

Partner-Assisted Learning for Few-Shot Image Classification

Jiawei Ma^{†,1}Hanchen Xie^{†,2}Guangxing Han¹Shih-Fu Chang¹Aram Galstyan²Wael Abd-Almageed²¹ Columbia University² USC Information Sciences Institute

{jiawei.m, gh2561, sc250}@columbia.edu, {hanchenx, galstyan, wamageed}@isi.edu

Abstract

Few-shot Learning has been studied to mimic human visual capabilities and learn effective models without the need of exhaustive human annotation. Even though the idea of meta-learning for adaptation has dominated the few-shot learning methods, how to train a feature extractor is still a challenge. In this paper, we focus on the design of training strategy to obtain an elemental representation such that the prototype of each novel class can be estimated from a few labeled samples. We propose a two-stage training scheme, Partner-Assisted Learning (PAL), which first trains a Partner Encoder to model pair-wise similarities and extract features serving as soft-anchors, and then trains a Main Encoder by aligning its outputs with soft-anchors while attempting to maximize classification performance. Two alignment constraints from logit-level and feature-level are designed individually. For each few-shot task, we perform prototype classification. Our method consistently outperforms the state-of-the-art methods on four benchmarks. Detailed ablation studies of PAL are provided to justify the selection of each component involved in training.

1. Introduction

Deep learning has achieved impressive success in many vision tasks, such as image classification [21, 38, 17], object detection [36, 34, 37], and image segmentation [26, 3, 16], especially when sufficient labeled data is available for training. However, data annotation can be expensive and large scale annotated data is not always available [13, 24, 43, 49].

Few-shot learning has been proposed to mimic human vision systems, which is capable of learning the visual appearance of new objects with only a few (*e.g.*, 1 or 5) instances [24, 46]. To facilitate few-shot learning for fast model adaptation, meta-learning has been employed to simulate few-shot tasks during training, by either designing an optimal algorithm for adaptation [10, 30] or learn-

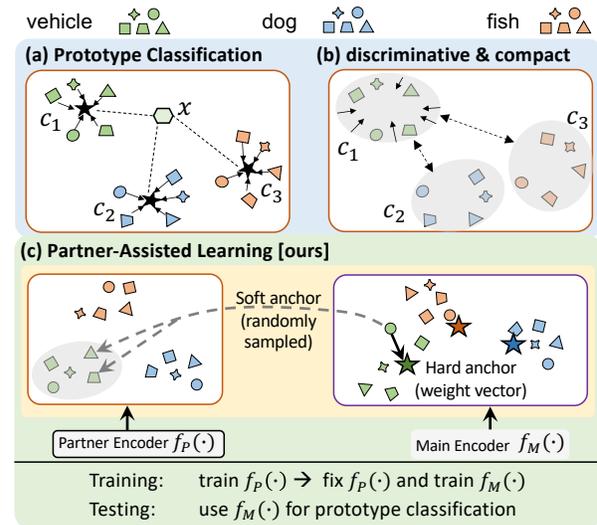


Figure 1: (a) Prototype classification calculates the few-shot prototypes and classifies a sample by comparing its similarity to each prototype. (b) The discriminative feature distribution with compact clusters benefits the prototype classification [47, 25]. (c) We propose a Partner-Assisted Learning framework, in which a pre-trained partner encoder, f_P , is used to generate soft-anchors to regularize the learning of main encoder, f_M , which will be used at the inference time.

ing a shared feature space for prototype-based classification [41, 31, 23]. As shown in Fig. 1(a), prototype classification methods [6, 8, 41, 45] estimate the few-shot prototypes by averaging the features of a few labeled samples (*i.e.*, support). A new sample (*i.e.*, query) is classified by comparing its cosine similarity with all prototypes using nearest neighborhood search. As illustrated in Fig. 1(b), in a classification context, the feature distribution is supposed to be (1) compact within each cluster (*i.e.*, supporting high intra-class similarities), and (2) discriminative between clusters (*i.e.* supporting large inter-class distances).

Recent work has shown that pretraining a model on a large scale (base) dataset with full supervision can serve as a strong baseline [6] for the novel few-shot tasks by perform-

[†]Equal contribution.

ing prototype classification [6, 8, 41, 45]. For each base class, conventional fully-supervised pretraining using class labels [45, 6] learns a unique weight vector, which serves as a hard-anchor. By minimizing the cross-entropy loss with respect to (w.r.t.) the class label, each image feature is pushed towards its corresponding class anchor. Therefore, for each class, the average of features is expected to represent the class during few-shot classification.

The feature extractor pretrained on the base classes for classification may suppress details irrelevant to the base domain [4], while these details could be discriminative for novel classes. Thus, incorporating instance comparison to preserve details can facilitate few-shot learning on novel domains. Knowledge distillation formulates a teacher-student setting and compares the outputs from two models for the same image [45]. For each image, the teacher model generates soft-labels to model the proximity between different classes. By comparing the outputs of teacher model and student model, the student model is trained with the soft-label so that more details indicating class relationship could be preserved. Thus, the student model achieves higher accuracy on few-shot tasks. Despite the success of knowledge distillation, the performance improvement is still limited since the teacher model has once been rigidly optimized according to hard-anchors of base classes.

Besides comparing the outputs of the same instance from two networks using cosine similarity, a single network can be trained for pair-wise comparison, so that its outputs of a few randomly selected support samples can dynamically represent the class center [41]. Metric-based meta-learning methods, such as prototypical network [41], have been proposed to learn to represent a class by aggregating support features. This way, representative centers are dynamically estimated according to a few labeled data. Similarly, supervised contrastive learning [19] performs pair-wise comparison, where each feature is sampled from the training set and individually represents the class without aggregation.

Inspired by the dynamic and individual representative employed in prototypical learning and supervised contrastive learning, to improve the generalization ability of feature extractor, we propose to extract features that can be used to dynamically represent classes, and set those features as soft-anchors to regularize the feature extractor trained with hard-anchors. Comparing with knowledge distillation, instead of aiming to iterate the feature extractor that has already been optimized w.r.t hard-anchors, our method uses diverse features on the base domain to regularize a new feature extractor which is trained with class label under cross-entropy loss from scratch. The contributions of this paper are as follows:

- We propose Partner-Assisted Learning (PAL): a framework for representation learning in few-shot classification setting, in which the *Partner Encoder* and *Main*

Encoder are trained sequentially such that the features from *Partner Encoder* are used as soft-anchors to regularize the training of *Main Encoder* from scratch.

- We propose two alignment approaches on both feature-level and logit-level, which utilizes the soft-anchors for regularization during training with class labels.
- We show that PAL consistently achieves state-of-the-art performance on four few-shot benchmarks, and improves the classification accuracy in a supervised learning setting. We also provide comprehensive ablation studies to justify the design of each component.

2. Related Work

Prototype Classification has been widely used in metric-based methods for few-shot classification. Prototypical networks [41] simulate few-shot tasks using episodes during meta-training. In each episode, a few labeled training samples are randomly sampled and then class prototype is estimated by averaging the extracted features. The quality of the estimated class prototype is evaluated by classifying the query features. Similarly, models trained by the supervised contrastive loss (SupCT) [19] learn to maximize the similarity between all instances of the same class so that all instances are clustered together and each class can be represented by every instance feature of that class. Furthermore, the concept of meta-learning is also employed to estimate the task-adaptive metric by learning to scale the metric or add margins for prototype classification, which has shown clear benefit on few-shot tasks [31, 23, 14, 15].

Recently, networks pretrained with fully-supervised classification tasks [6] have been treated as strong baselines for few-shot classification. A unique prototype for each class is learned through class labels, *i.e.*, one-hot vectors, to indicate the discrimination between classes. Furthermore, RFS [45] shows further improvements using knowledge distillation with soft labels based on the conventional network trained with cross-entropy loss.

Regularization to Cross-Entropy (CE) loss. CE loss [29] is widely used in fully-supervised tasks due to its simplicity, where it learns a classification hyperplane in a high-dimensional representation space. Regularization can be added to encourage the intra-class compactness, by setting large margin [25]. Various loss functions, such as center loss [48], L-GM loss [47], and Ring Loss [53] have been introduced to emphasize certain embedding distributions in the latent space. In face recognition, triplet loss [40] has widely been used where an image triplet is constructed by sampling a positive pair and a negative pair as anchors.

Knowledge Transfer between Networks: Knowledge Distillation [18] has been proposed to perform uni-direction knowledge transfer, which uses a strong teacher model to train a simple student model on the same task. Other work on knowledge distillation also show the advantages in semi-

supervised learning [44, 5]. The strong teacher model is pretrained with one-hot vectors, and then generates soft-labels for the student model. Although the teacher model can serve as a strong baseline, the large negative logits may hurt the distillation process and such logits need to be smoothed by modifying the Softmax operation [18]. Contrarily, mutual learning [52] studies bi-direction knowledge sharing, where two networks are trained jointly from scratch with the same objective. Both uni-direction and bi-direction knowledge transfer have demonstrated ability to learn a better representation than a single network.

3. Partner-Assisted Learning

In this section, we present the proposed Partner-Assisted Learning (PAL) to learn an embedding function. As illustrated in Fig. 2, PAL consists of *Partner* and *Main* Encoders. Task formulation and notation are defined in Section 3.1. The objective for the *Partner Encoder* is presented in Section 3.2. The framework of imposing alignment constraints for the *Main Encoder* is discussed in Section 3.3.

3.1. Learning-Task Formulation

In the few-shot learning, we are first given a *base* dataset \mathcal{D}_{base} , which consists of abundant amount of labeled samples. All sample labels in \mathcal{D}_{base} belong to the base class set \mathcal{C}_{base} . Then, we are given a *novel* set \mathcal{D}_{novel} , from which each episode \mathcal{D}_{epi} is sampled. All sample labels in \mathcal{D}_{novel} are from the novel class set \mathcal{C}_{novel} where the class sets for *base* and *novel* are disjoint, i.e., $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$. Each episode $\mathcal{D}_{epi} = (\mathcal{D}_S, \mathcal{D}_Q)$ is composed of a support set \mathcal{D}_S for prototype estimation and a query set \mathcal{D}_Q for evaluation. For an N -way K -shot task, the $\mathcal{D}_S \cup \mathcal{D}_Q$ in an episode contains N novel classes drawn from \mathcal{C}_{novel} , and \mathcal{D}_S contains K labeled samples for each class.

As illustrated in Fig. 2, \mathcal{D}_{base} is first used to train the *Partner Encoder* f_P to generate soft-anchors. Then, f_P is fixed and \mathcal{D}_{base} is used to train the *Main Encoder* f_M , which is regularized by the alignment constraints on either logit-level or feature-level from f_P under the PAL framework. During the few-shot evaluation, similar to [45, 6], we directly use the pre-trained f_M to estimate the prototype of each class using \mathcal{D}_S and classify the testing samples in \mathcal{D}_Q .

3.2. Partner Encoder

A *Partner Encoder* f_P is trained using supervised contrastive learning (SupCT) to do clustering and perform pairwise comparison among all feature instances. The features of the same class are pushed together while the features from different classes are pushed away. The detail of supervised contrastive learning is presented below.

Supervised Contrastive Learning: Given a batch \mathcal{D}_{raw} with B images, i.e., $|\mathcal{D}_{raw}| = B$, an augmented batch with

$2B$ samples is generated by performing two separate augmentations on each image,

$$\mathcal{D} = \text{Concat}(\text{Aug}(\mathcal{D}_{raw}), \text{Aug}(\mathcal{D}_{raw})), \quad (1)$$

where Aug indicates a data augmentation function-group which randomly transforms images. For each image $\mathcal{D}(i)$ where $i \in \mathcal{I} \equiv \{1 \dots 2B\}$, a positive index set $\mathcal{I}_{\text{pos}}(i) \subset \mathcal{I} \setminus \{i\}$ is selected, such that all images $\mathcal{D}(j)$ for $j \in \mathcal{I}_{\text{pos}}(i)$ are of the same class as $\mathcal{D}(i)$. Then, the supervised contrastive loss is defined as

$$\mathcal{L}_{SupCT}(\mathcal{D}) = \sum_{i \in \mathcal{I}} \frac{-1}{|\mathcal{I}_{\text{pos}}(i)|} \sum_{j \in \mathcal{I}_{\text{pos}}(i)} \Theta(i, j) \quad (2)$$

$$\Theta(i, j) = \log \frac{\exp(\mathbf{z}_{f_P, \mathcal{D}(i)} \cdot \mathbf{z}_{f_P, \mathcal{D}(j)} / \tau)}{\sum_{a \in \mathcal{I} \setminus \{i\}} \exp(\mathbf{z}_{f_P, \mathcal{D}(i)} \cdot \mathbf{z}_{f_P, \mathcal{D}(a)} / \tau)}$$

where $\mathbf{z}_{f_P, x}$ denotes the feature of image x extracted by the *Partner Encoder* f_P after l_2 -normalization, and τ is a temperature hyperparameter used to rescale the affinity score. Minimizing $\mathcal{L}_{SupCT}(\mathcal{D})$ trains the model to maximize the similarity between features of the same class (positive pair) while pushing away the features from different classes (negative pair). According to Eq. (2), as noted in [19], the disagreement between the two features in a positive pair is induced by the variation between image instances and difference resulting from data augmentation.

As an alternative to \mathcal{L}_{SupCT} , the unsupervised contrastive loss $\mathcal{L}_{CT}(\mathcal{D})$ shares the same formulation as $\mathcal{L}_{SupCT}(\mathcal{D})$ while semantic information of class label is excluded. Then, the positive index set for each $i \in \mathcal{I}$ is $\mathcal{I}_{\text{pos}}(i) = \{i + B\}$ for $i \leq B$ and $\mathcal{I}_{\text{pos}}(i) = \{i - B\}$ for $i > B$, i.e., the disagreement between the two features in a positive pair is only induced by exhaustive augmentation.

In PAL, we use \mathcal{L}_{SupCT} to train the *Partner Encoder*. \mathcal{L}_{CT} is used as an alternative to \mathcal{L}_{SupCT} for ablation study. Since \mathcal{L}_{SupCT} models instance-level similarity between features in positive pairs and push away features of different classes, as shown in our experiments in Section 4.3, among the considered alternative variants, the features extracted from \mathcal{L}_{SupCT} -trained *Partner Encoder* facilitates the training of *Main Encoder* most.

3.3. Main Encoder

In this section, we first review the soft-labels introduced in knowledge distillation, and then introduce the alignment constraints at the logit- and feature-level imposed by *Partner Encoder* during *Main Encoder* training.

3.3.1 Knowledge Distillation Preliminary

As discussed in [45], the teacher model provides soft-labels depicting the fact that some classes are relatively close to each other. The soft-labels $\mathbf{p} \in \mathcal{R}^{|\mathcal{C}_{base}|}$ are calculated from

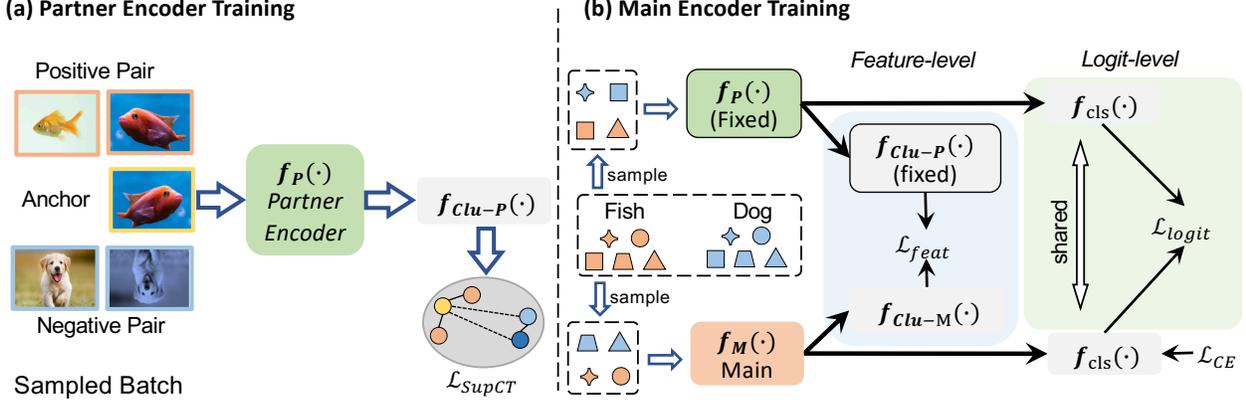


Figure 2: Training pipeline of Partner-Assisted Learning. (a) Train the *Partner Encoder* f_P by supervised contrastive learning \mathcal{L}_{SupCT} to model the pair-wise similarity among all features. (b) Train the *Main Encoder* f_M by imposing either logit-level or feature-level alignments using the pretrained f_P . Both f_P and f_M use ResNet-12[17] and each gray block denotes a fully-connected layer.

the output logits $\mathbf{v} \in \mathcal{R}^{|\mathcal{C}_{base}|}$ through softmax operation $\mathbf{p} = \text{Softmax}(\mathbf{v}/\tau)$, where the temperature τ can scale the logits and a higher τ produces a softer probability distribution over base classes.

In addition to the cross-entropy loss on student model, the KL-divergence \mathcal{L}_{KL} is used as objective in teacher-student setting for knowledge distillation and is defined as

$$\begin{aligned} \mathcal{L}_{KL}(\mathbf{p}_{t,x} || \mathbf{p}_{s,x}) &= \sum_{c \in |\mathcal{C}_{base}|} \mathbf{p}_{t,x}(c) \log \frac{\mathbf{p}_{t,x}(c)}{\mathbf{p}_{s,x}(c)} \quad (3) \\ &= \sum_{c \in |\mathcal{C}_{base}|} \mathbf{p}_{t,x}(c) \log \mathbf{p}_{t,x}(c) - \sum_{c \in |\mathcal{C}_{base}|} \mathbf{p}_{t,x}(c) \log \mathbf{p}_{s,x}(c) \\ &= -H(\mathbf{p}_{t,x}) + H(\mathbf{p}_{t,x}, \mathbf{p}_{s,x}), \end{aligned}$$

where $\mathbf{p}_{t,x}$ and $\mathbf{p}_{s,x}$ are the output probability distribution of the same image x by teacher model and student model. Minimizing \mathcal{L}_{KL} will minimize the cross-entropy $H(\mathbf{p}_{t,x}, \mathbf{p}_{s,x})$ between teacher soft-label and student prediction. When the teacher is also trained, the negative entropy of its output $-H(\mathbf{p}_{t,x})$ is minimized.

Since the teacher model is pretrained to learn hard-anchors for classification, and then predicts logits through the single linear mapping, the logits output is not well-constrained and negative logits with large absolute value exist [18]. As experimentally demonstrated in [18], a high temperature has to be set for cross-entropy $H(\mathbf{p}_{t,x}, \mathbf{p}_{s,x})$ between the student prediction and the soft-label during student model training, so that the effect of large negative logits from teacher model can be mitigated and the student model can work better.

Instead of first training a model that has once been rigidly optimized to hard-anchors and then setting a high temperature to reduce impact from large negative logits, when training a *Main Encoder* from scratch with class labels, we

propose to use the features extracted by *Partner Encoder* as soft-anchors for providing alignment regularization. To constrain the logit value of classifier, we first design the classifier as the cosine similarity function between feature representations and class weight vectors. Then, in addition to minimizing cross-entropy loss for each sample, we use the features of *Partner Encoder* to regularize *Main Encoder* and design the constraint method at either logit-level or feature-level alignment.

3.3.2 Logit-Level Alignment

During knowledge distillation, given a query feature of a target class, we calculate affinity score as the dot-product between the query feature and well-trained class weight vector for each base candidate class. The affinity scores can then be used to describe the relationship between all classes. A candidate class is close to the target class if the corresponding affinity score is high.

Similarly, we use the cosine similarity between the all class weight vectors and the well-trained query features to generate soft labels, and then minimize the cross-entropy,

$$\mathcal{L}_{logit} = H(\mathbf{p}_{p,x'}, \mathbf{p}_{m,x}),$$

between the soft label $\mathbf{p}_{p,x'}$ generated by *Partner Encoder* and the prediction $\mathbf{p}_{m,x}$ from *Main Encoder*, while the image x' and x are of the same class. As the *Partner Encoder* has been well trained for clustering by maximizing the cosine similarity between features in all positive pairs, we fix the *Partner Encoder* and then extracts features as soft-anchors. Since the *Main Encoder* is also trained by maximizing the cosine similarity between the query feature and the class weight vectors, we assume the features by *Partner* and *Main Encoders* share a common feature space. Thus,

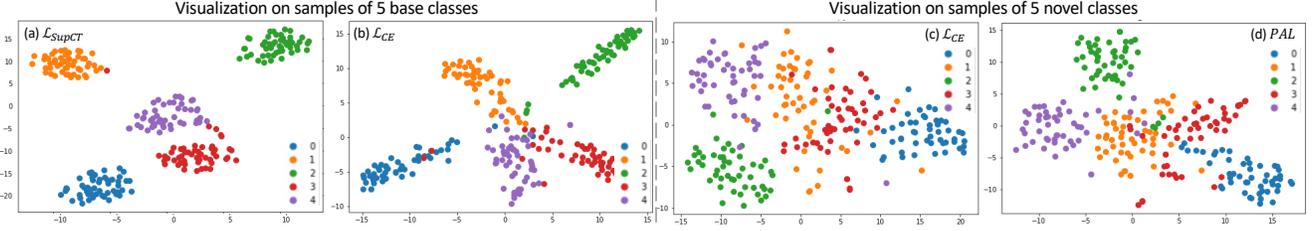


Figure 3: Visualization on five base classes using model trained by (a) supervised contrastive loss \mathcal{L}_{SupCT} and (b) cross-entropy loss \mathcal{L}_{CE} , and visualisation on five novel classes using model learned by (c) \mathcal{L}_{CE} and (d) PAL (our method). (a) \mathcal{L}_{SupCT} is used to cluster all features using class label, and the features in each cluster can be used as soft-anchors, while (b) \mathcal{L}_{CE} trains network to learn hard-anchors for classification. The feature distribution on novel classes by (d) PAL benefits the prototype classification compared with (c) the distribution of \mathcal{L}_{CE} . More visualization can be found in supp. material.

we feed features from *Partner Encoder* into the shared classifier and calculate $\mathbf{p}_{p,x'}$.

In terms of implementation, *Partner Encoder* shares the classifier of *Main Encoder* to generate cosine similarities as logits. Since the class weight vectors are randomly initialized before training, we adopt a warm-up strategy and increase the weight of \mathcal{L}_{logit} from 0 to 1 as the loss converges and the class weight vectors are gradually learned for each class. Notably, our logit-level constraint is different from knowledge distillation and does not minimize the negative entropy of *Partner Encoder*. Since the *Partner Encoder* is not trained to learn hard-anchors, minimizing the negative entropy of *Partner Encoder* is not needed. Meanwhile, as demonstrated by the ablation study in Table 5, since the classifier is randomly initialized at the beginning, comparing with our proposed logit-level alignment (Row₂), minimizing the negative entropy of $\mathbf{p}_{p,x'}$ (Row₃) will confuse the shared classifier and has negative impact on both classifier and *Main Encoder*.

3.3.3 Feature-Level Alignment

Feature-level alignment is achieved by pair-wise comparison between the features extracted by *Partner* and *Main Encoders*. As stated in Section 3.3.1, *Main Encoder* is trained with a classifier based on cosine similarity for generating logits and range of logits value is bounded. Equivalently, learning such a classifier effectively learns a unique anchor for each class and every feature is trained to maximize the cosine similarity with its corresponding class anchor.

During the training process of *Partner Encoder* with \mathcal{L}_{SupCT} , features that belong to the same class are clustered together and separated from features of other classes. Therefore, the clusters of base classes can be considered as a pool and each feature can be considered as a soft-anchor for alignment. For each image, in addition to the supervised classification signal provided class label, a subset of soft-anchors is sampled from the pools.

Specifically, similar to the idea in supervised contrastive

loss, we do pair-wise comparison for feature-level alignment. Given the batch \mathcal{D}_M consisting of $|\mathcal{D}_M|$ features extracted by *Main Encoder*, for each feature instance indexed by $i \in \mathcal{I}_M = \{1 \dots |\mathcal{D}_M|\}$, according to class labels, a set of positive features $\mathcal{D}_{P,D_M(i)}^+$ and negative features $\mathcal{D}_{P,D_M(i)}^-$ are randomly sampled from the pools by *Partner Encoder*, such that all features in $\mathcal{D}_{P,D_M(i)}^+$ are of the same class as $\mathcal{D}_M(i)$ and every feature in $\mathcal{D}_{P,D_M(i)}^-$ is of a different class from $\mathcal{D}_M(i)$. Then, we define feature-level constraint as

$$\mathcal{L}_{feat}(\mathcal{D}_M) = \sum_{i \in \mathcal{I}_M} \frac{-1}{|\mathcal{D}_{P,D_M(i)}|} \sum_{j \in \{1 \dots |\mathcal{D}_{P,D_M(i)}^+|\}} \Theta(i, j), \quad (4)$$

$$\Theta(i, j) = \log \frac{\exp(D_M(i) \cdot \mathcal{D}_{P,D_M(i)}^+(j)/\tau)}{\sum_{a \in \mathcal{I}_{P,D_M(i)}} \exp(D_M(i) \cdot \mathcal{D}_{P,D_M(i)}(a)/\tau)},$$

where $\mathcal{D}_{P,D_M(i)} = \mathcal{D}_{P,D_M(i)}^+ \cup \mathcal{D}_{P,D_M(i)}^-$, and $\mathcal{I}_{P,D_M(i)} = \{1 \dots |\mathcal{D}_{P,D_M(i)}|\}$.

3.3.4 Main Encoder Training

In summary, in addition to the cross-entropy loss for each sample $\mathcal{L}_{CE} = H(\mathbf{1}_y(x), \mathbf{p}_{m,x})$ where $\mathbf{1}_y(x)$ denotes one-hot vector of class label, we include both logit-level and feature-level alignments in the final training objective of *Main Encoder* while the *Partner Encoder* used to extract soft anchors is fixed. In practice, we found that using either \mathcal{L}_{logit} or \mathcal{L}_{feat} provides clear benefit in regularizing the *Main Encoder*, whereas summing up both of them produces the best performance.

4. Experimental Evaluation

We evaluated PAL on four benchmark datasets to demonstrate its robustness: miniImagenet [46], tieredImagenet [35], CIFAR-FS [2], and FC100 [31]. Results are shown in Table 1 and Table 2. Detailed ablation studies are discussed in Section 4.3.

Table 1: PAL Result on miniImageNet and tieredImageNet datasets. †: Results are generated on Train+Val set

Algorithm	Backbone	miniImageNet, 5-way		tieredImageNet, 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [10]	32-32-32-32	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75
Prototypical Networks [41]†	64-64-64-64	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
SNAIL [27]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92	-	-
AdaResNet [28]	ResNet-12	56.88 ± 0.62	71.94 ± 0.57	-	-
TADAM [31]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	-	-
Shot-Free [33]	ResNet-12	59.04 ± <i>n/a</i>	77.64 ± <i>n/a</i>	63.52 ± <i>n/a</i>	82.59 ± <i>n/a</i>
TEWAM [32]	ResNet-12	60.07 ± <i>n/a</i>	75.90 ± <i>n/a</i>	-	-
MTL [42]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80	-	-
Variational FSL [51]	ResNet-12	61.23 ± 0.26	77.69 ± 0.17	-	-
MetaOptNet [22]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
Fine-tuning [8]	WRN-28-10	57.73 ± 0.62	78.17 ± 0.49	66.58 ± 0.70	85.55 ± 0.48
LEO-trainval [39]†	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
Diversity w/Cooperation [9]	ResNet-18	59.48 ± 0.65	75.62 ± 0.48	-	-
Associative-Alignment [1]	ResNet-18	59.88 ± 0.67	80.35 ± 0.73	69.29 ± 0.56	85.97 ± 0.49
AdaMargin [23]	ResNet-12	67.10 ± 0.52	79.54 ± 0.60	-	-
DeepEMD [50]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
MABAS [20]	ResNet-12	65.08 ± 0.86	82.70 ± 0.54	-	-
RFS-simple [45]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
RFS-distill [45]	ResNet-12	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
PAL (Ours)	ResNet-12	69.37 ± 0.64	84.40 ± 0.44	72.25 ± 0.72	86.95 ± 0.47

4.1. Benchmark Datasets & Training Setup

Datasets derived from ImageNet [7]: miniImageNet [46, 12] and tieredImageNet [35]. MiniImageNet [46] contains 100 classes, the class split for (training, few-shot validation, few-shot testing) is (64,16,20). Each base class has 600 images for training and 300 images for fully-supervised classification evaluation [12]. TieredImageNet [35] contains 608 classes with the class split (351,97,160) and around 450K images from base dataset for network training. All images for the two sets are sized to 84×84 .

Dataset derived from CIFAR100: CIFAR-FS [2] and FC100 [31]. CIFAR-FS [2] contains 100 classes with the class split for (64,16,20). FC100 [31] contains 100 classes with the class split (60,20,20). Each class has 600 images and all images for the two sets are of 32×32 .

The hierarchical class structure, *i.e.*, some leaf classes can be ground together to a coarse class, are considered for the class split of the TieredImageNet and FC100. The leaf classes under the same coarse have more semantic correlation. As there is no overlap of coarse class between the base class set and the novel class set, the adaptation from the base class set to the novel class set will be more challenging.

Training Setup: On all benchmark datasets, we run experiments using ResNet12 [17] as the backbone optimized using stochastic gradient decent (SGD). We use an initial learning rate of 0.03 with a decay factor of 10 starting

at the 60-th epoch, and train for 90 epochs. The batch size is 64 on Mini-ImageNet, CIFAR-FS and FC100, and is 400 on tiered-ImageNet. The temperature scaling factor τ in \mathcal{L}_{SupCT} (Eq. (2)) and \mathcal{L}_{feat} (Eq. (4)) is 0.5 on ImageNet-derived datasets, and is 0.1 on CIFAR-derived datasets. We adopt the data augmentation methods used in SupCT [19] and include image rotation prediction for reducing the bias [11]. For each dataset, hyper-parameter settings are the same for both *Partner* and *Main Encoder*.

4.2. Comparison With State-Of-The-Art

We compare the performance of PAL with state-of-the-art (SOTA) methods. The idea of multi-task training is adopted in [31] and combines the objectives of classification task and 5-way few-shot task during training. Similarly, a strong baseline is first obtained through pretraining, and the idea of transfer learning is then used to perform training on hard-task [42] or through finetuning [8]. Recently, knowledge distillation has been implemented by [45] and improves the performance on few-shot tasks clearly. Despite the success of previous work, PAL outperforms SOTA methods on all four benchmark datasets in both 1-shot and 5-shot scenarios, which demonstrates the advantages of the proposed PAL learning scheme, where we train *Partner Encoder* and *Main Encoder* under different objective and impose constraint by logit-level alignment and feature-level alignment. Besides, AdaMargin [23] introduces external

Table 2: PAL Result on CIFAR-FS and FC100 datasets.

Algorithm	Backbone	CIFAR-FS, 5-way		FC100, 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [10]	32-32-32-32	58.9 ± 1.9	71.5 ± 1.0	-	-
Prototypical Networks [41]	64-64-64-64	55.5 ± 0.7	72.0 ± 0.6	35.3 ± 0.6	48.6 ± 0.6
Relation Networks [43]	64-96-128-256	55.0 ± 1.0	69.3 ± 0.8	-	-
R2D2 [2]	96-192-384-512	65.3 ± 0.2	79.4 ± 0.1	-	-
TADAM [31]	ResNet-12	-	-	40.1 ± 0.4	56.1 ± 0.4
Shot-Free [33]	ResNet-12	69.2 ± <i>n/a</i>	84.7 ± <i>n/a</i>	-	-
TEWAM [32]	ResNet-12	70.4 ± <i>n/a</i>	81.3 ± <i>n/a</i>	-	-
Prototypical Networks [41]	ResNet-12	72.2 ± 0.7	83.5 ± 0.5	37.5 ± 0.6	52.5 ± 0.6
Boosting [11]	WRN-28-10	73.6 ± 0.3	86.0 ± 0.2	-	-
MetaOptNet [22]	ResNet-12	72.6 ± 0.7	84.3 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
Associative-Alignment [1]	ResNet-18	-	-	45.8 ± 0.5	59.7 ± 0.6
DeepEMD [50]	ResNet-12	-	-	46.5 ± 0.8	63.2 ± 0.7
MABAS [20]	ResNet-12	73.5 ± 0.9	85.5 ± 0.7	42.3 ± 0.8	57.6 ± 0.8
RFS-simple [45]	ResNet-12	71.5 ± 0.8	86.0 ± 0.5	42.6 ± 0.7	59.1 ± 0.6
RFS-distill [45]	ResNet-12	73.9 ± 0.8	86.9 ± 0.5	44.6 ± 0.7	60.9 ± 0.6
PAL (Ours)	ResNet-12	77.1 ± 0.7	88.0 ± 0.5	47.2 ± 0.6	64.0 ± 0.6

structured semantic knowledge to model the relationship between classes, and meta-learn a discriminative feature space. However, our method still achieves better result on the few-shot tasks. Furthermore, by comparing the performance boost between mini-ImageNet and tieredImagenet, and between CIFAR-FS and FC100, we notice that PAL’s advantage could be better revealed on the dataset without hierarchical structure (mini-ImageNet, CIFAR-FS), but can still facilitate the few-shot tasks with hierarchical structure (tieredImagenet and FC100).

4.3. Discussion

Partner-Assisted Learning involves supervised contrastive loss and cross-entropy loss during training. A clear uni-direction from the *Partner Encoder* f_P to the *Main Encoder* f_M is set by sampling features of f_P as soft anchors to assist the training of f_M in addition to class labels. To this end, we study the alternatives of PAL by (1) altering the integration direction of the two objective types and (2) changing the objective of *Partner Encoder* training. We also studied (3) the impact of different alignment losses, which serves as a comparison with knowledge distillation.

Integration of Objectives: PAL uses \mathcal{L}_{SupCT} to pretrain f_P to extract soft-anchors, which are then used to regularize f_M trained by \mathcal{L}_{CE} , i.e., $\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$. We study the variants by altering the integration direction and have

- Uni-direction $\mathcal{L}_{CE} \rightarrow \mathcal{L}_{SupCT}$, $\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$,
- Mutual-learning $\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$: contemporarily train two networks from scratch and use the model under \mathcal{L}_{CE} for evaluation,
- Multi-task learning on one network: $\mathcal{L}_{SupCT} + \mathcal{L}_{CE}$,

Table 3: Ablation study of mini-ImageNet on the performance by different training scheme combining two objectives. Our method works best on the few-shot classification.

Train Scheme	5-Way Few-shot		Base
	1-Shot	5-Shots	
\mathcal{L}_{CE}	63.76	81.17	80.90
\mathcal{L}_{SupCT}	62.29	76.32	<i>n/a</i>
$\mathcal{L}_{CE} + \mathcal{L}_{SupCT}$	67.53	82.14	83.20
$\mathcal{L}_{SupCT} \leftrightarrow \mathcal{L}_{CE}$	65.21	81.53	80.13
$\mathcal{L}_{CE} \rightarrow \mathcal{L}_{SupCT}$	66.54	81.83	80.39
$\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$ [ours]	69.37	84.40	82.98

* Single objective \mathcal{L}_{SupCT} don’t train a base classifier.

- Single-Objective training: \mathcal{L}_{CE} , \mathcal{L}_{SupCT} .

As shown in Table 3, we observe that the network trained using PAL ($\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$) generalizes best with a clear margin on the few-shot classification on novel classes. On the base classes test samples, PAL continues to achieve high top-1 accuracy that is very close to the best score by multi-task learning, model trained with $(\mathcal{L}_{SupCT} + \mathcal{L}_{CE})$, and PAL clearly outperforms the rest methods. Meanwhile, compared to the two single-objective methods, even though the model trained by \mathcal{L}_{SupCT} is not as strong as the model trained by \mathcal{L}_{CE} , it can still be used to regularize the training under \mathcal{L}_{CE} and boost the performance.

Impact of Partner Encoder on Main Encoder is studied by evaluating the performance of *Main Encoder*, i.e., accuracy of few-shot tasks and conventional classification task on base test data. We select *Partner Encoders* trained by \mathcal{L}_{CE} , \mathcal{L}_{CT} , and \mathcal{L}_{SupCT} for comparison. Both \mathcal{L}_{feat} and \mathcal{L}_{logit} are used for all three methods for fair comparison.

As shown in Table 4, the alignment losses consistently introduce performance improvements on few-shot tasks, and our method ($\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$) continues to achieve the best performance on both few-shot and fully-supervised tasks. Since \mathcal{L}_{CT} does not utilize the class label information during training, the extracted soft-anchors are not as discriminative as the other two methods and the performance improvement is limited. Meanwhile, if both the *Main Encoder* and the *Partner Encoder* are trained under the \mathcal{L}_{CE} , the features are mainly learned w.r.t hard-anchors. Therefore, the knowledge is similar for the two networks and the performance improvement is limited. In contrast, supervised contrastive loss includes both the label information and pair-wise similarity comparison among all features, so that it could preserve more details to facilitate the training of *Main Encoder*.

Alignment versus Knowledge Distillation: Although teacher-student model with knowledge distillation can train a good feature extractor for few-shot prototype classification, a good teacher is a critical prerequisite for training the student. As discussed in Section 3.3.1, the negative entropy of teacher model is minimized if the teacher model is also tuned during distillation. To this end, as shown in Table 5, we quantitatively investigate the effectiveness of both \mathcal{L}_{feat} and \mathcal{L}_{logit} by applying various combinations. We also compare with the performance under teacher-student setting and set the KL-divergence as the logit-level loss.

By comparing Row_{1,2,4,5}, using either \mathcal{L}_{feat} or \mathcal{L}_{logit} can regularize the training of f_M , and clearly improve performance on both few-shot and fully-supervised tasks. Using \mathcal{L}_{feat} only can even outperform the multi-task training (Row₃ in Table 3). Meanwhile, adding these two losses can achieve the best scores on few-shot tasks. By comparing Row_{2,3} or Row_{5,6}, minimizing the negative entropy of outputs from f_P has negative impact on the training of f_M . As the f_P is pretrained to perform pair-wise comparison, the distance between all features and all feature clusters has already been modeled. Since f_P is not trained to do classification, the details irrelevant to classification on base domain could be preserved. Therefore, minimizing the negative entropy of f_P may make probability output of f_P less sensitive to the instance-level distances and then embed the soft-anchors with more uncertainty. Even though the *Partner Encoder* (trained under \mathcal{L}_{SupCT}) is not as strong as the teacher model (trained under \mathcal{L}_{CE}) in few-shot tasks, it can provide meaningful soft-anchors to assist the training of *Main Encoder* in our proposed PAL framework.

PAL versus Mutual Learning: Mutual learning trains two peer networks jointly from scratch either under the same task [52] or different tasks (Row₄ in Table 3). The alignment between probability distribution outputs is applied [52]. Even though the two networks are expected to learn in different direction, the optimization direction is still

Table 4: Ablation study of mini-ImageNet on the *Main Encoder* performance affected by different *Partner Encoder*. The *Partner Encoder* trained by \mathcal{L}_{SupCT} benefits the training of *Main Encoder* most.

Loss of f_P	5-Way Few Shot		Base
	1-Shot	5-Shots	
-	63.76 ± 0.62	81.17 ± 0.45	80.90
$\mathcal{L}_{CT} \rightarrow \mathcal{L}_{CE}$	65.89 ± 0.67	80.84 ± 0.48	80.66
$\mathcal{L}_{CE} \rightarrow \mathcal{L}_{CE}$	66.95 ± 0.65	81.54 ± 0.48	81.45
$\mathcal{L}_{SupCT} \rightarrow \mathcal{L}_{CE}$	69.37 ± 0.64	84.40 ± 0.44	82.98

Table 5: Ablation study on mini-ImageNet on alignment losses. Both feature-level and logit-level losses facilitates the training of *Main Encoder* consistently.

Alignment losses		5-way few-shot		Base
feature	logit	1-shot	5-shots	
-	-	63.76 ± 0.62	81.17 ± 0.45	80.90
-	\mathcal{L}_{logit}	66.35 ± 0.63	81.32 ± 0.46	82.50
-	\mathcal{L}_{KL}	64.76 ± 0.62	80.58 ± 0.47	81.98
\mathcal{L}_{feat}	-	68.03 ± 0.63	83.38 ± 0.44	83.94
\mathcal{L}_{feat}	\mathcal{L}_{logit}	69.37 ± 0.64	84.40 ± 0.44	82.98
\mathcal{L}_{feat}	\mathcal{L}_{KL}	68.75 ± 0.62	82.83 ± 0.45	83.40

not clearly modeled in such framework. In contrast, our method sets the features extracted by *Partner Encoder* as soft-anchors to model the distance of positive pair and negative pair, and then train *Main Encoder* through alignment.

5. Conclusion

In this paper, we have proposed the *Partner Assisted Learning* (PAL) to obtain an essential feature extractor for few-shot classification. We pre-train the *Partner Encoder* with supervised contrastive learning to obtain soft-anchors. Then, we fix the *partner* model and impose the constraints at either feature-level or logit-level to train the *Main Encoder* from scratch while seeking for classification. With the *main* model, both the classification accuracy on novel classes (few-shot) and on base classes (large-scale) are improved. Detailed ablation study is performed to compare potential variants of PAL and our method outperforms clearly all variants on few-shot tasks. Experiments on four benchmarks demonstrate the effectiveness of our method.

Acknowledgements

This material is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation therein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Laboratory, DARPA or the U.S. Government.

References

- [1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 18–35, Cham, 2020. Springer International Publishing. 6, 7
- [2] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. 5, 6, 7
- [3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 1
- [4] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1052, 2018. 2
- [5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 3
- [6] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020. 1, 2, 3
- [7] 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. 6
- [8] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019. 1, 2, 6
- [9] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6
- [10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 1, 6, 7
- [11] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8059–8068, 2019. 6, 7
- [12] 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. 6
- [13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. 1
- [14] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2
- [15] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. *arXiv preprint arXiv:2104.07719*, 2021. 2
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 4, 6
- [18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3, 4
- [19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 2, 3, 6
- [20] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 599–617, Cham, 2020. Springer International Publishing. 6, 7
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 1
- [22] 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. 6, 7
- [23] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12576–12584, 2020. 1, 2, 6
- [24] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. 1
- [25] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 507–516, 2016. 1, 2
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1
- [27] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. 6
- [28] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Learning Representations*, 2018. 6

- [29] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. 2
- [30] Alex Nichol and John Schulman. Reptile: a scalable meta-learning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018. 1
- [31] 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. 1, 2, 5, 6, 7
- [32] L. Qiao, Y. Shi, J. Li, Y. Tian, T. Huang, and Y. Wang. Transductive episodic-wise adaptive metric for few-shot learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3602–3611, 2019. 6, 7
- [33] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6, 7
- [34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1
- [35] Mengye Ren, Eleni Triantafyllou, 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 Representations ICLR*, 2018. 5, 6
- [36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1
- [37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 1
- [38] 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. 1
- [39] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. 6
- [40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2
- [41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 1, 2, 6, 7
- [42] 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. 6
- [43] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 7
- [44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017. 3
- [45] 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. 1, 2, 3, 6, 7
- [46] 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. 1, 5, 6
- [47] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2
- [48] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 499–515, Cham, 2016. Springer International Publishing. 2
- [49] Hanchen Xie, Mohamed E. Hussein, Aram Galstyan, and Wael Abd-Almageed. Muscle: Strengthening semi-supervised learning via concurrent unsupervised learning using mutual information maximization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2586–2595, January 2021. 1
- [50] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6, 7
- [51] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6
- [52] 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. 3, 8
- [53] Yutong Zheng, Dipan K. Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2