TENET: Beyond Pseudo-Labeling for Semi-supervised Few-shot Learning

Chengcheng Ma[1,2], Weiming Dong[2] and Changsheng Xu[2]

[1]School of Artificial Intelligence, UCAS, Beijing, 100049, China.
[2]NLPR, CASIA, Beijing, 100190, China.

Corresponding author: Weiming Dong (weiming.dong@ia.ac.cn);

Abstract

Few-shot learning attempts to identify novel categories by exploiting limited labeled training data, while the performances of existing methods still have much room for improvement. Thanks to a very low cost, many recent methods resort to additional unlabeled training data to boost performance, known as semi-supervised few-shot learning (SSFSL). The general idea of SSFSL methods is to first generate pseudo labels for all unlabeled data and then augment the labeled training set with selected pseudo-labeled data. However, almost all previous SSFSL methods only take supervision signal from pseudo-labeling, ignoring that the distribution of training data can also be utilized as an effective unsupervised regularization. In this paper, we propose a simple yet effective SSFSL method named *TENET*, which takes low-rank feature reconstruction as the unsupervised objective function and pseudo labels as the supervised constraint. We provide several theoretical insights on why TENET can mitigate overfitting on low-quality training data, and why it can enhance the robustness against inaccurate pseudo labels. Extensive experiments on four popular datasets validate the effectiveness of TENET.

Keywords: Semi-supervised few-shot learning, few-shot learning, pseudo-labeling, linear regression, low-rank reconstruction.

1 Introduction

Although deep learning has made great progress in a variety of visual recognition tasks in recent years [1–3], it heavily depends on a large amount of labeled training data, which is always costly and time-consuming to obtain in real-world situations [4, 5]. In comparison, humans can easily learn from one or just a few examples to identify novel objects. Motivated by this fact, few-shot learning (FSL) has recently attracted great research interest recently [6, 7], which aims to make recognition based on extremely limited training data similar to humans.

In general, the existing FSL methods can be summarized into meta-learning-based [8–13] and transfer-learning-based [14, 15]. However, their performances still have much room for improvement compared with the regular many-shot training. Inspired by semi-supervised learning [16], many recent methods [17–25] have been proposed to augment the few-shot labeled training set with additional unlabeled data, since the latter is much cheaper. Such methods are known as semi-supervised few-shot learning (SSFSL), whose general idea is to first generate pseudo labels for all unlabeled data, and then select the most credible

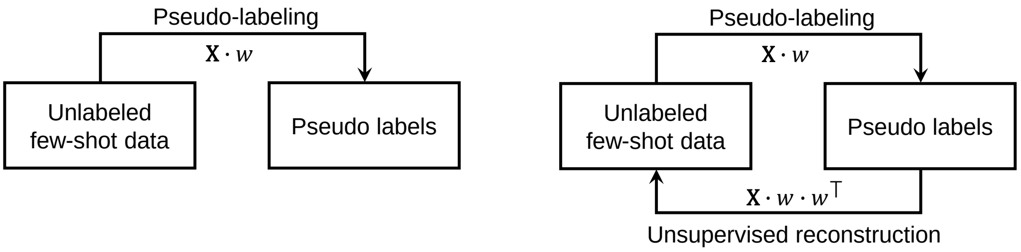


Fig. 1: The difference between the conventional pseudo-labeling method (left) and our TENET (right). In comparison, the pseudo labels are additionally utilized by TENET to reconstruct the unlabeled data, which acts as an unsupervised regularization to benefit the training process.

pseudo-labeled data for the augmentation. Referring to the few-shot classification leaderboard[1], the SSFSL methods can always outperform the regular FSL methods.

Although effective, almost all previous SSFSL methods only take the pseudo labels as supervision signal to train the classifier, which means that their performances are heavily affected by the quality of pseudo labels. In fact, most of these methods focus on how to select the samples with more accurate pseudo labels to augment the labeled training set, which can be complicated and empirical.

Therefore, beyond pseudo-labeling, can we extract an effective unsupervised signal for the SSFSL task? In this paper, we point out that the distribution of training data can also be utilized as an effective unsupervised regularization term. We propose a simple yet effective SSFSL method named TENET, which takes the low-rank feature reconstruction as the unsupervised objective function and pseudo labels as the supervised constraint. The core idea of TENET is illustrated in Fig. 1. As shown, the classifier weight projects the high-dimensional feature space into low-dimensional category space, while TENET hopes the transpose of the weight matrix can project the latter back to the former. In other words, TENET minimizes the total distance between low-rank reconstructed features and their original version, and such process is independent of pseudo labels. We will provide more theoretical insights in Sec 4.

In practice, we choose the simplest linear model, linear regression, for the task of SSFSL classification. We conduct extensive experiments on four widely used datasets and two few-shot settings. With a vanilla sample selection process, TENET can achieve comparable and even better performances than the previous state-of-the-art baselines, which validates the effectiveness of TENET.

In summary, the main contributions of this paper can be summarized as

- Based on the linear regression model, we analyze the non-robustness and overfitting issues in the existing SSFSL methods.
- We propose a novel yet effective method named TENET, which additionally utilizes the distribution of training data as unsupervised regularization term to benefit the SSFSL training.
- Extensive experimental results on four popular benchmarks including miniImageNet, tieredImageNet, CIFAR-FS and CUB, validates the effectiveness of TENET.

2 Related Work

Semi-supervised learning. When the costly labeled data is limited, semi-supervised learning (SSL) aims to improve the model performance by leveraging large amount of cheap unlabeled data, thereby saving the cost of data labeling. Currently, the mainstream SSL approaches are usually based on three assumptions, including the low-density separation [16, 26], cluster assumption [26], and manifold assumption [26]. Specifically, the low-density separation states that the decision boundary should pass through the low-density data regions, and avoid cutting a high density region into two different classes [26, 27].

[1] https://few-shot.yyliu.net/

Similarly, the cluster assumption states that when two samples are close to each other in the input space, they should belong to the same class [28–30]. Furthermore, after projecting the input space to the feature space, the manifold assumption states that the two close samples in the feature space should also belong to the same class [31, 32].

However, most of existing SSL approaches are usually not suitable for the semi-supervised few-shot problems, where the amounts of labeled and unlabeled data are both limited. As a result, it is difficult to distinguish whether two samples are close enough. For instance, [33] found that directly applying MixMatch [34] to few-shot learning leads to poor performance, especially in the low-shot settings.

Few-shot learning. Few-shot learning (FSL) aims to quickly adapt the deep models to novel tasks by exploiting only a few labeled samples. Existing approaches can be separated into two branches, *e.g.* meta-learning-based approaches [8–13] and transfer-learning-based approaches [14, 15]. The former branch usually designs a learning paradigm for task adaptation, and can be further divided into two categories: a) metric-based methods [8–10], which measure the sample distances between the query set and the support set, and classify images via nearest neighbors, and b) optimization-based methods [11–13], which design a specific optimization method for the few-shot training set,

Transfer-learning-based approaches [14, 15] usually pretrain a deep model on the base task, and conduct fine-tuning with the few training data from the novel task. Concretely, [15] pointed out that fixing the parameters of the feature extractor and only training a simple classifier can lead to similar performances with meta-learning-based methods, sometimes even better. Our method is based on the transfer-learning framework due to its simplicity and universality.

Semi-supervised few-shot learning. Thanks to a very low cost, many recent FSL methods [17–25] resort to additional unlabeled training data for boosting performance, which are known as semi-supervised few-shot learning (SSFSL). Similar to SSL, the core idea of SSFSL is to first generate pseudo labels for all unlabeled training samples, and then select the most credible pseudo-labeled samples to expand the few-shot labeled training set.

As pioneer works, [17] proposes advanced K-means clustering algorithms, while [18] and [20] adopt the label propagation or embedding propagation algorithms to generate pseudo labels via sample similarity in the manifold space, and [23] further resorts to the label denoising method [35] for refinement. However, all these methods need to compute graph matrices, so their running efficiencies are unsatisfactory in face of large-scale meta-testing sets. In addition, [19] adopts the self-training SSL method in a meta-learning manner, which cherry-picks credible samples for multiple times. [21] designs a linear classifier to evaluate the credibility for sample selection. Similar to [21], [24] utilizes Gaussian mixture models to fit the distribution of classification loss, and takes the fitness score for sample selection. However, all their performances are restricted by the quality of pseudo labels, and [21] is sensitive to the hyper-parameter choices [22, 23]. Recently, [25] applies negative learning [36], introducing an additional supervision signal to mitigate the adverse effects of inaccurate pseudo labels. However, our experiments show that the effectiveness of negative learning for the SSFSL problem is limited in low-shot settings, so an effective supervision signal is still needed.

In addition, [37] also conducts unsupervised, or namely self-supervised learning on unlabeled data to improve the FSL performance. The main difference lies in that [37] aims to improve the representation ability of the feature extractor, while our approach aims for the classifier, which is orthogonal to [37].

3 Background

In this section, we first introduce the problem formulation of semi-supervised few-shot learning (SSFSL), and then point out the flaws of the common objective function used in the existing SSFSL approaches.

3.1 Problem formulation

The SSFSL task includes five basic elements: a representation function, a few-shot labeled training set, a few-shot unlabeled training set, a test set, and a linear classifier. We now explain them in detail.

The representation function $f : \mathcal{X} \to \mathbb{R}^d$ is usually a feature extractor that maps the input space $\mathcal{X}$ to a discriminative feature space $\mathbb{R}^d$. As the size of the training set in the SSFSL task is too small to train $f(\cdot)$, transfer-learning-based methods [17, 19, 21, 23–25] usually pretrain $f(\cdot)$ on an auxiliary dataset with many-shot labeled data, denoted as $\mathcal{D}_{base} = \{(\boldsymbol{x}_i, y_i), y_i \in \mathcal{C}_{base}\}$. In this paper, we follow previous methods [17, 19, 21, 23–25] to freeze the parameters of $f(\cdot)$ after the pretraining stage.

The few-shot labeled training set, also called the support set, is denoted as $\mathcal{D}_{novel} = \{(\boldsymbol{x}_i, y_i), y_i \in \mathcal{C}_{novel}\}$, where $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \varnothing$. Specifically, the category set $\mathcal{C}_{novel}$ contains C categories, and each category contains K images, thus $|\mathcal{D}_{novel}| = C \times K$. Most of previous SSFSL works set C as 5 and set K as 1 or 5, which is known as the 5-way-1-shot problem or 5-way-5-shot problem. All image features extracted by the representation function are denoted as the matrix $\boldsymbol{X}_{\text{novel}} \in \mathbb{R}^{(C \times K) \times d}$.

The few-shot unlabeled training set $\mathcal{U}_{novel}$ shares the same category set with $\mathcal{D}_{novel}$, where each category contains U images, thus $|\mathcal{U}_{novel}| = C \times U$. Commonly, U is set as 30 or 50 when K equals 1 or 5. Many previous SSFSL works first generate pseudo labels for samples in $\mathcal{U}_{novel}$, and then expand $\mathcal{D}_{novel}$ as $\mathcal{D}_{novel} \cup \mathcal{U}_{novel}$ to train a better classifier. All extracted features are denoted as the matrix $\boldsymbol{X}_{\text{unlabel}} \in \mathbb{R}^{(C \times U) \times d}$.

The test set $\mathcal{D}_{test}$, also known as the query set, shares the same category set with $\mathcal{D}_{novel}$ and $\mathcal{U}_{novel}$, which is used for evaluation. Each category usually contains 15 images. The extracted features are denoted as $\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{(C \times 15) \times d}$. In particular, in the transductive few-shot setup, the test set is accessible during the training process.

Following previous SSFSL methods [17, 19, 21, 23–25], the general idea of the SSFSL method is to train a linear classifier $g(\cdot)$ on both $\mathcal{D}_{novel}$ and $\mathcal{U}_{novel}$ to classify images in $\mathcal{D}_{test}$ correctly. The inference process can be denoted as $\hat{\boldsymbol{Y}} = g(f(\boldsymbol{x})) = \sigma(\boldsymbol{X}\boldsymbol{w})$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ denotes the extracted features of n images, $\boldsymbol{w} \in \mathbb{R}^{d \times C}$ denotes the weight of the linear classifier $g(\cdot)$, $\hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times C}$ denotes the prediction results, and $\sigma(\cdot)$ denotes the activation function. Correspondingly, the true label is denoted as $\boldsymbol{Y} \in \mathbb{R}^{n \times C}$ in one-hot form.

Most of previous approaches [19, 21–25] are conducted in an iterative manner: at the 0-th iteration, the classifier g_0 is trained on $\mathcal{D}_{novel}$. At any t-th iteration, the classifier from the last iteration g_{t-1} predicts all remaining images in $\mathcal{U}_{novel}$, and some certain pseudo-labeled images are then removed from $\mathcal{U}_{novel}$ and put into $\mathcal{D}_{novel}$ (usually the same number of images per class). After that, a new classifier g_t is trained from scratch based on the updated $\mathcal{D}_{novel}$. The iteration process ends when $\mathcal{U}_{novel}$ is empty, and the last classifier g_T is utilized for evaluation (T is the iteration number).

3.2 Flaws of the common objective function

The existing SSFSL methods usually choose a single fully-connected layer [19, 25] or a logistic regression model [21–24] as the linear classifier $g(\cdot)$. The differences lie in the loss function, activation function and optimization method, but they all belong to the generalized linear model in essence. Without loss of generality, our experiments are all based on the simplest linear model, that is, the linear regression (LR) model. For the SSFSL problem, the objective function of the LR model is

$$\min \|\boldsymbol{w}\|_2 \quad \text{s.t. } \boldsymbol{X}\boldsymbol{w} = \boldsymbol{Y}. \tag{1}$$

The row number of feature matrix $\boldsymbol{X}$ is always less than the column number[2], so $\boldsymbol{X}$ is a row full-rank matrix, and $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{Y}$ is an under-determined equation. In other words, there are infinite solutions of the equation, while the objective function (1) aims to find the solution $\boldsymbol{w}^*$ with the least ℓ_2 norm. Using the Lagrange multiplier method, we can easily obtain the closed-form of $\boldsymbol{w}^*$ as

$$\boldsymbol{w}^* = \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{Y}. \tag{2}$$

Based on this, at the t-th iteration process of SSFSL, the optimal classifier weight is computed as

$$\boldsymbol{w}_t^* = \boldsymbol{X}_{\text{novel}}^\top(\boldsymbol{X}_{\text{novel}}\boldsymbol{X}_{\text{novel}}^\top)^{-1}\boldsymbol{Y}_{\text{novel}}. \tag{3}$$

[2] The feature dimension $d = 512$ with $f(\cdot)$ being ResNet-12 [38], while the largest n equals 5×(5+50)=275 (a 5-way-5-shot problem) and is still less than d.

However, there are two flaws of the common LR model. **(i)** If the feature matrix $\boldsymbol{X}_{\text{novel}}$ contains very similar row elements or outliers, the inverse matrix $(\boldsymbol{X}_{\text{novel}}\boldsymbol{X}_{\text{novel}}^{\top})^{-1}$ may suffer from numerical instability. Therefore, the LR model is greatly affected by the data quality and is not robust[3]. **(ii)** For any solution $\boldsymbol{w}_{\triangle}$ of equation $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{Y}$, let $\boldsymbol{w}_{\triangle} = \boldsymbol{w}^{*} + r$ and r be a residual vector in the null space $\mathcal{N}(\boldsymbol{X}_{\text{novel}})$, and we can know from Cauchy-Schwartz inequality that $\|\boldsymbol{w}^{*}\|_2 < \|\boldsymbol{w}_{\triangle}\|_2$, which means that optimizing the objective (1) will force $\boldsymbol{w}^{*}$ orthogonal to $\mathcal{N}(\boldsymbol{X}_{\text{novel}})$. However, if $\boldsymbol{X}_{\text{novel}}$ cannot represent the test set distribution well, then $\mathcal{N}(\boldsymbol{X}_{\text{novel}})$ may contain beneficial vectors for better generalization ability on the test set, so the orthogonality to $\mathcal{N}(\boldsymbol{X}_{\text{novel}})$ may lead to overfitting. For example, suppose that a few-shot training set $\boldsymbol{X}_1$ cannot represent the test set distribution well, while a larger training set $\boldsymbol{X}_{12} = \boldsymbol{X}_1 \cup \boldsymbol{X}_2$ can do (the number of training data contained by $\boldsymbol{X}_{12}$ is still less than the number of variables). Accordingly, we denote the optimal solution trained on $\boldsymbol{X}_{12}$ by $\boldsymbol{w}_{12}^{*} = \boldsymbol{w}_{1}^{*} + \Delta\boldsymbol{w}$. Because $\boldsymbol{w}_{12}^{*}$ still satisfies $\boldsymbol{X}_1\boldsymbol{w}_{12}^{*} = \boldsymbol{Y}_1$, we can know that the residual $\Delta\boldsymbol{w}$ lies in the null space $\mathcal{N}(\boldsymbol{X}_1)$, indicating that $\mathcal{N}(\boldsymbol{X}_1)$ contains beneficial vectors for better generalization ability onto the test set.

4 Methodology

As we discussed above, the common linear regression (LR) suffers from non-robustness and overfitting, so it is necessary to explore more supervision signals from the few-shot training set and modify the objective function. In this paper, we propose a simple yet quite effective approach for the SSFSL task, *i.e.* feaTurE recoNstruction based rEgression meThod (TENET). The idea of TENET and the comparison to LR are illustrated in Fig. 2.

Recalling the principal component analysis (PCA) algorithm [39], its purpose is to find a set of orthogonal basis that can preserve the data distribution to the maximum extent, and also make the low-rank reconstructed data closest to the original data. For the SSFSL task, since the weight matrix $\boldsymbol{w}$ can project the n-dim feature space into the C-dim category space, we encourage $\boldsymbol{w}$ to be able to reconstruct the feature samples in a C-rank space. Inspired by PCA, we propose to modify the objective function (1) as

$$\min \left\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top}\right\|_F \quad \text{s.t. } \boldsymbol{X}\boldsymbol{w} = \boldsymbol{Y} \tag{4}$$

Such an objective function has four advantages. **(i)** $\boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top}$ represents the C-rank reconstruction of feature $\boldsymbol{X}$, and minimizing $\left\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top}\right\|_F$ will no longer force the optimal solution to be orthogonal to $\mathcal{N}(\boldsymbol{X})$ anymore, which is because

$$\begin{aligned}\boldsymbol{X}(\boldsymbol{w}+r)(\boldsymbol{w}+r)^{\top} &= \boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top} + 2\boldsymbol{X}r\boldsymbol{w}^{\top} + \boldsymbol{X}rr^{\top} \\ &= \boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top} + \boldsymbol{0} + \boldsymbol{0},\end{aligned}$$

where $\boldsymbol{X}r = \boldsymbol{0}$. The overfitting issue can thus be mitigated. **(ii)** The feature reconstruction process implies an orthogonal regularization to the solution, because when $\left\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{w}\boldsymbol{w}^{\top}\right\|_F \to \boldsymbol{0}$, we have $\left\|\boldsymbol{Y}\boldsymbol{w}^{\top}\boldsymbol{w} - \boldsymbol{Y}\right\|_F \to \boldsymbol{0}$, which means $\boldsymbol{w}^{\top}\boldsymbol{w} \to \mathbf{I}$. Generally, the orthogonality of the network weight indicates better generalization capability and robustness [40, 41]. **(iii)** The feature reconstruction process aims to capture the distribution information of $\boldsymbol{X}$, which is naturally robust to the very similar samples and outliers [39]. **(iv)** The feature reconstruction process is independent of pseudo labels $\hat{\boldsymbol{Y}}$, so the adverse effects brought by inaccurate pseudo labels can be circumvented. In other words, the feature reconstruction process acts as an **unsupervised regularization** for the SSFSL task.

However, harder than linear regression, we cannot derive the closed-form solution of objective (4). Inspired by [42], we first rewrite the objective (4) as

$$\min \left\|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{w}^{\top}\right\|_F^2 + \lambda \cdot \left\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\right\|_F^2, \tag{5}$$

where λ is a trade-off parameter. By setting the first derivative of (5) as zero, we then apply the Bartels-Stewart algorithm [43] to solve the equation below to obtain the optimal solution $\boldsymbol{w}^{*}$

$$\lambda\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}^{*} + \boldsymbol{w}^{*}\boldsymbol{Y}^{\top}\boldsymbol{Y} = (1+\lambda)\boldsymbol{X}^{\top}\boldsymbol{Y}. \tag{6}$$

[3] We emphasize that (2) is not the closed-from solution of logistic regression, while the logistic regression model is also greatly affected by noisy data.

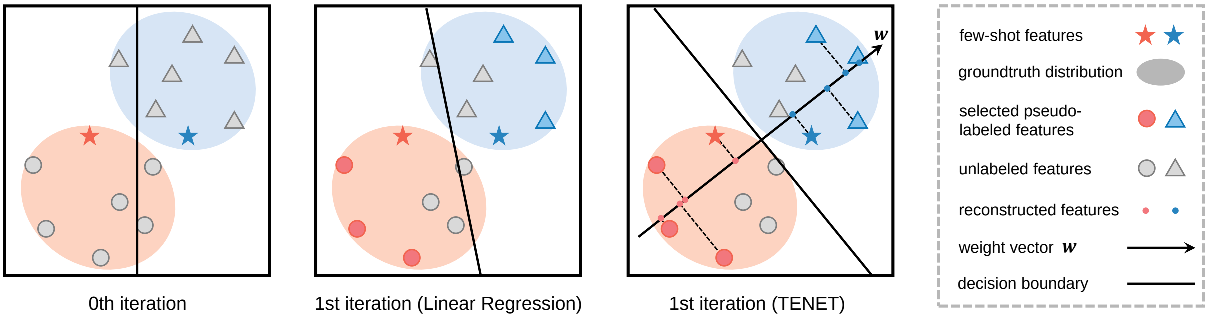


Fig. 2: A 2D toy example on a 2-way-1-shot SSFSL task, with 6 unlabeled samples per class. After the 0th iteration, the 3 most confident pseudo-labeled samples per class (farthest from the decision boundary) are selected. At the 1st iteration, our TENET works better than common linear regression, as TENET captures the distribution information of few-shot labeled and unlabeled data, by minimizing the total distance between the low-rank reconstructed training samples and their original version.

```
Algorithm 1 Inference process of TENET
Input: labeled training data $(\boldsymbol{X}_{\text{novel}}, \boldsymbol{Y}_{\text{novel}})$, unlabeled training data $\boldsymbol{X}_{\text{unlabel}}$, test data $\boldsymbol{X}_{\text{test}}$.
Obtain $\boldsymbol{w}_0$ by solving Equation (6) based on $(\boldsymbol{X}_{\text{novel}}, \boldsymbol{Y}_{\text{novel}})$.
$t \leftarrow 0$
while $t \leq T$ do
    Get prediction $\hat{\boldsymbol{Y}}_{\text{unlabel}} = \boldsymbol{X}_{\text{unlabel}} \cdot \boldsymbol{w}_t$
    Select a subset of $(\boldsymbol{X}_{\text{unlabel}}, \hat{\boldsymbol{Y}}_{\text{unlabel}})$ as $(\boldsymbol{X}_{\text{select}}, \hat{\boldsymbol{Y}}_{\text{select}})$ according to indices in (7).
    $\boldsymbol{X}_{\text{novel}} \leftarrow [\boldsymbol{X}_{\text{novel}};\ \boldsymbol{X}_{\text{select}}]$
    $\boldsymbol{Y}_{\text{novel}} \leftarrow [\boldsymbol{Y}_{\text{novel}};\ \hat{\boldsymbol{Y}}_{\text{select}}]$
    Obtain $\boldsymbol{w}_t$ by solving Equation (6) based on new $(\boldsymbol{X}_{\text{novel}}, \boldsymbol{Y}_{\text{novel}})$.
    $t \leftarrow t + 1$
end while
Output: Get prediction $\hat{\boldsymbol{Y}}_{\text{test}} = \boldsymbol{X}_{\text{test}} \cdot \boldsymbol{w}_T$.
```

In practice, our TENET approach follows the existing SSFSL approaches to be conducted in an iterative manner, as introduced in Sec 3.1. The entire process is summarized in Algorithm 1. Specifically, TENET selects the top-k confident pseudo-labeled samples per class to expand the labeled training set, and the selected indices are formulated as

$$I = \left\{ i \,\middle|\, i \in \text{topk}\left(\max_j \left[\text{softmax}(\hat{\boldsymbol{Y}}_{\text{unlabel}} \cdot s) \right]_{ij} \right) \right\}, \tag{7}$$

where s denotes the scaling factor. In addition, optimizing the objective (5) will lead to two forms of pseudo labels, as the first term leads to $\hat{\boldsymbol{Y}}_{\text{unlabel}} = \boldsymbol{X}_{\text{unlabel}} \cdot \boldsymbol{w}(\boldsymbol{w}^\top \boldsymbol{w})^{-1}$ and the second term leads to $\hat{\boldsymbol{Y}}_{\text{unlabel}} = \boldsymbol{X}_{\text{unlabel}} \cdot \boldsymbol{w}$. We find in experiments that both leads to similar performances, indicating that $\boldsymbol{w}$ has orthogonality.

5 Experiments

5.1 Experimental settings

Datasets. We evaluate the proposed approach on four publicly available benchmarks including miniImageNet [12], tieredImageNet [17], CIFAR-FS [44], and CUB [45], following previous SSFSL works [21–23, 25]. MiniImageNet is a subset of ImageNet [46] containing 64 base classes and 20 novel classes (600 images per class). We follow the commonly used data split proposed by [12]. TieredImageNet is also sampled from ImageNet but organized in a hierarchical label structure. It contains 351 base classes (448,695 images in total) and 160 novel classes (206,209 images in total). We follow the data split in [47]. CIFAR-FS is sampled from CIFAR-100 [48] containing 64 base classes and 20 novel classes (600 images per class). The data split is defined by [49]. CUB is a fine-grained bird dataset of 200 different species. We follow the split proposed by [49, 50] with 100 base classes and 50 novel classes. For fair comparisons with [21–23, 25], we crop all CUB images according to the bounding boxes provided by [45]. All images in four datasets are resized to 84×84 to fit the feature extractor network.

Note that there are multiple preprocessed versions of miniImageNet and tieredImageNet, and different versions can lead to large performance gaps[4]. Meanwhile, the versions of these two datasets chosen by different baseline methods are inconsistent (sometimes even unknown), so it is somehow unrealistic to make fair comparisons. In our experiments, we choose the original version of these two datasets without any preprocessing. Thus, the corresponding performances of TENET are shown less competitive. For the sake of fairness, we further make a comparison with the ICI baseline [21] under the same version of tieredImageNet in the following as an example, based on its own implementation[5].

Metrics. Similar to all baseline methods, our TENET is evaluated over 600 episodes with 15 test samples per class. We report the mean top-1 accuracy over 600 episodes.

Implementation Details. We choose the commonly used ResNet-12 [38] as the representation function $f(\cdot)$, and the network architecture follows [21–24]. For pretraining, we follow all the training settings in [22] and [25]: the total epoch number is 120, the optimizer is SGD with 0.9 momentum and $5e-4$ weight decay, and the learning rate is initialized as 0.1 and decayed as $6e-3$, $1.2e-3$ and $2.4e-4$ after the 60th, 70th and 80th epochs, respectively. For TENET, the trade-off parameter λ in objective function (5) is default set as 20, and the scaling factor s in the selection process (7) is 10. We will conduct ablation study on λ and s in the following. We utilize the scipy.linalg.solve_sylvester[6] API provided by scipy [62] to solve the Equation (6). Regarding the experimental settings, each of the features $\boldsymbol{X}$ extracted by $f(\cdot)$ is vectorized and ℓ_2-normalized, following [21–25]. The per-class number of unlabeled samples U is set as 30 and 50 in the 1-shot and 5-shot settings, respectively. We set the SSFSL iteration number T as 5 in the semi-supervised few-shot setup (we select 6 and 10 samples per class in the 1-shot and 5-shot setting, respectively), and set T as U (we select 1 sample per class) in the transductive few-shot setup. We utilize the last classifier for evaluation.

5.2 Experimental results

Semi-supervised few-shot setup. In the inductive setup, we compare our TENET with both regular few-shot learning (training without unlabeled samples) and semi-supervised few-shot learning methods. All the results are sourced from their original papers. As shown in Table 1, TENET outperforms all baselines on CIFAR-FS and CUB, and achieves comparable performances on miniImageNet and tieredImageNet. As discussed above, we further make a comparison with one of the previous methods ICI [21], based on the same dataset and pretrained representative function. As shown in Table 3, TENET leads to better performances most of the time. Such experimental results show the effectiveness of our TENET.

Transductive few-shot setup. The transductive few-shot setup is another important setup in the field of SSFSL, where the test set is accessible during the classifier training, and no unlabeled set is needed. In other words, the unsupervised information of the test set can be utilized to help train a better classifier. We compare TENET and baseline methods in Table 2 and 3. Again, the baseline performances are sourced from their papers. As shown, TENET almost always beats all baselines in both 1-shot and 5-shot settings.

Apart from the default ResNet-12 backbone, we follow [23] to utilize WRN-28-10 [13] and compare TENET with ICI [21], PT+MAP [61], EASE [60], and iLPC [23], and keep all training settings the same as [23], such as 1000 episodes and the same preprocessing on input features, and the same postprocessing on predictions. Table 4 shows that TENET achieves comparable performances with existing baselines, except for ICI with a $\sim 1.5\%$ improvement.

Variety-unlabeled semi-supervised few-shot setup. To evaluate the stability of TENET,

[4] https://github.com/Yikai-Wang/ICI-FSL/issues/8

[5] https://github.com/Yikai-Wang/ICI-FSL/tree/master/V1-CVPR20

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.solve_sylvester.html

Table 1: Comparisons of 5-way few-shot classification with the inductive setup. The light blue blocks represent that these methods are tested in the regular few-shot setup, and the light yellow blocks are tested in the semi-supervised few-shot setup. The best and second best performances are highlighted in **red** and **blue**, respectively. *: using the average of feature vectors under multiple times of data augmentations.

Method	Backbone	miniImageNet		tieredImageNet		CIFAR-FS		CUB	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet [9]	4 CONV	43.56	55.31	–	–	–	–	–	–
MAML [11]	4 CONV	48.70	63.11	51.67	70.30	58.90	71.50	54.73	75.75
ProtoNet [8]	4 CONV	49.42	68.20	53.31	72.69	55.50	72.00	50.46	76.39
LEO [13]	WRN-28-10	61.76	77.59	66.33	81.44	–	–	–	–
CAN [51]	ResNet-12	63.85	79.44	69.89	84.23	–	–	–	–
DeepEMD [10]	ResNet-12	65.91	82.41	71.16	86.03	74.58	86.92	75.65	88.69
FEAT [52]	ResNet-12	66.78	82.05	70.80	84.79	–	–	73.27	85.77
RENet [53]	ResNet-12	67.60	82.58	71.61	85.28	74.51	86.60	82.85	91.32
FRN [54]	ResNet-12	66.45	82.83	72.06	86.89	–	–	83.55	92.92
COSOC [55]	ResNet-12	69.28	85.16	73.57	87.57	–	–	–	–
SetFeat [56]	ResNet-12	68.32	82.71	73.63	87.59	–	–	79.60	90.48
MCL [57]	ResNet-12	69.31	85.11	73.62	86.29	–	–	85.63	93.18
STL DeepBDC [58]	ResNet-12	67.83	85.45	73.82	89.00	–	–	84.01	**94.02**
DC [59]	ResNet-12	68.57	82.88	78.19	89.90	–	–	79.56	90.67
TPN [18]	4 CONV	52.78	66.42	55.74	71.01	–	–	–	–
TransMatch [33]	WRN-28-10	60.02	79.30	72.19	82.12	–	–	–	–
LST [19]	ResNet-12	70.01	78.70	77.70	85.20	–	–	–	–
EPNet [20]	ResNet-12	70.50	80.20	75.90	82.11	–	–	–	–
ICI [21]	ResNet-12	69.66	80.11	84.01	89.00	76.51	84.32	89.58	92.48
iLPC [23]	ResNet-12	70.99	81.06	**85.04**	89.63	78.57	85.84	90.11	–
PLCM [24]	ResNet-12	71.58	83.44	83.05	89.55	77.48	85.56	–	–
MUSIC [25]	ResNet-12	**74.96**	**85.99**	**85.40**	**90.79**	78.96	**87.25**	90.76	93.27
TENET (ours)	ResNet-12	74.02	83.69	82.83	89.32	**80.11**	86.67	**91.74**	**94.11**
TENET* (ours)	ResNet-12	**74.58**	**84.19**	83.71	**90.14**	**80.62**	**86.91**	**91.74**	93.95

we follow previous SSFSL works [24, 25] to perform TENET with a varied number of unlabeled samples in the inductive semi-supervised few-shot setup. As shown in Fig. 3, TENET can always beat all baselines in the 1-shot setting, and obtain comparable results with PLCM [24] in the 5-shot setting, still surpassing all other baselines. We skip the dubious MUSIC [25] baseline due to its inconsistent performances reported by its original paper (in 1-shot, approximately 73% accuracy from its Fig. 2 but 74.96% from its Table 1).

5.3 Ablation study

Here we conduct ablation studies on the trade-off parameter λ in (5), the scaling factor s in sample selection (7), and the SSFSL iteration number T.

Recalling the objective function (5), a large λ means that more attention will be given to correct classification than to unsupervised feature reconstruction. On the miniImageNet and CUB datasets, we can see from Table 5 that λ being 10 or 20 leads to better semi-supervised 1-shot

Table 2: Comparisons of 5-way few-shot classification with the transductive few-shot setup. The best and second best performances are highlighted in **red** and **blue**, respectively. *: using the average of feature vectors under multiple times of data augmentations.

Method	Backbone	miniImageNet		tieredImageNet		CIFAR-FS		CUB	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TPN [18]	4 CONV	55.51	69.86	59.91	73.30	–	–	–	–
EPNet [20]	ResNet-12	66.50	81.06	76.53	87.32	–	–	–	–
ICI [21]	ResNet-12	66.80	79.26	80.79	87.92	73.97	84.13	88.06	92.53
iLPC [23]	ResNet-12	69.79	79.82	83.49	89.48	77.14	85.23	89.00	92.74
PLCM [24]	ResNet-12	70.92	82.74	82.61	89.47	–	–	–	–
EASE [60]	ResNet-12	70.47	80.73	**84.54**	**89.63**	**78.41**	85.67	90.11	93.13
MUSIC [25]	ResNet-12	**72.01**	**83.49**	**83.57**	**89.81**	77.56	85.49	89.40	92.91
TENET (ours)	ResNet-12	71.03	82.39	81.13	88.69	**78.15**	**86.16**	**90.37**	**94.14**
TENET* (ours)	ResNet-12	**71.58**	**82.82**	82.40	89.22	77.83	**85.72**	**90.76**	**94.15**

Table 3: Semi-supervised and transductive few-shot classification, in comparison with ICI [21]. †: our reproduction with official implementation on our datasets and pretrained representative function.

Setting	Method	Backbone	miniImageNet		tieredImageNet	
			1-shot	5-shot	1-shot	5-shot
Semi-supervised	ICI [21]	ResNet-12	69.66	80.11	**84.01**	89.00
	ICI† [21]	ResNet-12	73.27	82.56	82.28	88.80
	TENET (ours)	ResNet-12	**74.02**	**83.69**	82.83	**89.32**
Transductive	ICI [21]	ResNet-12	66.80	79.26	80.79	87.92
	ICI† [21]	ResNet-12	**71.23**	82.28	80.14	88.17
	TENET (ours)	ResNet-12	71.03	**82.39**	**81.13**	**88.69**

performances, while λ being 1 leads to worse performances. This reveals that category labels still provide more useful information for SSFSL than the unsupervised reconstruction. However, a λ that is too large (being 100 or 1000) will impair the effectiveness of TENET, leading to a degenerated performance. We set λ as 20 as default.

The scaling factor s affects the order of the vector elements obtained from the max operation in Equation (7), influences the sample selection process, and further impacts the final performances. Table 6 shows that the effect of s on semi-supervised 1-shot performances is insignificant. We set s as 10 as default.

Fig. 4 illustrates the semi-supervised 1-shot performances of TENET with different iteration numbers T. Increasing T from 1 to 5 can bring significant improvements, while increasing T from 5 to 15 leads to little-changed results. We note that T is usually set to 5 in previous SSFSL works [21–25].

5.4 Empirical comparison

Pseudo label accuracy. Now we make an empirical comparison of THE pseudo label accuracy and test set accuracy along SSFSL iterations between LR and TENET. We set the same pseudo label accuracy at the 1st iteration for both, while TENET can still achieve higher test accuracy than LR. The experimental result in Fig. 5 indicates that TENET is more robust against inaccurate

Table 4: Transductive few-shot classification based on WRN-28-10, in comparison with LR+ICI [21], PT+MAP [61], EASE [60], and iLPC [23].

Method	Backbone	miniImageNet		tieredImageNet		CIFAR-FS		CUB	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
LR+ICI [21]	WRN-28-10	80.61	87.93	86.79	91.73	84.88	89.75	90.18	93.35
PT+MAP [61]	WRN-28-10	82.88	88.78	88.15	92.32	86.91	90.50	91.37	93.93
EASE [60]	WRN-28-10	83.00	88.92	88.96	92.63	87.60	90.60	91.68	94.12
iLPC [23]	WRN-28-10	83.05	88.82	88.50	92.46	86.51	90.60	91.03	94.11
TENET (ours)	WRN-28-10	82.93	88.71	88.48	92.26	86.86	90.54	91.30	93.87

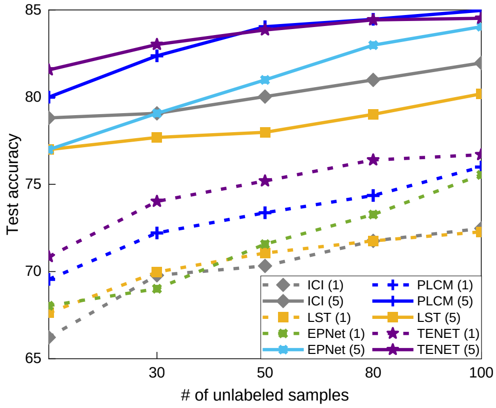


Fig. 3: Comparison results of varied unlabeled samples on miniImageNet. The number in the brackets denotes K-shot.

pseudo labels than LR, because the objective function of TENET is unsupervised. In addition, the performance gap between LR and TENET becomes larger as the iteration proceeds, indicating that TENET can generate more accurate pseudo labels.

Extend to logistic regression.

In the above sections, we take the linear regression model as an example to analyse the non-robustness and overfitting issues of generalized linear models. Similar to linear regression, the logistic regression model also suffers from the overfitting issue when the number of variables is greater than the number of data points, known as the $p > n$ problem. Through experimental validation, we show that the reconstruction-based regularization in TENET can also be applied to the logistic regression model. Over 600 episodes on miniImageNet and CUB, we compare the semi-supervised 1-shot performances among vanilla logistic regression, ℓ_2-regularized logistic regression, and reconstruction-regularized logistic regression, with the trade-off parameter λ being 1E-4 and the SSFSL iteration number T being 1.

Table 5: Ablation study on the trade-off parameter λ. Semi-supervised 1-shot classification, with iteration number T being 1.

λ	miniImageNet	CUB
1	65.31	83.41
5	67.56	85.89
10	**67.83**	86.48
20	67.65	**86.58**
100	66.04	85.86
1000	62.80	83.48

Table 6: Ablation study on the scaling factor s. Semi-supervised 1-shot classification, with iteration number T being 1.

s	miniImageNet	CUB
0.1	67.38	86.29
0.5	67.40	86.29
1	67.42	86.29
5	67.55	86.35
10	**67.58**	86.37
50	67.57	**86.40**

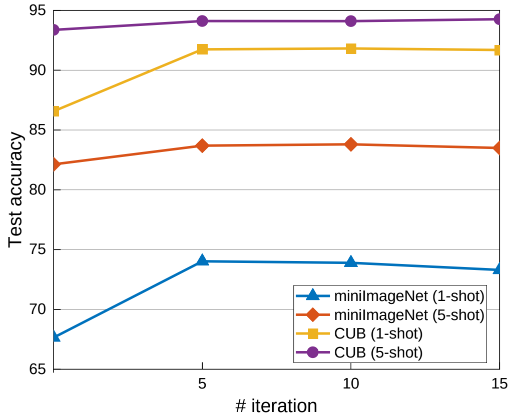


Fig. 4: Ablation study on SSFSL iteration number T. Semi-supervised 1-shot classification.

Table 7: Comparison of semi-supervised 1-shot performances among logistic regression models with different regularizations. The iteration number T is 1.

Regularization	miniImageNet	CUB
None	66.38	84.96
ℓ_2-norm	65.16	84.13
reconstruction	**66.91**	**85.92**

As Table 7 shows, the reconstruction-based regularization still performs best, which validates the effectiveness and generality of TENET.

Orthogonality of learned weights. As we state in Sec 4 that the feature reconstruction in TENET implies an orthogonal regularization to the classifier weights $\boldsymbol{w}$, here we check the orthogonality of $\boldsymbol{w}$ through experiments. Over 600 episodes of semi-supervised 1-shot learning on miniImageNet, we record all the weights $\boldsymbol{w}$ optimized by linear regression and TENET, respectively, and then compute the average $\boldsymbol{w}^\top \boldsymbol{w}$ results. As shown in Fig. 6b, TENET with the trade-off parameter λ in objective function (5) being 1 (equal consideration to correct classification and feature reconstruction) leads to the resulting matrix closest to an identity matrix. If further raising λ up to 20 (less consideration to feature reconstruction, and it is the default setting), the resulting matrix is still closer to an identity one than that of linear regression (see Fig. 6c *vs* Fig. 6a). Such empirical results demonstrate the achievability of orthogonal regularization in TENET.

Comparison with orthogonality regularization. To find out if the effectiveness of TENET arises from the orthogonal regularization or the reconstruction regularization, we conduct comparative experiments that solely use one of these two types of regularization. Formally, the objective function of linear regression with orthogonal regularization is formulated as

Table 8: Comparison of semi-supervised 1-shot between orthogonal regularization and reconstruction regularization. The iteration number T is 1.

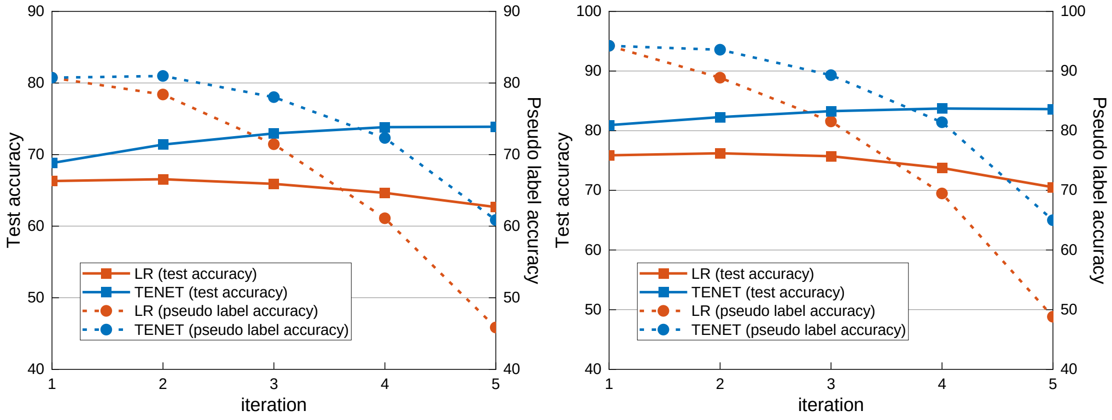
Regularization	miniImageNet	CUB
None	59.78	79.60
Orthogonal	64.75	82.98
Reconstruction	**66.96**	**84.82**

$$\min \ \lambda \cdot \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_F^2 + \left\|\boldsymbol{w}^\top \boldsymbol{w} - \boldsymbol{I}\right\|_F^2, \qquad (8)$$

and that of reconstruction regularization is consistent with Equation (4). For a fair comparison, we now utilize the SGD optimizer on Pytorch framework to minimize both Equations (8) and (4), and keep all training settings the same, such as a learning rate of 0.01, trade-off parameter $\lambda = 20$, and SSFSL iteration number $T = 1$. The average test set accuracy over 600 episodes on the miniImageNet and CUB datasets are shown in Table 8. As can be seen, the effectiveness of TENET arises more from the reconstruction regularization.

6 Conclusion

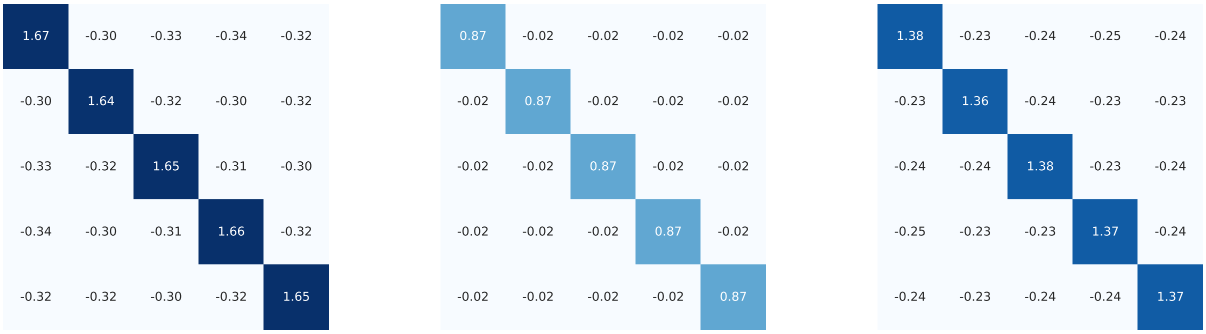
In this paper, we first analyse the flaws of the common objective function in previous SSFSL approaches, which are non-robustness and overfitting. The essence of these flaws is the lack of a reliable supervision signal. Motivated by this, we propose to capture the distribution information from both labeled and unlabeled training sets as unsupervised regularization to benefit the classifier training. We change the objective function of linear regression into low-rank feature reconstruction, and term this new approach as TENET.



(a) 1-shot

(b) 5-shot

Fig. 5: Comparison of pseudo label accuracy and test set accuracy during SSFSL iterations. The dataset is miniImageNet.



1.67	-0.30	-0.33	-0.34	-0.32
-0.30	1.64	-0.32	-0.30	-0.32
-0.33	-0.32	1.65	-0.31	-0.30
-0.34	-0.30	-0.31	1.66	-0.32
-0.32	-0.32	-0.30	-0.32	1.65

(a) Linear regression

0.87	-0.02	-0.02	-0.02	-0.02
-0.02	0.87	-0.02	-0.02	-0.02
-0.02	-0.02	0.87	-0.02	-0.02
-0.02	-0.02	-0.02	0.87	-0.02
-0.02	-0.02	-0.02	-0.02	0.87

(b) TENET ($\lambda = 1$)

1.38	-0.23	-0.24	-0.25	-0.24
-0.23	1.36	-0.24	-0.23	-0.23
-0.24	-0.24	1.38	-0.23	-0.24
-0.25	-0.23	-0.23	1.37	-0.24
-0.24	-0.23	-0.24	-0.24	1.37

(c) TENET ($\lambda = 20$, default)

Fig. 6: Comparison of the $\boldsymbol{w}^\top \boldsymbol{w}$ results. TENET leads to the resulting matrix being closer to an identity matrix, indicating better orthogonality than linear regression.

Experiments on four commonly used datasets validate the effectiveness of TENET. We believe that our proposed method can inspire the field of SSFSL.

Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation under Grant L221013, and in part by the National Science Foundation of China under Grant U20B2070 and Grant 61832016.

References

[1] W. Wu, H. Peng, and S. Yu. Yunet: A tiny millisecond-level face detector. *Machine Intelligence Research (MIR)*, pages 1–10, 2023.

[2] L. Wang, H. Xu, and W. Kang. Mvcontrast: Unsupervised pretraining for multi-view 3d object recognition. *Machine Intelligence Research (MIR)*, pages 1–12, 2023.

[3] R. Jiang, R. Zhu, H. Su, Y. Li, Y. Xie, and W. Zou. Deep learning-based moving object segmentation: Recent progress and research

prospects. *Machine Intelligence Research (MIR)*, 20(3):1–35, 2023.

[4] F.-Q. Liu and Z.-Y. Wang. Automatic "ground truth" annotation and industrial workpiece dataset generation for deep learning. *International Journal of Automation and Computing (IJAC)*, 17:539–550, 2020.

[5] D.-Y. She and K. Xu. Contrastive self-supervised representation learning using synthetic data. *International Journal of Automation and Computing (IJAC)*, 18(4):556–567, 2021.

[6] C. Yang, C. Liu, and X.-C. Yin. Weakly correlated knowledge integration for few-shot image classification. *Machine Intelligence Research (MIR)*, 19(1):24–37, 2022.

[7] M. Han, Y. Zhan, B. Yu, Y. Luo, H. Hu, B. Du, Y. Wen, and D. Tao. Region-adaptive concept aggregation for few-shot visual recognition. *Machine Intelligence Research (MIR)*, pages 1–15, 2023.

[8] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Neural Information Proceeding Systems (NeurIPS)*, pages 4080–4090, 2017.

[9] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Neural Information Proceeding Systems (NeurIPS)*, pages 3637–3645, 2016.

[10] C. Zhang, Y. Cai, G. Lin, and C. Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12203–12213, 2020.

[11] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135. PMLR, 2017.

[12] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[13] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

[14] H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5822–5830, 2018.

[15] 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 *European Conference on Computer Vision (ECCV)*, pages 266–282. Springer, 2020.

[16] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks (TNN)*, 20(3):542–542, 2009.

[17] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2018.

[18] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] X. Li, Q. Sun, Y. Liu, S. Zheng, Q. Zhou, T.-S. Chua, and B. Schiele. Learning to self-train for semi-supervised few-shot classification. In *Neural Information Proceeding Systems (NeurIPS)*, pages 10276–10286, 2019.

[20] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *European Conference on Computer Vision (ECCV)*, pages 121–138. Springer, 2020.

[21] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu. Instance credibility inference for few-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12836–12845, 2020.

[22] Y. Wang, L. Zhang, Y. Yao, and Y. Fu. How to trust unlabeled data? instance credibility inference for few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(10):6240–6253, 2022.

[23] M. Lazarou, T. Stathaki, and Y. Avrithis. Iterative label cleaning for transductive and semi-supervised few-shot learning. In *International Conference on Computer Vision (ICCV)*, pages 8751–8760, 2021.

[24] K. Huang, J. Geng, W. Jiang, X. Deng, and Z. Xu. Pseudo-loss confidence metric for semi-supervised few-shot learning. In *International Conference on Computer Vision (ICCV)*, pages 8671–8680, 2021.

[25] X.-S. Wei, H.-Y. Xu, F. Zhang, Y. Peng, and W. Zhou. An embarrassingly simple approach to semi-supervised few-shot learning. In *Neural Information Proceeding Systems (NeurIPS)*, volume 35, pages 14489–14500, 2022.

[26] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics (AISTATS)*, pages 57–64. PMLR, 2005.

[27] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Neural Information Proceeding Systems (NeurIPS)*, pages 529–536, 2004.

[28] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Neural Information Proceeding Systems (NeurIPS)*, 33:596–608, 2020.

[29] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

[30] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2020.

[31] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, pages 6438–6447. PMLR, 2019.

[32] V. Verma, M. Qu, K. Kawaguchi, A. Lamb, Y. Bengio, J. Kannala, and J. Tang. Graphmix: Improved training of gnns for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 10024–10032, 2021.

[33] Z. Yu, L. Chen, Z. Cheng, and J. Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12856–12864, 2020.

[34] D. Berthelot, N. Carlini, I. Goodfellow, A. Oliver, N. Papernot, and C. Raffel. Mixmatch: a holistic approach to semi-supervised learning. In *Neural Information Proceeding Systems (NeurIPS)*, pages 5049–5059, 2019.

[35] J. Huang, L. Qu, R. Jia, and B. Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *International Conference on Computer Vision (ICCV)*, pages 3326–3334, 2019.

[36] J. Chen, V. Shah, and A. Kyrillidis. Negative sampling in semi-supervised learning. In *International Conference on Machine Learning (ICML)*, pages 1704–1714. PMLR, 2020.

[37] J.-C. Su, S. Maji, and B. Hariharan. When does self-supervision improve few-shot learning? In *European Conference on Computer Vision (ECCV)*, pages 645–666. Springer, 2020.

[38] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[39] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[40] L. Huang, X. Liu, B. Lang, A. Yu, Y. Wang, and B. Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, pages 3271–3278, 2018.

[41] S. Li, K. Jia, Y. Wen, T. Liu, and D. Tao. Orthogonal deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(4):1352–1368, 2021.

[42] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3174–3183, 2017.

[43] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax+xb=c. *Communications of the ACM*, 15(9):820–826, 1972.

[44] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019.

[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. *Technical report*, 2011.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[47] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue. Self-supervised learning for few-shot image classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1745–1749. IEEE, 2021.

[48] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. *CIFAR-100*, 2009.

[49] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[50] N. Hilliard, L. Phillips, S. Howland, A. Yankov, C. D. Corley, and N. O. Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018.

[51] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen. Cross attention network for few-shot classification. In *Neural Information Proceeding Systems (NeurIPS)*, pages 4003–4014, 2019.

[52] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8808–8817, 2020.

[53] D. Kang, H. Kwon, J. Min, and M. Cho. Relational embedding for few-shot classification. In *International Conference on Computer Vision (ICCV)*, pages 8822–8833, 2021.

[54] D. Wertheimer, L. Tang, and B. Hariharan. Few-shot classification with feature

map reconstruction networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8012–8021, 2021.

[55] X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, and Q. Tian. Rectifying the shortcut learning of background for few-shot learning. In *Neural Information Proceeding Systems (NeurIPS)*, volume 34, pages 13073–13085, 2021.

[56] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, and C. Gagné. Matching feature sets for few-shot image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9014–9024, 2022.

[57] Y. Liu, W. Zhang, C. Xiang, T. Zheng, D. Cai, and X. He. Learning to affiliate: Mutual centralized learning for few-shot classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 14411–14420, 2022.

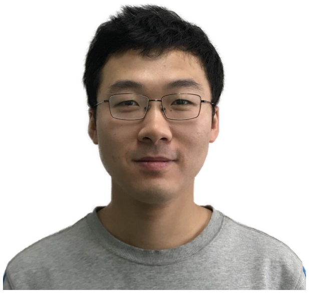
[58] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7972–7981, 2022.

[59] S. Yang, L. Liu, and M. Xu. Free lunch for few-shot learning: distribution calibration. In *International Conference on Learning Representations (ICLR)*, 2020.

[60] H. Zhu and P. Koniusz. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9078, 2022.

[61] Y. Hu, V. Gripon, and S. Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks (ICANN)*, pages 487–499. Springer, 2021.

[62] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.



Chengcheng Ma received the B.Sc. degrees in automation from Northwestern Polytechnical University, China in 2017. Currently, he is a Ph.D. candidate in the National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests include few-shot learning, semi-supervised learning and trustworthy machine learning.

E-mail: machengcheng2017@ia.ac.cn

ORCID iD: 0000-0002-0502-3960

Weiming Dong is a Professor in the National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). He received his BSc and MSc degrees in 2001 and 2004, both from Tsinghua University, China. He received his PhD in Computer Science from the University of Lorraine, France, in 2007. His research interests include image synthesis and image recognition. Weiming Dong is a member of the ACM and IEEE.

E-mail: weiming.dong@ia.ac.cn (Corresponding author)

ORCID iD: 0000-0001-6502-145X

Changsheng Xu is a Professor in National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition and computer vision. He has hold 30 granted/pending patents and published over 200 refereed research papers in these areas. Dr. Xu is an Associate Editor of ACM Trans. on Multimedia Computing, Communications and Applications and ACM/Springer Multimedia Systems Journal. He received the Best Associate Editor Award of ACM Trans. on Multimedia Computing, Communications and Applications in 2012 and the Best Editorial Member Award of ACM/Springer Multimedia Systems Journal in 2008. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops. He is IEEE Fellow, IAPR Fellow and ACM Distinguished Scientist.

E-mail: csxu@nlpr.ia.ac.cn

ORCID iD: 0000-0001-8343-9665