Partner-Assisted Learning for Few-Shot Image Classification (Supplementary Material)

Contents

1. Benchmark Dataset Introduction

Datasets derived from ImageNet [4]: miniImageNet [23] and tieredImageNet [17].
MiniImageNet contains 100 classes, each class has 600 images, all images are resized to 84×84. The classes are split into three sets [16], 64 classes for training, 16 classes for few-shot validation, and 20 classes for few-shot testing. For each base class, we only use the *600* images for network training (base train set), while another 300 images are provided in [6] to build the base test set for conventional fully-supervised image classification evaluation. We use the classifier trained from the large-scale base train set to classify all images in the base test set among all base classes and report the accuracy for evaluation.
TieredImageNet is a hierarchical dataset and contains 608 leaf-level classes, which can be further categorized into 34 super-classes. The class split is performed at super-class level, where 20 super-classes (351 leaf-classes) are used for training, 6 super-classes (97 leaf-classes) are used for few-shot validation, and 8 super-classes (160 leaf-classes) are used for few-shot testing. During network training and evaluation, the classification is performed on the leaf-level classes.
Dataset derived from CIFAR100: CIFAR-FS [1] and FC100 [15] are two few-shot datasets created by adapting different class-split protocols on CIFAR100. CIFAR100 contains 60K 32×32 RGB images from 100 classes and each class has 600 images. CIFAR-FS splits the classes by using 64 classes for training, 16 classes for few-shot validation, and 20 classes for few-shot testing. FC100 adopt the hierarchical concept, similar to tieredImageNet, which categorized 100 leaf-classes into 20 super-classes. Then, 12 super-classes (60 leaf-classes) are used for training, 4 super-class (20 leaf-classes) are used for few-shot validation, and 4 super-classes (20 leaf-classes) are used for few-shot testing. During network training and evaluation, the classification is performed on the leaf-level classes.

During network training, we use the few-shot learning accuracy by performing prototype classification on the tasks sampled from the few-shot validation set for model selection. During network evaluation, we follow [21] and samples 1000 few-shot tasks. Within each task, 15 samples for each class are selected as queries for classification evaluation. We report the mean accuracy as well as the 95% confidence intervals.

2. Meta-Dataset

Meta-Dataset [22] is recently proposed for evaluating the performance of the few-shot learning algorithms on various domains. The Meta-Dataset is a combinational dataset which consists of 10 commonly used vision dataset: ILSVRC [18], omniglot [10], Aircraft [13], CU-Birds [24], DTD [3], Quickdraw [8], Fungi [19], VGG Flower [14], Traffic Signs [7], and MSCOCO [12]. All classes in Traffic Sign and MSCOCO are used for either validation or testing. For the rest dataset, the classes of all datasets are splited to training, validation, and testing. More detailed information and discussion could be found in [22]. For evaluating PAL's performance on meta-dataset, we trained ResNet18 model with PAL scheme on ILSVRC's training split only and resized all images to 128×128.

Table S1: PAL results on Meta-Dataset by using ResNet18. We followed the protocol in [22].

Dataset	Test Acc	Dataset	Test Acc	Dataset	Test Acc	Dataset	Test Acc	Dataset	Test Acc
ILSVRC	61.68	Omniglot	65.50	Aircraft	64.75	Birds	77.20	Textures	81.07
Quick Draw	58.05	Fungi	48.80	VGG Flower	90.68	Traffic Sign	77.98	MSCOCO	59.64

3. Symbol and Experiment

First, We provide the detail of training methods compared in Table 3 and Table 5. Then, we extend Table 3 and provide the full results in Table. S2. For few-shot evaluation, we follow [21] and report the mean as well as the 95% confidence interval among the 1000 randomly sampled tasks. For fully-supervised image classification evaluation, we classify all samples from a fixed base test set in [6] among the base classes and then report the accuracy.

Single Objective Training: $\mathcal{L}_{CE}$, $\mathcal{L}_{SupCT}$.

For $\mathcal{L}_{CE}$, we train a feature extractor and a classifier. We use feature extractor to perform prototype classification for few-shot evaluation, and use the output logits from the classifier to classify the base testing images. For $\mathcal{L}_{SupCT}$, we only train a feature extractor for few-shot evaluation.

Multi-Task Training: $\mathcal{L}_{CE}$+$\mathcal{L}_{SupCT}$

We train a single network while using the sum of $\mathcal{L}_{CE}$ and $\mathcal{L}_{SupCT}$ as the final objective. The output of the feature extractor is feed into both a classifier and a projector. During the training, we calculate the $\mathcal{L}_{CE}$ on the classifier outputs and the $\mathcal{L}_{SupCT}$ on the projector outputs. During the testing, we discard the projector and then use the remaining modules for few-shot evaluation and classification on base test set.

Mutual Training on Two Objectives: $\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$:

We train the *Partner Encoder* and the *Main Encoder* from scratch at the same time. The *Partner Encoder* is trained under supervised contrastive learning. Both of the two networks share the same classifier and we calculated the $\mathcal{L}_{CE}$ for both of the two networks. We calculate both the logit-level and feature-level alignment constraints from the *Partner Encoder* to the *Main Encoder*. During the testing, we use the *Main Encoder* and the classifier for the evaluation.

Uni-Direction $\mathcal{L}_{\mathbf{CE}} \rightarrow \mathcal{L}_{\mathbf{SupCT}}$:

We first train the *Partner Encoder* and a classifier using $\mathcal{L}_{CE}$. Then we fix *Partner Encoder* and train the *Main Encoder* from scratch using $\mathcal{L}_{SupCT}$. Meanwhile, we finetune the classifier by using the objective logit-level alignment constraint only. The feature-level alignment from the *Partner Encoder* to the *Main Encoder* is also applied. During testing, we discard the *Partner Encoder*, and use the *Main Encoder* and the classifier for the evaluation on few-shot tasks and the base testing set.

Alternative of *Partner Encoder* in PAL $\mathcal{L}_{CE} \rightarrow \mathcal{L}_{CE}$, $\mathcal{L}_{CT} \rightarrow \mathcal{L}_{CE}$:

We use the symbol of the full training scheme to indicate the loss used to train the *Partner Encoder*, *i.e.*, we train the *Partner Encoder* using either $\mathcal{L}_{CE}$ for $\mathcal{L}_{CE} \rightarrow \mathcal{L}_{CE}$ or $\mathcal{L}_{CT}$ for $\mathcal{L}_{CT} \rightarrow \mathcal{L}_{CE}$. Then we fix *Partner Encoder* and train the *Main Encoder* from scratch using $\mathcal{L}_{CE}$ and the alignment constraints. During evaluation, we use the *Main Encoder* and a classifier. Notably, even though a classifier is involved in the training of *Partner Encoder* by using $\mathcal{L}_{CE}$, to evaluate the property of the extracted features, we still discard that classifier and train a new classifier from scratch jointly with the *Main Encoder*. In this way, the only difference among the methods compared in Table 5 is merely the training objective of *Partner Encoder*.

According to the comparison in Table S2, since the feature extractor trained by $\mathcal{L}_{CE}$ may have already been overfitted to the base class and lost information which is irrelevant to the base classes but critical for novel classes, the features extracted by the $\mathcal{L}_{CE}$-pretrained model is not as good as the features by $\mathcal{L}_{SupCT}$-pretrained models. Meanwhile, it even under-performs the multi-task training ($\mathcal{L}_{CE}$ + $\mathcal{L}_{SupCT}$).

Implementation Detail: We followed [11, 21] and set output dimension of residual blocks as 64-160-320-640 and the dropout blocks are used in the last two residual blocks. For few-shot testing, following [21], we augment the each support sample 5 times and extract the features separately. Then, we average the normalized feature instances and the prototype is obtained by normalizing the averaged feature. The logits of the base test set, generated from the classifier (if applicable), are used for fully-supervised classification.

Table S2: Extend Table 3. Ablation study on the training schemes of combining two objectives.

Train Scheme	5-Way Few-shot		Base
	1-Shot	5-Shots	
$\mathcal{L}_{CE}$	63.76 ± 0.62	81.17 ± 0.45	80.90
$\mathcal{L}_{SupCT}$	62.29 ± 0.67	76.32 ± 0.48	n/a
$\mathcal{L}_{CE}$ + $\mathcal{L}_{SupCT}$	67.53 ± 0.65	82.14 ± 0.48	*83.20*
$\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$	65.21 ± 0.63	81.53 ± 0.44	80.13
$\mathcal{L}_{CE} \rightarrow \mathcal{L}_{SupCT}$	66.54 ± 0.63	81.83 ± 0.44	80.39
$\mathcal{L}_{\mathbf{SupCT}} \rightarrow \mathcal{L}_{\mathbf{CE}}$ [ours]	$\mathbf{69.37 \pm 0.64}$	$\mathbf{84.40 \pm 0.44}$	82.98

$*$ $\mathcal{L}_{SupCT}$ doesn't train a base classifier.

4. Ablation Study Analysis

4.1. Novel Test Set Performance

As $\mathcal{L}_{CE} \rightarrow \mathcal{L}_{SupCT}$ first trains the f_P and classifier under $\mathcal{L}_{CE}$ only, the initial rigid optimization towards the hard anchors by $\mathcal{L}_{CE}$ may limit accuracy on base and novel sets.

$\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$ follows the setting in [25] where the f_P and f_M are jointly trained from scratch and the alignment loss is used to update both f_P and f_M simultaneously. As $\mathcal{L}_{CE}$ contributes to optimizing f_M and classifier, f_P can also refer to the guidance of hard anchors for a easier converging via $\mathcal{L}_{feat}$, which will compromise its generality for regularization and result in sub-optimal solution for f_M on few-shot task. Similarly, $\mathcal{L}_{SupCT}$+$\mathcal{L}_{CE}$ trains a single network with two objectives jointly may also lead to local optimal due to rigid optimization.

Besides the sub-optimal regularization, the under-developed soft anchors extracted from f_P trained without using class label may also hurt the training of f_M. For example, as the $\mathcal{L}_{CT}$-trained f_P excludes class label information, $\mathcal{L}_{CT} \rightarrow \mathcal{L}_{CE}$ underperforms PAL. In contrast, PAL first fully trains f_P and then fixes it to extract soft anchors, where the well-developed and generality-preserved soft anchors can serve as better regularization.

4.2. Base Test Set Performance

For Table 3 Row$_{1,4}$, $\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$ sets regularization on baseline $\mathcal{L}_{CE}$ as mutual learning. The classifier may overfit to base training set caused by the sub-optimal regularization to f_M mentioned above.

For table 6 Row$_{4,5,6}$, with presence of $\mathcal{L}_{feat}$, further imposing $\mathcal{L}_{logit}$ or $\mathcal{L}_{KL}$ at logit-level may overfit classifier on base train set and limit the accuracy on base test set.

4.3. Hard Sample Selection

With analysis in SupCT [9], $\mathcal{L}_{feat}$ enumerates all sample pairs, including hard samples which can provide larger gradient.

5. Visualization

Besides the quantitative evaluation, we provide eight visualization examples on the novel domain of MiniImageNet from Fig. S1 to S4. Each plot in the visualization is generated by applying tSNE function on the extracted image features.

The features are extracted by the network trained with different framework or losses:

- (a) *Main Encoder* trained under PAL ($\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$)
- Single Objective: a feature extractor trained under (b) $\mathcal{L}_{SupCT}$ (c) $\mathcal{L}_{CE}$ (d) $\mathcal{L}_{CT}$
- (e) the model trained under $\mathcal{L}_{CE}$ during mutual learning $\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$
- (f) a feature extractor trained under multi-task $\mathcal{L}_{SupCT} + \mathcal{L}_{CE}$
- *Main Encoder* aligned with the *Partner Encoder* trained under (g) $\mathcal{L}_{CE}$ (denoted as $\mathcal{L}_{CE} \rightarrow \mathcal{L}_{CE}$) and (h) $\mathcal{L}_{CT}$ (denoted as $\mathcal{L}_{CT} \rightarrow \mathcal{L}_{CE}$)

For feature extraction, we feed the raw images without any augmentation as input into the network. For each example, the images used for each plot are identical with each other. To best mimic the condition of few-shot evaluation and for fair comparison, for each example, we select 5 classes and each class has 30 images.

Prior works [5, 15, 20, 21] has shown that feature extractor trained for *fully supervised* classification over base classes has promising transferability from base to novel classes. However, there are still risks that such feature extractor overfits to the base classes and the ability of the adaptation can be compromised. To quantitatively evaluate the features extracted by the pre-trained model without any adaptation, we choose prototype classification over novel classes under few-shot settings. As such, a discriminative feature space with compact clusters over the novel domain is critical. To this end, we are specifically motivated to ensure the feature extractor trained on the base classes with additional constraints can preserve as much as useful information for object while maintain high classification accuracy. In this way, we expect such feature extractor can be robustly adapted to novel classes. According to the visualization, by comparing (a)-(h), the features extractor trained by $\mathcal{L}_{\mathbf{SupCT}} \rightarrow \mathcal{L}_{\mathbf{CE}}$ [ours] can generate the most discriminative feature space for the samples of novel classes, which works best for few-shot prototype classification.

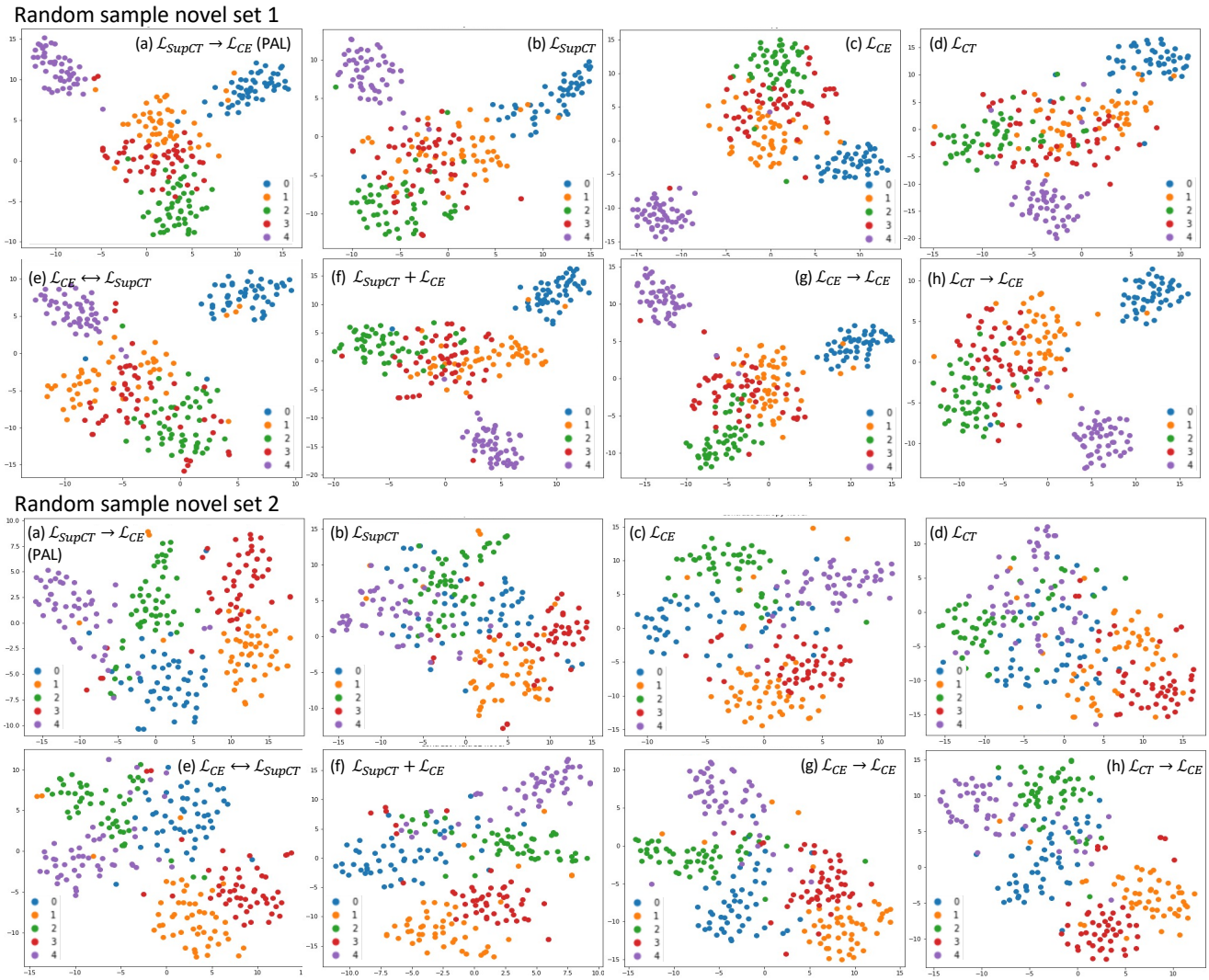

Figure S1: Two visualization examples on MiniImageNet novel set. For each example, the features are extracted by models trained with (a) PAL (ours), (b) Supervised contrastive learning $\mathcal{L}_{SupCT}$, (c) Cross-entropy loss $\mathcal{L}_{CE}$, (d) Unsupervised contrastive loss $\mathcal{L}_{CT}$ [2], (e) Mutual Learning ($\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$), (f) Multi-task learning ($\mathcal{L}_{SupCT} + \mathcal{L}_{CE}$), as well as the *Main Encoder*s constrained by *Partner Encoder* trained with (g) $\mathcal{L}_{CE}$ and (h) $\mathcal{L}_{CT}$. To best mimic the condition of few-shot evaluation and for fair comparison, for each row, only 5 classes are selected and the image samples used for all four plots are identical with each other.

References

[1] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. 2

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 6

[3] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[5] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv*

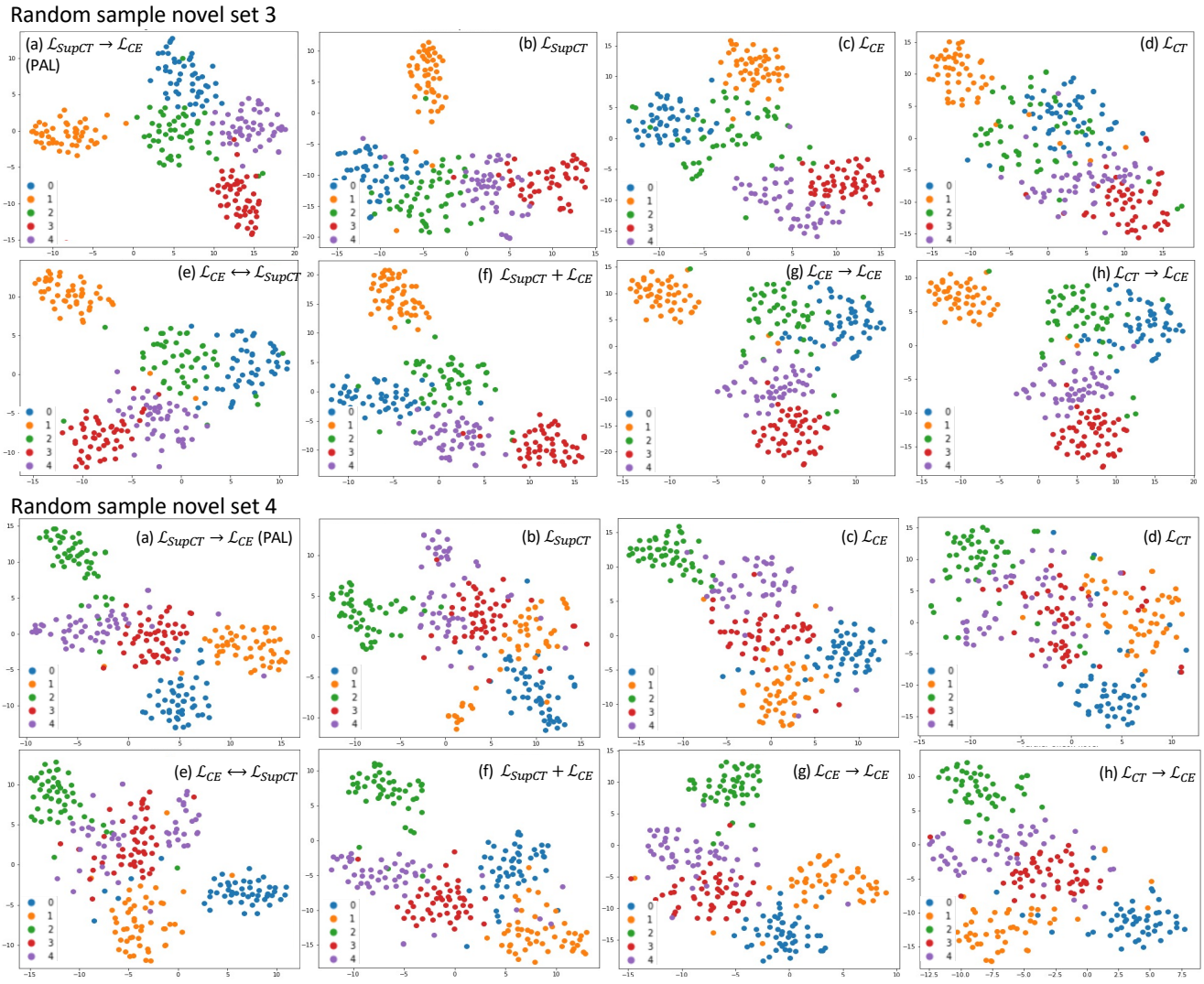

Figure S2: Two visualization examples on MiniImageNet novel set. Continued from Fig. S1. To best mimic the condition of few-shot evaluation and for fair comparison, for each row, only 5 classes are selected and the image samples used for all four plots are identical with each other.

preprint arXiv:1909.02729, 2019. 5

[6] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2, 3

[7] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013. 2

[8] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw! – a.i. experiment, 2016. https://quickdraw.withgoogle.com/. 2

[9] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 4

[10] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[11] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10649–10657, 2019. 3

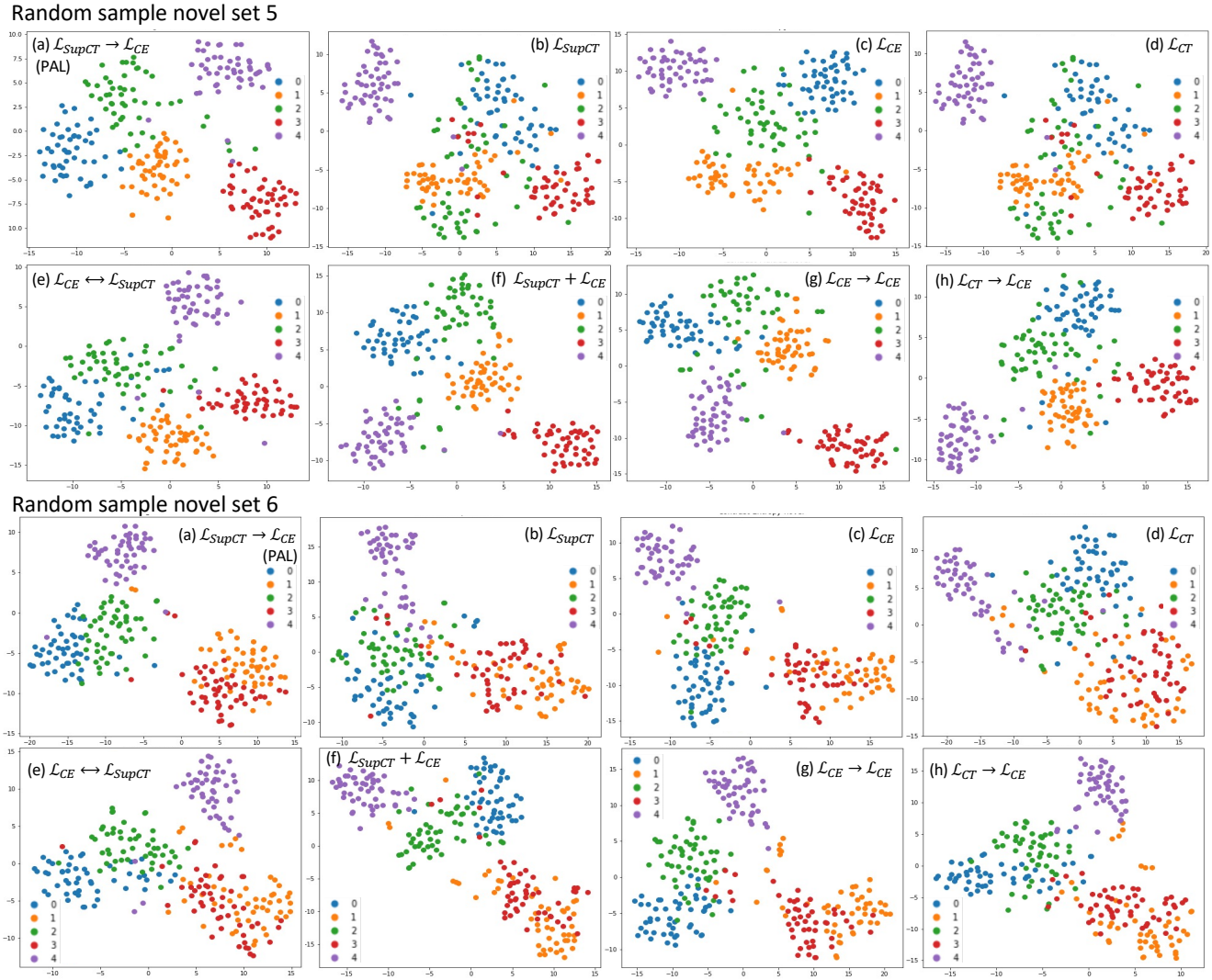

Figure S3: Two visualization examples on MiniImageNet novel set. Continued from Fig. S2. To best mimic the condition of few-shot evaluation and for fair comparison, for each row, only 5 classes are selected and the image samples used for all four plots are identical with each other.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 2

[13] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013. 2

[14] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729, 2008. 2

[15] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 721–731. Curran Associates, Inc., 2018. 2, 5

[16] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*, 2017. 2

[17] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Proceedings of 6th International Conference on Learning*

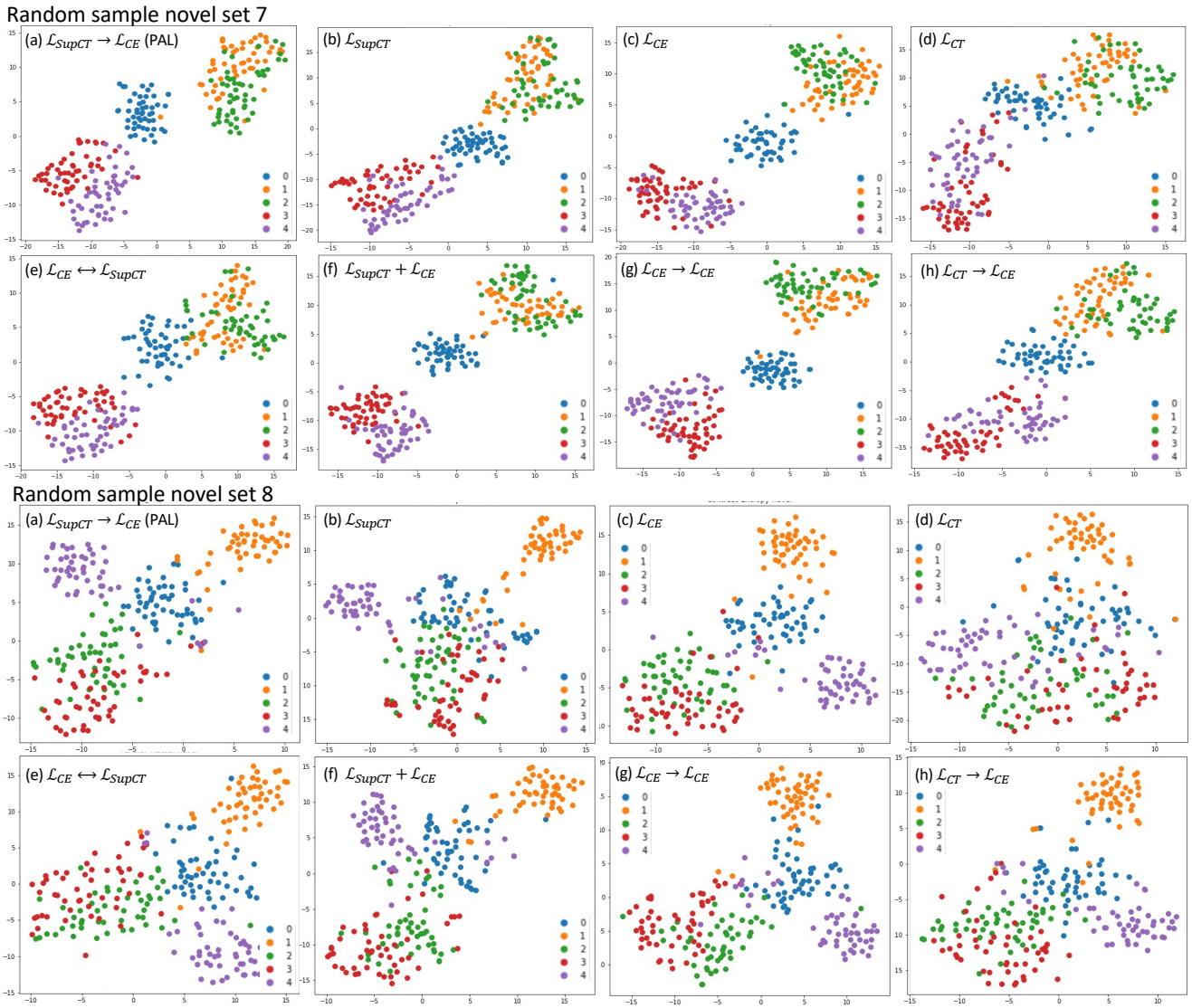

Figure S4: Two visualization examples on MiniImageNet novel set. Continued from Fig. S3. To best mimic the condition of few-shot evaluation and for fair comparison, for each row, only 5 classes are selected and the image samples used for all four plots are identical with each other.

Representations ICLR, 2018. 2

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[19] Brigit Schroeder and Yin Cui. Fgvcx fungi classification challenge 2018, 2018. github.com/visipedia/fgvcx_fungi_comp. 2

[20] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5

[21] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020. 2, 3, 5

[22] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020. 2

[23] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016. 2

[24] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[25] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 4