Attentive Prototype Few-shot Learning with Capsule Network-based Embedding

Anonymous ECCV submission

Paper ID 6066

Abstract. Few-shot learning, namely recognizing novel categories with a very small amount of training examples, is a challenging area of machine learning research. Traditional deep learning method requires massive training data to tune the huge number of parameters, which is often impractical and prone to over-fitting. In this work, we further work on the well-known few-shot learning method known as prototypical networks for better performance. Our contributions include (1) a new embedding structure to encode relative spatial relationships between features by applying capsule network; (2) a new triplet loss designated to enhance the semantic feature embedding where similar samples are close to each other while dissimilar samples are farther apart; and (3) an effective non-parametric classifier termed attentive prototypes in place of the simple prototypes in current few-shot learning. The proposed attentive prototype aggregates all of the instances in a support class which are weighted by their importance defined by the reconstruction error for a given query. The reconstruction error allows the classification posterior probability to be estimated, which corresponds to the classification confidence score. Extensive experiments on three benchmark datasets demonstrate that our approach is effective for the few-shot classification task.

Keywords: Few-shot learning · Meta learning · Capsule network · Feature embedding · Attentive prototype learning

1 Introduction

Deep learning has been greatly advanced in recent years, with many successful applications in image processing, speech processing, natural language processing and other fields. However, the successes usually rely on the condition to access a large dataset for training. If the amount of training data is not large enough, the deep neural network would not be sufficiently trained. Consequently, it is significant to develop deep learning for image recognition in the case of a small number of samples, and enhance the adaptability of deep learning models in different problem domains.

Few-shot learning is one of the most promising research areas targeting deep learning models for various tasks with a very small amount of training dataset [41], [36], [31], [39], [33], [26] , i.e., classifying unseen data instances (query examples) into a set of new categories, given just a small number of labeled instances in each class (support examples). The common scenario is a support set with only 1∼10 labeled examples per class. As a stark contrast, general classification problems with deep learning models [17], [40] often require thousands of examples per class. On the other hand, classes for training and testing sets are from two exclusive sets in few-shot learning while in traditional classification problems they are the same. A key challenge in few-shot learning, therefore, is to make best use of the limited data available in the support set in order to find the right generalizations as required by the task.

Few-shot learning is often elaborated as a meta-learning problem, with emphasis on learning prior knowledge shared across a distribution of tasks [41], [23], [36]. There are two sub-tasks for meta-learning: an embedding that maps the input into a feature space and a base learner that maps the feature space to task variables. As a simple, efficient and most popularly used few-shot learning algorithm, prototypical network [36] tries to solve the problem by learning the metric space to perform classification. A query point (new point) is classified based on the distance between the created prototypical representation of each class and the query point. While the approach is extensively applied, there are a number of limitations that we'd like to address and seek better solutions.

Firstly, the prototypical representations [41], [36] generated by deep Convolutional Neural Networks, can not account for the spatial relations between the parts of the image and are too sensitive to orientation. Secondly, a prototypical network [36] divides the output metric space into disjoint polygons where the nearest neighbor of any point inside a polygon is the pivot of the polygon. This is too rough to reflect various noises effects in the data, thus compromising the discrimination and expressiveness of the prototype. It has been well-known that the performance of such a simple distance-based classification is severely influenced by the existing outliers, especially in the situations of small training sample size [8].

From the aforementioned discussion, we intend to improve the prototype network by proposing a capsule network [34] based embedding model and reconstruction-based prototypical learning within the framework of the meta-learning. There are two main components in the proposed scheme: a capsule network-based embedding module which create feature representation, and an improved non-parametric classification scheme with an attentive prototype for each class in the support set, which is obtained by attentive aggregation over the representations of its support instances, where the weights are calculated using the reconstruction error for the query instance. The training of the proposed network is based on the metric learning algorithm with an improved triplet-like loss, which generalizes the triplet network [35] to allow joint comparison with K negative prototypes in each mini-batch. This makes the feature embedding learning process more tally with the few-shot classification problem. We further propose a semi-hard mining technique to sample informative hard triplets, thus speeding up the convergence and stabilize the training procedure.

In summary, we proposed a new embedding approach for few-shot learning based on capsule network, which features of the capability to encode the part-whole relationships between various visual entities. An improved routing procedure with DeepCaps mechanism [29] is designed to implement the embedding. With class-specific output capsule, the proposed network can better preserve the semantic feature representation, and reduce the disturbance of irrelevant noisy information. The proposed attentive prototype scheme is query-dependent, rather than just averaging the feature points of a class for the prototype as in the vanilla prototype network, which means all of the feature points from the support set are attentively weighted in advance, and then the weighting values completely depend on the affinity relations between two feature points from the support set and the query set. By using reconstruction as an efficient expression of the affinity relation, the training points near the query feature point acquire more attention in the calculation of the weighting values.

The proposed approach has been experimentally evaluated on few-shot image classification tasks using three benchmark datasets, i.e., the *mini*ImageNet, *tiered*ImageNet and Fewshot-CIFAR100 datasets. The empirical results verify the superiority of our method over the state-of-the-art approaches. The main contributions of our work are two-fold:

- We put forward a new few-shot classification approach with a capsule-based model, which combines the 3D convolution based dynamic routing procedure to obtain semantic feature representation while preserving the spatial information between visual entities.
- We propose a novel attentive prototype concept to take account of all the instances in a given support class, with each instance being weighted by the reconstruction errors between the query and prototype candidates from the support set. The attentive prototype is robust to outliers by design and also allows the performance to be improved by refraining from making predictions in the absence of sufficient confidence.

2 Related work

2.1 Few-shot learning

Few-shot learning aims to classify novel visual classes with very few labeled samples available [4], [5]. Current methods usually tackle the challenge using meta-learning approaches or metric-learning approaches, with representative works elaborated below:

Metric learning methods aim to learn a task-invariant metric, which provide an embedding space for learning from few-shot examples. Vinyals et al. [41] introduced the concept of episode training in few-shot learning, where metric learning-based approaches learn a distance metric between a test example and the training examples. Prototypical networks [36] learn a metric space in which classification can be performed by computing distances to prototype representations of each class. The learned embedding model maps the images of the same class closer to each other while different classes are spaced far away. The mean of embedded support samples are utilized as the prototype to represent the class.

The work in [19] goes beyond this by incorporating the context of the entire support set available by looking between classes and identifying task-relevant features.

There are also interesting works that explore different metrics for the embedding space to provide more complex comparisons between support and query features. For example, the relation module proposed in [39] calculates the relation score between query images to identify unlabeled images. Some recent approaches [9], [21] have also explored graph-based similarity for few-shot learning with the node-labeling framework, which implicitly models the intra-cluster similarity and the inter-cluster dissimilarity. Kim et al. [14] proposed an edge-labeling GNN (EGNN) for few-shot classification. Metric-based task-specific feature representation learning has also been presented in many related works. Our work is a further exploration of the prototype based approaches [36], [39], aiming to enhance the performance of learning an embedding space by encoding the spatial relationship between features. Then the embedding space generates attentive prototype representations in a query-dependent scheme.

2.2 Capsule Networks

The capsule network [13] is a new type of neural network architecture proposed by Geoffrey Hinton, with the main motivation to address some of the shortcomings of Convolutional Neural Networks (CNNs). For example, the pooling layers of CNNs lose the location information of relevant features, one of the so-called instantiation parameters that characterize the object. Other instanced parameters include scale and rotation, which are also poorly represented in the CNNs. Capsule network handles these instantiation parameters explicitly by representing an object or a part of an object. More specifically, a capsule network replaces the mechanisms of the convolution kernel in CNNs by implementing a group of neurons to encode the spatial information and the probability of the existence of objects. The length of the Capsule vector is the probability of the features in the image, and the orientation of the vector will represent its instantiation information.

Sabour et al. [34] first proposed a dynamic routing algorithm for capsule network in 2017 for the bottom-up feature integration, the essence of which is the realization of clustering algorithm for the information transmission in the model. In [34], Gaussian mixture model (GMM) was integrated into the feature integration process to adjust network parameters through EM routing. Since the seminal works [13], [34], a number of approaches have been proposed to implement and improve the capsule architecture [15], [45], [18],[29].

Many applications have been attempted by applying the capsule networks, for example, intent detection [42], text classification [27] and computer vision [43], [44]. A sparse, unsupervised capsules network [30] was proposed showing that the network generalizes better than supervised masking, while potentially enabling deeper capsule networks. A group equivariant capsule network [18] was proposed as a framework to introduce guaranteed equivariance and invariance properties. Rajasegaran et al. [29] proposed a deep capsule network architecture called DeepCaps that adapts the original routing algorithm for 3D convolutions and increases its performance on more complex datasets. A variant of capsule network was investigated in [3] in which all capsules are divided into different groups and perform group reconstruction routing algorithm to obtain the corresponding high-level capsules.

In this work, we further work on the prototype network to improve the few-shot learning performance, with novel contributions including (1) extending the concept of prototype from simple average for each category to attentive prototype which takes account of all of the instances, and (2) establishing an embedding space with capsule network which provides a description of the images components at various 'levels' of semantics. With the aid of 3D convolution based dynamic routing, our model can capture part-whole relationships between the corresponding deeper and shallower capsules.

3 Method

3.1 Problem definition: Few-shot classification

Few-shot learning is to recognize novel categories with only one or few labeled examples by transferring visual patterns obtained from base categories to describe the novel categories. The problem is usually formulated with three datasets: a training set D_{train}, a support set $D_{support}$ and a query set D_{query}. The categories in D_{train} are defined as base categories C_{base}. The categories in $D_{support}$ and D_{test} are novel categories which are exclusive with the training set D_{train}. If the support set contains M categories and each category has K image examples, this few-shot learning problem is defined as M-way K-shot learning. We follow the practice of episodic training in [41] which is the most popular and effective meta learning methodology [36], [39].

3.2 Approach Details

In this section, we first revisit the DeepCaps network [29], which designed for more complex image datasets. We then extend it to the scenario of few-shot learning, and describe the proposed algorithm in detail.

DeepCaps Revisit DeepCaps is a deep capsule network architecture proposed in [29] to improve the performance of the capsule networks for more complex image datasets. It extends the dynamic routing algorithm in [34] to stacked multiple layers, which essentially uses a 3D convolution to learn the spatial information between the capsules. The model consists of four main modules: skip connected CapsCells, 3D convolutional CapsCells, a fully-connected capsule layer and a decoder network. The skip-connected CapsCells have three ConvCaps layers, the first layer output is convolved and skip-connected to the last layer output. The motivation behind skipping connections is to borrow the idea from

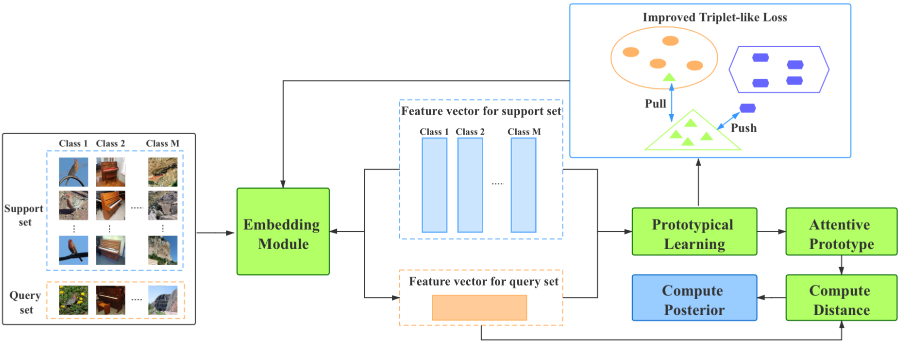

Fig. 1. Framework of the proposed method for few-shot learning. We perform joint end-to-end training of the Embedding Module (modified DeepCaps) together with the Prototypical Learning via an improved triplet-like loss from the training dataset. The well-learned embedding features are used to compute the distances among the query image and attentive prototype generated from the support set. The final classification is performed by calculating the posterior probability for the query instance.

residual network to sustain a sound gradient flow in a deep model. The element-wise layer is used to combine the outputs of the two capsule layers after skipping the connection.

DeepCaps has a unit with a ConvCaps3D layer, in which the number of route iterations is kept at 3. Then, before dynamic routing, the output of ConvCaps is flattened and connected with the output of the capsule, which is then followed by 3D routing (in CapsCell 3). Intuitively, this step helps to extend the model to a wide range of different datasets. For example, for a dataset composed of images with less rich information, such as MNIST, the low-level capsule from cell 1 or cell 2 is sufficient, while for a more complex dataset, we need the deeper 3D ConvCaps to capture rich information content. Once all capsules are collected and connected, they are routed to the class capsule through the fully-connected capsule layer.

Network Architecture As explained in Section 1, our proposed model has two parts: (1) a modified DeepCaps network with improved triplet-like loss that learns the deep embedding space, and (2) an non-parameter classification scheme that produces a prototype vector for each class candidate, which is derived from the attentive aggregation over the representations of its support instances, where the weights are calculated using the reconstruction errors for the query instance from respective support instances in the embedding space. The final classification is performed by calculating the posterior probability for the query instance based on the distances between the embedding vectors of the query and the attentive prototype. Figure 1 schematically illustrates an overview of our approach to few-shot image classification. We describe each of the parts in detail below.

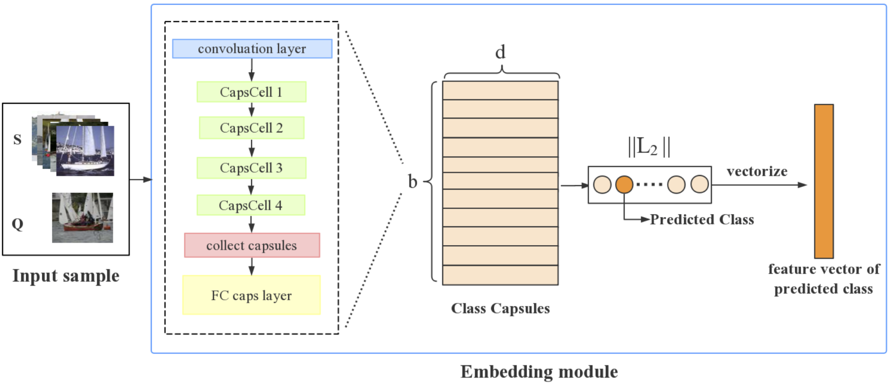

Fig. 2. The architecture of embedding module in which obtain only the activity vectors of the predicted class.

Embedding module. As mentioned in Section 3.1, we construct support set S and query set Q from D_{train} in each episodes to train the model.

$$\begin{aligned} S &= \{s_1, s_2, .., s_K\}, \\ Q &= \{q_1, s_2, ..., q_N\}, \end{aligned} \tag{1}$$

where K and N represent the number of samples in the support set and query set for each class, respectively. As shown in Fig. 2, we first fed the samples S and Q into the convolution layer and CapsCells, then the collected capsules are routed to the class capsules after the Flat Caps layer. Here, the decision making happens via L_2 and the input image is encoded into the final capsule vector. The length of the capsule's output vector represents the probability that the object represented by the capsule exists in the current input. We assume the class capsules as $P \in Y^{b\times d}$ which consists of the activity vectors for all classes, where b and d represents the number of classes in the final class capsule and capsule dimension, respectively. Then, we only fed the activity vector of predicted class $P_m \in Y^{1\times d}$ into the final embedding space in our setting, where $m = argmax_i(||P_i||_2^2)$. The embedding space acts as a better regularizer for the capsule networks, since it is forced to learn the activity vectors jointly within a constrained Y^d space. The function of margin loss used in DeepCaps enhances the class probability of the true class, while suppressing the class probabilities of the other classes. In this paper, we propose the improved triplet-like loss based on an attentive prototype to train the embedding module and learn more discriminative features.

Attentive prototype. The prototypical network in [36] computes a D dimensional feature representation $p_i \in \mathbb{R}^D$, or prototype, of each class through an embedding function $f_\phi : \mathbb{R}^D \to \mathbb{R}^M$ with learnable parameters ϕ. Each prototype is the mean vector of the embedded support points belonging to its class:

$$p_i = \frac{1}{|s_i|} \sum_{(x_i, y_i) \in s_i} f_\phi(x_i) \tag{2}$$

where each $x_i \in s_i$ is the D-dimensional feature vector of an example from class i. Given a distance function $d : \mathbb{R}^D \times \mathbb{R}^D \rightarrow [0, +\infty)$, prototypical networks produce a distribution over classes for a query point x based on a softmax over distances to the prototypes in the embedding space:

$$p_\phi(y = t|x) = \frac{exp(-d(f_\phi(x), p_t))}{\sum_{t'} exp(-d(f_\phi(x), p_{t'}))} \tag{3}$$

Learning proceeds by minimizing the negative log-probability $J(\phi) = -logp_\phi(y = t|x)$ of the true class t via SGD. Most prototypical networks for few-shot learning use some simple non-parametric classifiers, such as kNN. It is well known that non-parametric classifiers are usually affected by existing outliers [7], which is particularly serious when the number of samples is small, the scenario addressed by few-shot learning. A practical and reliable classifier should be robust to outliers. Motivated by this observation, we propose an improved algorithm based on the local mean classifier [24]. Given all prototype instances of a class, we calculate their reconstruction errors for the query instance, which are then used for the weighted average of prototype instances. The new prototype aggregates attentive contributions from all of the instances. The reconstruction error between the new prototype and the query instance not only provides a discrimination criteria for the classes, but also serves as a reference for the reliability of the classification.

More specifically, with K support samples $\{x_{i1}, x_{i2}, ..., x_{iK}\}$ selected for class i, a membership γ_{ij} can be defined for a query instance q by employing normalized Gaussian functions with the samples in support sets, e.g.,

$$\gamma_{ij} = \frac{exp(\frac{||q - x_{ij}||^2}{2\sigma_i^2})}{\sum_{l=1}^{K} exp(\frac{||q - x_{il}||^2}{2\sigma_i^2})}, j = 1, ..., K, i = 1, ..., M \tag{4}$$

where x_{ij} are the j-th samples in class i, and σ_i is the width of Gaussian defined for class i, and we set the value σ_i relatively small (e.g, σ_i=0.1).

Then, for each class i, an attentive prototype pattern $\hat{q}_i$ can be defined for a query sample q as below

$$\hat{q}_i = \frac{\sum_{j=1}^{K} \gamma_{ij} x_{ij}}{\sum_{l=1}^{K} \gamma_{ij}}, i = 1, ..., M \tag{5}$$

where γ_{ij} is defined in Eq. 4 and $\hat{q}_i$ can be considered as the generalized support samples from class i for the query instance q. Here we want to ensure that an image q^a (anchor) of a specific class in the query set is closer to the attentive prototype of the positive class $\hat{q}^p$ (positive) than it is to multiple $\hat{q}^n$ (negative) attentive prototypes.

$$||q^a - \hat{q}^p||_2^2 + \alpha < ||q^a - \hat{q}^n||_2^2, \forall q^a \in Q. \tag{6}$$

f where α is a margin that is enforced between positive and negative pairs, Q is the query set cardinality MN. The loss that is being minimized is then:

$$\sum_{m=1}^{MN} \left[||f(q_m^a) - f(\hat{q}_m^p))||_2^2 - ||f(q_m^a) - f(\hat{q}_m^n)||_2^2 + \alpha \right]_+ \tag{7}$$

For image classification, a query image can be classified based on the comparison of the errors between the reconstructed vectors and the presented image. That is, a query image q is assigned to m^* class if

$$m^* = \underset{m}{argmin}\, err_m \tag{8}$$

where $err_m = ||q - \hat{q}_m||, m = 1, ..., M$.

Improved Triplet-like loss. In order to ensure fast convergence it is crucial to select triplets that violate the triplet constraint in Eq. 7. The traditional triplet loss interacts with only one negative sample (and equivalently one negative class) for each update in the network, while we actually need to compare the query image with multiple different classes in few-shot classification. Hence, the triplet loss may not be effective for the feature embedding learning, particularly when we have several classes to handle in the few-shot classification setting. Inspired by [1], [37], we generalize the traditional triplet loss with E-negatives prototypes to allow simultaneous comparisons jointly with the E negative prototypes instead of just one negative prototype, in one mini-batch. This extension makes the feature comparison more effective and faithful to the few-shot learning procedure, since in each update, the network can compare a sample with multiple negative classes.

In particular, we randomly choose the E negative prototypes $\hat{q}^{n_e}, e = \{1, 2, ..., E\}$ to form into a triplet. Accordingly, the optimization objective evolves to:

$$\begin{aligned}\mathcal{L}(q_m^a, \hat{q}_m^p, \hat{x}_m^n) = \quad & \sum_{m=1}^{MN} \frac{1}{E} \sum_{e=1}^{E} \left[||f(q_m^a) - f(\hat{q}_m^p))||_2^2 \right. \\ & \left. - ||f(q_m^a) - f(\hat{q}_m^{n_e})||_2^2 + \alpha \right]_+ \end{aligned} \tag{9}$$

For the sample q_m^a in the query set, the optimization shall maximize the distance to the negative prototype q_m^n to be larger than the distance to the positive prototypes q_m^p in the feature space. For each anchor sample q_m^a, we then learn the positive prototype q_m^p from the support set of the same class as q_m^a and further randomly select E other negative prototypes whose classes are different from q_m^a. Compared with the traditional triplet loss, each forward update in our improved Triplet-like loss includes more inter-class variations, thus making the learnt feature embedding more discriminative for samples from different classes.

Mining hard triplets is an important part of metric learning with the triplet loss, as otherwise training will soon stagnate [12]. This is because when the model begins to converge, the embedding space learns how to correctly map the triples relatively quickly. Thus most triples satisfying the margin will not contribute to the gradient in the learning process. To speed up the convergence and stabilize the training procedure, we propose a new hard-triplet mining strategy to sample more informative hard triplets in each episode. Specifically, triplets will be randomly selected in each episode as described above, we then check whether the sampled triplets satisfy the margin. The triplets that have already met the margin will be removed and the network training will proceed with the remaining triplets.

4 Experiments

Extensive experiments have been conducted to evaluate and compare the proposed method for few-shot classification using on three challenging few-shot learning benchmarks datasets, *mini*ImageNet [41], *tiered*ImageNet [31] and Fewshot-CIFAR100 (FC100) [26]. All the experiments are implemented based on PyTorch and run with NVIDIA 2080 GPUs.

4.1 Datasets

***mini*ImageNet** is the most popular few-shot learning benchmark proposed by [41] and derived from the original ILSVRC-12 dataset [32]. It contains 100 randomly sampled different categories, each with 600 images of size 84×84 pixels. The ***tiered*ImageNet** [31] is is a larger subset of ILSVRC-12 [32] with 608 classes and 779,165 images in total. The classes in *tiered*ImageNet are grouped into 34 categories corresponding to higher-level nodes in the ImageNet hierarchy curated by human [2]. Each hierarchical category contains 10 to 20 classes, which are divided into 20 training (351 classes), 6 validation (97 classes) and 8 test (160 classes) categories. **Fewshot-CIFAR100 (FC100)** is based on the popular object classification dataset CIFAR100 [16]. Oreshkin et al. [26] offer a more challenging class split of CIFAR100 for few-shot learning. The FC100 further groups the 100 classes into 20 superclasses. Thus the training set has 60 classes belonging to 12 superclasses, the validation and test data consist of 20 classes belonging to 5 superclasses each.

4.2 Implementation Details

Following the general few-shot learning experiment settings [36], [39], we conducted 5-way 5-shot and 5-way 1-shot classifications. The Adam optimizer is exploited with an initial learning rate of 0.001. The total training episodes on *mini*ImageNet, *tiered*ImageNet and FC100 are 600,000, 1,000,000 and 1,000,000, respectively. The learning rate is dropped by 10% every 100,000 episodes or when the loss enters a plateau. The weight decay is set to 0.0003. We report the mean accuracy (%) over 600 randomly generated episodes from the test set.

Few-shot learning method	5-Way 1-Shot	5-Way 5-Shot
Matching Networks [41]	43.56 ± 0.84	55.31±0.73
MAML [6]	48.70±1.84	63.11±0.92
Relation Net [39]	50.44±0.82	65.32±0.70
REPTILE [25]	49.97±0.32	65.99±0.58
Prototypical Net [36]	49.42±0.78	68.20±0.66
Predict Params [28]	59.60±0.41	73.74 ± 0.19
LwoF [10]	60.06±0.14	76.39 ± 0.11
TADAM [26]	58.50±0.30	76.70±0.30
EGNN [14]	–	66.85
EGNN+Transduction [14]	–	76.37
CTM [19]	62.05±0.55	78.63±0.06
wDAE-GNN [11]	62.96±0.15	78.85±0.10
CTM, data augment [19]	64.12±0.82	80.51±0.13
Baseline	59.71±0.35	75.21±0.43
Ours	63.23±0.26	80.17±0.33
Ours, data augment	**66.43±0.26**	**82.13±0.21**

Table 1. Few-shot classification accuracies (%) on *mini*ImageNet.

4.3 Results Evaluation

Comparison with baseline model. Using the training/testing data split and the procedure described in Section 3, the baseline in Table 1, Table 2 and Table 3 evaluate a model with modified DeepCaps, without the attentive prototype. The accuracy is 75.21±0.43%, 78.41±0.34% and 59.8±1.0% and in the 5-way 5-shot setting on *mini*ImageNet, *tiered*ImageNet and FC100 respectively. Our baseline results are on par with those reported in [39], [36]. As shown in Table 1, Table 2 and Table 3, using the attentive prototype strategy in the model training with improved triplet-like loss, our method significantly improves the accuracy on all of the three datasets. There are obvious improvements of approximately +4.96% (from 75.21% to 80.17%), +4.83% (from 78.41% to 83.24%), +2.5% (from 57.3% to 59.8%) under the 5-way 5-shot setting for *mini*ImageNet, *tiered*ImageNet and FC100, respectively. For the one-shot setting, we further analyze the improved triplet-like loss which is based on samples. On *mini*ImageNet, *tiered*ImageNet and FC100, we achieve improvement of +3.52% (from 59.71% to 63.23%), +2.28% (from 63.25% to 65.53%) and +3.3% (from 44.2% to 47.5%) in the 5-way 1-shot setting, respectively. These results indicate that the proposed approach is tolerant to large intra- and inter-class variations and the improved triplet-like loss produces marked improvements over the baseline.

Comparison with the state-of-the-art methods. We also compare our method with some state-of-the-art methods on *mini*ImageNet,*tiered*ImageNet in Table 1 and Table 2, respectively. On *mini*ImageNet, we achieve at a **5-way 1-shot accuracy =63.23±0.26, 5-way 5-shot accuracy =80.17 ± 0.33%** when using the proposed method, which has a highly competitive performance

Few-shot learning method	5-Way 1-Shot	5-Way 5-Shot
MAML [6]	51.67±1.81	70.30±0.08
Meta-SGD [20], reported by [33]	62.95±0.03	79.34±0.06
LEO [33]	66.33±0.05	81.44±0.09
Relation Net [39]	54.48±0.93	71.32±0.78
Prototypical Net [36]	53.31±0.89	72.69±0.74
EGNN [14]	–	70.98
EGNN+Transduction [14]	–	80.15
CTM [19]	64.78±0.11	81.05±0.52
CTM, data augment [19]	68.41±0.39	84.28±1.73
Baseline	63.25±0.31	78.41±0.34
Ours	65.53±0.21	83.24±0.18
Ours, data augment	**69.87±0.32**	**86.35±0.41**

Table 2. Few-shot classification accuracies (%) on *tiered*ImageNet.

compared with the state-of-the-art. On *tiered*ImageNet, we arrive at **5-way 1-shot accuracy = 65.53±0.21, 5-way 5-shot accuracy =83.24 ± 0.18%** which is also very competitive. The previous best result is produced by introducing a Category Traversal Module [19] and data augment that can be inserted as a plug-and-play module into most metric-learning based few-shot learners. We further investigate whether the data augment could work on our model. By training a version of our model with basic data augmentation, we obtain the improved results **5-way 5-shot accuracy = 82.13±0.21%** on *mini*ImageNet. On *tiered*ImageNet, we also observe a performance **5-way 5-shot accuracy = 86.35±0.41%**. For FC100 dataset, our proposed method is superior to all the other methods [6], [26], [38] in accuracy. The comparisons consistently confirm the competitiveness of the proposed method on few-shot image classification. In

Few-shot learning method	5-Way 1-Shot	5-Way 5-Shot	5-Way 10-Shot
MAML [6]	38.1±1.7	50.4±1.0	56.2±0.8
TADAM [26]	40.1±0.4	56.1±0.4	61.6±0.5
MTL [38]	45.1±1.8	57.6±0.9	63.4±0.8
Baseline	44.2±1.3	57.3±0.8	62.8±0.6
Ours	**47.5±0.9**	**59.8±1.0**	**65.4±0.5**

Table 3. Few-shot classification accuracies (%) on FC100 dataset.

summary, our proposed attentive prototype learning scheme obviously improve over the previous methods, mainly due to the better embedding space provided by the capsule network and the attentive prototyping scheme. The importance value is used as the weighting value for the support set instances, which is completely dependent on the affinity relationship between the two feature points from the support set and the query. The importance weighting values vary exponentially, with larger value reflecting nearby pairs of feature points and a smaller value for the distant pair. This conforms that the feature points from the support set that are nearer to the query feature point should be paid more attention.

Ablation Study: To verify the effectiveness of components in proposed method, we conducted ablation experiments on the *mini*ImageNet and *tiered*ImageNet dataset. First, to investigate the contribution of the designed attentive prototype method, we compare the performance of the proposed method with vanilla prototypical networks [36]. Then, we verify the effectiveness of our proposed feature embedding module by embedding it into the metric-based algorithm Relation Net [39].Table 4 summarizes the performance of the different variants of our method.

Method	*mini*ImageNet		*tiered*ImageNet	
	5-Way 5 shot	10-Way 5 shot	5-Way 5-shot	10-Way 5-shot
Prototypical Net [36]	68.20	-	72.69	-
Ours (average mechanism)	76.32	58.41	80.31	62.17
Ours (attentive prototype)	80.17	63.12	83.24	66.33
Relation Net [39]	65.32	–	71.32	–
Relation Net [39] (our implementation)	80.91	64.34	83.98	67.86

Table 4. Ablation study on attentive prototype and embedding module.

1)*Attentive prototype:* In vanilla prototypical networks [36], the prototypes are defined as the averages the embed features of each class in the support set. Such a simple class-wise feature takes all instances into consideration equally. Our attentive prototype scheme is a better replacement. A variant of DeepCaps is applied with improved triplet-like loss to learn the feature embedding instead of a shallow CNN network. To further verify the effectiveness of our attentive prototype, we also compared the average-based prototypes created from our embedding framework. The experimental results on *mini*ImageNet and *tiered*ImageNet are summarized in Table 3. It can be observed that the attentive prototype gains an approximately 3%-4% increase after replacing the average mechanism. This shows that the attentive prototypes can be more 'typical' when compared to the original average vectors by giving different weights for different instances.

2)*Embedding module:* The embedding is switched from four convolutional blocks in Relation Net [39] to the modified DeepCaps model and the supervision loss is changed to the improved triplet-like loss. Table 3 shows the results obtained by the improvements over the Relation Net. We find that the improved Relation Net exceeds the original model by approximately +10%. This shows

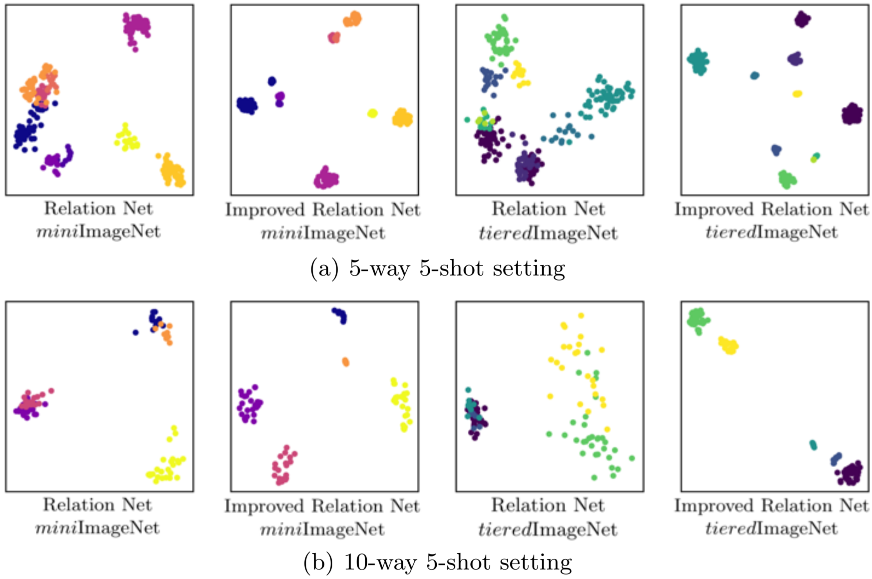

Fig. 3. The t-SNE visualization [22] of the improved feature embeddings learned by our proposed approach..

the ability of the proposed capsule network-based embedding network to improve the performance of the metric based method. Fig. 3 visualizes the feature distribution using t-SNE [22]. The features computed in 5-way 5-shot setting and 10-way 5-shot setting. As can be clearly observed, the improved Relation Net model has more compact and separable clusters, indicating that features are more discriminative for the task. This descends from the design of the embedding module.

5 Conclusion

In this paper, we proposed a new few-shot learning scheme aiming to improve the metric learning-based prototypical network. Our proposed scheme has the following novel characteristics: (1) the new embedding space created by a capsule network, which is unique in its capability to encode the relative spatial relationship between features. The network is trained with a novel triple-loss designed to learn the embedding space; (2) an effective and robust non-parameter classification scheme, named attentive prototypes, to replace the simple feature average for prototypes. The instances from the support set are taken into account to generate prototypes, with their importance being calculated by the reconstruction error for a given query. Experimental results showed that the proposed method outperforms other few-shot learning algorithms on all of miniImageNet, tieredImageNet and Fewshot-CIFAR100 datasets.

References

1. Arik, S.O., Pfister, T.: Attention-based prototypical learning towards interpretable, confident and robust deep neural networks. arXiv preprint arXiv:1902.06292 (2019)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
3. Ding, X., Wang, N., Gao, X., Li, J., Wang, X.: Group reconstruction and max-pooling residual capsule network. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 2237–2243 (2019)
4. Fe-Fei, L., et al.: A bayesian approach to unsupervised one-shot learning of object categories. In: IEEE InternationalConference on Computer Vision (ICCV). pp. 1134–1141 (2003)
5. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence (TPAMI) **28**(4), 594–611 (2006)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML). pp. 1126–1135 (2017)
7. Fukunaga, K.: Introduction to statistical pattern recognition. Elsevier (2013)
8. Gao, T., Han, X., Liu, Z., Sun, M.: Hybrid attention-based prototypical networks for noisy few-shot relation classification. In: AAAI Conference on Artificial Intelligence (AAAI) (2019)
9. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043 (2017)
10. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4367–4375 (2018)
11. Gidaris, S., Komodakis, N.: Generating classification weights with gnn denoising autoencoders for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
13. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: International Conference on Artificial Neural Networks. pp. 44–51. Springer (2011)
14. Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling graph neural network for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11–20 (2019)
15. Kosiorek, A.R., Sabour, S., Teh, Y.W., Hinton, G.E.: Stacked capsule autoencoders. arXiv preprint arXiv:1906.06818 (2019)
16. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NIPS). pp. 1097–1105 (2012)
18. Lenssen, J.E., Fey, M., Libuschewski, P.: Group equivariant capsule networks. In: Advances in neural information processing systems (NIPS). pp. 8844–8853 (2018)
19. Li, H., Eigen, D., Dodge, S., Zeiler, M., Wang, X.: Finding task-relevant features for few-shot learning by category traversal. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–10 (2019)

20. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-sgd: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835 (2017)
21. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Transductive propagation network for few-shot learning. In: International Conference on Learning Representations (ICLR) (2019)
22. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
23. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. arXiv preprint arXiv:1707.03141 (2017)
24. Mitani, Y., Hamamoto, Y.: A local mean-based nonparametric classifier. Pattern Recognition Letters **27**(10), 1151–1159 (2006)
25. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
26. Oreshkin, B., López, P.R., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. In: Advances in neural information processing systems (NIPS). pp. 721–731 (2018)
27. Peng, H., Li, J., Gong, Q., Wang, S., He, L., Li, B., Wang, L., Yu, P.S.: Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. arXiv preprint arXiv:1906.04898 (2019)
28. Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7229–7238 (2018)
29. Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R.: Deepcaps: Going deeper with capsule networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10725–10733 (2019)
30. Rawlinson, D., Ahmed, A., Kowadlo, G.: Sparse unsupervised capsules generalize better. arXiv preprint arXiv:1804.06094 (2018)
31. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. In: International Conference on Learning Representations (ICLR) (2018)
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
33. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: International Conference on Learning Representations (ICLR) (2018)
34. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in neural information processing systems (NIPS). pp. 3856–3866 (2017)
35. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015)
36. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in neural information processing systems (NIPS). pp. 4077–4087 (2017)
37. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Advances in neural information processing systems (NIPS). pp. 1857–1865 (2016)
38. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 403–412 (2019)
39. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1199–1208 (2018)

40. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–9 (2015)
41. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems (NIPS). pp. 3630–3638 (2016)
42. Xia, C., Zhang, C., Yan, X., Chang, Y., Yu, P.S.: Zero-shot user intent detection via capsule neural networks. arXiv preprint arXiv:1809.00385 (2018)
43. Zhang, W., Tang, P., Zhao, L.: Remote sensing image scene classification using cnn-capsnet. Remote Sensing **11**(5), 494 (2019)
44. Zhang, X., Zhao, S.G.: Cervical image classification based on image segmentation preprocessing and a capsnet network model. International Journal of Imaging Systems and Technology **29**(1), 19–28 (2019)
45. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3d point capsule networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1009–1018 (2019)