while the upper rows are the same as those of Tab.5.

	Small-scale	
Dataset	ImageNet-S	ImageNet-SS
CNN	ResNet-10	ResNet-10
Softmax	55.53 31.58	70.52 48.47
L-Softmax [7]	53.41 29.60	65.83 41.74
ArcFace [3]	53.95 29.68	65.18 40.69
Center Loss [9]	55.11 31.24	70.03 47.72
Classifier Loss [5]	55.36 31.55	70.21 48.05
Virtual Softmax [1]	60.85 33.30	70.90 43.93
DropOut [8]	52.69 28.21	66.41 42.78
t-vMF (A.2) ($\kappa = 16$)	52.06 27.54	64.77 40.67
DropOut+		
L-Softmax [7]	51.35 27.43	65.55 41.55
ArcFace [3]	53.72 29.00	64.67 39.88
Center Loss [9]	52.14 28.22	65.97 42.22
Classifier Loss [5]	52.29 28.10	66.09 42.17
Virtual Softmax [1]	62.49 34.77	71.89 45.39
t-vMF (A.2) ($\kappa = 16$)	50.98 26.64	63.00 37.97

classifier loss [5] with the weight parameter λ; it is set to $\lambda \in \{0.1, 0.01, 0.001\}$ for those two types of additional regularization losses.

Others. The virtual softmax loss [1] favorably contains no hyper-parameter to be tuned. The method of DropOut [8] is applied to the (final) feature representation $\boldsymbol{x}$ with the drop-out ratio of $p \in \{0.1, 0.2, 0.3\}$ based on the analysis [6].

D. Combination with DropOut

As shown in Tab.5, DropOut works well for learning CNNs on the *small-scale* dataset due to the simple yet effective feature perturbation. In the small-scale situation, we can expect that such a perturbation in input features $\boldsymbol{x}$ is compatible with the regularization to reduce within-class variance like ours. The DropOut perturbation would provide *adversarial* regularization [4] for our methods from the viewpoint of feature representation; the feature perturbation might be regarded as enlarging within-class variance *adversarially* to our regularization.

Performance results by the combination methods with the DropOut are shown in Tab. C. While the regularization losses [9, 5] to reduce the within-class variance via additional loss term is improved by the DropOut, the proposed method of t-vMF (A.2) is also further improved to produce better performance.

References

[1] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *NeurIPS*, page 1946–1956, 2018. 2, 3

[2] Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *NeurIPS*, pages 342–350, 2009. 1, 2

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

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv,* 1412.6572, 2014. 3

[5] 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. 2, 3

[6] 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. 3

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

[8] 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. 2, 3

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