Pseudo Label Guided Unsupervised Meta-Learning for Low-Shot Image Classification

Jinghan Sun, Futian Weng, Chenyu Lian, Qian Dai, Meng Su

24520210157071, 24520210157072, 23020211153895, 23020211153926, 24520211154659
Xiamen University, Xiamen 361005, China

Abstract

Low-shot machine learning is inspired by the observation that humans, based on prior experience, can learn new concepts given only a few examples. One way to acquire this prior knowledge is by meta-learning on tasks analogous to the low-shot learning task. Especially, unsupervised meta-learning waives the demanding requirement that the auxiliary dataset for task construction must be labeled. However, most existing approaches focus on pretraining a classifier for diverse tasks, but omit the practically relevant situation where the target task is relatively fixed and the goal is optimal performance on this specific task, e.g., recognition of few known rare diseases. In this work, we propose Unsupervised Meta-learning with tasks constructed with Pseudo Labels and Augmentation (UMPLA), for improving performance on a specific target task. UMPLA innovatively connects the model-agnostic meta-learning (MAML) process with the target task, by generating pseudo labels for the unlabeled data using a sentinel model fit to the target task. These pseudo labels are then used during MAML to steer the model towards learning the classes in the target task by exploiting interclass visual correlations.

1 Introduction

Deep convolutional neural networks (DCNNs) have surpassed human on various large-scale recognition tasks, *e.g.*, the ILSVRC-12 image classification challenge (Russakovsky et al. 2015). The superb performance, however, is fundamentally driven by the emergence of various large-scale labeled datasets. The demanding requirement for massive labeled training data could impede the development of DCNN-based solutions to a new problem. Humans, in contrast, can learn new concepts quickly given only a few examples. This intriguing characteristic of biological learning has inspired the rise of low-shot machine learning (Fei-Fei, Fergus, and Perona 2006).

It is widely accepted that such learning capability of humans is granted by prior knowledge accumulated from life experience (Lake et al. 2011). Similarly, low-shot machine learning assumes the access to a large pool of auxiliary dataset, which is drawn from the same distribution but different classes from the target task. Then, prior knowledge can be acquired from this auxiliary set, and generalized for the low-shot learning task (Lake et al. 2011). A particular group of methods that has been proven effective in acquiring this prior knowledge is by meta-learning on tasks constructed from the auxiliary set, similarly to the target task (Finn, Abbeel, and Levine 2017; Snell, Swersky, and Zemel 2017). Taking a step further, unsupervised meta-learning (Hsu, Levine, and Finn 2018; Khodadadeh, Boloni, and Shah 2019) eliminates the requirement of labeling the auxiliary set entirely.

While promising, most existing meta-learning approaches to low-shot learning focus on pretraining a classifier that can be quickly adapted for diverse tasks. As a result, the meta-learning process and the target task are often isolated (Finn, Abbeel, and Levine 2017; Hsu, Levine, and Finn 2018), and the meta-learner has little knowledge about its end task. We hypothesize that, by connecting the meta-learning process with the target task, the performance can be boosted on the specific task. Our hypothesis is based on the observation that, even if the unlabeled set and the target task consist of disjoint classes, the data in the unlabeled set can still help in learning task-specific category semantics by exploiting interclass correlations.

In addition, we notice that UMTRA (Khodadadeh, Boloni, and Shah 2019)—a state-of-the-art (SOTA) unsupervised meta-learning algorithm for low-shot image recognition—assumed the number of classes in the auxiliary dataset to be large. This assumption, however, may be difficult to satisfy in practice at times. Again taking rare disease recognition for example, the number of common disease types (to comprise the auxiliary dataset) for a certain organ is limited. In this work, we loose this constraint and only require the auxiliary dataset to be large in amount of raw data, which is easier to satisfy.

Motivated by the analysis above, we propose an algorithm coined Unsupervised Meta-learning with tasks constructed with Pseudo Labels and Augmentation (UMPLA). Most notably, UMPLA bridges the unsupervised meta-learning process and the target task with a sentinel model.

2 Related work

Transfer learning has been proven an effective strategy when the data is insufficient for training DCNNs from scratch (Tan et al. 2018). When applied to low-shot learning, however, finetuning on the target task with a few samples would severely overfit (Snell, Swersky, and Zemel 2017). To cope with the challenge, low-shot learning has developed into a dedicated subfield in machine learning (Fei-Fei, Fergus, and Perona 2006). In recent years, encouraging results have been achieved in DCNN-based low-shot classification (Gidaris and Komodakis 2018; Hariharan and Girshick 2017), detection (Hu et al. 2019; Kang et al. 2019; Karlinsky et al. 2019), and segmentation tasks (Fan et al. 2020; Shaban et al. 2017; Wang et al. 2019; Zhang et al. 2019). Although the tasks and methods are versatile, a common key component across the above-enumerated works is the access to a relatively large auxiliary dataset, which is assumed to be drawn from the same distribution as the target low-shot learning task. It is on this dataset that the prior knowledge about the target task is built, analogous to the life experience of human when facing a new learning task. In this work we focus on low-shot image classification, a major task in computer vision.

One way of effective knowledge building from the large auxiliary dataset is by training the network with a large quantity of tasks constructed in a way similar to the target low-shot learning task. Notably, Vinyals (Vinyals et al. 2016) proposed to sample mini-batches called *episodes* during training, where each episode was designed to mimic the target task by subsampling classes as well as data points. The use of episodes makes the training process more faithful to the test environment and thereby improves generalization (Ravi and Larochelle 2017). A big group of works belong to the genre of metric learning (Kulis 2012; Koch, Zemel, and Salakhutdinov 2015; Li et al. 2019). These methods aim to learn a (set of) projection function(s) such that when projected in the embedding space, images can be easily classified using simple nearest neighbour or linear classifiers. In this case the learned transferrable prior knowledge is the projection functions. Another group of works are the model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018; Nichol and Schulman 2018). MAML aims at learning the initial parameters of a deep network from a variety of different tasks, such that one or a few gradient descending steps lead to effective generalization on a new low-shot learning task (easy to finetune in effect). Being model-agnostic, these methods are compatible with any differentiable network architecture, as opposed to a third group of works which use a custom network architecture for encoding the knowledge acquired during the meta-learning phase, *e.g.*, in fast weights (Ba et al. 2016), neural plasticity values (Miconi, Clune, and Stanley 2018), the state of temporal convolutions (Mishra et al. 2017) or in the long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997; Munkhdalai and Yu 2017; Ravi and Larochelle 2017; Santoro et al. 2016). Although promising, the methods described above all relied on extensive labeling of the large auxiliary dataset. This demanding requirement may hinder their practical applications to low-show learning problems in real world.

Unsupervised meta-learning takes a step further by waiving this demanding requirement and building up prior knowledge on a large but unlabeled dataset (Hsu, Levine, and Finn 2018; Khodadadeh, Boloni, and Shah 2019), where a key issue is to determine labels for the unlabeled samples. Assuming knowledge of an upper bound on the number of classes present in potential target tasks, CACTUs (Hsu, Levine, and Finn 2018) leveraged unsupervised embeddings (Berthelot et al. 2018; Caron et al. 2018) to partition all the unlabeled images into many clusters and assigned pseudo labels accordingly. These pseudo labels were subsequently used for low-shot learning with established frameworks including ProtoNet (Snell, Swersky, and Zemel 2017) and MAML (Finn, Abbeel, and Levine 2017). Different from CACTUs, UMTRA (Khodadadeh, Boloni, and Shah 2019) assumed that the number of classes in the unlabeled dataset was large. Based on this assumption, a distinct numerical class label was randomly assigned to each sample consecutively. Both CACTUs and UMTRA achieved competitive performance on the Omniglot (Lake et al. 2011) and mini-ImageNet (Vinyals et al. 2016) benchmarks. However, CACTUs relied on repeated clustering of all the unlabeled images which can be computationally costly, while UMTRA's assumption about the number of classes sometimes cannot be satisfied in practice. In contrast, our method requires no clustering just like UMTRA, while only assuming the unlabeled dataset is large in quantity, which is much easier to satisfy. Hence, this work further broadens practical applications of low-shot machine learning.

Last but not least, the meta-learning process and end task were isolated in most existing meta-learning approaches to low-shot learning ((Finn, Abbeel, and Levine 2017; Hsu, Levine, and Finn 2018; Khodadadeh, Boloni, and Shah 2019; Snell, Swersky, and Zemel 2017), to name a few). As introduced earlier, such isolation may be suboptimal for performance on a specific target task. In this work, we propose to bridge the meta-learning process and the target task with a "sentinel" model for performance improvement.

3 Methods

3.1 Problem statement

In a formal definition of low-shot image classification, a learning task $\mathcal{T}$ comprises a labeled support set $S = \{(x, y)\}$, where x is an image and y is its label; and similarly a query set $Q = \{(x, y)\}$. Hence, $\mathcal{T} = (S, Q)$. An N-way K-shot task includes N classes for recognition, each with K instances in S, where K is small. Thus $y \in [1, ..., N]$ and $|S| = N \times K$. The goal is to learn a classifier f_θ on S that can differentiate the N classes by outputting desirable probability distributions over the classes. Here, f is parameterized by θ. Eventually the learned classifier is evaluated on Q. Note that the number of instances per class in Q does not have to be K. Meanwhile, we assume the access to an unlabeled dataset: $U = \{x\}$, where $|U| = M$, drawn from the same distribution as the target task. Further, we assume much more images in U than in S, *i.e.*, $M \gg N \times K$. Lastly, following the convention in literature (*e.g.*, (Finn, Abbeel, and Levine 2017; Khodadadeh, Boloni, and Shah 2019; Snell, Swersky, and Zemel 2017)), we require that U and $\mathcal{T}$ comprise disjoint classes from each other. We aim to optimize f_θ's performance on $\mathcal{T}$ by effectively utilizing U.

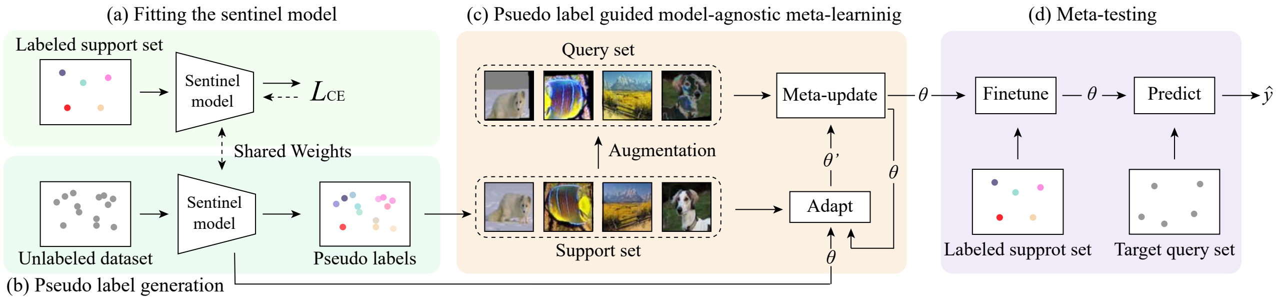

Figure 1: Pipeline of the proposed UMPLA. Note the labeled support set in (a) and (d) is the same set from the target task.

3.2 Model

The pipeline of our UMPLA is illustrated in Fig. 1. Given a target task $\mathcal{T} = (S, Q)$, it starts by fitting a "sentinel" model to the limited data in the support set S. Next, this sentinel model is utilized to generate pseudo labels for data points in the unlabeled dataset U. Then, the pseudo labeled data are used in a MAML-style learning process to meta-train the classifier f_θ. Lastly, f_θ is finetuned on S, and evaluated on the query set Q. In the following, we describe the steps in detail.

Fitting the sentinel model. In UMPLA, the sentinel model is the bridge across the MAML process and the target task $\mathcal{T}$. It fits the model parameters θ for recognition of the classes in $\mathcal{T}$, and then is utilized to generate pseudo labels for images in the unlabeled set U. These pseudo labels will be used in the MAML process, injecting domain knowledge of the specific task. Concretely, we train f_θ on the support set S of $\mathcal{T}$ until convergence (judged by the loss on S), using the cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}}(y, f_\theta(x)) = \mathcal{L}_{\mathrm{CE}}(\boldsymbol{y}, \boldsymbol{p}) = -\sum\nolimits_{n=1}^{N} y_n \log p_n, \quad (1)$$

where $\boldsymbol{p} = f_\theta(x) = [p_1, ..., p_N]$ is the predicted probability distribution over the N classes, $\boldsymbol{y} = [y_1, ..., y_N]$ is the probability distribution of the label y, and $\sum_n y_n = \sum_n p_n = 1$. For a hard label, $\boldsymbol{y}$ is a one-hot vector, *i.e.*, $y_n = 1$ if $n = y$, otherwise $y_n = 0$.

Pseudo label generation. Given the fitted sentinel model, we consider a straightforward pseudo label generating scheme by directly utilizing the output of f_θ. Concretely, for a data point x in the unlabeled set U, we first feed it through f_θ to obtain its probability distribution over the classes in the target task $\mathcal{T}$: $\boldsymbol{p} = f_\theta(x)$. Then, we consider two alternatives for the pseudo label generating function $\hat{y} = G(\boldsymbol{p})$: hard and soft labeling. In effect, hard labeling is the same as classifying x to one of the N classes: $\hat{y}_{\mathrm{hd}} = G_{\mathrm{hd}}(\boldsymbol{p}) = \mathrm{argmax}_n\, p_n$. In this case, $\hat{y}_{\mathrm{hd}}$ indicates the class that x *resembles* most (recall that U and S do not share any common class). For soft labeling, we use an identity function: $\hat{\boldsymbol{y}}_{\mathrm{sf}} = G_{\mathrm{sf}}(\boldsymbol{p}) = \boldsymbol{p}$. The motivation is that $\boldsymbol{p}$ actually reflects the extents of resemblance between x and all the classes in $\mathcal{T}$. By exploiting interclass correlations (Lake et al. 2011; Wei et al. 2020), $\hat{\boldsymbol{y}}_{\mathrm{sf}}$ may help in learning the semantics of multiple classes simultaneously.

Pseudo label guided model-agnostic meta-learning. In MAML (Finn, Abbeel, and Levine 2017), a meta-batch (N_{meta}) of tasks analogous to the target task are firstly constructed from the auxiliary dataset. Then, the network parameters θ are *adapted* multiple times (N_{adpt}) using each constructed task alone, yielding θ' (the adaptation;). Iterating through all the constructed tasks for adaptation comprises the inner loop of the meta-learning. Next, the network parameters are *meta-updated* by collectively considering all the constructed tasks and their individually adapted θ''s. This entire process described above comprises the outer loop of the meta-learning. The loss function in the adaptation and meta-update steps can be summarised in the form below:

$$\mathcal{L}_{\mathcal{T}_i}(f_\theta) = \sum\nolimits_{(x,y)} \mathcal{L}(y, f_\theta(x)), \quad (2)$$

where the data points belong to the support set for the adaptations: $(x, y) \in S_i$, or to the query set for the meta-update: $(x, y) \in Q_i$. Note that in Equation (2) the label can either be a hard one (a scalar) or a soft one (a vector), although we use a scalar symbol here without losing generality.

By employing the pseudo labels, we inject task-specific domain knowledge into the MAML process, and better prepare the model for the target task. Specifically, for the hard labeling scheme we employ the cross-entropy loss in Equation (1), whereas for the soft labeling scheme we employ the Kullback-Leibler divergence loss:

$$\mathcal{L}_{\mathrm{KL}}(\boldsymbol{y}, f_\theta(x)) = \mathcal{L}_{\mathrm{KL}}(\boldsymbol{y}||\boldsymbol{p}) = \sum\nolimits_n y_n \log(y_n/p_n). \quad (3)$$

After the pseudo label guided MAML, f_θ is finetuned on the support set of the target learning task, and finally evaluated on its query set.

4 Experiments

Experimental protocol. The Omniglot (Lake et al. 2011) and miniImageNet (Vinyals et al. 2016) image recognition tasks are the most commonly used low-shot learning benchmarks recently. We adopt the train/validation/test splits and N-way K-shot settings used in (Khodadadeh, Boloni, and Shah 2019) for a direct comparison with existing unsupervised meta-learning approaches. We follow the convention to train on the training set, and use the validation set only for performance generalization. No network engineering is involved, as our focus is to validate the effect of bridging

Table 1: The effects of the hyperparameters meta-batch size (N_{meta}) and number of adaptations (N_{adpt}) on Omniglot (left) and miniImageNet (right) validation accuracies (in %; 5-way 1-shot with soft pseudo labels).

N_{adpt}	N_{meta}: Omniglot			N_{meta}: miniImageNet		
	16	32	48	16	24	32
1	79.71	**82.07**	81.06	33.32	33.04	34.68
5	78.23	76.48	77.74	33.96	**34.72**	34.64
10	74.18	74.27	72.77	32.88	34.56	33.92

Table 2: The effects of (i) soft versus hard pseudo labeling and (ii) different sentinel models on Omniglot validation accuracy (in %). Results of UMTRA (our reimplementation) are also presented for reference. (N, K) indicates N-way K-shot learning.

Pseudo labeling	(N, K)			
	(5, 1)	(5, 5)	(20, 1)	(20, 5)
Sentinel: pretrained with representative classes				
Soft	**78.23**	**93.09**	57.96	83.45
Hard	73.86	90.34	**66.26**	**88.22**
Sentinel: pretrained with random classes				
Soft	76.78	92.18	56.70	82.40
Hard	75.10	90.60	56.85	84.29
Sentinel: randomly initialized				
Soft	75.26	91.52	36.41	50.50
Hard	71.92	89.54	53.00	81.93
UMTRA				
N/A	76.14	90.92	64.46	87.21

the unsupervised meta-learning and the target task. Thus, we employ the same architecture as broadly used in the low-shot learning literature (Khodadadeh, Boloni, and Shah 2019; Snell, Swersky, and Zemel 2017; Vinyals et al. 2016).

Implementation. All the experiments are conducted with PyTorch (Paszke et al. 2017). The NVIDIA GeForce RTX 2080 Ti GPU is used. Except for few crucial ones to be studied in the next section, we determine the values of most hyperparameters by referring to related works (Khodadadeh, Boloni, and Shah 2019; Finn, Abbeel, and Levine 2017) or our empiricism. When hyperparameter tuning is involved, the preferred values are determined on the validation set, then directly applied on the test set.

4.1 UMPLA on the Omniglot dataset

The Omniglot (Lake et al. 2011) is a dataset of handwritten characters, with 1623 characters from 50 alphabets. Every character has 20 instances, each drawn by a different person.

Important meta-learning hyperparameters. Like Khodadadeh (Khodadadeh, Boloni, and Shah 2019), we notice that the meta-batch size (N_{meta}) and number of adaptations (N_{adpt}) are two important hyperparameters in the meta-learning process. Likewise, we empirically study the effect of varying them on the validation accuracy of the network. The experimental results are shown in Table 1. Based on the results, we fix $N_{\text{meta}} = 32$ and $N_{\text{adpt}} = 1$ for subsequent experiments.

Soft versus hard pseudo labeling. In Section 3.2, we present both soft and hard pseudo labeling strategies. Now we compare them on the Omniglot validation set, by investigating their performance and behavior in both 5- and 20-way low-shot tasks. The classification accuracies are charted in Table 2. Interestingly, the soft labels consistently outperform the hard ones on 5-way tasks in both 1- and 5-shot settings. Inversely, the hard labels consistently outperform the soft ones on 20-way tasks. Especially when using the sentinel model fit to the most representative classes of the validation set (to be detailed next), the performance gaps are considerable. We conjecture the reason is that, for high-way tasks (*e.g.*, 20 classes), the soft labels may be distributed too much and thus become flat and sparse. These flat and sparse distributions cannot provide effective supervision for network training using the Kullback-Leibler divergence loss (Equation (3)). In this case, the one-hot hard labels succeed. In contrast, for low-way tasks (*e.g.*, five classes), the soft labels are able to maintain the prominence as needed, and meanwhile exploit the full spectrum of interclass correlations (Lake et al. 2011; Wei et al. 2020). Accordingly, better performance is achieved by the soft labels.

Based on this experiment, we will use soft labels for 5-way tasks, and hard labels for 20-way tasks when later evaluating our proposed method on the test set.

Effect of connecting with target tasks. As aforementioned, it is prohibitive to repeat the entire meta-learning process for each of the randomly sampled low-shot learning tasks for performance evaluation. Instead, we opt to pick the most representative classes as delegates of the classes in these target tasks. These representative classes are used to pretrain the sentinel model for pseudo label generation. To study the effect of connecting with the (delegate) classes of the target tasks on performance, we experiment with two alternative sentinel model settings: (i) the sentinel model is fit to randomly selected classes, and (ii) it is directly used for pseudo-label generation after random initialization (*i.e.*, untrained). The results are presented in Table 2. As expected, the best performance is achieved with the sentinel model fit to most representative classes, whereas the worst is produced by the randomly initialized model. In addition, the performance gaps range from $\sim$1% to above 10%—the more difficult the tasks, the larger the gaps. Lastly, our best results are better than those of UMTRA. There results validate our motivation in connecting with the target tasks: by bridging the meta-learning process and target learning task, low-shot performance on the latter can be boosted.

Comparison with other methods. Finally on the test set, we evaluate the performance of the proposed UMPLA. The results are presented in Table 3. Above all, UMPLA substantially outperforms training from scratch and vanilla deep learning with the pseudo labels (note the latter mostly performs worse than the former). This indicates that UMPLA makes effective use of the unlabeled dataset as well as the pseudo labels. In addition, UMPLA achieves the best results

Table 3: Omniglot test accuracies (in %) of various methods. (N, K) indicates N-way K-shot learning. "Embedding" column indicates the adopted unsupervised embedding algorithm. "N/A": not applicable or available.

		(N, K)			
Method	Embedding	(5, 1)	(5, 5)	(20, 1)	(20, 5)
Training from scratch	N/A	52.50	74.78	24.91	47.62
Vanilla pseudo label	N/A	29.80	44.91	25.92	45.32
k_{nn}-nearest neighbors	BiGAN	49.55	68.06	27.37	46.70
Linear classifier	BiGAN	48.28	68.72	27.80	45.82
MLP with dropout	BiGAN	40.54	62.56	19.92	40.71
Cluster matching	BiGAN	43.96	58.62	21.54	31.06
CACTUs-MAML	BiGAN	58.18	78.66	35.56	58.62
CACTUs-ProtoNets	BiGAN	54.74	71.69	33.40	50.62
k_{nn}-nearest neighbors	ACAI	57.46	81.16	39.73	66.38
Linear classifier	ACAI	61.08	81.82	43.20	66.33
MLP with dropout	ACAI	51.95	77.20	30.65	58.62
Cluster matching	ACAI	54.94	71.09	32.19	45.93
CACTUs-MAML	ACAI	68.84	87.88	48.09	73.36
CACTUs-ProtoNets	ACAI	68.12	83.58	47.75	66.27
MAML (supervised)	N/A	98.7	99.9	95.8	98.9
ProtoNets (supervised)	N/A	98.8	99.7	96.0	98.9

for all the four task settings (*i.e.*, different combinations of N and K for N-way K-shot learning), surpassing the existing SOTA (UMTRA (Khodadadeh, Boloni, and Shah 2019)) with clear margins. The performance gaps range from 1.53% to 5.18%, and are larger for the more difficult, one-shot learning tasks. These results validate the benefit of connecting the meta-learning process with target tasks, especially considering that our method is built on top of UMTRA. Lastly, the supervised approaches yield the best results, followed by the semi-supervised, and the unsupervised. This is expected, given different quantities of ground truth labels used for representation learning. However, the gaps are narrowing, especially for the 5-shot tasks.

4.2 UMPLA on the miniImageNet dataset

Settings. The miniImageNet dataset, originally introduced by Vinyals (Vinyals et al. 2016), is derived from the larger ILSVRC-12 dataset (Russakovsky et al. 2015). It consists of 100 classes each with 600 color images of size 84×84 pixels. Similar to the Omniglot dataset, we conduct a coarse grid search on the validation set for values of the hyperparameters N_{meta} and N_{adpt}. It turns out that UMPLA is relatively insensitive to these two hyperparameters on miniImageNet, and we set $N_{\mathrm{meta}} = 24$ and $N_{\mathrm{adpt}} = 5$ for testing. Meanwhile, based on the experience on Omniglot, we decide to use most representative classes for pretraining the sentinel model. In addition, as all the experiments on miniImageNet are 5-way, we choose to use soft pseudo labels.

Results. The miniImageNet test-set accuracies are presented in Table 4. Again, the results validate that our proposed UMPLA can make effective use of the unlabeled data (comparing against training from scratch and vanilla deep learning with the pseudo labels), and that it generally improves performance upon randomly assigned labels (overall better performance over UMTRA (Khodadadeh, Boloni, and Shah 2019) in three of the four low-shot settings). Meanwhile, UMPLA achieves competitive performance against variants of CACTUs (Hsu, Levine, and Finn 2018). Yet, our pseudo labeling scheme does not impose the overhead of producing unsupervised embeddings for all the unlabeled images, nor of the repeated clustering.

Table 4: MiniImageNet test accuracies (in %) of various methods. (N, K) indicates N-way K-shot learning. "Embedding" column indicates the adopted unsupervised embedding algorithm. "N/A": not applicable or available.

		(N, K)			
Method	Embedding	(5, 1)	(5, 5)	(5, 20)	(5, 50)
Training from scratch	N/A	27.59	38.48	51.53	59.63
Vanilla pseudo label	N/A	27.56	35.32	47.89	54.70
k_{nn}-nearest neighbors	BiGAN	25.56	31.10	37.31	43.60
Linear classifier	BiGAN	27.08	33.91	44.00	50.41
MLP with dropout	BiGAN	22.91	29.06	40.06	48.36
Cluster matching	BiGAN	24.63	29.49	33.89	36.13
CACTUs-MAML	BiGAN	36.24	51.28	61.33	66.91
CACTUs-ProtoNets	BiGAN	36.62	50.16	59.56	63.27
k_{nn}-nearest neighbors	DC	28.90	42.25	56.44	63.90
Linear classifier	DC	29.44	39.79	56.19	65.28
MLP with dropout	DC	29.03	39.67	52.71	60.95
Cluster matching	DC	22.20	23.50	24.97	26.87
CACTUs-MAML	DC	39.90	53.97	63.84	69.64
CACTUs-ProtoNets	DC	39.18	53.36	61.54	63.55
UMTRA	N/A	**39.93**	50.73	61.11	67.15
UMPLA (ours)	N/A	38.56	**53.98**	**64.93**	**69.97**
MAML (supervised)	N/A	48.70	63.11	N/A	N/A
ProtoNets (supervised)	N/A	49.42	68.20	N/A	N/A

5 Conclusion and future work

This work presented UMPLA, a novel algorithm for unsupervised meta-learning for low-shot image recognition. To improve performance on a target low-shot learning task, UMPLA innovatively bridged the model-agnostic meta-learning (MAML) process with the target task. The bridging was achieved by fitting a sentinel model to the limited training data in the low-shot task, and using the model to generate pseudo labels for construction of MAML tasks. Our experiments validated that, even fit to very limited data, the sentinel model was able to produce reasonable pseudo labels for this purpose. Experiments showed that UMPLA not only could effectively utilize the unlabeled dataset, but also clearly improved performance upon randomly assigned pseudo labels, as we hypothesized. In addition, UMPLA achieved superior/competitive performance to recent SOTA approaches on the Omniglot and miniImageNet benchmarks. Besides, UMPLA also expanded the practical application range of existing approaches by only requiring the unlabeled dataset to be large in quantity.

A particular phenomenon that caught our attention was that the soft and hard pseudo labels succeeded in different N-way settings. In the future, it would be interesting to explore strategies to adaptively select either of them or to effectively integrate them for varying scenarios.

References

Ba, J.; Hinton, G. E.; Mnih, V.; Leibo, J. Z.; and Ionescu, C. 2016. Using fast weights to attend to the recent past. In *Adv. Neural Inform. Process. Syst.*, 4331–4339.

Berthelot, D.; Raffel, C.; Roy, A.; and Goodfellow, I. 2018. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *Int. Conf. Learn. Represent.*

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Eur. Conf. Comput. Vis.*, 132–149.

Fan, Z.; Yu, J.-G.; Liang, Z.; Ou, J.; Gao, C.; Xia, G.-S.; and Li, Y. 2020. FGN: Fully Guided Network for Few-Shot Instance Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 9172–9181.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4): 594–611.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Int'l Conf. on Machine Leanring*.

Gidaris, S.; and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4367–4375.

Hariharan, B.; and Girshick, R. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *Int. Conf. Comput. Vis.*, 3018–3027.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780.

Hsu, K.; Levine, S.; and Finn, C. 2018. Unsupervised learning via meta-learning. In *Int. Conf. Learn. Represent.*

Hu, T.; Mettes, P.; Huang, J.-H.; and Snoek, C. G. 2019. SILCO: Show a Few Images, Localize the Common Object. In *Int. Conf. Comput. Vis.*, 5067–5076.

Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; and Darrell, T. 2019. Few-shot object detection via feature reweighting. In *Int. Conf. Comput. Vis.*, 8420–8429.

Karlinsky, L.; Shtok, J.; Harary, S.; Schwartz, E.; Aides, A.; Feris, R.; Giryes, R.; and Bronstein, A. M. 2019. RepMet: Representative-based metric learning for classification and few-shot object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5197–5206.

Khodadadeh, S.; Boloni, L.; and Shah, M. 2019. Unsupervised meta-learning for few-shot image classification. In *Adv. Neural Inform. Process. Syst.*, 10132–10142.

Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Kulis, B. 2012. Metric learning: A survey. *Found. Trends Mach. Learn.*, 5(4): 287–364.

Lake, B.; Salakhutdinov, R.; Gross, J.; and Tenenbaum, J. 2011. One shot learning of simple visual concepts. In *Annu. Mtg. Cogn. Sci. Soc.*, volume 33.

Li, H.; Eigen, D.; Dodge, S.; Zeiler, M.; and Wang, X. 2019. Finding task-relevant features for few-shot learning by category traversal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1–10.

Miconi, T.; Clune, J.; and Stanley, K. O. 2018. Differentiable plasticity: Training plastic neural networks with backpropagation. *arXiv preprint arXiv:1804.02464*.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Munkhdalai, T.; and Yu, H. 2017. Meta networks. *Proc. of Mach. Learn Res.*, 70: 2554.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Nichol, A.; and Schulman, J. 2018. Reptile: A scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *Adv. Neural Inform. Process. Syst. Workshop*.

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *Int. Conf. Learn. Represent.*

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3): 211–252.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-learning with memory-augmented neural networks. In *Int. Conf. Mach. Learn.*, 1842–1850.

Shaban, A.; Bansal, S.; Liu, Z.; Essa, I.; and Boots, B. 2017. One-Shot Learning for Semantic Segmentation. In Tae-Kyun Kim, G. B., Stefanos Zafeiriou; and Mikolajczyk, K., eds., *Brit. Mach. Vis. Conf.*, 167.1–167.13. BMVA Press. ISBN 1-901725-60-X.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, 4077–4087.

Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; and Liu, C. 2018. A survey on deep transfer learning. In *Int. Conf. Artificial Neural Networks*, 270–279. Springer.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Adv. Neural Inform. Process. Syst.*, 3630–3638.

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. PANet: Few-shot image semantic segmentation with prototype alignment. In *Int. Conf. Comput. Vis.*, 9197–9206.

Wei, D.; Cao, S.; Ma, K.; and Zheng, Y. 2020. Learning and Exploiting Interclass Visual Correlations for Medical Image Classification. In *Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 106–115. Springer.

Zhang, C.; Lin, G.; Liu, F.; Yao, R.; and Shen, C. 2019. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5217–5226.