Supplementary Material of Unleash the Power of Local Representations: Feature Calibration and Adaptive Metric for Few-Shot Learning

Shi Tang[1], **Chaoqun Chu**[1], **Guiming Luo**[1*], **Xinchen Ye**[2], **Zhiyi Xia**[1], **Haojie Li**[2]

[1]School of Software, Tsinghua University

[2]International School of Information Science&Engineering, Dalian University of Technology

The supplementary material of the paper titled "Unleash the Power of Local Representations: Feature Calibration and Adaptive Metric for Few-Shot Learning" is organized as follows:

- Section A presents the proof of Eq. (4) of the main text;
- Section B describes our experimental setup in detail;
- Section C shows some additional experimental results, including cross-domain experiments (Section C.1), some analysis on computational time (Section C.2) and more visualized transport matrices (Section C.3) as a supplement to Figure 7 of the main text.

A Proof of Eq. (4)

Let $\mathbf{z} = [z_1, z_2, ..., z_{n_c}] \in \mathbb{R}^{1\times n_c}$ denote the network output for a local patch where n_c is the total number of base classes and z_i represents the logit of the i-th base class. Denote the softmax function as σ and the probabilities of the i-th binary classification as $\mathbf{b}_i = [p_i, p_{\neg i}]$ where

$$p_i = \frac{\exp(z_i)}{\sum_{j=i}^{n_c}\exp(z_j)}, \quad p_{\neg i} = \frac{\sum_{k=i+1}^{n_c}\exp(z_k)}{\sum_{j=i}^{n_c}\exp(z_j)}$$

Using the superscripts $\mathcal{T}$ and $\mathcal{S}$ to mark the variables calculated using the output of the teacher $\mathbf{z}^{\mathcal{T}}$ and that of the student $\mathbf{z}^{\mathcal{S}}$, respectively, we want to prove that

$$KL(\sigma(\mathbf{z}^{\mathcal{T}})||\sigma(\mathbf{z}^{\mathcal{S}})) = \sum_{i=1}^{n_c-1}\mathbf{w}_i \cdot KL(\mathbf{b}_i^{\mathcal{T}}||\mathbf{b}_i^{\mathcal{S}}),$$
$$\text{where } \mathbf{w}_i = \frac{\sum_{k=i}^{n_c}\exp(z_k^{\mathcal{T}})}{\sum_{j=1}^{n_c}\exp(z_j^{\mathcal{T}})}$$

Given the definition of $p_i, p_{\neg i}$, we need to additionally introduce the ordinary probability on full set q_i as

$$q_i = \frac{\exp(z_i)}{\sum_{j=1}^{n_c}\exp(z_j)}$$

We have

$$\begin{aligned} p_i &= \frac{\exp(z_i)}{\sum_{j=i}^{n_c}\exp(z_j)} \\ &= \frac{\exp(z_i)}{\sum_{j=1}^{n_c}\exp(z_j)} \cdot \frac{\sum_{j=1}^{n_c}\exp(z_j)}{\sum_{j=i}^{n_c}\exp(z_j)} \\ &= \frac{q_i}{\sum_{k=i}^{n_c} q_k} \end{aligned} \tag{A}$$

$$\mathbf{w}_i = \sum_{k=i}^{n_c} q_k^{\mathcal{T}}$$

$$q_i^{\mathcal{T}} = \mathbf{w}_i \cdot p_i^{\mathcal{T}} \tag{B}$$

Therefore we have

$$p_{\neg i} = 1 - p_i = \frac{\sum_{k=i+1}^{n_c} q_k}{\sum_{k=i}^{n_c} q_k}$$

$$\sum_{k=i+1}^{n_c} q_k = p_{\neg i} \cdot \sum_{k=i}^{n_c} q_k$$

$$\sum_{k=i+1}^{n_c} q_k^{\mathcal{T}} = \mathbf{w}_i \cdot p_{\neg i}^{\mathcal{T}} \tag{C}$$

From the above equation, it can be concluded that

$$\begin{aligned} \sum_{k=i}^{n_c} q_k =& p_{\neg(i-1)} \cdot \sum_{k=i-1}^{n_c} q_k \\ =& (\prod_{k=1}^{i-1} p_{\neg k}) \cdot (\sum_{k=1}^{n_c} q_k) \\ =& \prod_{k=1}^{i-1} p_{\neg k} \end{aligned} \tag{D}$$

*Corresponding author, gluo@tsinghua.edu.cn.

Preprint. Under review.

The KL-Divergence can be reformulated as

$$
\begin{aligned}
& KL(\sigma(\mathbf{z}^{\mathcal{T}})||\sigma(\mathbf{z}^{\mathcal{S}})) && \text{(E)} \\
&= \sum_{i=1}^{n_c} q_i^{\mathcal{T}} \log \frac{q_i^{\mathcal{T}}}{q_i^{\mathcal{S}}} \\
&= \sum_{i=1}^{n_c} q_i^{\mathcal{T}} (\log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + \log \frac{\sum_{k=i}^{n_c} q_k^{\mathcal{T}}}{\sum_{k=i}^{n_c} q_k^{\mathcal{S}}}) && \text{(Eq. (A))} \\
&= \sum_{i=1}^{n_c} q_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + \sum_{i=1}^{n_c} q_i^{\mathcal{T}} \log (\prod_{k=1}^{i-1} \frac{p_{\neg k}^{\mathcal{T}}}{p_{\neg k}^{\mathcal{S}}}) && \text{(Eq. (D))} \\
&= \sum_{i=1}^{n_c} \mathbf{w}_i \cdot p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + \sum_{i=1}^{n_c} \sum_{k=1}^{i-1} q_i^{\mathcal{T}} \log \frac{p_{\neg k}^{\mathcal{T}}}{p_{\neg k}^{\mathcal{S}}} && \text{(Eq. (B))} \\
&= \sum_{i=1}^{n_c} \mathbf{w}_i \cdot p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + \sum_{k=1}^{n_c-1} \sum_{i=k+1}^{n_c} q_i^{\mathcal{T}} \log \frac{p_{\neg k}^{\mathcal{T}}}{p_{\neg k}^{\mathcal{S}}} \\
&= \sum_{i=1}^{n_c} \mathbf{w}_i \cdot p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + \sum_{k=1}^{n_c-1} \mathbf{w}_k \cdot p_{\neg k}^{\mathcal{T}} \log \frac{p_{\neg k}^{\mathcal{T}}}{p_{\neg k}^{\mathcal{S}}} && \text{(Eq. (C))} \\
&= \sum_{i=1}^{n_c-1} \mathbf{w}_i \cdot (p_i^{\mathcal{T}} \log \frac{p_i^{\mathcal{T}}}{p_i^{\mathcal{S}}} + p_{\neg i}^{\mathcal{T}} \log \frac{p_{\neg i}^{\mathcal{T}}}{p_{\neg i}^{\mathcal{S}}}) \\
&= \sum_{i=1}^{n_c-1} \mathbf{w}_i \cdot KL(\mathbf{b}_i^{\mathcal{T}}||\mathbf{b}_i^{\mathcal{S}})
\end{aligned}
$$

B Detailed Experimental Setup

Following the "pretraining + episodic training" paradigm, the training process of our method can be divided into two stages. For the pretraining stage, we pre-train the encoder in the form of self-distillation based on the proxy task of standard multi-classification on D_{base} and select the model with the highest validation accuracy. For the episodic training stage, each epoch involves 50 iterations with a batch size of 4. We first pre-train RM with the parameters of the encoder fixed. Then, the parameters of both the encoder and RM are optimized jointly. Globally, we set $n_p = 4$, $m = 0.999$, and $n = 25$. The cropped patches are resized to 84×84 before being embedded. For evaluation, we randomly sample 5000/600 episodes for testing and report the average accuracy with the 95% confidence interval for 1-shot/5-shot experiment following (Zhang et al. 2022). In the following, we describe our detailed experimental setup according to different benchmarks:

(1) ***mini*ImageNet.** The encoder is first pre-trained for 360 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. $\mathcal{L}_{SKD}$ will not be used for the first 120 epochs to ensure the teacher has well-converged before being used. For the latter 240 epochs, the learning rate is set to 0.01, and λ is set to 0.1. Then, in an episodic manner, RM is pre-trained for 100 epochs with the parameters of the encoder fixed, in which the Adam optimizer with a weight decay of $5e$-4 is adopted. The learning rate starts from $1e$-3 and decays by 0.1 at epoch 60 and 90. Finally, all the parameters will be optimized jointly for another 100 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. The learning rate starts from $5e$-4 and decays by 0.5 every 10 epochs.

(2) ***tiered*ImageNet.** The encoder is first pre-trained for 240 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. $\mathcal{L}_{SKD}$ will not be used for the first 120 epochs to ensure the teacher has well-converged before being used. For the latter 120 epochs, the learning rate is set to 0.001, and λ is set to 0.05. Then, in an episodic manner, RM is pre-trained for 100 epochs with the parameters of the encoder fixed, in which the Adam optimizer with a weight decay of $5e$-4 is adopted. The learning rate starts from $1e$-3 and decays by 0.1 at epoch 60 and 90. Finally, all the parameters will be optimized jointly for another 100 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. The learning rate starts from $1e$-4 and decays by 0.5 every 10 epochs.

(3) **CUB-200-2011.** Each image is first cropped with the provided human-annotated bounding box as many previous works (Triantafillou, Zemel, and Urtasun 2017; Liu et al. 2022; Zhang et al. 2022). The encoder is first pre-trained for 360 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. $\mathcal{L}_{SKD}$ will not be used for the first 120 epochs to ensure the teacher has well-converged before being used. For the latter 240 epochs, the learning rate is set to 0.03, and λ is set to 0.5. Then, in an episodic manner, RM is pre-trained for 100 epochs with the parameters of the encoder fixed, in which the Adam optimizer with a weight decay of $5e$-4 is adopted. The learning rate starts from $1e$-3 and decays by 0.1 at epoch 60 and 90. Finally, all the parameters will be optimized jointly for another 100 epochs where the SGD optimizer with a momentum of 0.9 and a weight decay of $5e$-4 is adopted. The learning rate starts from $1e$-3 and decays by 0.5 every 10 epochs.

C Additional Experimental Results

C.1 Cross-Domain Experiments

For the cross-domain setting which poses a greater challenge for novel-class generalization, we perform an experiment where models are trained on *mini*Imagenet and evaluated on CUB-200-2011 following the setups in (Chen et al. 2019). The cross-domain setting allows us to better evaluate the models' ability to handle novel classes with significant domain differences from the base classes, due to the large domain gap. As a result, it better reflects novel-class generalization. As shown in Table A, our method outperforms the previous state-of-the-art approach, demonstrating the superiority of our method in improving novel-class generalization.

C.2 Analysis on Computational Time

Although RM will inevitably bring additional time overhead during inference as a parameterized module, the proposed AM still costs less time for measuring two feature sets compared to EMD since the solving of the OT problem is accel-

Method	1-shot	5-shot
ProtoNet† (Snell, Swersky, and Zemel 2017)	50.01 ± 0.82	72.02 ± 0.67
MatchNet† (Vinyals et al. 2016)	51.65 ± 0.84	69.14 ± 0.72
cosine classifier (Chen et al. 2019)	44.17 ± 0.78	69.01 ± 0.74
linear classifier (Chen et al. 2019)	50.37 ± 0.79	73.30 ± 0.69
KNN (Li et al. 2019)	50.84 ± 0.81	71.25 ± 0.69
DeepEMD v2 (Zhang et al. 2022)	$\underline{54.24 \pm 0.86}$	$\underline{78.86 \pm 0.65}$
FCAM (ours)	$\mathbf{58.20 \pm 0.30}$	$\mathbf{80.92 \pm 0.65}$

† results are reported in (Zhang et al. 2022). The second best results are underlined.

Table A: Cross-domain experiments (*mini*ImageNet→CUB) following the setting of (Chen et al. 2019). Average 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals.

erated. Specifically, the introduced entropic regularization makes the OT problem a strictly convex problem (Cuturi 2013). Thus, it can be solved by the Sinkhorn-Knopp algorithm (Sinkhorn and Knopp 1967) which is known to have a linear convergence (Franklin and Lorenz 1989; Knight 2008). We conduct an experiment on an RTX-3090 (Linux, PyTorch 3.6) using the same $10,000$ randomly sampled episodes to compare the time cost empirically. The average time spent to process an episode is reported in Table B. It can be seen that our Adaptive Metric (dSD + RM) spends way less time than EMD (26.33% faster) even in the presence of a parameterized module, demonstrating its superiority in both accuracy and speed.

Metric	Average time per episode (ms)
EMD	378.42
Adaptive Metric (ours)	**278.78**

Table B: Time spent processing an episode for methods with different metrics. 9 patches are used to represent a sample.

C.3 Visualization of Solved Transport Matrices

For Figure 7 of the main text, we provide more results in Figure A.

For sets consisting of similar local patches, the transport matrices solved by EMD (Figure A (a)) tend to be very sparse, which is not a desired property because it tries to match a patch with few "most" similar opposite patches, neglecting the fact that the opposite patches are homogeneous. In contrast, dSD (Figure A (b)) generates smoother transport matrices. By allowing "one-to-many" matching, it enables a comprehensive utilization of opposite patches and reduces the dependency on a few opposite patches.

By making ε a learnable parameter, RM can control the smoothness of the transport matrix self-adaptively. For sets consisting of similar local patches, RM produces a relatively larger ε, resulting in a smoother transport matrix (Figure A (b)). While for sets consisting of dissimilar local patches, a relatively smaller ε is predicted, making the transport matrix moderately sparse (Figure A (c)). RM makes it possible for our method to handle various local feature sets by realizing an Adaptive Metric.

References

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; and Huang, J.-B. 2019. A Closer Look at Few-shot Classification. In *International Conference on Learning Representations*.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26.

Franklin, J.; and Lorenz, J. 1989. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114-115: 717–735.

Knight, P. A. 2008. The Sinkhorn–Knopp Algorithm: Convergence and Applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1): 261–275.

Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; and Luo, J. 2019. Revisiting Local Descriptor Based Image-To-Class Measure for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.; Zhang, W.; Xiang, C.; Zheng, T.; Cai, D.; and He, X. 2022. Learning To Affiliate: Mutual Centralized Learning for Few-Shot Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14411–14420.

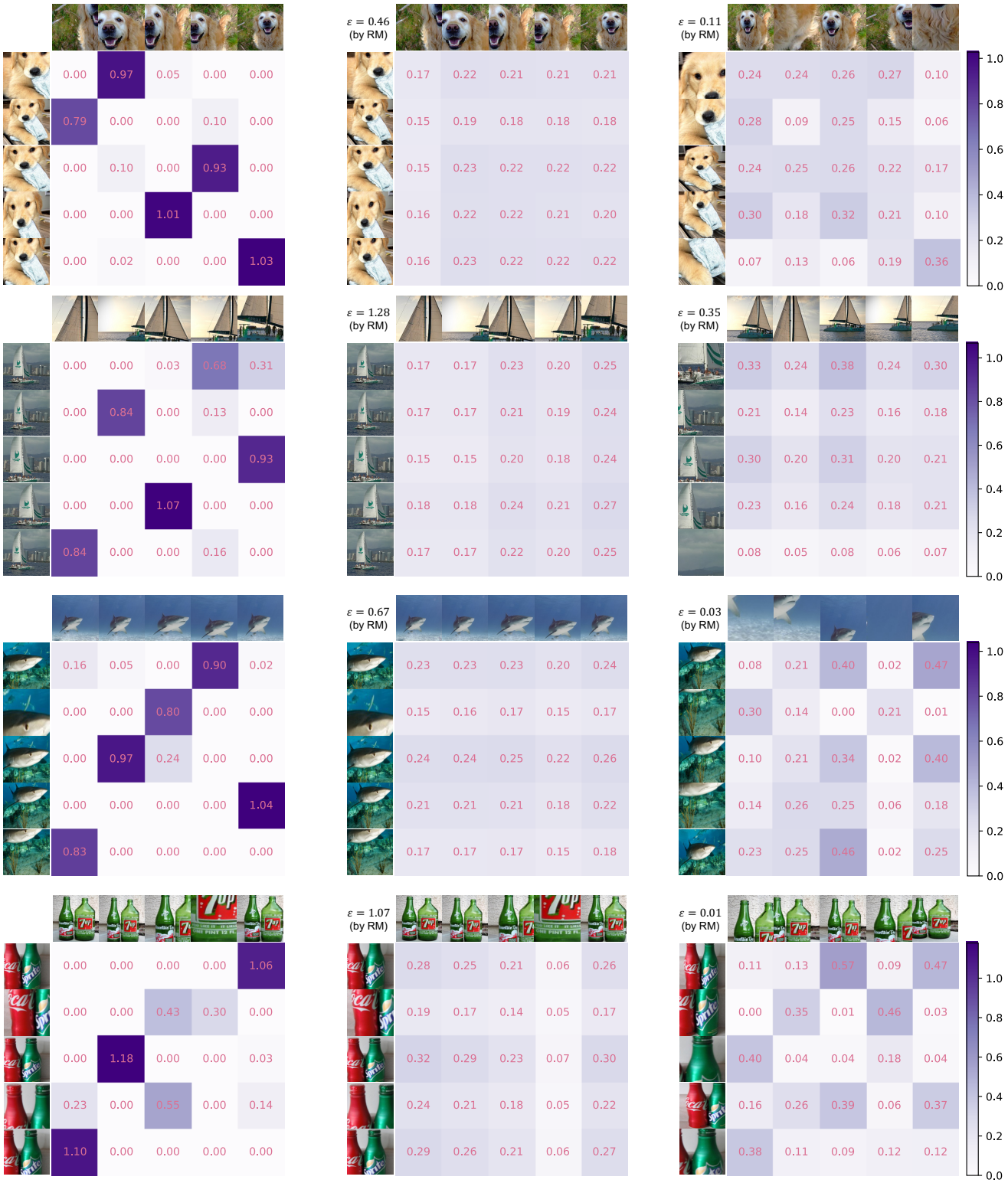
Sinkhorn, R.; and Knopp, P. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2): 343–348.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, volume 30.

Triantafillou, E.; Zemel, R.; and Urtasun, R. 2017. Few-Shot Learning Through an Information Retrieval Lens. In *Advances in Neural Information Processing Systems*, volume 30.

Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, volume 29.

Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2022. DeepEMD: Differentiable Earth Mover's Distance for Few-Shot Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–17.



(a) (b) (c)

Figure A: Visualization of solved transport matrices. Results of (a) EMD and (b) AM for sets consisting of similar local patches, and the result of (c) AM for sets consisting of dissimilar local patches.