Pseudo-Labeling Based Practical Semi-Supervised Meta-Training for Few-Shot Learning

Xingping Dong, Member, IEEE, Tianran Ouyang®, Shengcai Liao®, Senior Member, IEEE, Bo Du®, Senior Member, IEEE, and Ling Shao®, Fellow, IEEE

Abstract—Most existing few-shot learning (FSL) methods require a large amount of labeled data in meta-training, which is a major limit. To reduce the requirement of labels, a semi-supervised meta-training (SSMT) setting has been proposed for FSL, which includes only a few labeled samples and numbers of unlabeled samples in base classes. However, existing methods under this setting require class-aware sample selection from the unlabeled set, which violates the assumption of unlabeled set. In this paper, we propose a practical semi-supervised meta-training setting with truly unlabeled data to facilitate the applications of FSL in realistic scenarios. To better utilize both the labeled and truly unlabeled data, we propose a simple and effective meta-training framework, called pseudo-labeling based meta-learning (PLML). Firstly, we train a classifier via common semi-supervised learning (SSL) and use it to obtain the pseudo-labels of unlabeled data. Then we build few-shot tasks from labeled and unlabeled data and design a novel finetuning method with feature smoothing and noise suppression to better learn the FSL model from noise labels. Surprisingly, through extensive experiments across two FSL datasets, we find that this simple meta-training framework effectively prevents the performance degradation of various FSL models under limited labeled data, and also significantly outperforms the representative SSMT models. Besides, benefiting from meta-training, our method also improves several representative SSL algorithms as well. We provide the training code and usage examples at https://github.com/ouyangtianran/PLML.

Index Terms—Few-shot learning, semi-supervised, pseudo-labeling.

I. INTRODUCTION

Recent Advances in few-shot learning (FSL), which aims to recognize novel (unseen) objects by only learning from one or a few examples (support set) in each category, with some exploration and accumulation of related techniques [18], [38]. Existing FSL methods are grouped into three types in terms of the inference manner: (i) the traditional inductive FSL methods [6], [36] (IFSL) do not use additional information during inference and evaluate the test data one by one; (ii) the transductive inference [12] for FSL (TFSL) [22], [27], [32] supposes that we can utilize all test data to learn the model for inference; (iii) Semi-supervised FSL (SSFSL) [15], [17], [19], [31], [32], [47] assumes we can have additional unlabeled data for inference.

A. Limitation 1: Heavy Reliance on Base Dataset With Many Labeled Data

Although they achieve impressive performance on novel classes with few labels, most FSL methods still need to pre-train their models on a disjoint but relevant base dataset with many labeled samples, even for SSFSL approaches. Specifically, IFSL and TFSL methods assume that they have access to abundant labeled examples in the base dataset for training. The SSFSL methods based on transfer-learning [21], [32] also use all base labels for pre-training, and employ the semi-supervised inference on the novel dataset. To investigate the impact of limitation 1, we reduce the number of labeled data and retrain three representative FSL models with different inference manners (inductive Proto [36], transductive T-EP, and semi-supervised S-EP [32]), which require full labels during meta-training. As shown in Fig. 1, they suffer from obvious performance degradation with label reduction. This indicates that most FSL methods still heavily rely on a mass of labeled data, which is one major limit for the applications with expensive data annotation (e.g. medical data).

Unlike transfer-learning based SSFSL methods [21], [32], which use semi-supervised inference only on the novel dataset, the meta-training based SSFSL methods split the base dataset into labeled and unlabeled sets, and design new meta-learning methods to reduce the requirement of labeled data and fully utilize the unlabeled data. As shown in Fig. 1d, we evaluate two representative models: SPN [31], and M-PL [17], with different numbers of labeled data. We observe that the evaluation scores still drop sharply with the reduction of...
Illustrations of performance degradation with the reduction of labels in few-shot learning (FSL). The horizontal and vertical axes represent the number of labels per class and accuracy on ImageNet [30]. We demonstrate the performance degradation of three representative Few-Shot Learning (FSL) models (inductive Proto (a), transductive T-EP [32] (b), and semi-supervised S-EP [32] (c)) when using different numbers of training labels in various tasks. “Full” denotes training with fully labeled data, while “PLML” indicates incorporation with our pseudo-labeling based meta-learning approach. In (d), we compare two models: SPN [31], and M-PL [17], in both the original setting and our new setting (*-N).

B. Limitation 2: Class-Prior Selecting Strategy in Meta-Training Based SSFSL Models

Existing meta-training based SSFSL models [17], [31] require a class-prior selecting strategy for training, which is not suitable for truly unlabeled data. As shown in Fig. 2, during task construction, meta-training based SSFSL models first select M classes from both labeled and unlabeled sets, then randomly pick up several samples from these classes to construct the tasks. To ensure the selected unlabeled samples are from these M classes, they actually need to know all the labels of samples in the unlabeled set, which violates the principle of an unlabeled set in practice. Is this strategy necessary for SSFSL? Can we remove it? These are still unexplored questions.

C. The New Setting

To overcome the limitation 2, we propose a new and practical semi-supervised meta-training setting for FSL. As shown in Fig. 2, our practical task only requires selecting M classes from the labeled set and using any class-not-known selection strategy to pick up unlabeled samples. Specifically, the base dataset is split into labeled and unlabeled sets, and we forbid the class-prior selecting strategy for the unlabeled set to make sure we do not have access to the true labels of the unlabeled data. For the novel dataset, we adopt the same inference manners of previous works to focus on the impact of the number of labeled data.

Accordingly, we retrain SPN and M-PL with different numbers of labels on our new setting, noted as SPN-N and M-PL-N. As shown in Fig. 1d, SPN-N suffers from obvious performance degradation while M-PL-N obtains a slight reduction in accuracy. This indicates that different models have different sensitivity to the class-prior selecting strategy. In other words, we cannot simply infer their performance in a more practical new setting based on their performance in the original setting. Thus, we use the new setting for experimental evaluation to design a more practical FSL approach.

D. The Two-Stage Meta-Training Framework

To overcome aforementioned limitations, we propose a two-stage meta-training framework, called pseudo-labeling based meta-learning (PLML), to fully use unlabeled data. In the first stage, we pre-train a base classifier on both the labeled and unlabeled sets, by using a semi-supervised learning approach. In the second stage, the pre-trained classifier firstly generates pseudo-labels of unlabeled data. Then we construct the episodic tasks from both the labeled data and pseudo-labeled data, and propose a novel finetuning method with feature smoothing and noise suppression to train the FSL model over these constructed tasks containing noisy labels. We also finetune the base classifier during the second learning stage, to explore the impact of meta-learning for semi-supervised classification.

Our core idea is to integrate SSL with meta-learning based SSFSL to mitigate the performance degradation rising from the reduction of labeled data in the base dataset, while also avoiding the impractical class-prior selecting strategy in task construction. We instantiate our PLML with two representative meta-learning algorithms: Proto [10] (IFSL) and EP [32] (TFSL and SSFSL), and compare to recent semi-supervised learning (SSL) methods, including SemCo [24], FlexMatch [50] and MarginMatch [39]. Extensive experiments are conducted on two standard benchmarks: miniImageNet [30], and tieredImageNet [31]. The results demonstrate that our simple meta-training framework can significantly alleviate the performance degradation with label reduction for the existing FSL models and narrow the huge gaps between meta-training and transfer-learning based SSFSL methods. As shown in Fig. 1c and 1d, our S-EP-PLML (based on Semco) trained with only 20 labeled samples per class achieves the accuracy of 69.19% on miniImageNet for 5-way 5-shot tasks, which respectively obtains huge gains of 13.55% and 15.41% than the original model (S-EP) and state-of-the-art (SOTA) meta-training based model (M-PL-N). Notably, we find that our meta-learning stage can further improve the performance of recent SSL models, which indicates that combining meta-learning might be a new direction for SSL.

In summary, our key contributions include:

- We propose a more practical semi-supervised meta-training setting for FSL, as a platform to facilitate the development of training with a few labeled base data for...
FSL models. This is the first attempt to investigate and remove the class-prior selection in SSFSL.

- A simple and effective semi-supervised training algorithm, named pseudo-labeling based meta-learning, is proposed to reduce the requirement of the labeled base data for most FSL models. Our approach is the first work to introduce the semi-supervised pre-training manner to meta-training based SSFSL.

- We find that our meta-learning strategy can further improve the performance of recent SSL models, which indicates that using meta-training could be a promising direction to develop new SSL approaches.

- Our approach is model-agnostic and successfully incorporates SSL to FSL. We significantly improve the performance of FSL models (EP and Proto) with limited labeled base training data.

II. RELATED WORK

A. Few-Shot Learning (FSL)

1) Inductive: Usually, inductive methods are based on meta-learning, which can be grouped as model-, optimization-, and metric-based algorithms. The first group attempts to extract meta knowledge by memory models, such as recurrent and memory-augmented networks [30], [34]. The second one aims to learn a generative model initiation that enables fast adaption for novel tasks [3], [6], [7], [16], [28], [53]. The last group attempts to learn powerful representations to discriminate novel classes [13], [32], [36], [43]. Recently, transfer-learning based methods [4], [8], [41] utilize the entire training set to learn a powerful representation instead of episodic training.

2) Transductive: Common approaches combine label propagation [22] or embedding propagation [32] by smoothing embedding. Some researchers focus on using unlabeled data better, e.g. subspace [20], [35], optimal-transport [11], and cross-attention [9]. Recently, accurate prototype estimation has gained attention. For example, [49] introduced a prototype completion based meta-learning framework, while [52] proposed a prototype-based label propagation method.

3) Semi-Supervised: Existing methods are roughly split into two groups. 1) Meta-training based: The pioneering work [31] proposes an adaptive prototypical network to utilize unlabeled data. The following works introduce label propagation [22] and pseudo-labeling [19]. TACO [47] adds a task-level smoothness constraint on unlabeled data, thereby smoothing the meta-model space. Recently, the model-agnostic meta-learning [6] is extended to SSFSL by selecting reliable training subsets from unlabeled data [17]. 2) Transfer-learning based: These methods [15], [21], [32], [44], [48] have attracted main attention due to the high accuracy. However, they require abundant base labels, limiting them for realistic scenarios.

As aforementioned, most FSL models rely heavily on many training labels. This paper aims to design a new framework to facilitate FSL training with fewer labels.

B. Semi-Supervised Learning (SSL)

Since the pioneering work, II-Model [29], pseudo-labeling and consistency regularization techniques have attracted much attention in the SSL community and generated various algorithms [14], [23], [33], [40]. Recently proposed models [1], [2], [24], [37], [39], [45], [50] combine these two techniques by using unsupervised data augmentation [45]. The main idea is to use the prediction of a weakly augmented image as the pseudo-label, and then enforce it to be consistent with the one with strong augmentation. For instance, UDA [45] and ReMixMatch [1] apply confidence-based strategies and sharpened soft pseudo labels to select sufficiently confident...
data for training, whereas FixMatch [37] uses one-hot hard labels. More recently, SemCo [24] successfully complements FixMatch by introducing semantic prior knowledge from language models [26]. FlexMatch [50] applies curriculum learning to design a dynamic threshold function and select high confident pseudo labels according to learning status. MarginMatch [39] utilizes margin-based strategies to select high-confidence pseudo labels. We adopt three SSL algorithms: SemCo, FlexMatch and MarginMatch for pre-training, and investigate the impact of meta-training on SSL models.

III. PRACTICAL SEMI-SUPERVISED META-TRAINING

We first introduce the dataset settings of both common FSL and SSL, and then unify these two settings to define our practical semi-supervised meta-training setting.

A. Few-Shot Learning (FSL)

An FSL dataset is usually split as base and novel class sets, where there is no overlap class between these two sets. The first one is used to train an FSL model, and the latter one is utilized to evaluate the recognition ability of the FSL model for novel objects. Formally, we denote the base set as \( D_{\text{base}} = \{ \mathbf{x}_i, y_i \}_{i=1}^{N_b} \), where \( \mathbf{x}_i \in \mathbb{R}^D \) is the sample (image), \( y_i \in \mathcal{Y}_b = \{ y^1, y^2, \ldots, y^{C_b} \} \) is the class label, and \( N_b \) and \( C_b \) are the numbers of sample and class, respectively. Similarly, the novel set is denoted as \( D_{\text{novel}} = \{ \mathbf{x}_i, y_i \}_{i=1}^{N_n} \), \( y_i \in \mathcal{Y}_n = \{ y^1, y^2, \ldots, y^{C_n} \} \), respectively, where \( C_n \) represents the novel class number. \( \mathcal{Y}_b \) and \( \mathcal{Y}_n \) are denoted as the base and novel label sets, respectively.

Most FSL methods use a meta-learning manner (episodic sampling) to generate many few-shot tasks for training. Specifically, to generate a \( M \)-way \( K \)-shot task, we randomly select \( M \) classes from the base set and then randomly choose \( N_k + N_q \) samples from each selected class, where \( M \) \( N_k \) and \( M \) \( N_q \) samples are used to build the support set \( S \) and query set \( Q \), respectively. The goal is to find a model \( f_{\text{ssl}} \) that classifies a query sample \( \mathbf{x}_q \) in \( Q \) by \( \hat{y}_q = f_{\text{ssl}}(\mathbf{x}_q; \theta_{\text{ssl}}; S) \in \mathcal{Y}_i \subset \mathcal{Y}_b \), where \( \mathcal{Y}_i \) is the class label set of \( S \), and \( \theta_{\text{ssl}} \) is the parameter set of \( f_{\text{ssl}} \). Formally, the model \( f_{\text{ssl}} \) is learned to minimize the averaged loss over those generated tasks. The loss is defined:

\[
L_f = \sum_{(\mathbf{x}, y) \in Q} l_f(\mathbf{x}, y; \theta_{\text{ssl}}, S),
\]

where the loss \( l \) measures the discrepancy between the prediction and the true label. Finally, we evaluate the model on thousands of few-shot tasks on \( D_{\text{novel}} \) and take the average classification accuracy as the final performance.

B. Semi-Supervised Learning (SSL)

Different from FSL, SSL does not have a novel class set for evaluation or testing. The classes in the testing set of SSL are the same as the training set. Thus, in this setting, we do not need to use \( D_{\text{novel}} \). To evaluate the performance on both few-shot and semi-supervised tasks, we split the original base set \( D_{\text{base}} \) into three sets: \( D_l \), \( D_u \), and \( D_t \), which are respectively corresponding to the labeled, unlabeled, and testing set in the semi-supervised setting. Specifically, we randomly select \( N_l \) (less than 20%) and \( N_t \) (near 20%) samples from each class to build the labeled and testing set, respectively. The remaining samples in each class are used to build the unlabeled set. The goal of SSL is to learn a model \( f_{\text{ssl}} \) that classifies the testing instance of base class \( \mathbf{x}_i \) by \( y_i = f_{\text{ssl}}(\mathbf{x}_i; \theta_{\text{ssl}}) \in \mathcal{Y}_i \). Formally, we can learn the model by minimizing the following loss [46],

\[
\sum_{(\mathbf{x}, y) \in D_l} l_f(\mathbf{x}, y; \theta_{\text{ssl}}) + \alpha \sum_{\mathbf{x} \in D_u} l_u(\mathbf{x}; \theta_{\text{ssl}}) + \beta \sum_{\mathbf{x} \in D_{\text{train}}} r(\mathbf{x}; \theta_{\text{ssl}}),
\]

where \( l_f \) is the per-example supervised loss, \( l_u \) denotes the unsupervised loss, \( r \) is the regularization, e.g., the consistency loss or a regularization term, \( D_{\text{train}} = D_l \cup D_u \).

C. Semi-Supervised Meta-Training for Few-Shot Learning

Given the splitting sets, we can use \( D_{\text{train}} \) as the training (base) set in the newly proposed semi-supervised meta-training setting. The novel set is the same as the common few-shot setting to maintain the same evaluation for existing methods. In our experiments, we evaluate our model on the few-shot novel sets \( D_{\text{novel}} \), but also test it on semi-supervised testing set \( D_t \) to explore the impact of the meta-learning manner for common semi-supervised approaches.

Our new training setting is totally different to the previous SSFSL methods, including meta-training [19], [31] and transfer-learning [32], [44] based algorithms. Firstly, in meta-training based works, there exists an additional unlabeled set \( U \) for each \( M \)-way \( K \)-shot task. \( U \) unlabeled samples from each of \( M \) classes are selected to build this unlabeled set. However, this selection strategy requires prior information for the labels of these ‘unlabeled’ samples, which is not reasonable during the training phase and can not utilize the real unlabeled samples. To remove the requirement of prior labels and make a practical semi-supervised few-shot setting, we split the training set into labeled (\( D_l \)) and unlabeled (\( D_u \)) sets, and forbid the above selection strategy on the unlabeled set during the training phase. Secondly, the algorithms based on transfer-learning require all labeled data in the base class set for training, while our setting only needs a few labels.

IV. PSEUDO-LABELING BASED META-LEARNING

Under the proposed semi-supervised few-shot setting, we firstly ignore the unlabeled set \( D_u \) and use the label set \( D_l \) to train existing few-shot models. We change the label number \( N_l \) and observe the performance of three representative methods (Proto [36]), T-EP, and S-EP [32] on several few-shot tasks. As shown in Fig. 1, the evaluation scores of all models dramatically drop with the reduction of \( N_l \). For example, when we use a very small number of labeled data i.e. \( N_l = 20 \) (4\% \( |D_{\text{train}}| \)), the accuracy of S-EP model (with Conv4) on miniImageNet [30] decreases sharply from 72.45\% to 55.64\% in 5-way 5-shot task. Can we use the unlabeled data to fill up the huge gap caused by the reduction of labeled data? To answer this question, we propose a novel semi-supervised few-shot training framework to make full use of the unlabeled...
data. Inspired by the recent two-stage method [32], we also decompose the training method as pre-training and finetuning stages. We illustrate our framework in Fig. 3.

A. Pre-Training With SSL

The FSL model \( f_{\text{fs}} \) is often decomposed as a feature extractor \( \phi \), which is used to predict the discriminative feature of a sample \( x \), i.e., \( z = \phi(x; \theta_b) \), and a \( M \)-way classifier \( c_f \), which is used to predict the class label according to the support set \( S \), i.e., \( \hat{y} = c_f(z; \theta_f, S) \in \mathbb{R}^M \), where \( \hat{y} \) is the predicting one-hot label of \( \gamma_i \). The aim of pre-training is to achieve a good initialization of \( \phi \), which contains sufficient class prior knowledge. Thus, the previous methods [32], [44] add a new base classifier \( c_b \) for the base set, i.e., \( y = c_b(z; \theta_b) \in \mathbb{R}^C \), where \( y \) is the predicting one-hot label of \( \gamma_i \), in contrast to \( c_f \) only focusing classes in the support set. Then \( \phi \) and \( c_b \) are trained via the supervised learning method. Surprisingly, this strategy obtains significant improvement in the few-shot task.

Inspired by this, we also use \( c_b \) for pre-training in our new semi-supervised FSL approach. Our goal is not only to train a good feature extractor \( \phi \) but also to learn a discriminative base classifier \( c_b \) to predict the base class labels of the unlabeled samples. These predicted labels are used as pseudo labels in the finetuning stage. We wish the pre-training method could improve the discriminate ability of both \( \phi \) and \( c_b \). This goal is the same as the common semi-supervised learning, which aims to learn a model that classifies the instances with the base class the same as the common semi-supervised learning, which aims to smooth feature embeddings through self attention.

B. Finetuning With Noise Suppression

To utilize the unlabeled data, we firstly use the pre-trained model to generate pseudo-labels as follows,

\[
\tilde{y}_i = \pi(x_i) = \gamma_b[\text{arg max}_j c_b(\phi(x_i))][j], \quad x_i \in D_u. \quad (3)
\]

We denote \( D_p = \{(x_i, \tilde{y}_i)|x_i \in D_u\} \) as the pseudo-labeling dataset. Combined with the labeled set, we obtain the new training set \( D_{\text{train}} = D_l \cup D_p \), which might contain noisy labels. To reduce the negative impact of noisy labels, we propose a novel noise-suppressed finetuning framework based on label consistency with similar features. Our core idea is inspired by the assumption that samples with similar features have the same label with a high possibility. Accordingly, we design a learnable smoothing module for feature adjustment and a noise dropout strategy for finetuning.

1) Learnable Smoothing Module: For each few-shot task, we aim to adjust the noisy samples by utilizing all samples in the few-shot task, in order to smooth their features to be more similar to the samples with correct labels. Specifically, first, we adopt a Transformer encoder layer of depth one \( T \) [42], so as to smooth feature embeddings through self attention between all samples of a few-shot task. Specifically, denote the feature matrix as \( Z = [z_1; z_2; \cdots; z_N] \in \mathbb{R}^{N_f \times D_f} \), where \( D_f \) is the feature dimension, \( N_f = |\bar{Q} \cup \bar{S}| \), and \( \bar{Q} \) and \( \bar{S} \) respectively represent the support and query sets produced from \( D_{\text{train}} \). Then the transformer operation is formulated as \( Z_t = T(Z; \theta_t) \), where \( N_f \) in \( Z \) is interpreted as the sequence length or the number of words in Transformers, so that the self attention mechanism in Transformers can be considered as a feature propagation and smoothing layer. Furthermore, to make this module learn more from similar samples, we add another feature (embedding) propagation mechanism [32], [51] to guide the learning procedure and adjust noisy features to be more smooth. Specifically, firstly we compute the adjacency matrix as \( A_{ij} = \exp(-d_{ij}^2/\sigma^2) \), if \( i \neq j \), while \( A_{ii} = 0 \), where \( d_{ij}^2 = \|x_i - x_j\|_2^2 \), \( \sigma^2 = \text{Var}(d_{ij}^2) \), \( \forall x_i, x_j \in Z_t \). Given the adjacency matrix, we obtain the propagator matrix \( P = (I - \alpha L)^{-1} \), where \( L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \) is the Laplacian matrix [51], \( D_{ii} = \sum_j A_{ij} \), \( \alpha_p = 0.2 \) is a scaling factor, and \( I \) is the identity matrix. Then, the features are adjusted as...
follows, \( Z_p = PZ_t \). The learnable smoothing module is:

\[
z_i^p = m_z(z_i; \theta_t), \forall z_i^p \in Z_p. \tag{4}
\]

2) Noise Dropout Strategy: To further suppress the noise, we drop out some labels according to the assumption of \( M \) in an \( M \)-way \( K \)-shot task, we randomly select a sample as an anchor and denote its feature as \( z_i \). We compute the distances between the anchor and the other samples, \( d_{ij}^a = \| z_i^a - z_j \|^2, \forall z_j \in Z, \hat{y}_j = i \). Then we sort these samples in ascending order of distance and select the top \( N_q \) samples (including the anchor) to rebuild the support set, which will contain less noise. Since the samples with high distances are more likely to have inconsistency labels with the anchor, we drop out the last \( \lfloor N_q \rfloor \) samples and select the remaining samples to rebuild the query set, where \( \alpha_d \) is the dropout rate (experimentally set as 0.1), and \( N_q \) is the query shot. Finally, we obtain more clean support set \( \tilde{S} \) and query set \( \tilde{Q} \).

3) Training Loss: Our learnable smoothing module and noise dropout strategy are model-agnostic, and we can incorporate them with most few-shot models. Combined Eq. 4 with Eq. 1, we obtain the general formulation of loss as follows,

\[
L_f = \sum_{(x, y) \in \tilde{Q}} l_f(x, \hat{y}; \theta_f, \theta_t, \tilde{S}_c),
\tag{5}
\]

where \( l_f(x_i, \hat{y}_i) = H(c_f(m_i(\phi(x_i))), \hat{y}_i) \), \( H \) represents the cross-entropy loss and \( \hat{y}_i \in \mathbb{R}^M \) is the one-hot label of \( M \) classes. Besides, we finetune the classifier \( c_b \) to preserve the discriminative ability on the basic classes by

\[
L_b = \sum_{(x, y) \in \tilde{Q} \cup \tilde{S}} l_b(x, \hat{y}; \theta_b),
\tag{6}
\]

where \( l_b(x_i, \hat{y}_i) = H(c_b(\phi(x_i)), \hat{y}_i) \), and \( \hat{y}_i \in \mathbb{R}^{C_b} \) is the one-hot label. The final loss is \( L_p = L_f + \gamma L_b \), where the trade-off parameter \( \gamma \) is empirically set as 0.1.

We present the pseudo-code for our proposed framework in Algorithm 1. This pseudo-code outlines the key steps and components, providing a clear and concise guide for applying our method. The specific code can be found at the GitHub address mentioned in the abstract.

V. EXPERIMENTS

A. Experimental Setting

1) Datasets: We evaluate our approach in two standard datasets: miniImageNet [30], and tieredImageNet [31]. miniImageNet is a subset of ImageNet [5] composed of 100 classes with 600 images per class. Classes are divided into 64 base, 16 validation, and 20 novel categories. tieredImageNet is a more challenging and larger dataset than miniImageNet, where classes are split into 20 base (351 classes), 6 for validation (97 classes), and 8 novel (160 classes) super categories, which are selected from supersets of the wordnet hierarchy. Each class contains nearly 900–1200 images. To adjust the above datasets for our new SSFSL setting, we set different \( N_q \) and \( N_l \) proposed in §III to split labeled, unlabeled, and testing sets for different datasets, according to the number of images per class. Specifically, for miniImageNet, we set \( N_l = [20, 50, 100] \) and \( N_q = 100 \). We increase the numbers of tieredImageImageNet, i.e., \( N_l = [20, 100, 200] \) and \( N_q = 200 \). Notice that our results with all base training data are lower than the ones in the original papers since we use fewer labels.

2) Training and Inference Details: We use two common feature extractors (backbones): (i) Conv4: a 4-layer convolutional network [36] with 64 channels per layer, and (ii) ResNet12: a 12-layer residual network [25]. For all datasets, we resize images to \( 84 \times 84 \). In the pre-training stage, we use the same strategy in EP [32] as the Base training method and train the SSL models (SemCo [24], FlexMatch [50] and MarginMatch [39]) by using the default hyperparameter setting in official codes. In the episodic finetuning stage, for miniImageNet and tieredImageNet, 5 classes are randomly
sampled per episode, where in each class we select 5 and 15 instances for the support and query set, respectively. We optimize the models by using SGD with an initial learning rate of 0.01 for 200 epochs. When the validation loss had not decreased for 10 epochs, we reduce the learning rate by a factor of 10. We use 15 instances in the query set of 1- or 5-shot tasks for inference. We adopt 100 unlabeled data per class (same as EP) for semi-supervised inference.

### B. Previous Semi-Supervised FSL Methods With Fewer Labels

To investigate the effect of the number of training labels for previous Semi-supervised FSL methods, we select two kinds of mainstream algorithms including semi-supervised meta-training based and supervised pre-training based methods, as the representation for comparison. The former methods, such as SPN [31] and M-PL [17], split the base training set as labeled and unlabeled sets, and only apply semi-supervised meta-training to learn FSL models. The latter methods use all labels in the base training set for pre-training by simple supervised learning to obtain discriminative feature extractors.

As shown in Table I, we evaluate all approaches on the novel class set of miniImageNet regarding 5-way 1-shot and 5-shot tasks. All previous methods suffer from large performance degradation with reduced labeled data. The pioneering work SPN requires a large number of labels for training. In its original setting, the labeled set contains 40% labeled samples of the base training set. Thus, when we reduce the labels, its performance dramatically drops. M-PL achieves more stable performance with the reduction of labels, since it uses submodular mutual information to select reliable unlabeled set for meta-training. However, these two methods still have a large gap with EP and MFC, because they lack of pre-training phase for better feature representation. EP and MFC also have large performance degradation with a few training labels. For example, on the setting of 20 labels per class, EP suffers from huge performance degradation on four tasks by decrements of 15.27% (1 shot with Conv4), 16.80% (5 shot with Conv4), and 23.86% (1 shot with ResNet12), and obtains a similar reduction of 19.95% in the last task. MFC also has similar dramatical decrements on four FSL tasks.

In summary, some performance gaps still exist between semi-supervised meta-training based methods and pre-training based semi-supervised FSL methods. Most semi-supervised models still require abundant labels to achieve the expected performance. These results inspired us to design a new semi-supervised FSL model to effectively utilize the labels.

### C. Inductive and Transductive Methods With Fewer Labels

To investigate the effect of the number of training labels in the other two kinds of FSL models, we select Proto [36] as the representation for inductive FSL methods, and choose EP [32] for transductive FSL algorithms. We first use the pre-training method of EP (e.g. Base in Table II) to train models on the subsets of miniImageNet with different numbers of labeled data. Then we finetune these models by the episodic training (meta-learning) manners of Proto or EP on the same subsets. Finally, we evaluate these models on the novel class set by using commonly used two inferences in terms of 5-way 1-shot and 5-shot tasks. To further investigate the impact of pre-training, we remove the finetuning phase and directly evaluate the pre-trained model, such as Base+N/A in Table II.

When we reduce the number of labels, the finetuning phase can not obtain improvement in most tasks. Only Base+EP with ResNet12 achieves better performance (1/4) than the pre-training model. This indicates that under the situation of a few base labels, existing methods are not stable. Thus, we need to seek new FSL finetuning methods for this new situation. Moreover, as shown in Table II, all models suffer from huge performance degradation with the reduction of labeled data. For example, under the setting of 20 labels per class, the accuracy of inductive Proto with Conv4 (Base+Proto) declines by 11.19% and 14.29% in terms of 1-shot and 5-shot tasks, respectively. For the models of ResNet12 (Base+EP), amounts of reduction are larger and reach 19.23% and 21.49%. The performance of transductive EP with Conv4 drops sharply by 12.67% and 14.14% in two tasks. Similarly, the ResNet12 version loses more accuracy, declining by 20.98% and 21.03%.

**Table I**

| Methods                  | Pre-training | Backbone | 5-way-1-shot | 5-way-5-shot |
|--------------------------|--------------|----------|--------------|--------------|
|                          | Labels Per Class |          | 20 | 50 | 100 | Full | 20 | 50 | 100 | Full |          |
| SPN [31]                 | N/A          | Conv4    | 28.03 | 32.44 | 40.44 | N/A  | 38.37 | 47.94 | 54.78 | N/A |          |
| M-PL [17]                | N/A          | Conv4    | 40.76 | 44.72 | 45.85 | N/A  | 54.93 | 58.51 | 61.37 | N/A |          |
| EP [32]                  | Supervised   | Conv4    | 45.06 | 51.91 | 55.33 | 60.33 | 55.64 | 64.08 | 66.27 | 72.45 |          |
| MFC [21]                 | Supervised   | Conv4    | 42.91 | 48.14 | 51.41 | 55.49 | 50.94 | 57.10 | 60.90 | 64.92 |          |
| SemCo-PLML-EP (ours)     | Semi-supervised | Conv4   | 57.53 | 58.78 | 62.67 | N/A  | 69.19 | 70.24 | 73.45 | N/A |          |
| FlexMatch-PLML-EP (ours) | Semi-supervised | Conv4  | 56.46 | 56.04 | 59.81 | N/A  | 68.36 | 68.14 | 70.77 | N/A |          |
| MarginMatch-PLML-EP (ours)| Semi-supervised | Conv4 | 55.13 | 56.96 | 60.13 | N/A  | 67.58 | 69.15 | 71.25 | N/A |          |
| B. Previous Semi-Supervised FSL Methods With Fewer Labels

In summary, some performance gaps still exist between semi-supervised meta-training based methods and pre-training based semi-supervised FSL methods. Most semi-supervised models still require abundant labels to achieve the expected performance. These results inspired us to design a new semi-supervised FSL model to effectively utilize the labels.
TABLE II
ACCURACY (%) OF M-WAY K-SHOT TASKS WITH INDUCTIVE (in), TRANSDUCTIVE (trans) AND SEMI-SUPERVISED (semi) INFERENCE ON miniImageNet. WE SHOW THE RESULTS OF TWO FSL METHODS: Proto [36] AND EP [32]. BASED ON A BASE PRE-TRAINING METHOD AND TWO SSL APPROACHES: SemCo [24] AND FlexMatch [50]. THE BEST VALUES ARE IN BOLD. RED VALUES INDICATE OUR PLML OUTPERFORMS THE MODELS TRAINED WITH FULL LABELS.

| Pre-training | Pinetuning | Inference | Backbone | S-way 1-shot | Full | S-way 5-shot | Full |
|--------------|------------|-----------|----------|--------------|------|--------------|------|
|              | Labels Per Class | in-Proto | Conv4    | 20 | 50 | 100 | Full | 20 | 50 | 100 | Full |
| Base         | N/A        | 39.86     | 43.37    | 45.53 | 48.64 | 55.09 | 60.26 | 63.22 | 67.45 | 68.92 |
| Base         | Proto      | 38.58     | 43.37    | 44.49 | 49.77 | 54.29 | 59.62 | 62.36 | 68.58 |
| SemCo        | N/A        | 42.68     | 44.98    | 47.01 | N/A  | 61.71 | 63.55 | 65.71 | N/A  |
| SemCo        | PLML-Proto (ours) | 45.24     | 45.13    | 47.57 | N/A  | 64.94 | 66.16 | 68.32 | N/A  |
| FlexMatch    | N/A        | 40.57     | 42.35    | 44.91 | N/A  | 59.01 | 61.69 | 62.92 | N/A  |
| FlexMatch    | PLML-Proto (ours) | 42.55     | 43.75    | 46.63 | N/A  | 62.83 | 63.62 | 67.01 | N/A  |
| MarginMatch  | N/A        | 41.46     | 41.10    | 44.23 | N/A  | 58.93 | 60.28 | 62.64 | N/A  |
| MarginMatch  | PLML-Proto (ours) | 42.46     | 44.47    | 46.93 | N/A  | 62.87 | 64.49 | 66.75 | N/A  |
|              | trans-EP   | 43.33     | 47.01    | 49.34 | 53.06 | 59.92 | 60.36 | 63.74 | 67.23 |
| Base         | EP         | 42.31     | 46.78    | 49.01 | 54.99 | 54.79 | 61.07 | 63.60 | 68.92 |
| SemCo        | N/A        | 47.56     | 49.44    | 52.10 | N/A  | 61.39 | 64.18 | 66.57 | N/A  |
| SemCo        | PLML-EP (ours) | 52.59     | 54.03    | 56.23 | N/A  | 66.30 | 67.92 | 70.05 | N/A  |
| FlexMatch    | N/A        | 45.35     | 47.77    | 49.57 | N/A  | 58.75 | 61.76 | 64.02 | N/A  |
| FlexMatch    | PLML-EP (ours) | 50.59     | 52.22    | 55.14 | N/A  | 64.47 | 65.29 | 68.78 | N/A  |
| MarginMatch  | N/A        | 45.47     | 48.32    | 49.83 | N/A  | 59.48 | 60.99 | 63.13 | N/A  |
| MarginMatch  | PLML-EP (ours) | 49.93     | 52.13    | 53.49 | N/A  | 63.81 | 65.88 | 67.85 | N/A  |

The above results indicate that current FSL models still rely heavily on mass labels in the base training set, which limits them from being used for practical applications.

D. Benefits of Our PLML
To evaluate the effectiveness of our PLML, we select SemCo [24] as the pre-training method and then use our PLML-Proto and PLML-EP methods (based on Proto and EP) for pinetuning. As shown in Tables I and II, all models with our PLML (SemCo + PLML-*) achieve significant performance improvement in all evaluation tasks with three inference manners and fewer labels, compared with their baselines (Base + Proto or EP), especially on the dataset with the least labels (20). For example, under the setting of 20 labels per class, our models with ResNet12 obtain huge gains of 14.77%–22.31% in terms of 1-shot tasks with three inference manners. In the 5-shot task, the proposed models outperform baselines by large gains of 19.09%–21.63%. For the Conv4 backbone, our models also achieve significant gains of 6.66%–12.27% in the 1-shot task and 10.65%–13.54% in the 5-shot task. The above performance increments clearly demonstrate the benefits of our approach for FSL with a few training labels. In several settings, our models with fewer labels achieve superior performance than the models with full labels. This suggests that the gap between full-supervised and semi-supervised training has been largely closed in FSL tasks.

E. Different SSL Methods
To evaluate the generalization of our PLML, we use the other SSL method, FlexMatch [50] and MarginMatch [39], as the pre-training method. As shown in Table I and II, our models based on FlexMatch and MarginMatch [39] also obtain significant improvement over their baselines in most settings. This indicates that our PLML is robust for different SSL pre-training methods. The overall performance of our FlexMatch-based and MarginMatch-based models is slightly lower than the corresponding SemCo-based models since SemCo introduces the semantic embedding features to provide better prior knowledge. This suggests that our approach can transfer the advantages of SSL methods to the semi-supervised training of FSL, and bridge two topics. Furthermore, we directly evaluate these pre-training methods on the novel set to explore the effectiveness of SSL models. Compared with baselines (Base), both SemCo-, FlexMatch- and MarginMatch-based models achieve large gains in most settings. However, there still exist gaps in the models with full labels, especially for the ResNet12-based models. Incorporated our PLML, the gaps have been largely closed in most tasks.
TABLE III

Accuracy (%) of M-way K-shot Tasks With Inductive (in), Transductive (trans) and semi-supervised (semi) Inference on tieredImageNet [31]. We show the results of two FSL methods: Proto [36] and EP [32], based on a base pre-training method and two SSL approaches SemCo [24] and FlexMatch [50]. PLML represents our approach. The best values are in bold. Red values indicate our models outperform the original models trained with all labels.

| Pre-training | Finetuning | Inference | Backbone | 5-way 1-shot | 5-way 5-shot |
|--------------|------------|-----------|----------|--------------|--------------|
|              | Labels Per Class |          |          | 20 100 200 All | 20 100 200 All |
| Base         | Proto      | in-Proto  | Conv4    | 4.51 50.80 51.07 52.50 | 63.21 68.49 70.12 71.50 |
|              | SemoCo     | PLML-Proto| in-Proto  | 51.73 51.86 52.04 - | 72.27 72.43 72.64 - |
|              | FlexMatch  | PLML-Proto| in-Proto  | 44.94 49.33 50.60 - | 64.80 70.54 71.32 - |
| Base         | EP         | trans-EP  | Conv4    | 49.33 55.33 56.08 58.10 | 67.51 68.67 70.01 71.44 |
|              | SemoCo     | PLML-EP   | trans-EP | 61.67 60.52 61.09 - | 74.08 73.90 74.29 - |
|              | FlexMatch  | PLML-EP   | trans-EP | 54.41 58.88 60.56 - | 68.43 72.71 73.54 - |
| Base         | EP         | semi-EP   | Conv4    | 54.02 61.19 61.96 63.28 | 65.03 70.87 71.98 74.14 |
|              | SemoCo     | PLML-EP   | semi-EP  | 66.69 67.71 68.20 - | 76.84 75.74 76.66 - |
|              | FlexMatch  | PLML-EP   | semi-EP  | 60.50 65.93 66.37 - | 71.38 74.85 75.28 - |

For example, our SemCo-based models with ResNet12 obtain close or superior performance in 5-way 5-shot tasks, compared with models trained by full labels.

F. Our PLML for Different Datasets

To further demonstrate the generalization, we evaluate our approach in the tieredImageNet dataset. We show the results of 1- and 5-shot tasks in Table III. Our PLML facilitates both SemCo- and FlexMatch-based models to achieve significant improvement in most tasks and approach the baselines with full labels. Especially, in the 5-way 5-shot tasks on 20 labels subset, the highest gain reaches 11.81% (SemCo+semi-EP). Furthermore, our models outperform the baselines with full labels in 26 tasks (highlighted by red). This demonstrates our generalization ability for the large-scale dataset.

G. Different SS-Meta-Training Setting

To investigate the impact of class-prior selection strategy i.e. first pick up some classes and then select several samples from these classes, we retrain two representative approaches, SPN [31] and M-PL [17], by using the original and our new setting with different label numbers. Then, we evaluate them by using the semi-supervised inference method with 15 queries and 100 unlabeled samples on miniImageNet and tieredImageNet. As shown in Table IV, the accuracy of SPN-N drops sharply, while M-PL-N obtains a slight reduction. These indicate we cannot correctly infer the performance in the new setting based on the results obtained in the original setting. Therefore, we need a new setting to reevaluate the methods.

H. Benefits of Meta-Learning for SSL

To investigate the effect of meta-learning on SSL models, we finetune SemCo and FlexMatch by our PLML, and evaluate them on the testing set of base classes in miniImageNet, and tieredImageNet. As shown in Fig. 4, our approaches achieve superior performance than the baselines in most evaluation tasks of three datasets. Specifically, the highest gains are up to 3.04% in miniImageNet (SemCo-trans with 20 labels) and 2.83% in tieredImageNet (FlexMatch-trans with 200 labels) over state-of-the-art SSL algorithms. This indicates the huge potency of meta-learning for SSL, since our approach is an initial and simple attempt to utilize meta-learning for SSL. More techniques can be introduced to combine two topics.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
I. Relation Between the Pseudo-Labeling Accuracy and Few-Shot Learning Performance

To investigate the impact of Pseudo-labeling (PL) accuracy, we provide the classification accuracy on the training set of base dataset, and the corresponding few-shot learning (FSL) accuracy in 5-way 5-shot tasks. As shown in Table V, PL and FSL accuracy scores generally co-improve with the increasing number of labels per class. Besides, our approach with only 20 labels achieves comparable performance with fully supervised models, even though the PL accuracy is low. A better backbone obtains a smaller gap of the full model (C4: 2.62% vs. R12: 1.49%), indicating an excellent backbone with our approach can improve the accuracy with a few labels.

J. Ablation Study of Our PLML

To evaluate the key components of our PLML, we conduct experiments on miniImageNet with extremely few labels (20) per class. We select the transductive EP with Conv4 as the baseline. As shown in Table VI, we first pre-train the model with SemCo and achieve significant improvement (6.69% and 5.25% in terms of 5-shot and 1-shot tasks) than EP. Then, we add our Learnable Smoothing Module for feature adjustment and also obtain large gains in both 5-shot (4.20%) and 1-shot (4.30%) tasks. Finally, we additionally apply the noise dropout strategy to remove noisy samples during training and observe further improvements in both tasks. Besides, we also test only using a transformer layer or feature propagation for feature adjustment. However, both of them achieve worse performance than our LSM, clearly demonstrating our model’s effectiveness.

K. Benefits of Pseudo-Labeling

To demonstrate the benefits of our pre-training method based on pseudo-labeling, we conducted experiments on miniImageNet. As shown in Table VII, we replaced the pre-training method in EP with our pseudo-labeling training stage (PL-EP) and achieved significant improvement in all tasks compared to the original EP. This indicates our pseudo-labeling can easily incorporated with other SSFSSL methods to improve the performance with fewer labels. How-
ever, the simple incorporation still suffers from the problem of noisy samples. Thus, we proposed finetuning with noise suppression to alleviate this issue and our PLML-EP achieves significant improvements under all conditions. Specifically, the highest gains are up to 10.28% with Conv4 as the backbone (20 labels per class) and 19.68% with ResNet12 as the backbone (20 labels per class).

L. Ablation Study for Hyperparameter

One of our main hyperparameters is the dropout rate $\alpha_d$ (in the noise dropout strategy of §IV-B). We observe the performance variation on the 5-way 5-shot task on miniImageNet with 20 labels, by using different $\alpha_d$. As shown in table VIII, when we only resort to our support set and do not dropout any samples ($\alpha_d = 0$), we achieve a slight improvement compared with the original set (LSM). Then, we increase $\alpha_d$ from 0.1 to 0.3 and obtain similar improvements. This indicates that dropping out a small number of samples will improve performance. When we increase it to 0.5, the accuracy reduces. This means removing too many samples will hurt the final training performance, because of the lacking of various samples. Therefore, we experimentally set our dropout rate $\alpha_d$ with a small value of 0.1.

The other main hyperparameter is the trade-off parameter $\gamma$ (in the training loss of §IV-B) of the base and few-shot learning (FSL) classification losses. To investigate its impact for both two tasks, we use different $\gamma$ for finetuning. As shown in table IX, when we do not use base classification loss ($\gamma = 0$), we improve both FSL performance compared with the pre-training model (65.69% vs. 61.39%) and the base classification accuracy (44.28% vs. 42.23). Then we try different $\gamma$ and find when $\gamma = 0.1$, we achieve the best performance in terms of both two tasks. Thus, $\gamma$ is experimentally set as 0.1.

M. Limits and Discussion

There are still some shortcomings to address. Firstly, achieving performance equivalent to models trained with full labeled data still necessitates a relatively large number of labels, typically around 100 labels, leaving substantial room for improvement in tasks with fewer labels, such as those with 20 or 50 (see Table I). Secondly, experimental results (e.g. Table II) suggest that the efficacy of the final few-shot learning method is closely intertwined with the effectiveness of semi-supervised learning. From this standpoint, enhancing performance further entails designing better training mechanisms to fully exploit meta-learning for the training of semi-supervised models.

VI. Conclusion

We proposed a novel semi-supervised setting for FSL to explore effective training solutions with a few labels. Under this setting, we found that most FSL approaches suffered

| Table VIII | Accuracy (%) of 5-way 5-shot tasks on miniImageNet [30] with 20 labels per class. We show the results with different trade-off parameter $\alpha_d$, based on the SemCo [24] pre-training and PLML-EP finetuning methods. The backbone is Conv4. The best values are in bold.
|-------------|-------------|-------------|-------------|-------------|-------------|
| Dropout Rate $\alpha_d$ | 0 | 0.1 | 0.2 | 0.3 | 0.5 | LSM |
| 5-way 5-shot | 65.74 | **66.3** | 66.14 | 66.18 | 65.78 | 65.59 |

| Table IX | Accuracy (%) of 5-way 5-shot tasks and base classification on miniImageNet [30] with 20 labels per class. We show the results with different trade-off parameter $\gamma$, based on the SemCo [24] pre-training and PLML-EP finetuning methods. “Pre” refers to pre-training. The backbone is Conv4. The best values are in bold.
|-------------|-------------|-------------|-------------|-------------|-------------|
| Parameter $\gamma$ | 0 | 0.1 | 0.2 | 0.3 | 1 | 2 | Pre-training |
| 5-way 5-shot | 65.69 | **66.3** | 65.06 | 65.62 | 64.77 | 64.39 | 61.39 |
| Base Classification | 44.28 | **45.28** | 45.23 | 44.61 | 44.61 | 43.33 | 42.23 |

| Table X | List of Symbols and Abbreviations
|-------------|-------------|-------------|
| Symbol/Abbreviation | Definition |
| FSL | Few-shot learning |
| IFSL | Inductive few-shot learning |
| TFSL | Transductive few-shot learning |
| SSFSL | Semi-supervised learning |
| *PLML | Our method |
| $D_{\text{base}}$ | Base set |
| $D_{\text{novel}}$ | Novel set |
| $D_s$ | Semi-supervised setting |
| $D_u$ | Unlabeled set in the semi-supervised setting |
| $D_t$ | Testing set in the semi-supervised setting |
| $D_p$ | Pseudo-labeling dataset |
| $f_{\text{finetune}}$ | Semi-supervised learning model |
| $f_{\text{rl}}$ | Few-shot learning model |
| $\phi$ | Feature extractor |
| $r_f$ | M-way classifier |
| $c_b$ | Base classifier |
| $Y_r$ | Predicting label of $c_f$ |
| $Y_p$ | Predicting label of $c_b$ |
| $T$ | Transformer operation |
| $Z$ | Original feature matrix |
| $Z_t$ | Feature matrix adjusted by $T$ |
| $A_{i,j}$ | Adjacency matrix element |
| $d_i^2$ | Squared Euclidean distance |
| $\sigma_i^2$ | Variance |
| $P$ | Propagator matrix |
| $I$ | Identity matrix |
| $L$ | Laplacian matrix |
| $D_{r}$ | Degree of node i |
| $\alpha_p$ | Scaling factor |
| $Z_p$ | Feature matrix adjusted by propagation |
| $z_{si}$ | Feature of the anchor sample |
| $\alpha_d$ | Dropout rate |
from huge performance degradation with the reduction of labels. To alleviate this issue, we presented a simple yet effective two-stage learning framework, called meta-learning with pseudo-labeling (PLML), which successfully utilizes the advantages of SSL and FSL for effective learning from a few labels. We integrated PLML into three SSL and FSL models. Extensive results on two datasets have demonstrated that our approach not only provided significant improvement over FSL baselines but also achieved superior performance than SSL approaches not only provided significant improvement over FSL learning via multi-factor clustering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 14544–14553.

[21] J. Ling, L. Liao, M. Yang, and J. Shuai, “Semi-supervised few-shot learning via feature alignment,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 7237–7246.

[22] Y. Liu et al., “Learning to propagate labels: Transductive propagation network for few-shot learning,” 2018, arXiv:1805.10002.

[23] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[24] I. Nasar, S. Herath, E. Abbasnejad, W. Buntine, and G. Hafari, “All labels are not created equal: Enhancing semi-supervision via label grouping and co-training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 7237–7246.

[25] B. Oreshkin, P. R. López, and A. Lacoste, “TADAM: Task dependent adaptive metric for improved few-shot learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR), Jun. 2018, pp. 719–729.

[26] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.

[27] L. Qiao, Y. Shi, J. Li, Y. Tian, T. Huang, and Y. Wang, “Transductive episodic-wise adaptive metric for few-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3603–3612.

[28] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine, “Meta-learning with implicit gradients,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 113–124.

[29] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, “Semi-supervised learning with ladder networks,” 2015, arXiv:1507.02672.

[30] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in Proc. Int. Conf. Learn. Represent., 2017. [Online], Available: https://openreview.net/forum?id=dYJo-KcIl.

[31] M. Ren et al., “Meta-learning for semi-supervised few-shot classification,” in Proc. Int. Conf. Learn. Represent., 2018. [Online], Available: https://openreview.net/forum?id=HcKsz-CZ.

[32] P. Rodriguez, I. Laradji, A. Drouin, and A. Lacoste, “Embedding propagation: Smoother manifold for few-shot classification,” in Proc. Eur. Conf. Comput. Vis., Aug. 2020, pp. 121–138.

[33] L. Samuli and A. Timo, “Temporal ensembling for semi-supervised learning,” in Proc. Int. Conf. Learn. Represent., 2017. [Online], Available: https://openreview.net/forum?id=BJ6oQgge.

[34] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in Proc. Int. Conf. Mach. Learn., Jun. 2016, pp. 1842–1850.

[35] C. Simon, P. Koniusz, R. Nock, and M. Harandi, “Adaptive subspaces for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4136–4145.

[36] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 30, 2017, pp. 4077–4087.

[37] K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2020, pp. 596–608.

[38] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, “A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities,” ACM Comput. Surv., vol. 55, no. 13s, pp. 1–40, Dec. 2023.

[39] T. Sosea and C. Caragea, “MarginMatch: Improving semi-supervised learning with pseudo-margins,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2023, pp. 15773–15782.

[40] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 1195–1204.
Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, “Rethinking few-shot image classification: A good embedding is all you need?” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 266–282.

A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 5998–6008.

O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, “Matching networks for one shot learning,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2016, pp. 3630–3638.

Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, “Instance credibility inference for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 12836–12845.

Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” in Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 6256–6268.

X. Yang, Z. Song, I. King, and Z. Xu, “A survey on deep semi-supervised learning,” 2021, arXiv:2103.00550.

H.-J. Ye, X.-C. Li, and D.-C. Zhan, “Task cooperation for semi-supervised few-shot learning,” in Proc. AAAI Conf. Artif. Intell., May 2021, vol. 35, no. 12, pp. 10682–10690.

Z. Yu, L. Chen, Z. Cheng, and J. Luo, “TransMatch: A transfer-learning scheme for semi-supervised few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12856–12864.

B. Zhang et al., “FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 18408–18419.

Z. Zhu, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, “Learning with local and global consistency,” in Adv. Neural Inf. Process. Syst., 2003, pp. 321–328.

H. Zhu and P. Koniusz, “Transductive few-shot learning with prototype-based label propagation by iterative graph refinement,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2023, pp. 23996–24006.

L. M. Zitnograf, K. Shiarlis, V. Kurin, K. Hofmann, and S. Whiteson, “Fast context adaptation via meta-learning,” in Proc. Int. Conf. Mach. Learn. (ICML), Jun. 2019, pp. 7693–7702.

Xingping Dong (Member, IEEE) received the B.S. degree from Xiamen University in 2012 and the Ph.D. degree from Beijing Institute of Technology in 2019. From 2016 to 2018, he was a joint Ph.D. Student with the College of Engineering and Computer Science, The Australian National University. He is currently a Professor with the School of Computer Science, Wuhan University. His current research interests include generative AI, vision and language, and AI for healthcare. He is a fellow of IAPR, BCS, and IET.

Tianran Ouyang received the B.S. degree from the Huazhong University of Science and Technology in 2018 and the master’s degree from Peking University in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University. His current research interests include computer vision, meta-learning, and deep learning.

Shengcai Liao (Senior Member, IEEE) received B.S. degree in mathematics from Sun Yat-sen University in 2005 and the Ph.D. degree from CASIA in 2010. He is an Associate Professor with the College of Information Technology (CIT) in UAEU. He was a Principal Scientist in IIAI, G42 in UAE during 2018-2024, and an Associate Professor with the Institute of Automation, Chinese Academy of Sciences (CASIA), from 2014 to 2018. He was a Postdoc with Michigan State University during 2010-2012. He is interested in face and person detection, recognition, analysis, and image and video generation. He was awarded Best Paper/Best Student Paper in ICB 2006/2007/2015, and CCBR 2016, Best/Outstanding Reviewer in IJCB 2014, CVPR 2019/2021. He served as an Assistant Editor for “Encyclopedia of Biometrics, 2nd Ed.”, Guest Associate Editor for IEEE T-BIOM, and Associate Editor for IEEE TIP and TNNLS. He served as Program Chair for IJCB 2022 and Area Chair for ICPR 2016, ICB 2016/2018, CVPR 2022/2023/2025, ICCV 2023, NeurIPS 2023/2024, ECCV 2024, and WACV 2024/2025.

Bo Du (Senior Member, IEEE) is currently a Professor with the School of Computer Science and the Institute of Artificial Intelligence, Wuhan University, Wuhan, China. He is also the Director of the National Engineering Research Center for Multimedia Software, Wuhan University. He has more than 80 research articles published in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. 24 of them are ESI hot articles or highly cited articles. His research interests include machine learning, computer vision, and image processing. He regularly serves as a Senior PC Member for IJCAI and AAAI. He won the IEEE Geoscience and Remote Sensing Society 2020 Transactions Prize Paper Award, the International Joint Conferences on Artificial Intelligence (IJCAI) Distinguished Paper Prize, the IEEE Data Fusion Contest Champion, and the IEEE Workshop on Hyperspectral Image and Signal Processing Best Paper Award in 2018. He served as the Area Chair for IJCB.