Rethinking Clustering-Based Pseudo-Labeling for Unsupervised Meta-Learning

Xingping Dong[1][0000−0003−1613−9288], Jianbing Shen[2⋆][0000−0003−1883−2086], and Ling Shao[3][0000−0002−8264−6117]

[1] Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
[2] SKL-IOTSC, Computer and Information Science, University of Macau
[3] Terminus Group, China
{xingping.dong, shenjianbingcg}@gmail.com, ling.shao@ieee.org

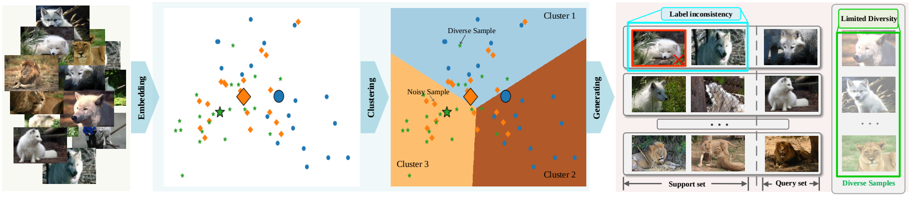
Abstract. The pioneering method for unsupervised meta-learning, CACTUs, is a clustering-based approach with pseudo-labeling. This approach is model-agnostic and can be combined with supervised algorithms to learn from unlabeled data. However, it often suffers from label inconsistency or limited diversity, which leads to poor performance. In this work, we prove that the core reason for this is lack of a clustering-friendly property in the embedding space. We address this by minimizing the inter- to intra-class similarity ratio to provide clustering-friendly embedding features, and validate our approach through comprehensive experiments. Note that, despite only utilizing a simple clustering algorithm (k-means) in our embedding space to obtain the pseudo-labels, we achieve significant improvement. Moreover, we adopt a progressive evaluation mechanism to obtain more diverse samples in order to further alleviate the limited diversity problem. Finally, our approach is also model-agnostic and can easily be integrated into existing supervised methods. To demonstrate its generalization ability, we integrate it into two representative algorithms: MAML and EP. The results on three main few-shot benchmarks clearly show that the proposed method achieves significant improvement compared to state-of-the-art models. Notably, our approach also outperforms the corresponding supervised method in two tasks. The code and models are available at https://github.com/xingpingdong/PL-CFE.

Keywords: Meta-learning; Unsupervised learning; Clustering-friendly

1 Introduction

Recently, few-shot learning has attracted increasing attention in the *machine learning* and *computer vision* communities [32,34,14,40,42]. It is also commonly used to evaluate meta-learning approaches [22,44,16]. However, most of the existing literature focuses on the supervised few-shot classification task, which is built upon datasets with human-specified labels. Thus, most previous works cannot

[⋆] Corresponding author: *Jianbing Shen*



(a) Images (b) PCA Space (c) Clusters (d) N-ways-K-shots task

Fig. 1. Illustration of the *label inconsistency* and *limited diversity* issues in the clustering-based CACTUs [25]. (a) The unlabeled images. **(b)** The 2D mapping space of the embedding features, generated via principal component analysis (PCA). Each mark (or color) represents one class, and the larger marks are the class centers (*i.e.* the average features in a class). **(c)** The noisy clustering labels generated by *k-means*, with many inconsistent or noisy samples in each cluster. For example, we use the green class label for cluster 3, since cluster 3 has the most green-samples. Thus, the samples with inconsistent labels, like the orange samples in cluster 3, are regarded as noisy samples. Further, in the unsupervised setting, the green samples in other regions (cluster 1 and cluster 2) cannot be used to build few-shot tasks. This leads to the unsupervised few-shot tasks being less diverse than the supervised tasks. We term this issue *limited diversity* and refer to these green samples as diverse samples. **(d)** The *label inconsistency* and *limited diversity* problems in the few-shot task, which are caused by the noisy clustering labels, as shown in (c). The red box highlights an incorrect sample in the few-shot task, and the green box illustrates the limited diversity problem, *i.e.*, the diverse samples cannot be used to build few-shot tasks.

make use of the rapidly increasing amount of unlabeled data from, for example, the internet.

To solve this issue, the pioneering work CACTUs [25], was introduced to automatically construct the few-shot learning tasks by assigning pseudo-labels to the samples from unlabeled datasets. This approach partitions the samples into several clusters using a clustering algorithm (*k-means*) on their embedding features, which can be extracted by various unsupervised methods [21,4,38,49,15]. Unlabeled samples in the same clusters are assigned the same pseudo-labels. Then, the supervised meta-learning methods, MAML [16] and ProtoNets [46], are applied to the few-shot tasks generated by this pseudo-labeled dataset. It is worth mentioning that CACTUs is model-agnostic and any other supervised methods can be used in this framework.

However, this approach based on clustering and pseudo-label generation suffers from *label inconsistency*, *i.e.* samples with the same pseudo-label may have different human-specified labels in the few-shot tasks (*e.g.* Fig. 1(d)). This is caused by noisy clustering labels. Specifically, as shown in Fig. 1(c), several samples with different human-specified labels are partitioned into the same cluster, which leads to many noisy samples (with different labels) in the few-shot tasks. This is one reason for performance degeneration. Besides, noisy clustering labels will also partition samples with the same label into different clusters, which results in a lack of diversity for the few-shot tasks based on pseudo-labels compared with supervised methods. As shown in Fig. 1 (c)(d), CACTUs ignores the diversity among partitioned samples with the same human label. This is termed the *limited diversity* problem. How to utilize the diversity is thus one critical issue for performance improvement.

To overcome the above problems, we first analyze the underlying reasons for the noisy clustering labels in depth, via a qualitative and quantitative comparison bewteen the embedding features of CACTUs. As shown in Fig. 1(b), we find that the embedding features extracted by unsupervised methods are not clustering-friendly. In other words, most samples are far away from their class centers, and different class centers are close to each other in the embedding space. Thus, a standard clustering algorithms cannot effectively partition samples from the same class into one cluster, leading to noisy clustering labels. For example, cluster 3 in Fig. 1(c) contains many noisy samples from different classes. Furthermore, we propose an inter- to intra-class similarity ratio to measure the clustering-friendly property. In the quantitative comparison, we observe that the accuracy of the few-shot task is inversely proportional to the similarity ratio. This indicates that reducing the similarity ratio is critical for performance improvement.

According to these observations, a novel pseudo-labeling framework based on a clustering-friendly feature embedding (PL-CFE) is proposed to construct few-shot tasks on unlabeled datasets, in order to alleviate the *label inconsistency* and *limited diversity* problems. Firstly, we introduce a new unsupervised training method to extract clustering-friendly embedding features. Since the similarity ratio can only be applied to labeled datasets, we simulate a labeled set via data augmentation and try to minimize the similarity ratio on this to provide a clustering-friendly embedding function. Given our embedding features, we can run *k-means* to generate several clusters for pseudo-labeling and build clean few-shot tasks, reducing both the *label inconsistency* and *limited diversity* problems. Secondly, we present a progressive evaluation mechanism to obtain more divisive samples and further alleviate the *limited diversity*, by utilizing additional clusters for the task construction. Specifically, for each cluster, which we call a base cluster, in the task construction, we choose its *k-nearest* clusters as candidates. We use an evaluation model based on previous meta-learning models to measure the entropy of the candidates, and select the one with the highest entropy as the additional cluster for building a hard task, as it contains newer information for the current meta-learning model.

To evaluate the effectiveness of the proposed PL-CFE, we incorporate it into two representative supervised meta-learning methods: MAML [25] and EP [42], termed as PL-CFE-MAML and PL-CFE-EP, respectively. We conduct extensive experiments on Omniglot [30], *mini*ImageNet [39], *tiered*ImageNet [41]. The results demonstrate that our approach achieves significant improvement compared with state-of-the-art model-agnostic unsupervised meta-learning methods. In particular, our PL-CFE-MAML outperforms the corresponding supervised MAML method on *mini*ImageNet in both the 5-ways-20-shots and 5-ways-50-shots tasks. Notably, we achieve a gain of 1.75% in the latter task.

2 Related Work

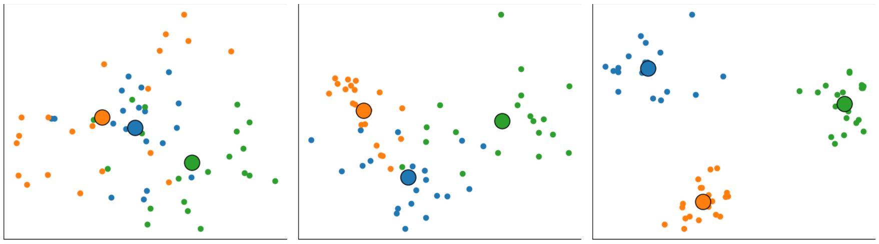
2.1 Meta-Learning for Few-Shot Classification

Meta-learning, whose inception goes as far back as the 1980s [22,44], is usually interpolated as *fast weights* [22,3], or *learning-to-learn* [44,48,23,1]. Recent meta-learning methods can be roughly split into three main categories. The first kind of approaches is metric-based [29,50,46,42,7], which attempt to learn discriminative similarity metrics to distinguish samples from the same class. The second kind of approaches are memory-based [43,39], investigating the storing of key training examples with effective memory architectures or encoding fast adaptation methods. The last kind of approaches are optimization-based methods [16,17], searching for adaptive initialization parameters, which can be quickly adjusted for new tasks. Most meta-learning methods are evaluated under supervised few-shot classification, which requires a large number of manual labels. In this paper, we explore a new approach of few-shot task construction for unsupervised meta-learning to reduce this requirement.

2.2 Unsupervised Meta-Learning

Unsupervised learning aims to learn previously unknown patterns in a dataset without manual labels. These patterns learned during unsupervised pre-training can be used to more efficiently learn various downstream tasks, which is one of the more practical applications [21,4,38,49,15]. Unsupervised pre-training has achieved significant success in several fields, such as image classification [55,19], speech recognition [54], machine translation [37], and text classification [10,24,36].

Hsu *et al.* [25] proposed an *unsupervised meta-learning method* based on clustering, named CACTUs, to explicitly extract effective patterns from small amounts of data for a variety of tasks. However, as mentioned, this method suffers from the *label inconsistency* and *limited diversity* problems, which our approach can significantly reduce. Subsequently, Khodadadeh *et al.* [27] proposed the UMTRA method to build synthetic training tasks in the meta-learning phase, by using random sampling and augmentation. Stemming from this, several works have been introduced to synthesize more generative and divisive meta-training tasks via various techniques, such as introducing a distribution shift between the support and query set [35], using latent space interpolation in generative models to generate divisive samples [28], and utilizing a variational autoencoder with a mixture of Gaussian for producing meta tasks and training [31]. Compared to these synthetic methods, our approach can obtain harder few-shot tasks.Specifically, these previous methods only utilize the differences between augmented samples to increase the diversity of tasks, while our method can use the differences inside the class. Besides, some researchers explore to apply clustering methods for meta-training, by prototypical transfer learning [33] or progressive clustering [26]. These methods are not model-agnostic, while our approach can be incorporated into any supervised meta-learning method. This will bridge the unsupervised and supervised methods in meta-learning.



(a) BiGAN (b) ACAI (c) CFE (ours)

Fig. 2. Illustrations of 2D mapping of different embeddings, including BiGAN [11], ACAI [5], and our clustering-friendly embedding (CFE) features. We randomly select three classes from Omniglot [30] and map the embeddings of their samples in 2D space via principal component analysis (PCA). Each color represents one class, and large circles are the class centers.

3 In-Depth Analysis of Clustering-Based Unsupervised Methods

CACTUs [25] is a clustering-based and model-agnostic algorithm, which is easily incorporated into supervised methods for the unsupervised meta-learning task. However, there is still a large gap in performance between CACTUs and the supervised methods. We believe that the main reason is that the unsupervised embedding algorithms, such as InfoGAN [8], BiGAN [11], ACAI [5] and DC [6], in CACTUs are not suitable for the clustering task, which is a core step of CACTUs. This is because these unsupervised methods were originally designed for the pre-training stage, where the extracted features can be further fine-tuned in the downstream tasks. Thus, they do not need to construct a clustering-friendly feature space, where the samples in the same class are close to their class center and each class center is far away from the samples in other classes. However, the clustering-friendly property is very important for CACTUs, since it directly clusters the embedding features without fine-tuning.

We first provide an intuitive analysis by visualizing the embedding features. We collect samples from three randomly selected classes of a dataset and map them into 2D space with principal component analysis (PCA). For example, we observe the BiGAN and ACAI embedding features on Omniglot [30]. As shown in Fig. 2(a)(b), many samples are far away from their class centers, and the class centers are close to each other. Thus, we can see that many samples in different classes are close to each other. These samples are difficult for a simple clustering method to partition.

In order to observe the embedding features in the whole dataset, we define two metrics, *intra-similarity* and *inter-similarity*, to measure the clustering performance. We define the *intra-similarity* for each class as follows:

$$s_i^{\text{intra}} = exp(\sum\nolimits_{j=1}^{N_s} \boldsymbol{\mu}_i \cdot \mathbf{z}_{ij} / (\tau N)), \tag{1}$$

where $\mathbf{z}_{ij}$ is the embedding feature of the j-th sample in class i, $\boldsymbol{\mu}_i = \frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{z}_{ij}$ is the class center, $\cdot$ is the dot product, N_s is the number of samples in a class,

Embedding	Dataset	$s^{\text{inter}} \downarrow$	$s^{\text{intra}} \uparrow$	$R \downarrow$	Acc (%) $\uparrow$
BIGAN [11]	Omni	**1.032**	2.690	0.449	58.18
ACAI [5]	Omni	5.989	16.33	0.413	68.84
CFE (ours)	Omni	1.316	**20.10**	**0.081**	**91.32**
DC [6]	Mini	**0.999**	1.164	0.862	39.90
CFE (ours)	Mini	1.048	**1.680**	**0.641**	**41.99**

Table 1. Illustrations of the relationship between similarity ratio R and classification accuracy (Acc). We present the average *intra-similarity* s^{intra}, *inter-similarity* s^{inter}, and similarity ratio R of different embedding methods on Omniglot [30](Omni) and *mini*ImageNet [39](Mini). We also report the accuracy of MAML [16] based on these embeddings for the 5-ways-1-shot task. All Acc values except ours are sourced from CACTUs [25].

and τ is a temperature hyperparameter [53]. The *intra-similarity* is the average similarity between the samples and their class center. It is used to evaluate the compactness of a class in the embedding space. A large value indicates that the class is compact and most samples are close to the class center.

The other metric, *inter-similarity*, is defined as follows:

$$s_{ij}^{\text{inter}} = exp(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j / \tau), \ j \neq i. \tag{2}$$

A low value of s_{ij}^{inter} indicates that two classes are far away from each other.

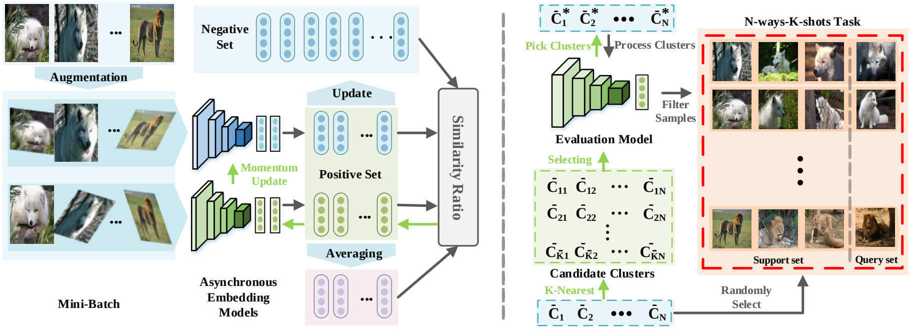
To combine the above two similarities, we use the inter- to intra-class similarity ratio $r_{ij} = s_{ij}^{\text{inter}} / s_i^{\text{intra}}$ to represent the clustering performance. Finally, the average similarity ratio R over the whole dataset is denoted as:

$$R = \frac{1}{C} \sum\nolimits_{i=1}^{C} \frac{\sum\nolimits_{j \neq i}^{C} s_{ij}^{\text{inter}}}{(C-1) s_i^{\text{intra}}}, \tag{3}$$

where C is the number of classes. The lower the value of R, the better the clustering performance. In addition to R, we also calculate the average *intra-similarity* $s^{\text{intra}} = \sum s_i^{\text{intra}} / C$ and average *inter-similarity* $s^{\text{inter}} = \sum s_{ij}^{\text{inter}} / (N_s C)$ for complete analysis, where N_s is the samples' number of a class.

As shown in Table 1, we apply the above three criteria, R, s^{intra} and s^{inter}, to the different embedding features (BiGAN [11] and ACAI [5]) in the Omniglot [30] dataset, and report the accuracy on the 5-ways-1-shot meta-task with the CACTUs-MAML [25] method. We find that the accuracy is inversely proportional to the similarity ratio R, but has no explicit relationship with the individual similarities s^{intra} or s^{inter}. This indicates that minimizing R is critical for better accuracy.

Therefore, we propose a novel clustering-friendly embedding method (CFE) to reduce the similarity ratio in the following subsection §4.1. In the visualization comparison of Fig. 2, the proposed method provides more compact classes than the previous BiGAN and ACAI. As shown in Table 1, our CFE approach also has significantly reduced R on both the Omniglot and *mini*ImageNet. These results indicate that our method is more clustering-friendly. To investigate the relationship between the clustering-friendly property and the accuracy on the meta-task,


Negative Set
Augmentation
Update
Momentum Update
Positive Set
Averaging
Similarity Ratio
Mini-Batch
Asynchronous Embedding Models
$\bar{C}_1^*$ $\bar{C}_2^*$ ••• $\bar{C}_N^*$
Pick Clusters
Process Clusters
N-ways-K-shots Task
Evaluation Model
Filter Samples
Selecting
$\bar{C}_{11}$ $\bar{C}_{12}$ ⋯ $\bar{C}_{1N}$
$\bar{C}_{21}$ $\bar{C}_{22}$ ⋯ $\bar{C}_{2N}$
$\bar{C}_{\bar{K}1}$ $\bar{C}_{\bar{K}2}$ ⋯ $\bar{C}_{\bar{K}N}$
Candidate Clusters
K-Nearest
$\bar{C}_1$ $\bar{C}_2$ ••• $\bar{C}_N$
Randomly Select
Support set
Query set

(a) Clustering-Friendly Feature Embedding (b) Progressive Evaluation Mechanism

Fig. 3. The frameworks of our clustering-friendly feature embedding and progressive evaluation mechanism. (a) The embedding feature extraction. We first augment the samples in one mini-batch and feed them into the asynchronous embedding models to build a positive set. We also use the negative set storing the historical embedding features to obtain more varied features. Then we update our main embedding model (green) with backpropagation (green arrows), by minimizing the similarity ratio between the average features (purple ellipses), and the features in the positive and negative sets. The historical encoder is momentum updated with the main model. **(b) The progressive evaluation mechanism.** We randomly select N clusters as base clusters ($\bar{C}_i$), and choose the $\bar{K}$-nearest neighbors as the candidates. Then, we use an evaluation model to select the clusters ($\bar{C}_i^*$) with the highest entropy and filter out noisy samples.

we firstly use *k-means* to split the samples into different clusters, and assign the same pseudo-label to samples in one cluster. Then we run the supervised MAML method on the 5-ways-1-shot task constructed by pseudo-labeled samples. Compared with CACTUs, which is based on previous embedding methods, our approach achieves significant improvement on both Omniglot and *mini*ImageNet. Notably, the gain on Omniglot is more than 20%. The results clearly support our claim: a clustering-friendly embedding space can benefit clustering-based unsupervised meta-learning methods.

4 Our Approach

We propose a novel clustering-based pseudo-labeling for unsupervised meta-learning, which includes a clustering-friendly feature embedding and a progressive evaluation mechanism. Fig. 3 shows the frameworks of our approach.

4.1 Clustering-Friendly Feature Embedding

Optimization Objective. We aim to learn a feature extraction function to map each sample to a clustering-friendly embedding space (with low similarity ratio). In this space, samples with the same class label are close to their class center, and each class center is far away from samples with different class labels.

We can assign most samples from the same class into the same cluster, even with a simple clustering algorithm, such as *k-means*.

In the unsupervised setting, we do not have access to the class labels. Thus, we need to simulate a labeled dataset. To do so, we first randomly select $N_p \ll N_{all}$ samples from the unlabeled dataset $\mathcal{D}$ to build a positive set $\mathcal{D}_p$, where N_{all} is the size of $\mathcal{D}$. For each sample $\mathbf{x}_i \in \mathcal{D}_p$, we produce N_a augmented samples $\{\mathbf{x}^+_{ij}, j \in [1, N_a]\}$ via data augmentation methods, such as random color jittering, random horizontal flipping, and random grayscale conversion. Samples augmented from the same original sample are regarded as belonging to the same class. To involve more samples for training, we also construct a negative set by randomly selecting N_n samples from $\mathcal{D}$, without including the N_p positive samples. We augment each sample once to obtain the final negative set $\{\mathbf{x}^-_k, k \in [1, N_n]\}$. Given the positive ('labeled') set and negative set, we can reformulate the *intra-similarity* in Eq. (1) and *inter-similarity* in Eq. (2), for the final optimization objective. Maximizing the former will force the samples to be close to their class center, while minimizing the latter will pull the class centers far away from the other class center and negative samples.

We rewrite the *intra-similarity* for each class:

$$s^{\text{intra}}_i = exp(\sum\nolimits_{j=1}^{N_a} \boldsymbol{\mu}_i \cdot \mathbf{z}^+_{ij}/(\tau N_a)), \tag{4}$$

where $\mathbf{z}^+_{ij} = \phi(\mathbf{x}^+_{ij}; \boldsymbol{\omega})$ is the embedding feature and $\boldsymbol{\mu}_i = \frac{1}{N_a}\sum\nolimits_j^{N_a} \mathbf{z}^+_{ij}$ is the class center. ϕ represents the embedding function and $\boldsymbol{\omega}$ is its parameter. This embedding function can be any learnable module, such as a convolutional network. The rewritten *inter-similarity* includes a negative-similarity measuring the similarity between the class center and negative sample, and a center-similarity calculating the similarity bewteen the class centers. The formulations are as follows:

$$s^{\text{neg}}_{ik} = exp(\boldsymbol{\mu}_i \cdot \mathbf{z}^-_k/\tau),\ \ s^{\text{cen}}_{ij} = exp(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j/\tau), \tag{5}$$

where $\mathbf{z}^-_k = \phi(\mathbf{x}^-_k; \boldsymbol{\omega})$ is the negative embedding feature.

To utilize the common losses from standard deep learning tools, such as PyTorch [47], we incorporate the logarithm function with the similarity ratio R in Eq. (3) to obtain the minimization objective, as follows:

$$L = \frac{1}{N_p}\sum\nolimits_{i=1}^{N_p} \log \frac{1}{N_o}(1 + \frac{\sum\nolimits_{j\neq i}^{N_p} s^{\text{cen}}_{ij} + \sum\nolimits_{k=1}^{N_n} s^{\text{neg}}_k}{s^{\text{intra}}_i}), \tag{6}$$

where $N_o = N_p + N_n - 1$. Similar to R in Eq. (3), a low value of L in Eq. (6) means that the current embedding space is clustering-friendly. Thus, we can partition most samples from the same class into the same cluster, using a simple clustering algorithm. A low value of L can also reduce the label inconsistency problem in the clustering-based unsupervised meta-learning.

Asynchronous Embedding. If we only use a single encoder function ϕ for embedding learning, it is difficult to produce various embedding features for the samples augmented from the same original sample. Thus, we propose to utilize two asynchronous encoders, $\phi(\cdot; \boldsymbol{\omega})$ and $\bar{\phi}(\cdot; \bar{\boldsymbol{\omega}})$ for training to avoid fast convergence caused by similar positive samples. The first function ϕ is the main encoder, which is updated with the current samples (in a mini-batch). In other words, we use gradient backpropagation to update its parameter $\boldsymbol{\omega}$. The latter $\bar{\phi}$ is the history encoder, which collects information from the main encoders in previous mini-batches. To ensure that the history encoders on-the-fly are smooth, *i.e.*, the difference among these encoders can be made small [19], we use a momentum coefficient $m \in [0, 1)$ to update the encoder parameter $\bar{\boldsymbol{\omega}} = m\bar{\boldsymbol{\omega}} + \boldsymbol{\omega}$.

To reduce the computational load, only one sample from each class is encoded by the main encoder ϕ, and the others are encoded by the history encoder $\bar{\phi}$. Without loss of generality, we encode the first augmented sample in each class. Then, the embedding features of the positive dataset $\{\mathbf{x}_{ij}^{+}, i \in [1, N_p], j \in [1, N_a]\}$ are reformulated as follows:

$$\mathbf{z}_{ij}^{+} = \begin{cases} \phi(\mathbf{x}_{ij}^{+}; \boldsymbol{\omega}), j = 1, \\ \bar{\phi}(\mathbf{x}_{ij}^{+}; \bar{\boldsymbol{\omega}}), j \neq 1. \end{cases} \tag{7}$$

For the negative set, a naive approach would be to randomly select samples from the unlabeled dataset and then encode them with the history encoder. However, this solution is time-consuming and does not make full use of the different history encoders from previous mini-batches. Inspired by [19], we use a queue to construct the negative set by inserting the embedding features encoded by the history encoders. Specifically, we only insert one historical embedding of each class into the queue to maintain the diversity of the negative set. We remove the features of the oldest mini-batch to maintain the size of the negative set, since the oldest features are most outdated and least inconsistent with the new ones. Using the queue mechanism can reduce the computational cost and also remove the limit on the mini-batch size.

Clustering and Meta-Learning. With the above training strategy, we can provide embedding features for the samples in the unlabeled dataset. Similar to CACTUs [25], we can then apply a simple clustering algorithm, *e.g k-means*,to these features to split the samples into different clusters, denoted as $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{N_s}\}$. We assign the same pseudo-label to all samples in a cluster. Thus, we obtain the pseudo-labeled dataset $\{(\mathbf{x}_i, \bar{y}_i), \mathbf{x}_i \in \mathcal{D}\}$, where $\bar{y}_i$ is the pseudo-label obtained by the clusters, *i.e.* $\bar{y}_i = k$ if $\mathbf{x}_i \in \mathcal{C}_k$. Then, we can use any supervised meta-learning algorithm on this dataset to learn an efficient model for the meta-tasks.

4.2 Progressive Evaluation Mechanism

Although our embedding approach yields promising improvements, it still suffers from the *limited diversity* problem. In other words, we cannot generate meta-tasks using the divisive samples, which are far away from their class center, and easily assigned the wrong label by the simple clustering algorithm. However, the supervised methods can utilize these samples to generate hard meta-tasks to enhance the model's discrimination ability. Thus, we propose a progressive evaluation mechanism to produce similar hard meta-tasks with supervised methods.

As mentioned before, our embedding approach will push samples close to their class centers. For each cluster, denoted as the base cluster, we assume that the divisive samples are in its $\bar{K}$-nearest-neighbors, which are denoted as the candidate clusters. Thus, we can try to use the samples in the candidate clusters to generate hard meta-tasks. To do so, we build an evaluation model using the meta-learning models in previous iterations to evaluate the candidates. To obtain a stable evaluation model, only the meta-training models trained at the end of an epoch are utilized. We denote the evaluation model as $f(\mathbf{x};\bar{\boldsymbol{\theta}})$, where $\bar{\boldsymbol{\theta}}$ is the corresponding model parameter. Then, we select those with high information entropy as the final clusters, and choose the samples from the base and final clusters to generate hard meta-tasks, by filtering out noise.

Selecting Clusters with High Information Entropy. To construct an N-ways-K-shots meta-task, we need to randomly choose N_c clusters denoted as base clusters $\bar{\mathcal{C}}_i$, $i \in [1, N_c]$. We use the cluster center similarity $\bar{s}_{ik} = \bar{\boldsymbol{\mu}}_i \cdot \bar{\boldsymbol{\mu}}_k$ to obtain the $\bar{K}$ most similar neighbors as the candidate clusters $\bar{\mathcal{C}}_{i\bar{k}}$, $k \in [1, \bar{K}]$, where $\bar{\boldsymbol{\mu}}_i$ is the average embedding feature of cluster $\bar{\mathcal{C}}_i$, and $\bar{K} = 5$ in this paper. We randomly select K samples from each base cluster to build the support set, which we then use to finetune the evaluation model $f(\cdot;\bar{\boldsymbol{\theta}})$ with the same meta-training method.

For simplification, we first define the entropy of a cluster $\mathcal{C}$ based on the fine-tuned evaluation model. The N-ways classification label of each sample $\mathbf{x}_i \in \mathcal{C}$ is computed as $l_i = \arg\max_{j=1}^{N} \mathbf{l}_i[j]$, where $\mathbf{l}_i = f(\mathbf{x}_i;\bar{\boldsymbol{\theta}})$. Then we can provide the probability p_j of selecting a sample with label j from the cluster $\mathcal{C}$ by computing the frequency of the label j, *i.e.*, $p_j = N_j/\dot{N}_l$, where N_j is the occurrence number of label j and $\dot{N}_l$ is the length of cluster $\mathcal{C}$. Then the entropy of $\mathcal{C}$ is formulated as:

$$H(\mathcal{C}) = -\sum\nolimits_{j=1}^{N} p_j \log p_j. \tag{8}$$

According to Eq. 8, we choose the cluster $\bar{\mathcal{C}}_i^*$ with the highest entropy as the final cluster to construct the query set. The formulation is as follows:

$$\bar{\mathcal{C}}_i^* = \bar{\mathcal{C}}_{ik^*}, k^* = \arg\max\nolimits_{\bar{k}=1}^{\bar{K}} H(\bar{\mathcal{C}}_{i\bar{k}}) \tag{9}$$

In our case, a low entropy for a cluster indicates that it is certain or informationless for the evaluation model, since the label outcome of a cluster can be regarded as a variable based on our evaluation model. In other words, the information in a cluster with low entropy has already been learned in the previous training epochs and is thus not new for the current evaluation model. In contrast, a cluster with high entropy can provide unseen information for further improvement.

Filtering Out Noisy Samples. Notice that the cluster $\bar{\mathcal{C}}_i^*$ contains many noisy samples (with inconsistent labels compared with the support samples). Thus, we use the proposed evaluation model $f(\cdot;\bar{\boldsymbol{\theta}})$ to filter out several noisy samples and build the query set. First, we run $f(\cdot;\bar{\boldsymbol{\theta}})$ on each sample $\mathbf{x}_{ij}$ in $\bar{\mathcal{C}}_i^*$ to provide the *probabilities* $\mathbf{l}_{ij}$ of N-ways classification, *i.e.*, $\mathbf{l}_{ij} = f(\mathbf{x}_{ij};\bar{\boldsymbol{\theta}})$. We take the *probability* of the i-th classification $\mathbf{l}_{ij}[i]$ as the evaluation score of the sample $\mathbf{x}_{ij}$. According to these scores, we re-sort the cluster $\bar{\mathcal{C}}_i^*$ in descending order to obtain a new cluster $\dot{\mathcal{C}}_i$. The noisy samples are placed at the end of the cluster.

Thus, we can filter out the noisy samples by removing the ones at the end of the new cluster $\dot{\mathcal{C}}_i^*$. The *keep rate* $\beta \in (0, 1)$ is used to control the number of samples removed. Specifically, we define the removing operation as $\dot{\mathcal{C}}_i = \dot{\mathcal{C}}_i[1 : \lfloor \beta \bar{N}_l \rfloor]$, where $\bar{N}_l$ is the length of $\dot{\mathcal{C}}_i$, and $\lfloor \cdot \rfloor$ is the *floor* operation. Finally, we randomly select Q samples from the cluster $\dot{\mathcal{C}}_i$ as the query set.

In particular, during training, we employ a random value $\eta \in (0, 1)$ for each mini-batch. Only when $\eta > 0.9$, we use the progressive evaluation mechanism to build the meta-tasks. Since our pseudo-labels contain the most useful information for training, we need to utilize this information as much as possible.

5 Experiments

5.1 Datasets and Implementation Details

Datasets. We evaluate the proposed approach on three popular datasets: Omniglot [30], *mini*ImageNet [39], and *tiered*ImageNet [41]. Following the setting in [25], for all three datasets, we only use the unlabeled data in the training subset to construct the unsupervised few-shot tasks.

Embedding Functions and Hyperparameters. For Omniglot, we adopt a 4-Conv model (the same as the one in MAML [16]), as the backbone, and add two fully connected (FC) layers with 64 output dimensions (64-FC). For *mini*ImageNet and *tiered*ImageNet, we use the same embedding function, which includes a ResNet-50 [20] backbone and two 128-FC layers, and train it on the whole ImageNet. The other training details are the same for the two models. The number of training epochs is set to 200. We use the same data augmentation and cosine learning rate schedule as [9]. The other hyperparameters for training are set as $N_p = 256$, $N_a = 2$, $N_n = 65536$, $\tau = 0.2$, $m = 0.999$. We use the same number of clusters as CACTUs for fair comparison, *i.e.* $N_c = 500$. The *keep rate* β in the progressive evaluation is set as 0.75 to filter out very noisy samples.

Supervised Meta-Learning Methods. We combine the proposed pseudo-labeling based on clustering-friendly embedding (PL-CFE) with two supervised methods including classical model-agnostic meta-learning (MAML) [16], and the recently proposed embedding propagation (EP) [42]. The corresponding methods are denoted as PL-CFE-MAML and PL-CFE-EP, respectively. In the following experiments, we only train the meta-learning models (MAML and EP) on the one-shot task and directly test them on the other few-shot tasks.

5.2 Ablation Study

To analyze the effectiveness of the key components in our framework, we conduct extensive experiments on Omniglot and *mini*ImageNet with MAML and EP. First, we choose the unsupervised method ACAI [5]/DC [6] as the baselines of the embedding function, which achieves the best accuracy for CACTUs. Specifically, we apply *k-means* to the ACAI/DC embedding features to provide the pseudo-labels, and then run MAML and EP over eight few-shot tasks. These

						Omniglot				*mini*ImageNet			
	Pos	Neg	AE	SC	FN	(5,1)	(5,5)	(20,1)	(20,5)	(5,1)	(5,5)	(5,20)	(5,50)
A/D-M						68.84	87.78	48.09	73.36	39.90	53.97	63.84	69.64
MC-M						-	-	-	-	39.83	55.62	67.27	73.63
CFE-M	✓					89.97	97.50	74.33	92.44	39.75	57.31	69.68	75.64
	✓	✓				69.88	89.68	40.50	72.50	25.35	34.91	47.80	55.92
	✓		✓			90.67	97.90	75.18	92.75	41.04	57.86	69.62	75.67
	✓	✓	✓			91.32	97.96	76.19	93.30	41.99	58.66	69.64	75.46
	✓	✓	✓	✓		92.01	98.27	78.29	94.27	43.39	59.70	70.06	75.38
	✓	✓	✓		✓	91.40	97.94	78.61	94.46	42.85	59.31	69.61	75.25
	✓	✓	✓	✓	✓	**92.81**	**98.46**	**80.86**	**95.05**	**43.60**	**59.84**	**71.19**	**75.75**
A/D-E						78.23	88.97	53.50	72.37	39.73	52.69	59.75	62.06
MC-E						-	-	-	-	43.26	56.67	63.19	65.07
CFE-E	✓					93.31	97.31	78.59	90.09	43.44	56.98	64.06	66.65
	✓	✓				60.70	73.37	37.93	49.33	26.68	32.36	37.17	38.86
	✓		✓			93.48	97.40	79.75	90.67	43.55	57.73	64.17	66.17
	✓	✓	✓			93.88	97.46	79.55	91.13	47.55	62.13	68.89	71.11
	✓	✓	✓	✓		95.31	98.01	83.17	92.50	48.25	62.54	69.76	71.58
	✓	✓	✓		✓	94.93	98.00	82.57	91.97	48.02	62.56	69.85	**72.31**
	✓	✓	✓	✓	✓	**95.48**	**98.04**	**83.67**	**92.54**	**49.13**	**62.91**	**70.47**	71.79

Table 2. Influence of key components in our model. We evaluate different variants in terms of the accuracy for N-ways-K-shots (N,K) tasks. The A/D labels represent ACAI [5] on Omniglot [30] and DC [6] on *mini*ImageNet [39]. MC and M are short names of MoCo-v2 [9] and MAML [16]. Similarly, CFE is our embedding method and E represents the EP [42]. The results of A/D-M are taken from CACTUs [25].

two baselines are denoted as A/D-M and A/D-E, respectively. The results of A/D-M come from the original CACTUs paper.

The key components in our clustering-friendly embedding (CFE) are the positive set (Pos), negative set (Neg), and asynchronous embedding (AE). We also investigate the main parts of our progressive evaluation, including selecting the cluster (SC) and filtering out noise (FN). First, we minimize the similarity ratio on the positive set. As shown in Table 2, our approach achieves impressive improvement for both MAML and EP in terms of all four tasks on Omniglot, compared with the baselines. In particular, in the 5-ways-1-shot task, the highest accuracy gain reaches 21.15%. Then, we add the negative set for training. However, the performance drops dramatically. The reason may be that the rapidly changing embedding models reduces the consistency of negative embedding features. Thus, we add AE to alleviate this issue. The increased performance (Pos+Neg+AE vs. Pos) indicates that Neg+AE can obtain more divisive samples for better training. We also use AE on the positive set (Pos+AE) and achieve expected accuracy gains for the two supervised methods in terms of all tasks. Finally, we use the SC strategy to select diverse samples for constructing few-shot tasks, yielding performance gains in all tasks. Then, the FN strategy is added to filter out noisy samples (SC+FN), which further improves the accuracy scores of all tasks. We also directly use our FN strategy on the original clusters (Pos+Neg+AE+FN) and achieve promising improvement in all tasks.

		Omniglot				*mini*ImageNet			
Algorithm (N, K)	Clustering	(5,1)	(5,5)	(20,1)	(20,5)	(5,1)	(5,5)	(5,20)	(5,50)
Training from scratch [25]	-	52.50	74.78	24.91	47.62	27.59	38.48	51.53	59.63
CACTUs-MAML [25]	BiGAN	58.18	78.66	35.56	58.62	36.24	51.28	61.33	66.91
CACTUs-ProtoNets [25]	BiGAN	54.74	71.69	33.40	50.62	36.62	50.16	59.56	63.27
CACTUs-MAML [25]	A/D	68.84	87.78	48.09	73.36	39.90	53.97	63.84	69.64
CACTUs-ProtoNets [25]	A/D	68.12	83.58	47.75	66.27	39.18	53.36	61.54	63.55
AAL-ProtoNets [2]	-	84.66	89.14	68.79	74.28	37.67	40.29	-	-
AAL-MAML++ [2]	-	88.40	97.96	70.21	88.32	34.57	49.18	-	-
ULDA-ProtoNets [35]	-	91.00	98.14	78.05	94.08	40.63	56.18	64.31	66.43
ULDA-MetaOptNet [35]	-	90.51	97.60	76.32	92.48	40.71	54.49	63.58	67.65
LASIUM-MAML [28]	-	83.26	95.29	-	-	40.19	54.56	65.17	69.13
LASIUM-ProtoNets [28]	-	80.15	91.1	-	-	40.05	52.53	59.45	61.43
CACTUs-EP	A/D	78.23	88.97	53.50	72.37	39.73	52.69	59.75	62.06
PL-CFE-MAML (ours)	CFE	92.81	**98.46**	80.86	**95.05**	43.38	60.00	**70.64**	**75.52**
PL-CFE-EP (ours)	CFE	**95.48**	98.04	**83.67**	92.54	**49.13**	**62.91**	70.47	71.79
MAML (supervised)	-	94.64	98.90	87.90	97.50	48.03	61.78	70.20	73.77
EP (supervised)	-	98.24	99.23	92.38	97.18	57.15	71.27	77.46	78.77

Table 3. Accuracy (%) of N-ways-K-shots (N,K) tasks. A/D represents ACAI [5] on Omniglot [30] and DC [6] on *mini*ImageNet [39]. The best values are in bold.

This indicates that the proposed FN strategy can reduce the *label inconsistency* issue for improved performance.

Besides, we also compare the recent contrast learning method MoCo-v2 [9], since our model has a similar training strategy but different training loss. We apply the pretrained MoCo-v2 model (200 epochs) to extract embeddings and provide the pseudo-labels, and then run MAML and EP over four few-shot tasks on *mini*ImageNet. These are denoted as MC-M and MC-E, respectively. As shown in Table 2, even our methods without progress evaluation (Pos+Neg+AE) can outperform the MoCo-based methods in all tasks. This clearly demonstrates the effectiveness of our clustering-friendly embedding. Notice that our progress evaluation (Pos+Neg+AE+SC+FN) can further improve the performance.

5.3 Comparison with Other Algorithms

Results on Omniglot and *mini*ImageNet. We compare our PL-CFE-MAML and PL-CFE-EP with model-agnostic methods: CACTUs [25], AAL [2], ULDA [35], and LASIUM [28].

As shown in Table 3, our PL-CFE-MAML outperforms all model-agnostic methods in eight few-shot tasks on two datasets. Specifically, for the 5-ways-5-shots task on the Omniglot dataset, our method achieves a very high score of 98.46%, which is very close to the supervised result of 98.90%. Surprisingly, the proposed PL-CFE-MAML even outperforms the corresponding supervised method under the 5-ways-20-shots and 5-ways-50-shots settings on the *mini*ImageNet dataset. In addition, compared with the baseline CACTUs-MAML, we achieve significant accuracy gains, which reach 32.77% (for 20-ways-1-shot) on Omniglot and 6.8% (for 5-ways-20-shots) on *mini*ImageNet.

	(5,1)	(5,5)	(5,20)	(5,50)
Training from scratch [35]	26.27	34.91	38.14	38.67
ULDA-ProtoNets [35]	41.60	56.28	64.07	66.00
ULDA-MetaOptNet [35]	41.77	56.78	67.21	71.39
PL-CFE-MAML (ours)	43.60	59.84	**71.19**	**75.75**
PL-CFE-EP (ours)	**49.51**	**64.31**	70.98	73.06
MAML (supervised) [25]	50.10	66.79	75.61	79.16
EP (supervised) [42]	58.21	71.73	77.40	78.78

Table 4. Accuracy (%) of N-ways-K-shots (N,K) tasks on *tiered*ImageNet [41]. We show the results of supervised methods MAML [25] and EP [42] for complete comparison. The best values are in bold.

In the experiments for our PL-CFE-EP, we first create the CACTUs-EP baseline by combining CACTUs with EP. Our PL-CFE-EP provides impressive improvements in performance compared with CACTUs-EP. In particular, in the 20-ways-1-shot task on Omniglot, we achieve a huge gain of 30.17%. On *mini*ImageNet, the gains reach 10.72% (in 5-ways-20-shots). Compared with the other existing methods, our PL-CFE-EP obtains superior performance in six tasks and comparable performance in the other two tasks. In particular, we achieve a significant accuracy gain of 8.42% in 5-ways-1-shot on *mini*ImageNet.

Results on *tiered*ImageNet. We compare our PL-CFE-MAML and PL-CFE-EP with the recent ULDA [35] on *tiered*ImageNet. As shown in Table 4, our models outperform the compared methods in four few-shot tasks. Specifically, for the 5-ways-1-shots and 5-ways-5-shots tasks, our PL-CFE-EP achieves the highest scores. Compared with the best previous method, ULDA-MetaOptNet, our method obtains accuracy gains of 7.74% and 7.35% on these two tasks, respectively. Our PL-CFE-MAML outperforms all compared methods in the 5-ways-20-shots and 5-ways-50-shots tasks. It also achieves significant improvements, with gains of 3.98% and 4.36%, respectively, compared with ULDA-MetaOptNet.

In summary, the results demonstrate the effectiveness of our method.

6 Conclusion

In this paper, we introduce a new framework of pseudo-labeling based on clustering-friendly embedding (PL-CFE) to automatically construct few-shot tasks from unlabeled datasets for meta-learning. Specifically, we present an unsupervised embedding approach to provide clustering-friendly features for few-shot tasks, which significantly reduces the *label inconsistency* and *limited diversity* problems. Moreover, a progressive evaluation is designed to build hard tasks to further alleviate *limited diversity* issue. We successfully integrate the proposed method into two representative supervised models to demonstrate its generality. Finally, extensive empirical evaluations clearly demonstrate and the effectiveness of our PL-CFE, which outperforms the corresponding supervised meta-learning methods in two few-shot tasks. In the future, we will utilize our model to more computer vision tasks, such as object tracking [12,18,45] and segmentation [52,51,13], to explore label-free or label-less solutions.

References

1. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: NeurIPS (2016) 4
2. Antoniou, A., Storkey, A.: Assume, augment and learn: Unsupervised few-shot meta-learning via random labels and data augmentation. arXiv preprint arXiv:1902.09884 (2019) 13
3. Ba, J., Hinton, G.E., Mnih, V., Leibo, J.Z., Ionescu, C.: Using fast weights to attend to the recent past. In: NeurIPS (2016) 4
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: NeurIPS (2007) 2, 4
5. Berthelot, D., Raffel, C., Roy, A., Goodfellow, I.: Understanding and improving interpolation in autoencoders via an adversarial regularizer. In: ICLR (2019) 5, 6, 11, 12, 13
6. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018) 5, 6, 11, 12, 13
7. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. ICLR (2019) 4
8. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NeurIPS (2016) 5
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 11, 12, 13
10. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: NeurIPS (2015) 4
11. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: ICLR (2017) 5, 6
12. Dong, X., Shen, J., Shao, L., Porikli, F.: Clnet: A compact latent network for fast adjusting siamese trackers. In: ECCV. Springer (2020) 14
13. Dong, X., Shen, J., Shao, L., Van Gool, L.: Sub-markov random walk for image segmentation. IEEE T-IP (2015) 14
14. Dvornik, N., Schmid, C., Mairal, J.: Diversity with cooperation: Ensemble methods for few-shot classification. In: ICCV (2019) 1
15. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? JMLR pp. 625–660 (2010) 2, 4
16. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017) 1, 2, 4, 6, 11, 12
17. Finn, C., Xu, K., Levine, S.: Probabilistic model-agnostic meta-learning. In: NeurIPS (2018) 4
18. Han, W., Dong, X., Khan, F.S., Shao, L., Shen, J.: Learning to fuse asymmetric feature maps in siamese trackers. In: CVPR (2021) 14
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 4, 9
20. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV (2015) 11
21. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18**(7), 1527–1554 (2006) 2, 4
22. Hinton, G.E., Plaut, D.C.: Using fast weights to deblur old memories. In: CCSS (1987) 1, 4

23. Hochreiter, S., Younger, A.S., Conwell, P.R.: Learning to learn using gradient descent. In: ICANN (2001) 4
24. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: ACL (2018) 4
25. Hsu, K., Levine, S., Finn, C.: Unsupervised learning via meta-learning. In: ICLR (2019) 2, 3, 4, 5, 6, 9, 11, 12, 13, 14
26. Ji, Z., Zou, X., Huang, T., Wu, S.: Unsupervised few-shot feature learning via self-supervised training. Frontiers Comput. Neurosci. (2020) 4
27. Khodadadeh, S., Boloni, L., Shah, M.: Unsupervised meta-learning for few-shot image classification. In: NeurIPS (2019) 4
28. Khodadadeh, S., Zehtabian, S., Vahidian, S., Wang, W., Lin, B., Bölöni, L.: Unsupervised meta-learning through latent-space interpolation in generative models. In: ICLR (2021) 4, 13
29. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML workshop (2015) 4
30. Lake, B., Salakhutdinov, R., Gross, J., Tenenbaum, J.: One shot learning of simple visual concepts. In: CogSci (2011) 3, 5, 6, 11, 12, 13
31. Lee, D.B., Min, D., Lee, S., Hwang, S.J.: Meta-GMVAE: Mixture of Gaussian VAE for Unsupervised Meta-Learning. In: ICLR (2021) 4
32. Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.S., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. In: NeurIPS (2019) 1
33. Medina, C., Devos, A., Grossglauser, M.: Self-supervised prototypical transfer learning for few-shot classification. arXiv preprint arXiv:2006.11325 (2020) 4
34. Peng, Z., Li, Z., Zhang, J., Li, Y., Qi, G.J., Tang, J.: Few-shot image recognition with knowledge transfer. In: ICCV (2019) 1
35. Qin, T., Li, W., Shi, Y., Gao, Y.: Unsupervised few-shot learning via distribution shift-based augmentation. arXiv preprint arXiv:2004.05805 (2020) 4, 13, 14
36. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Preprint (2018) 4
37. Ramachandran, P., Liu, P.J., Le, Q.V.: Unsupervised pretraining for sequence to sequence learning. In: EMNLP (2017) 4
38. Ranzato, M., Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: NeurIPS (2007) 2, 4
39. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR (2017) 3, 4, 6, 11, 12, 13
40. Ravichandran, A., Bhotika, R., Soatto, S.: Few-shot learning with embedded class models and shot-free meta training. In: ICCV (2019) 1
41. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. In: ICLR (2018) 3, 11, 14
42. Rodríguez, P., Laradji, I., Drouin, A., Lacoste, A.: Embedding propagation: Smoother manifold for few-shot classification. In: ECCV (2020) 1, 3, 4, 11, 12, 14
43. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: ICML (2016) 4
44. Schmidhuber, J.: Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. Ph.D. thesis, Technische Universität München (1987) 1, 4
45. Shen, J., Liu, Y., Dong, X., Lu, X., Khan, F.S., Hoi, S.C.: Distilled siamese networks for visual tracking. IEEE T-PAMI (2021) 14

46. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: NeurIPS (2017) 2, 4
47. Steiner, B., DeVito, Z., Chintala, S., Gross, S., Paszke, A., Massa, F., Lerer, A., Chanan, G., Lin, Z., Yang, E., Desmaison, A., Tejani, A., Kopf, A., Bradbury, J., Antiga, L., Raison, M., Gimelshein, N., Chilamkurthy, S., Killeen, T., Fang, L., Bai, J.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS. pp. 8026–8037 (2019) 8
48. Thrun, S., Pratt, L.: Learning to learn: Introduction and overview. In: Learning to learn, pp. 3–17. Springer (1998) 4
49. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: ICML (2008) 2, 4
50. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NeurIPS (2016) 4
51. Wang, W., Shen, J., Dong, X., Borji, A., Yang, R.: Inferring salient objects from human fixations. IEEE T-PAMI (2019) 14
52. Wu, D., Dong, X., Shao, L., Shen, J.: Multi-level representation learning with semantic alignment for referring video object segmentation. In: CVPR (2022) 14
53. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018) 6
54. Yu, D., Deng, L., Dahl, G.: Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition. In: NeurIPS Workshop (2010) 4
55. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: CVPR (2017) 4