Self-supervised Network Evolution for Few-shot Classification

Xuwen Tang[1] , **Zhu Teng**[1] , **Baopeng Zhang**[1*] , **Jianping Fan**[2]
[1]Beijing Jiaotong University
[2]Lenovo Research
{19120402, zteng, bpzhang}@bjtu.edu.cn, jfan1@lenovo.com

Abstract

Few-shot classification aims to recognize new classes by learning reliable models from very few available samples. It could be very challenging when there is no intersection between the already-known classes (base set) and the novel set (new classes). To alleviate this problem, we propose to evolve the network (for the base set) via label propagation and self-supervision to shrink the distribution difference between the base set and the novel set. Our network evolution approach transfers the latent distribution from the already-known classes to the unknown (novel) classes by: (a) label propagation of the novel/new classes (novel set); and (b) design of dual-task to exploit a discriminative representation to effectively diminish the overfitting on the base set and enhance the generalization ability on the novel set. We conduct comprehensive experiments to examine our network evolution approach against numerous state-of-the-art ones, especially in a higher way setup and cross-dataset scenarios. Notably, our approach outperforms the second best state-of-the-art method by a large margin of 3.25% for one-shot evaluation over miniImageNet.

1 Introduction

By learning from large-scale labeled samples, deep learning methods have upgraded performances of many computer vision tasks such as classification, detection, etc. Unfortunately, it is hard to acquire and manually-annotate mass samples. In contrast, humans can learn from very limited labeled samples and recognize new classes accurately. For instance, children can recognize a horse by learning from only few pictures in a book. Many researchers have tried to enable AI models to learn from few samples and one major research area is few-shot learning: the model, which is pre-trained on large-scale samples for already-known classes, is further extended to classify new classes with only few labeled examples.

To enable few-shot learning, some existing methods adopt the meta-learning framework to reduce the gap between the training samples and the test samples. Metric-based methods pay too much attention to the type of embedding space and overlook how to extract more transferable and discriminative representation. On the other hand, transfer based methods learn a good embedding on entire base set, but most of these methods assume that the base set and the novel set share the same embedding space, which is obviously not valid. They learn the embedding on the base set whose already-known classes are quite different from new classes in the novel set, the gap between the base set and the novel set makes such embedding not being generalized to the novel set.

To overcome the gap between the base set and the novel set, some regularization techniques emerge such as mixup and manifold mixup. They enhance the generalization through the mixed images in a batch or feature mixture in the convolutional layer, which smooths the feature space and decision boundaries. For example, EPNet proposes a simple embedding propagation to regularize the feature representation. But none of these methods considers the distribution difference between the already-known (base) classes and the novel classes. In summary, there are two issues for the existing methods: (1) they assume the base set and the novel set share the same embedding space. (2) existing regularization methods in few-shot learning have not yet made full use of the information provided by the unlabeled data in the novel set.

Based on these observations, we propose to evolve the network via label propagation and self-supervision to shrink the distribution difference. Self-supervised Network Evolution involves the images for the novel classes to generate a domain-specific network from the base network. A deep clustering method is employed to propagate the labels of the novel classes to further learn latent distribution from the known classes to the unknown classes. Because a progressive clustering algorithm is adopted, the incorrect pseudo labels are inevitably generated. To alleviate the negative effects on label propagation while the network evolves, self-supervised learning is designed to combine with the supervised learning in the Network Evolution to force the model to learn richer semantic information of the sample itself. Note that manual annotations of the images for the novel classes are not required in our model.

Our main contributions are summarized as follows:

(1) A Self-supervised Network Evolution (SNE) model is developed to deal with the distribution difference between the already-known (base) classes and the novel/new classes.

*Contact Author

(2) A dual-task is designed to combine a self-supervised task and a supervised task to exploit a discriminative representation.

(3) Extensive experiments are conducted on miniImageNet, CIFAR-FS, and FC-100 to verify the performance of our proposed method. In particular, our method can achieve superior performance on a higher way setup and the cross-dataset scenario evaluation.

2 Related Work

In this section, we provide a review for two most relevant researches: few-shot learning and self-supervised learning.

Few-shot Learning: Few-shot learning approaches can be roughly categorized into three divisions: meta-learning based methods, metric-learning based methods, and transfer-learning based methods.

Meta-learning-based methods aim to learn a set of commonly-shared parameters, so that the model can adapt to the new tasks in few steps. The most classic method is MAML [Finn *et al.*, 2017], which learns a set of the initialization parameters to adapt to a new task in very few gradient steps. However, this kind of method usually needs to compute the costly higher-order gradients. To reduce the computation load, LEO [Rusu *et al.*, 2019] uses an encoder and relation network to project the sample into a low-dimensional space and utilizes a decoder to transfer to high-dimension parameters. In our work, we employ a conventional classification setting to avoid massive computation.

Metric-learning-based methods aim to learn a metric space. For instance, MatchingNet [Vinyals *et al.*, 2016] is the first deep metric method to enable few-shot classification. It predicts the similarity between the support and query embedding by cosine distance space. ProtoNet [Snell *et al.*, 2017] computes the average of the support set as prototypes to predict similarity in the Euclidean distance space, while RelationNet [Sung *et al.*, 2018] creates a learnable distance space by CNN. [Bateni *et al.*, 2020] and [Zhang *et al.*, 2020] propose to use the Mahalanobis distance and Earth Mover distance in the few-shot task. Metric-based methods focus more on the choice of the metric space but ignore the feature embedding. We propose to learn a good feature embedding and address the distribution difference between the base and novel set.

The key difference between transfer-based methods and other methods is the setting in the training stage. The methods using the meta-learning framework in the training stage mimic the test set to reduce the gap between training and test sets. In contrast, the transfer-based methods [Chen *et al.*, 2019] generally train a feature extractor under the conventional classification setting on the base set, and then finetune a cosine classifier. RFS [Tian *et al.*, 2020] learns a logistic regression classifier instead of a cosine classifier and obtains competitive performance compared with the meta-learning based methods. Different from these methods, our SNE model proposes label propagation and network evolution to learn more generalization embeddings, which reduces the distribution difference between the base and novel set.

Self-supervised Learning: Self-supervised learning is used in many applications, which mainly uses pretext tasks to mine its own supervised information from large-scale unsupervised data. In computer vision, most works focus on the context information to construct a pre-text task. For example, [Doersch *et al.*, 2015] splits an image into nine pieces and then predicts the relative position to learn the semantic information. [Noroozi and Favaro, 2016] further extends this method to predict the permutation of the nine patches, which makes the pretext task more difficult and learns more positive information. Similar to the context prediction, [Pathak *et al.*, 2016] erases a part of the image and lets the model reconstruct the whole image. [Zhang *et al.*, 2016] leverages the color information by predicting the color of the image given the gray-scale image. [Gidaris *et al.*, 2018] constructs the pretext task by predicting the angle of the image provided with the rotated version of the original image before they are input to the feature extractor. [Gidaris *et al.*, 2019] employs the self-supervised technique in the training process on the base set to enhance the representation ability. In contrast, we adopt the rotation self-supervision to alleviate the incorrect label propagation in our network evolution process.

3 Our Proposed Method

In this section, we elaborate the proposed Self-supervised Network Evolution (SNE) model for the few-shot classification task. The task setup is described in Section 3.1 and the SNE model is described in Section 3.2.

3.1 Few-shot Classification Setup

The few-shot classification dataset is divided into three parts: *base* set (D_b), *validation* set (D_v), and *novel* set (D_n), where categories from these three sets are distinct (e.g., a category in the *base* set cannot be found in the *novel* set). The base set consists of a large number of labeled images $D_b = \{(x_i, y_i), i = 1, 2, \cdots, m_b\}$ where $y_i \in y_{base}$. The novel set is composed by relatively small amount of labeled data $D_n = \{(x_j, y_j), j = 1, 2, \cdots, m_n\}$ where $y_j \in y_{novel}$. Notice that $y_{base} \cap y_{novel} = \emptyset$. The validation set D_v consists of the classes different from both D_b and D_n, and is employed to determine the hyperparameters. For the episode setting, we follow the N-way K-shot task. Each episode consists of n classes randomly selected from the dataset, a labeled support set (S) containing k images per class, and an unlabeled query set (Q) including q images per class.

3.2 Self-supervised Network Evolution

Our proposed SNE model is evolved through three stages by adding various ingredients. First, a base network is learned from base classes, where a base embedding space is constructed. Secondly, to learn latent distributions evolved from known classes to unknown classes, deep clustering is employed to propagate pseudo labels of novel classes. Thirdly, the network is evolved by a designed dual-task, which consists of a self-supervised task and a supervised task constrained by pseudo labels of novel classes. The entire architecture is described in Figure 1.

Embedding Space: The embedding space is built by learning a base network N_{base} through a linear classifier C_B on

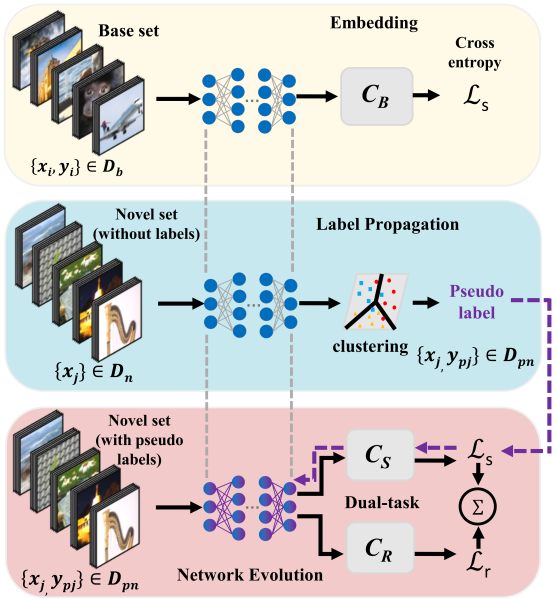

Figure 1: The architecture of the proposed Self-supervised Network Evolution model.

the base set $D_b = \{(x_i, y_i), i = 1, 2, \cdots, m_b\}$. The classifier is trained to predict the label of the images in the base set and is formulated by minimizing the standard cross-entropy objective as shown in Eq. 1 where z_{x_i} denotes for the embedding of the input image x_i. $p(y_i|z_x, C_B)$ represents the class probability and is acquired by appending a softmax layer on the output of the linear classifier.

$$\mathcal{L}_s(x_i, y_i; C_B, N_{base}) = -\ln p(y_i|z_{x_i}, C_B) \tag{1}$$

Label Propagation: In the few-shot setting, the classes from the base set are never mingled with the classes of the novel set. On this issue, currently many few-shot methods directly assume that the base set and the novel set share the embedding network to extract features, which is obviously not valid. Transfer learning-based methods can employ the data and corresponded labels of the target domain to fine-tune the network, but the utilization of labels from the target domain (novel set) shatters the strict few-shot setting. Some Unsupervised Domain Adaptation (UDA) methods align the source and target domain by matching the data distribution using MMD (Maximum Mean Discrepancy), and others fix the classifier and fine-tune the feature extractor to adapt the target domain. These UDA methods require the source and target set have common categories, which is not suitable for the few-shot problem. To tackle this issue, we propose to evolve the network from known categories to unknown categories by learning latent distributions. Specifically, we first utilize the base embedding network N_{base} to extract features $\mathcal{F}_n$ from the penultimate layer on the image x_j of the novel set $D_n = \{(x_j, y_j), j = 1, 2, \cdots, m_n\}$. Then, all the features are clustered into groups and a pseudo label is propagated to each group. These pseudo labels construct a pseudo novel set named $D_{pn} = \{(x_j, y_{pj}), j = 1, 2, \cdots, m_n\}$. In the process of clustering, we employ SCAN [Gansbeke *et al.*, 2020] to decouple the feature learning and clustering. Features from the embedding space are utilized to find C nearest neighbors of each image. Then, we apply a loss function (Eq. 2) to maximize the dot product between each image and its mined neighbors so that images can be automatically grouped into semantically meaningful clusters. Here, Φ_η is clustering function parameterized by a neural network with weights η. $\mathcal{N}_X$ stands for the neighbors of sample X. $\mathcal{K}$ is the clusters $\mathcal{K} = \{1, ...K\}$.

$$\begin{aligned}\mathcal{L}_{scan} &= -\frac{1}{|D_n|}\sum_{X\in D_n}\sum_{n\in\mathcal{N}_X} log\langle\Phi_\eta(X), \Phi_\eta(n)\rangle \\ &+ \lambda\sum_{k\in\mathcal{K}}\Phi'^k_\eta log\Phi'^k_\eta, \quad \Phi'^k_\eta = \frac{1}{|D_n|}\sum_{X\in D_n}\Phi^k_\eta(X)\end{aligned} \tag{2}$$

Network Evolution: In the third stage, we first learn a Convolutional Network N_{novel_S} with a single task, which minimizes the standard cross-entropy objective with the pseudo novel set D_{pn} as described in Eq. 3, where y_{p_j} is the pseudo label of the image in the novel set and $p(y_{p_j}|z_{x_j}, C_S)$ is the probability that the input image is predicted as y_{p_j}. Compared with N_{base}, the network N_{novel_S} evolves to adapt to the novel set. However, there might exist a mass of incorrect label propagation, which may confuse the network and finally damage the performance.

$$\mathcal{L}_s(x_j, y_{p_j}; C_S, N_{novel_S}) = -\ln p(y_{p_j}|z_{x_j}, C_S) \tag{3}$$

To restrain the inaccurate label propagation, we design a dual-task by collaborating a supervised task and a self-supervised task simultaneously. By orienting input images, the self-supervised task is defined as the prediction of oriented angles on these images. This enforces the network to learn more semantic information and focuses on high-level embedding. In our work, we construct four oriented angles denoted by $r \in R = \{0°, 90°, 180°, 270°\}$. On each input image, a rotation with angles r in R is operated. We represent the oriented image by x^r_j and the corresponding label is y^r. A rotation classifier C_R predicts the angle of an image, which is formulated by $\hat{y} = C_R \circ \hat{z}$, where $(a \circ b)$ indicates b is input into a, $\hat{y}$ is the predicted angle, and $\hat{z}$ is the feature of input image extracted by the network N_{novel_S}. Further, we define the objective of the self-supervision task in Eq. 4, where $p(y^r_j|z_{x^r_j}, C_R)$ is the probability that the input image x^r_j is predicted to be oriented with an angle of r by C_R.

$$\mathcal{L}_r(x^r_j, y^r_j; C_R, N_{novel_D}) = -\ln p(y^r_j|z_{x^r_j}, C_R) \tag{4}$$

In the network evolution, we first cluster the features extracted from the novel set by the network N_{base} and propagate a pseudo label to each cluster to construct the pseudo novel set D_{pn}. With the same architecture to N_{base}, the network is further evolved to N_{novel_D} by training under the dual-task, which is encoded by linear layers named C_S and C_R, respectively. The first task aims to predict the label of samples on the pseudo novel set and the optimization objective is described in Eq. 3. The second task aims to predict the oriented angle of the input image and the optimization objective is presented in Eq. 4. Due to the joint learning of a supervised task

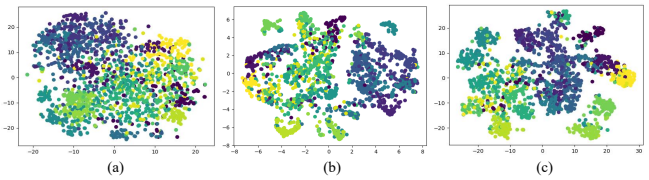

Figure 2: The T-SNE visualization of feature distribution executed on images from the novel set of miniImageNet. The feature embedding is extracted by (a) Base Network; (b) Novel-S Network trained with a single task; (c) Novel-D Network trained with the dual-task.

and a self-supervised task, multiple objectives are involved, including an image rotation classification and a standard image classification. Conventionally, linear weighting can be employed to balance multiple tasks, as formulated in Eq. 5.

$$\mathcal{L}_{all} = (1-w)\mathcal{L}_s + w\mathcal{L}_r \tag{5}$$

But it is difficult to tune this weight because an optimal weighting of each task is impacted by many factors such as the measurement scale, the magnitude of the noise in each task, etc. To deal with this multi-task problem, we adopt an adaptive way (as described in Eq. 6) that considers the homoscedastic uncertainty of each task to combine multiple loss functions [Kendall *et al.*, 2018].

$$\mathcal{L}_{all} = \frac{1}{2\delta_1^2}\mathcal{L}_s + \frac{1}{2\delta_2^2}\mathcal{L}_r + \log\delta_1 + \log\delta_2 \tag{6}$$

where δ_1, $\delta_2 \in \mathbb{R}$ are parameters learnt through the back-propagation of the network in the training process. $\mathcal{L}_s$ is a standard classification task as formulated in Eq. 3 and $\mathcal{L}_r$ is a rotation classification task described in Eq. 4.

In addition, we utilize the T-SNE technique [Maaten and Hinton, 2008] to visualize the distribution of features (2-dim) extracted from images in the novel set during the network evolution. The results are reported in Figure 2. The base network N_{base} first learns the knowledge from seen classes in the base set (Figure 2 (a)) and it has the preliminary ability to generate vaguely discernible feature representation. However, it has a large inter-class variance, and the embedding of different classes is mostly mixed. As the network evolves to N_{novel_S}, the distribution of unseen classes is slightly separated into clusters (as shown in Figure 2 (b)) compared with the feature distribution performed by N_{base} in Figure 2 (a). When the network evolves to N_{novel_D}, the clusters are rather segregated, as visualized in Figure 2 (c), which demonstrates the representation ability of our SNE model.

4 Experiments

In this section, we first introduce our experimental setting including datasets, implementation details, and evaluation criteria. Extensive experiments are conducted on three widely used benchmarks for the few-shot classification task and comparisons with a number of state-of-the-art methods are reported in Section 4.2. We execute ablation studies in Section 4.3 where the contributions of components in the SNE model are analyzed. To further verify the robustness of our SNE model, we evaluate SNE with several other approaches in a higher way setup and cross-dataset scenarios in Section 4.4.

4.1 Experimental Settings

Datasets: Experiments are executed on three widely used datasets for few-shot classification: miniImageNet, CIFAR-FS, and FC100. The **miniImageNet** dataset is a subset of ImageNet, which contains 100 classes with 600 images per class randomly selected from the 1000 classes in ImageNet. The **CIFAR-FS** dataset is constructed from the standard CIFAR-100 dataset, which includes 100 classes with 600 images per class. Both miniImageNet and CIFAR-FS are randomly split into 3 parts: 64 base classes, 16 validation classes, and 20 novel classes. The **FC100** dataset is also built from the standard CIFAR-100 dataset with 100 classes with 600 images per class. Different from the above two datasets, the classes in FC100 are split based on the superclass. Base classes contain 12 superclasses (60 classes), validation classes incorporate 4 superclasses (20 classes), and novel classes comprise 4 superclasses (20 classes). Images in miniImageNet, CIFAR-FS, and FC100 are resized to 84x84, 32x32, 32x32.

Implementation Details: We use ResNet-12 as our backbone in all experiments. The ResNet-12 contains 4 Residual blocks, and each residual block consists of 3 convolutional layers with a 3x3 kernel followed by a Batchnorm2d layer and a ReLu layer. The first three residual blocks apply a 2x2 max-pooling layer, and the last residual block employs an adaptive pooling to ensure the adaptation of different input scales. The ResNet-12 finally outputs a 640-dimensional embedding. We adopt SGD optimizer with a momentum of 0.9 and a weight decay of $5e^{-4}$. We train 100 epochs for all the datasets, with a batch size of 128. The learning rate is set to 0.05 at first and is declined at the 60^{th} and 80^{th} epoch by a factor of 0.1. In the training process, the baseline method RFS needs 2.7 hours and our SNE requires 5.7 hours. In evaluation, one episode evaluation needs 0.1s for 1-shot and 0.2s for 5-shot. In all the experimental tables, we use the following denotations. †: the WRN-28-10 backbone. ♣: the Conv-32F backbone. ♠: the Conv-64F backbone. ♢: the Capsule Network backbone. Others: the ResNet12 backbone.

Episode Evaluation Criteria: We use the N-way K-shot episode evaluation setting. 5-way 1-shot and 5-way 5-shot are widely used for few-shot classification. Each episode randomly selected 5 classes in the novel set and sample 1/5 image(s) per class as the support set and Q images per class as the query set. For all the three datasets (miniImageNet, CIFAR-FS, and FC100), 1000 episodes with $Q = 15$ are executed and we repeat the experiments 10 times and record the accuracy by averaging these results.

4.2 Comparisons with State-of-the-arts

In this section, we compare our method with state-of-the-art approaches in the 5-way 1-shot task and the 5-way 5-shot task for few-shot classification.

Results on MiniImageNet: We compare multiple classic and state-of-the-art methods on the miniImageNet benchmark in Table 1. Our method achieves the best accuracy for 1-shot (71.02±0.08) and ranks number two for 5-shot (84.56±0.05

Methods	1-shot	5-shot
MAML♣ [Finn *et al.*, 2017]	48.70±1.84	63.11±0.92
ProtoNet♠ [Snell *et al.*, 2017]	49.42±0.78	68.20±0.66
TADAM [Oreshkin *et al.*, 2018]	58.50±0.3	76.60±0.3
RFS [Tian *et al.*, 2020]	62.02±0.63	79.64±0.44
MetaOptNet [Lee *et al.*, 2019]	62.64±0.61	78.63±0.46
S2M2† [Mangla *et al.*, 2020]	64.92±0.18	83.18±0.11
DeepEMD [Zhang *et al.*, 2020]	65.91±0.82	82.41±0.56
EPNet [Rodríguez *et al.*, 2020]	66.50±0.89	81.06±0.60
FEAT [Ye *et al.*, 2020]	66.78±0.20	82.05±0.14
ICI [Wang *et al.*, 2020]	66.8±n/a	79.26±n/a
DSN-MR [Simon *et al.*, 2020]	67.09±0.68	81.65±0.69
DPGN [Yang *et al.*, 2020]	67.77±0.32	84.60±0.43
SNE (Ours)	71.02±0.08	84.56±0.05

Table 1: Comparisons of average accuracies (%) with 95% confidence intervals against state-of-the-art methods for 1-shot and 5-shot classification on the **miniImageNet** benchmark.

Methods	1-shot	5-shot
ProtoNet♠ [Snell *et al.*, 2017]	55.5±0.7	72.0±0.6
MAML♣ [Finn *et al.*, 2017]	58.9±1.9	71.5±1.0
RFS [Tian *et al.*, 2020]	71.5±0.8	86.0±0.5
ProtoNet [Snell *et al.*, 2017]	72.2±0.7	83.5±0.5
MetaOptNet [Lee *et al.*, 2019]	72.6±0.7	84.3±0.5
J.Kim [Kim *et al.*, 2020]	73.51±0.92	85.65±0.65
ICI [Wang *et al.*, 2020]	73.97±n/a	84.13±n/a
S2M2† [Mangla *et al.*, 2020]	74.81±0.19	87.47±0.13
DSN-MR [Simon *et al.*, 2020]	75.6±0.9	86.2±0.6
Fine-tune† [Dhillon *et al.*, 2020]	76.58±0.68	85.79±0.50
DPGN [Yang *et al.*, 2020]	77.9±0.5	90.2±0.4
SNE (Ours)	79.53±0.05	88.56±0.05

Table 2: Comparisons of average accuracies (%) with 95% confidence intervals against state-of-the-art methods for 1-shot and 5-shot classification on the **CIFAR-FS** benchmark.

). Among all the comparative methods, the MAML and ProtoNet are the pioneering works in few-shot learning. RFS employs the same cross-entropy loss to train the feature extractor, which provides a baseline for ours. RFS obtains an accuracy of 62.02% for 1-shot, which is 9% behind ours. This proves the effectiveness of network evolution. The best performer for the 5-shot task is DPGN, and it outperforms our SNE model by a slight gain of 0.04% (84.60% VS. 84.56%). However, when focusing on the 1-shot task, ours outperforms DPGN by a large margin of 3.25% (71.02% VS. 67.77%). Compared to the metric-based methods TADAM and DeepEMD, our SNE model leads them by 12.52% and 2.25% for 1-shot, respectively. ICI is a semi-supervised method that employs the unseen query set to enhance the classifier. Both our SNE model and ICI employ unlabelled data, but ICI only achieves an accuracy of 66.8% for 1-shot, which is 4.42% fewer than ours. This demonstrates that our SNE model possesses a better ability to transfer latent distribution from seen classes to unseen classes.

Results on CIFAR-100 Derivatives: We perform experiments on two CIFAR-100 derivatives, including CIFAR-FS and FC100. Table 2 and Table 3 reflect the results of CIFAR-FS and FC100, respectively. On CIFAR-FS, ours ranks number one for the 1-shot task and obtains a slightly lower performance behind DPGN on the 5-shot task. Compared with the baseline method RFS, ours enhances the 1-shot task by a gain of +8.03% and a margin of +2.56% on the 5-shot task. Our method attempts to learn a good embedding of images rather than a prototype of a set of images. For FC100, our method achieves a new state-of-the-art performance both on 1-shot and 5-shot tasks, which outperforms the second-best SOTA method by a large margin of 3.01% on the 1-shot evaluation.

Methods	1-shot	5-shot
ProtoNet♠ [Snell *et al.*, 2017]	35.3±0.6	48.6±0.6
MAML♣ [Finn *et al.*, 2017]	38.1±1.7	50.4±1.0
TADAM [Oreshkin *et al.*, 2018]	40.1±0.4	56.1±0.4
MetaOptNet [Lee *et al.*, 2019]	41.1±0.6	55.5±0.6
J.Kim et al [Kim *et al.*, 2020]	42.31±0.73	58.16±0.78
RFS [Tian *et al.*, 2020]	42.6±0.7	59.1±0.6
E^3BM [Liu *et al.*, 2020]	43.2±0.3	60.2±0.3
Centroid [Afrasiyabi *et al.*, 2020]	45.83±0.48	59.74±0.56
DeepEMD [Zhang *et al.*, 2020]	46.47±0.78	63.22±0.71
F.Wu et al♢ [Wu *et al.*, 2020]	47.5±0.9	59.8±1.0
SNE (Ours)	50.51±0.05	64.89±0.05

Table 3: Comparisons of average accuracies (%) with 95% confidence intervals against state-of-the-art methods for 1-shot and 5-shot classification on **FC100** benchmark.

			miniImageNet		CIFAR-FS	
ST	DT	AL	1-shot	5-shot	1-shot	5-shot
×	×	×	61.19	79.91	66.69	82.46
✓	×	×	66.78	81.15	73.87	84.93
✓	✓	×	70.27	84.19	79.35	88.33
✓	✓	✓	71.02	84.56	79.53	88.56

Table 4: Ablation studies on the components of the proposed SNE model on miniImageNet and CIFAR-FS. The baseline (the first row) is the direct utilization of N_{base} without network evolution. ST indicates the network evolution with label propagation (Eq. 3), DT stands for the network evolution by dual-task (Eq. 5), and AL suggests the adaptive multi-task loss (Eq. 6).

Parameter C	15	20	25
1-shot	70.27%	71.02%	69.71%
5-shot	84.64%	84.56%	83.89%

Table 5: Ablation study of the Parameter C on miniImageNet.

Backbone	Conv-64F	ResNet12	SEResNet12
1-shot	51.78%	71.02%	71.21%
5-shot	65.34%	84.56%	84.35%

Table 6: Ablation study of different backbones on miniImageNet

4.3 Ablation Study

In this section, an ablation study is conducted on miniImageNet and CIFAR-FS to analyze the impacts of different components and parameters in our SNE model. Four settings are compared: (1) the direct utilization of N_{base} without network

Methods	5-way		10-way		15-way		20-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline++ [Chen *et al.*, 2019]	57.53	72.99	40.43	56.89	31.96	48.2	26.92	42.8
LEO [Rusu *et al.*, 2019]	61.76	77.59	45.26	64.36	36.74	56.26	31.42	50.48
S2M2† [Mangla *et al.*, 2020]	64.93	83.18	50.4	70.93	41.65	63.32	35.5	58.36
EPNet† [Rodríguez *et al.*, 2020]	70.74	84.34	53.70	72.17	44.55	64.44	38.55	59.01
SNE (Ours)	71.02	84.56	59.32	76.06	52.7	70.46	47.45	65.94

Table 7: Evaluations on a higher way setup. Different values of N (N-way K-shot) are set for the few-shot classification task on miniImageNet.

evolution (baseline); (2) network evolution with label propagation; (3) network evolution with dual-task; (4) SNE model.

The results are summarized in Table 4. Network Evolution brings a significant improvement of 5.59% on the 1-shot task compared with the baseline (66.78% VS. 61.19%) on miniImageNet. It verifies that the network evolution effectively transfers and adapts the source knowledge (base classes) to the target domain (novel classes). By bringing the self-supervised task in the network evolution, a gain of 3.49% is observed because the impacts of incorrect label propagation are effectively mitigated. To analyze the effect of the adaptive multi-task loss, we set the conventional multi-loss as $\mathcal{L}_{all} = 0.5\mathcal{L}_s + 0.5\mathcal{L}_r$. Our full SNE model adopts the adaptive multi-task loss technique, which boosts the performance by a margin of 0.75% against the conventional loss on miniImageNet. The improvement is not obvious on CIFAR-FS, and a possible reason is the uncertainty of the task on CIFAR-FS is not as high as that on miniImageNet.

Besides, we also execute an ablation study to analyze the effects of C in the C-nearest neighbor of our clustering stage (see also Section 3.2). Three settings of C are examined, including 15, 20, and 25. Table 5 reveals the results, where 20 achieves the best performance in the 1-shot evaluation and occupies the second place in the 5-shot evaluation, but it only lags the best performer (C=15) by 0.09% (5-shot). When C drops to 15, the diversity of C-nearest images decreases, which results in a decline of 1-shot accuracy. As C increases to 25, the quality of deep clustering is impacted, which may also influence the accuracy of the few-shot classification.

Lastly, we analyze the impacts of backbones (Conv-64F, ResNet12, SEResNet12) on miniImageNet. The results are shown in the Table 6. ResNet12 and SEResNet12 obtain a similar result while Conv-64F shows relatively worse results due to its limited representation ability.

4.4 Evaluations on a Higher Way Setup and Cross-dataset Scenarios

Impacts of N in N-way K-shot: To testify the robustness of our model, we evaluate our method in the few-shot scenario with more categories. We increase the value of N in N-way K-shot from 5 to 10, 15, and 20. This makes the evaluation more complicated and closer to a real scenario. The results are summarized in Table 7. Compared with other state-of-the-art algorithms, our method achieves the best accuracy in all scenarios. With the increase of N, the difficulty of the evaluation gradually increases and the advantages of our method over other SOTA methods become more obvious. In the 20-way scenario, our method improves the second-best performer EPNet by a large gain of 8.9% (38.55% VS. 47.45%) in 1-shot and 6.93% (59.01% VS. 65.94%) in 5-shot, which proves the generalization ability to more classes of our SNE model.

Methods	miniIN ⇒ CIFAR-FS		miniIN ⇒ FC100	
	1-shot	5-shot	1-shot	5-shot
baseline++	42.23	61.62	33.74	47.46
RFS	58.2	74,8	41.9	55.63
S2M2	52.42	72.9	39.99	56.06
SNE (Ours)	67.79	82.99	63.08	81.41

Table 8: Evaluations on cross-dataset scenarios.

Cross-dataset Evaluation: Each dataset has a unique data distribution. MiniImageNet has higher image resolution and lower inner-class similarity, while CIFAR-100 derivatives have lower image resolution and higher inner-class similarity. In fact, domain differences not only exist in the same dataset but also exist between datasets in real scenarios. Therefore, we further evaluate the few-shot classification accuracy over cross-dataset: **miniImageNet ⇒ CIFAR-FS** and **miniImageNet ⇒ FC100**. The experimental results are reported in Table 8 where miniIN stands for miniImageNet. It is clear from the results that our method has a great advantage over other methods in the cross-dataset scenario. This suggests that our method can transfer the knowledge of base classes to novel classes even they are from different datasets.

5 Conclusion

In this work, a Self-supervised Network Evolution (SNE) model is developed to deal with the problem of few-shot classification. The network evolution encodes the latent distribution transferring from the already-known classes to the novel/new classes by label propagation and self-supervised learning (dual-task design). The dual-task exploits a discriminative representation to effectively alleviate the propagation of incorrect pseudo labels in the network. We have conducted extensive experiments to demonstrate our SNE model in various few-shot scenarios. In the standard few-shot evaluation, our method can achieve state-of-the-art performance on miniImageNet and CIFAR-FS. Furthermore, our SNE model has presented a superiority in a higher way setup and the cross-dataset evaluation as well.

Acknowledgments

This work was supported by the Beijing Municipal Natural Science Foundation (Grant No. 4212041) and the Natural Science Foundation of China (61972027).

References

[Afrasiyabi *et al.*, 2020] Arman Afrasiyabi, Jean Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *ECCV*, pages 18–35, 2020.

[Bateni *et al.*, 2020] Peyman Bateni, Raghav Goyal, Vaden Masrani, and et al. Improved few-shot visual classification. In *CVPR*, pages 14481–14490, 2020.

[Chen *et al.*, 2019] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, and et al. A closer look at few-shot classification. In *ICLR*, 2019.

[Dhillon *et al.*, 2020] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and et al. A baseline for few-shot image classification. In *ICLR*, 2020.

[Doersch *et al.*, 2015] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[Gansbeke *et al.*, 2020] Wouter Van Gansbeke, Simon Vandenhende, and et al. SCAN: learning to classify images without labels. In *ECCV*, pages 268–285, 2020.

[Gidaris *et al.*, 2018] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[Gidaris *et al.*, 2019] Spyros Gidaris, Andrei Bursuc, and et al. Boosting few-shot visual learning with self-supervision. In *ICCV*, pages 8058–8067, 2019.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, June 2018.

[Kim *et al.*, 2020] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *ECCV*, pages 599–617, 2020.

[Lee *et al.*, 2019] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and et al. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019.

[Liu *et al.*, 2020] Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *ECCV*, pages 404–421, 2020.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mangla *et al.*, 2020] Puneet Mangla, Mayank Singh, Abhishek Sinha, and et al. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, pages 2207–2216, 2020.

[Noroozi and Favaro, 2016] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016.

[Oreshkin *et al.*, 2018] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *NIPS*, pages 719–729, 2018.

[Pathak *et al.*, 2016] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, and et al. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[Rodríguez *et al.*, 2020] Pau Rodríguez, Issam H. Laradji, Alexandre Drouin, and et al. Embedding propagation: Smoother manifold for few-shot classification. In *ECCV*, pages 121–138, 2020.

[Rusu *et al.*, 2019] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, and et al. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

[Simon *et al.*, 2020] Christian Simon, Piotr Koniusz, Richard Nock, and et al. Adaptive subspaces for few-shot learning. In *CVPR*, pages 4135–4144, 2020.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, and et al. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

[Tian *et al.*, 2020] Yonglong Tian, Yue Wang, and et al. Rethinking few-shot image classification: A good embedding is all you need? In *ECCV*, pages 266–282, 2020.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Tim Lillicrap, and et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[Wang *et al.*, 2020] Yikai Wang, Chengming Xu, Chen Liu, and et al. Instance credibility inference for few-shot learning. In *CVPR*, pages 12833–12842, 2020.

[Wu *et al.*, 2020] Fangyu Wu, Jeremy S. Smith, and et al. Attentive prototype few-shot learning with capsule network-based embedding. In *ECCV*, pages 237–253, 2020.

[Yang *et al.*, 2020] Ling Yang, Liangliang Li, and Zilun Zhang. DPGN: distribution propagation graph network for few-shot learning. In *CVPR*, pages 13387–13396, 2020.

[Ye *et al.*, 2020] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and et al. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, pages 8805–8814, 2020.

[Zhang *et al.*, 2016] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016.

[Zhang *et al.*, 2020] Chi Zhang, Yujun Cai, Guosheng Lin, and et al. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, pages 12200–12210, 2020.