ECKPN: Explicit Class Knowledge Propagation Network for Transductive Few-shot Learning

Chaofan Chen[1], Xiaoshan Yang[2,3], Changsheng Xu[2,3†], Xuhui Huang[4], Zhe Ma[4]
[1]School of Information Science and Technology, University of Science and Technology of China (USTC)
[2]National Lab of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA)
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[4]X Lab, The Second Academy of CASIC, Beijing China
chencfbupt@gmail.com, {xiaoshan.yang,csxu}@nlpr.ia.ac.cn, {starhxh,mazhe_thu}@126.com

Abstract

Recently, the transductive graph-based methods have achieved great success in the few-shot classification task. However, most existing methods ignore exploring the class-level knowledge that can be easily learned by humans from just a handful of samples. In this paper, we propose an Explicit Class Knowledge Propagation Network (ECKPN), which is composed of the comparison, squeeze and calibration modules, to address this problem. Specifically, we first employ the comparison module to explore the pairwise sample relations to learn rich sample representations in the instance-level graph. Then, we squeeze the instance-level graph to generate the class-level graph, which can help obtain the class-level visual knowledge and facilitate modeling the relations of different classes. Next, the calibration module is adopted to characterize the relations of the classes explicitly to obtain the more discriminative class-level knowledge representations. Finally, we combine the class-level knowledge with the instance-level sample representations to guide the inference of the query samples. We conduct extensive experiments on four few-shot classification benchmarks, and the experimental results show that the proposed ECKPN significantly outperforms the state-of-the art methods.

1. Introduction

Recent deep learning methods rely on a large amount of labeled data to achieve high performance, which may have problems in some scenarios, where the cost of data collection is high, and thus it is difficult to obtain a large amount of labeled data. The learning schema of these deep methods is different from that of humans. After being exposed to a few data/samples, human beings can use their prior knowledge to learn quickly so as to successfully recognize new classes. Therefore, how to reduce the gap between deep learning methods and human learning abilities has aroused the interest of many researchers. Few-shot learning [17, 21, 45], which simulates the human learning schema, has attracted much attention in the field of computer vision and machine learning.

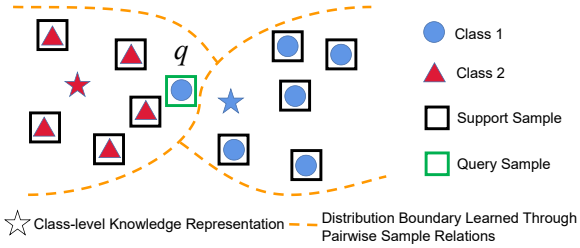

Figure 1. An illustration of the role of class-level knowledge representations (e.g., class centers).

As a straightforward method to solve the few-shot learning task, traditional fine-tuning techniques [10] can utilize the samples of the new classes to update the parameters of the network pretrained on the classes with sufficient samples. However, these methods always lead to over-fitting, since only a few training samples are not enough to represent the data distributions of the corresponding classes and learn effective classifiers. A successful attempt to solve the over-fitting problem is to apply the meta-learning mechanism [20] in few-shot learning task. The meta-learning based methods [3, 43, 30, 31, 23, 37, 7, 28, 12, 11, 52, 47, 2, 29] are composed of two steps: meta-train and meta-test. Each step (meta-train or meta-test) consists of multiple episodes (sub-tasks), and the data of each episode are composed of support set and query set. These methods keep the meta-train environment consistent with the meta-test to help improve the generalization ability of the models, thereby solving the problem of over-fitting. Nowadays, meta-learning has become a general training mechanism in most of the few-shot learning methods. In this paper, we also follow this training mechanism.

† indicates corresponding author: Changsheng Xu.

Recently, inspired by the success of graph networks in modeling structure information [14, 8, 42], researchers began to propose the graph-based meta-learning approaches for few-shot learning and obtain the state-of-the-art performances [38, 12, 28, 29, 47, 27]. These methods treat the samples as nodes to construct the graph and utilize the adjacency matrix to model the relations of images. There are two settings of the graph-based meta-learning approaches: transductive setting and inductive setting. The transductive methods characterize the relations of samples from both the support set and the query set for joint prediction, and thus obtain better performances than inductive methods, which can only learn a network based on the relations of support samples and classify each query sample individually.

Existing transductive graph-based methods learn to propagate the class label from the support set to the query set by comprehensively considering the instance-level relations of samples. However, these methods ignore the global context knowledge from the perspective of a category. In contrast, people can learn richer representations of a new category from just a handful of samples, using them for creating new exemplars, and even creating new abstract categories based on existing categories [18]. **This inspires us to consider how to explicitly learn the richer class knowledge to guide the graph-based inference of query samples.** As illustrated in Figure 1, if we only utilize the sample representations and relations to conduct the few-shot classification task, we may misclassify the query sample q into class 2. However, if we learn the class-level knowledge representations explicitly to guide the inference procedure, we can classify q correctly, because q is closer to the representation of the class 1.

In order to address the above problem, we propose an end-to-end transductive graph neural network, which is called Explicit Class Knowledge Propagation Network (ECKPN). The proposed ECKPN is composed of the **comparison**, **squeeze** and **calibration** modules, which can be flexibly stacked **to explicitly learn and propagate the class-level knowledge.** **(1)** Firstly, the comparison module captures the rich representations of samples based on the pairwise relations in a instance-level graph. The visual features are always structured vectors and many factors (e.g., frequency, shapes, illumination, textures) could lead to grouping [46, 51] (i.e., a group of dimensions represents a semantic aspect or a piece of knowledge). Thus, we adopt multi-head relations in the message passing of the comparison module to characterize the group-wise relations of samples, which provides fine-grained comparison of different samples. Each node feature is divided into groups along the dimension, and adjacency matrices are computed for different groups to obtain multiple relation measurements, which are then aggregated to compute the new node features of the samples. **(2)** Then, the squeeze module explores the intra-class context knowledge by clustering samples with similar features from the instance-level graph, which results in a class-level graph. The number of nodes in the class-level graph is same as the total number of classes. Thus, each node represents the visual knowledge of a specific class. **(3)** Finally, the calibration module explicitly captures the relationships between different classes and learn more discriminative class knowledge to guide the graph-based inference of query samples. Since the word embeddings of the class names can provide rich semantic knowledge that may not be contained in the visual contents, we combine them with the visual knowledge to obtain the multi-modal knowledge representations of different classes. Based on the multi-modal knowledge representations, a class-level message passing is adopted to exploit the relationship of different classes. The new class-level knowledge representations obtained by message passing are combined with the corresponding instance-level sample representations to guide the inference of the query samples.

To sum up, the main contributions of this paper are fourfold:

- To the best of our knowledge, we are the first to propose an end-to-end graph-based few-shot learning architecture, which can explicitly learn the rich class knowledge to guide the graph-based inference of query samples.
- We build multi-head sample relations to explore the fine-grained comparison of pairwise samples, which can facilitate the learning of richer class knowledge based on the pairwise relations.
- We leverage the semantic embeddings of the class names to construct the multi-modal knowledge representations of different classes, which can provide more discriminative knowledge to guide the inference of the query samples.
- We conduct extensive experiments on four benchmarks (i.e., miniImageNet, tieredImageNet, CIFAR-FS and CUB-200-2011) for the transductive few-shot classification task, and the results show that the proposed method achieves the state-of-the-art performances.

2. Related Work

In recent years, researchers have proposed many novel approaches to address the few-shot learning problems and achieved great success. As illustrated in [7], we can divide the existing few-shot learning methods into two categories: gradient-based [3, 37, 31, 11, 30, 19, 26, 40, 53, 16, 49, 4] and metric-based [41, 39, 43, 38, 52, 28, 29, 23, 47, 27, 12, 7, 50, 32, 25].

Gradient-based Approaches. These approaches try to adapt to new classes within a few optimization steps. The well-known model-agnostic meta-learning [3] (MAML)

method relies on the meta-learner [20] to realize the fine-tune updates. Reptile [31] is a first-order gradient-based meta-learning approach, which points out that MAML can be simply implemented. It is trained on the sampled tasks and does not need a training-testing split for each task. Latent embedding optimization [37] (LEO) is an encoder-decoder architecture, which utilizes the encoder to explore the low-dimensional latent embedding space for updating the representations and the decoder to predict the high-dimensional parameters. Conditional class-aware meta-learning [11] (CAML) conditionally transforms embeddings to explore the inter-class dependencies. However, these gradient-based approaches usually fail to learn the effective sample representations for inference.

Metric-based Approaches. These methods usually embed the support and query samples into the same feature space at first, and then compute the similarity of features for prediction. Relation Networks [41] exploit the pair-wise relations between support samples and query samples using the distance metric network. Matching Networks [43] combine the attention mechanism and memory together to present an end-to-end differentiable nearest-neighbor classifier. Prototypical Networks [39] leverage the mean of the sample features of each class to build the prototype representations at first, and then compute the similarity between the query sample representation and the prototype representation for inference. Recently, task-dependent adaptive metric [32] (TADAM) and task-adaptive projection network [50] (TapNet) have been proposed to explore the task-dependent metric space to enhance the performance of existing few-shot models.

The core of the metric-based approaches is exploring the relations between the query samples and the support samples/classes. Inspired by the success of graph neural networks (GNNs) [14, 8, 42] on modeling the relationships and propagating information among points, researchers proposed many graph-based methods [38, 12, 28, 29, 47, 27] to conduct few-shot learning tasks and have achieved great success. For example, GNN-FSL [38] is the first work to build an end-to-end trainable graph neural network architecture to conduct the few-shot classification task. Transductive Propagation Network [28] (TPN) is the first to employ the GNNs to conduct the transductive inference. It utilizes a closed-form solution to perform iterative label propagation. Edge-Labeling Graph Neural Network [12] (EGNN) exploits the similarity/dissimilarity between nodes to dynamically update edge-labels. Transductive relation-propagation graph neural network (TRPN) explicitly considers the relations of support-query pairs for few-shot learning. Recent distribution propagation graph network (DPGN) [47] builds a dual graph to model the distribution-level relations of samples and outperforms most existing methods in the classification task. However, existing graph-based methods ignore exploring the class-level knowledge explicitly, which may limit their inference ability as illustrated in Figure 1.

3. Method

3.1. Problem Statement

As illustrated in Section 1, we utilize the meta-learning mechanism to conduct the few-shot classification task. For each episode in the meta-train, we sample N classes from C_{train} (the class set of the training data D_{train}) to construct the support and query set. The support set $S \subset D_{train}$ contains K samples for each class (i.e., the N-way K-shot setting), which can be denoted as $S = \{(x_1, y_1), (x_2, y_2), ..., (x_{N\times K}, y_{N\times K})\}$, where x_i represents the i-th sample and y_i denotes the label of x_i. The query set includes T samples from the N classes in total, which can be denoted as $Q = \{(x_{N\times K+1}, y_{N\times K+1}), ..., (x_{N\times K+T}, y_{N\times K+T})\}$.

For the transductive setting, we need to train a classification model which can leverage the $N \times K$ labeled support samples and the T unlabeled query samples to correctly predict the labels of the T query samples. The training procedure is employed episode by episode until convergence. Given the test data set D_{test} and its corresponding class set C_{test}, we construct the support and query set for the episode (in the meta-test) in a similar way as in the meta-train. Note that $C_{train} \cap C_{test} = \varnothing$. In the meta-test, we utilize the model learned in the meta-train to predict the labels of the query set samples. The prediction/classification results are used to estimate the effectiveness of the model.

Notations. In this paper, $X_{i;m}$ denotes the m-th row of the matrix X_i and $X_{i;m,n}$ denotes the element located in the m-th row and n-th column of the matrix X_i.

3.2. Explicit Class Knowledge Propagation Network

In this section, we introduce the technical details of the proposed Explicit Class Knowledge Propagation Network (ECKPN). As illustrated in Figure 2, we first utilize the support and query samples to build an instance-level graph. Then, we leverage the comparison module to update the sample representations based on the pairwise node relations in the instance-level graph. In this module, we construct the multi-head relations to help model the fine-grained relations of the samples to learn rich sample representations. Next, we squeeze the instance-level graph to the class-level graph to explore the class-level visual knowledge explicitly. In the calibration module, we perform the class-level message passing operation based on the relationships of the classes to update the class-level knowledge representations. Since the semantic word embeddings of the classes can provide rich prior knowledge, we combine them with the class-level visual knowledge to construct the multi-modal class knowledge representations before the message passing of the calibration module. Finally, the class-level knowledge

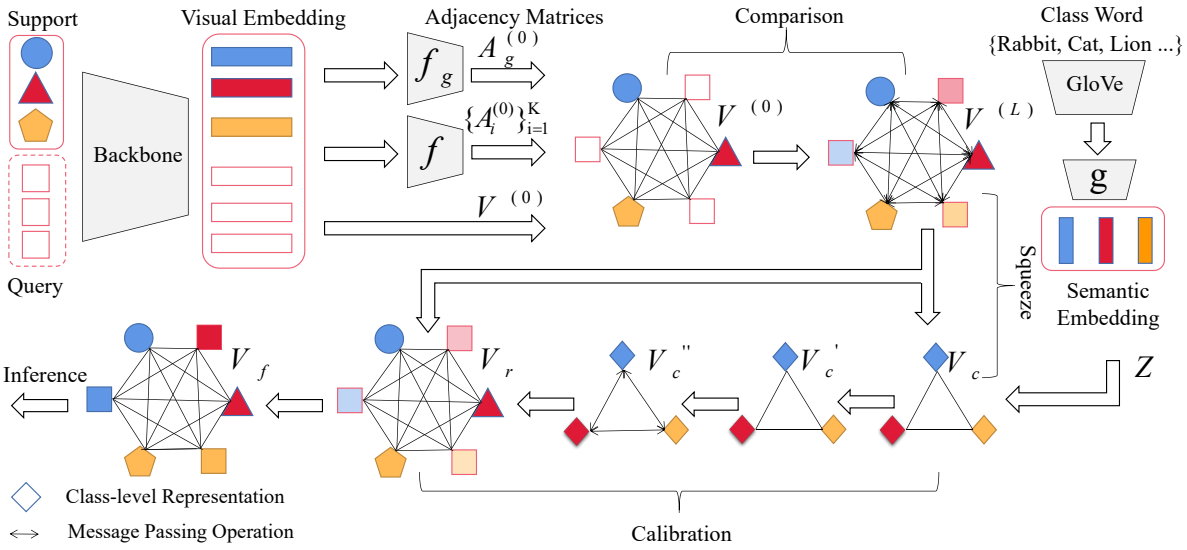

Figure 2. The overall framework of our proposed ECKPN. We take the 3-way 1-shot classification task as an example in this figure. Our ECKPN is composed of the comparison, squeeze and calibration modules, which can learn and propagate the class-level knowledge explicitly. Note that our comparison module contains L message passing layers, but we just illustrate one layer for simplicity.

representations are combined with the instance-level sample representations to guide the inference of the query samples.

3.2.1 Comparison Module: Instance-level Message Passing with Multi-head Relations

For an image i, we employ a deep CNN model as backbone to extract its d-dimensional visual feature $v_i^{(0)} \in R^d$. In each episode, we treat the support and query set samples as nodes to build the graph $G = (V^{(0)}, A^{(0)})$, where $V^{(0)}$ is the initial node feature matrix and $A^{(0)}$ is the initial adjacency matrix set which represents the sample relations. As illustrated in [46, 51], the visual features always contain some concepts that could lead to grouping, i.e., the feature dimensions from the same group represent similar knowledge. However, existing graph-based few-shot learning methods usually directly utilize the global visual features to compute the similarities of the samples to construct the adjacency matrix, which cannot characterize the fine-grained relations well. In this paper, we separate the visual features into K chunks (i.e., $V^{(l)} = [V_1^{(l)}, V_2^{(l)}, ..., V_K^{(l)}] \in R^{r \times d}$) and compute the similarities in each chunk to explore the multi-head relations of samples (i.e., K adjacency matrices $A_1^{(l)}, A_2^{(l)}, ..., A_K^{(l)} \in R^{r \times r}$), where r denotes the number of samples in each episode, $[*, *]$ denotes the concatenation operation and l denotes that the matrix is generated in the l-th graph layer. Note that each chunk $V_i^{(l)}$ has the dimension of d/K. We also compute the global relation matrix $A_g^{(l)} \in R^{r \times r}$ based on the unchunked visual features.

We utilize the global ($A_g^{(l)}$) and multi-head ($\{A_i^{(l)}\}_{i=1}^K$) relations jointly (i.e., $A^{(l)} = \{A_g^{(l)}, A_1^{(l)}, ..., A_K^{(l)}\}$) to propagate the information in the instance-level graph to update the sample representations. In this way, we can explore the relations of samples more sufficiently and learn richer sample representations. In the l-th layer, we leverage the updated sample representations $V^{(l)}$ to construct the new adjacency matrices $A_g^{(l)}$ and $A_i^{(l)}$ as follows:

$$A_{g;m,n}^{(l)} = f_g((V_m^{(l)} - V_n^{(l)})^2), \quad A_{i;m,n}^{(l)} = f_i((V_{i;m}^{(l)} - V_{i;n}^{(l)})^2) \tag{1}$$

where V_m denotes the visual feature of the m-th image, $V_{i;m}$ denotes the i-th chunk of V_m, and $(*)^2$ denotes the element-wise square operation. $f_i : R^{d/K} \rightarrow R^1$ and $f_g : R^d \rightarrow R^1$ are the mapping functions.

Inspired by the recent success of TRPN [29] in the few-shot classification task, we utilize the following matrix to mask the adjacency matrix:

$$M_{m,n} = \begin{cases} -1 & if\ m, n \in S\ and\ y_m \neq y_n \\ 1 & otherwise \end{cases} \tag{2}$$

where m and n are the samples in $S \cup Q$ and y_m is the label of sample m. This ensures that, for two samples from different categories, the higher the feature similarity of them, the more commonality decreases in the message passing process. For the two samples from the same category, the results are exactly the opposite.

In the l-th layer, we utilize the $A^{(l-1)}$, $V^{(l-1)}$ and M to generate the $V^{(l)}$ as follows:

$$V^{(l)} = Tr([\|_{i=1}^K ((A_i^{(l-1)} \odot M)V_i^{(l-1)}), (A_g^{(l-1)} \odot M)V^{(l-1)}]) \tag{3}$$

where $\|$ denotes the concatenation operation, $\odot$ denotes the element-wise multiplication operation and Tr denotes the transformation function: $R^{r \times 2d} \rightarrow R^{r \times d}$. We repeat the above message passing L times and obtain the new sample features $V^{(L)}$ which will be used in the squeeze module.

3.2.2 Squeeze Module: Class-level Visual Knowledge Learning

In order to obtain the class-level knowledge representations, we squeeze the instance-level graph to generate the class-level graph, where the nodes represent the visual knowledge of the classes. For instance, we squeeze the nodes in the instance-level graph into 5 clusters/nodes so as to obtain the visual knowledge of the classes for the 5-way classification task. Specifically, we first utilize the ground truth to supervise the assignment matrix generation, and then squeeze samples according to the assignment matrix to obtain the class-level knowledge representations $V_c \in R^{r_1 \times d}$, where r_1 denotes the number of classes in each episode.

In this paper, we feed $V^{(L)}$ and $A_g^{(L)}$ into the standard graph neural network [14] to compute the assignment matrix $P \in R^{r \times r_1}$ for simplicity.

$$P = softmax((A_g^{(L)} \odot M)V^{(L)}W) \tag{4}$$

where $W \in R^{d \times r_1}$ denotes the trainable weight matrix and the $softmax$ operation is applied in a row-wise fashion. Each element P_{uv} in the assignment matrix P represents the probability that node u in the original graph is allocated to node v in the class-level graph. After generating the assignment matrix P, we utilize the following equation to generate the initial class-level knowledge representations:

$$V_c = P^T V^{(L)} \tag{5}$$

where T denotes transpose operation. In the class-level graph, each node feature can be considered as the weighted sum of the node features with the same label in the instance-level graph. In this way, we obtain the class-level visual knowledge representations, which will facilitate modeling the relations of different classes in the calibration module.

3.2.3 Calibration Module: Class-level Message Passing with Multi-modal Knowledge

Since the class word embeddings can provide the information that may not be contained in the visual content, we combine them with the generated class-level visual knowledge to construct the multi-modal knowledge representations. Specifically, we first leverage the GloVe (pretrained on a large text corpora with self-supervised constraint) [33] to obtain the d_1-dimensional semantic embeddings of class labels. The Common Crawl version of the GloVe is used in this paper, which is trained on 840B tokens. More details can be found in [33]. After obtaining the word embeddding $e_i \in R^{d_1}$ of the i-th class, we employ a mapping network $g : R^{d_1} \rightarrow R^d$ to map it into a semantic space which has the same dimension with the visual knowledge representation, i.e., $z_i = g(e_i) \in R^d$. Finally, we obtain the multi-modal class representations as follows:

$$V_c^{'} = [V_c, Z] \tag{6}$$

where $Z \in R^{r_1 \times d}$ is the matrix of semantic word embeddings. In this way, we can obtain richer class-level knowledge representations.

The adjacency matrix (A_c) of the class-level graph represents the relations of the class representations and its value denotes the connectivity strength of the class pairs. In this paper, we leverage the following equations to compute the adjacency matrix A_c and the new class-level knowledge representations $V_c^{''}$.

$$A_c = P^T A_g P, \quad V_c^{''} = A_c V_c^{'} W^{'} \tag{7}$$

where $W^{'} \in R^{2d \times 2d}$ is a trainable weight matrix. In order to make each sample contain the corresponding class knowledge learned in (7), we utilize the assignment matrix to map the class knowledge back to the instance-level graph as follows:

$$V_r = PV_c^{''} \tag{8}$$

where $V_r \in R^{r \times 2d}$ denotes the refined features. Finally, we combine V_r with $V^{(L)}$ by concatenation to generate the sample representations V_f for query inference.

3.3. Inference

To infer the class labels of query samples, we utilize V_f to compute the corresponding adjacency matrix A_f as follows:

$$A_{f;m,n} = f_l((V_{f;m} - V_{f;n})^2) \tag{9}$$

where $V_{f;m}$ and $V_{f;n}$ are the representations of the m-th sample and the n-th sample, respectively. $f_l : R^{3d} \rightarrow R^1$ is a mapping function. For each query sample, we leverage the class labels of the support samples to predict its label:

$$\widetilde{y_v} = softmax\Big(\sum_{u=1}^{N \times K} A_{f;u,v} \cdot one\text{-}hot(y_u)\Big) \tag{10}$$

where one-hot denotes the one-hot encoder.

3.4. Loss Function

The overall framework of the proposed ECKPN can be optimized in an end-to-end form by the following loss function:

$$\mathcal{L} = \lambda_0 \mathcal{L}_0 + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 \tag{11}$$

where λ_0, λ_1 and λ_2 are hyper-parameters that are set to 1.0, 0.5 and 1.0 in the experiment. $\mathcal{L}_0$, $\mathcal{L}_1$ and $\mathcal{L}_2$ are adjacency loss, assignment loss and classification loss respectively, that will be introduced as follows.

Adjacency Loss: As illustrated in Section 3.2.1, for each graph network layer $l \in \{1, ..., L\}$ in the comparison module, we have multiple adjacency matrices $A_g^{(l)}$ and $\{A_i^{(l)}\}_{i=1}^K$ for message passing between support and query samples. In addition, we have the adjacency matrix A_f for query inference in Section 3.3. To ensure these adjacency matrices to be able to capture the correct sample relations, we use the following loss function:

$$\mathcal{L}_0 = -\sum_{A_* \in A_s} \Big(\frac{sum(A_* H G_t)}{sum(H G_t)} + \frac{sum((1 - A_*)H(1 - G_t))}{sum(H(1 - G_t))}\Big) \tag{12}$$

where $A_s = \{A_g^{(1)}, ..., A_g^{(L)}\} \cup \{A_f\} \cup \{A_i^{(1)}, ..., A_i^{(L)}\}_{i=1}^{K}$ and $sum(*)$ denotes the sum of all elements in the matrix. $H \in R^{r \times r}$ is the query mask and $G_t \in R^{r \times r}$ is the ground truth matrix which are defined as follows:

$$H_{m,n} = \begin{cases} 0 & if\ m \in S \\ 1 & otherwise \end{cases}, \quad G_{t;m,n} = \begin{cases} 1 & if\ y_m = y_n \\ 0 & otherwise \end{cases} \tag{13}$$

where m and n denote the nodes in the graph.

Assignment Loss: To ensure that the assignment matrix P computed in the squeeze module (illustrated in Section 3.2.2) can correctly cluster the samples with the same label, we utilize the following cross-entropy loss function:

$$\mathcal{L}_1 = \mathcal{L}_{ce}(P, one\text{-}hot([C_s, C_q])) \tag{14}$$

where $one\text{-}hot([C_s, C_q])$ denotes the ground truth one-hot class vectors of the support and query samples.

Classification Loss: To constrain that the proposed ECKPN can predict the correct query labels, we use the following loss function:

$$\mathcal{L}_2 = \sum_{v \in Q} \mathcal{L}_{ce}(\widetilde{y_v}, y_v) \tag{15}$$

where $\mathcal{L}_{ce}$ denotes the cross-entropy loss function.

4. Experiments

4.1. Datasets

MiniImageNet [43] and tieredImageNet [35] are two popular few-shot benchmarks derived from the ILSVRC-12 dataset [36]. The miniImageNet contains 100 classes with 600 images per class. Each image is RGB-colored and has the size of 84×84. The tieredImageNet contains 779165 images of size 84×84 sampled from 608 classes. CIFAR-FS [1] is reorganized from the CIFAR-100 dataset for the few-shot classification task. It contains 100 classes with 60000 images in total. Each image has the size of 32×32. CUB-200-2011 [44] is a medium-scale dataset used for fine-grained classification. It has 11788 images of size 84×84 from 200 bird categories. We follow the popularly used train/val/test setting proposed in [35, 36, 1, 47, 29]. The statistics of these benchmarks are shown in Table 1.

Dataset	Classes	Images	Train/Val/Test
miniImageNet	100	60000	64/16/20
tieredImageNet	608	779165	351/97/160
CIFAR-FS	100	60000	64/16/20
CUB-200-2011	200	11788	100/50/50

Table 1. The statistics of the four few-shot classification benchmarks.

4.2. Experimental Setup

Architectures. We utilize two popular backbones (Conv-4 [3, 12] and ResNet-12 [9, 30, 47]) to encode the input images into 128 dimensions. Both Conv-4 and ResNet-12 consist of four blocks. Each block in Conv-4 is composed of 3×3 convolutions, a batch normalization (BN) and a LeakyReLU activation. Each residual block in ResNet-12 contains 3 convolutional layers with the size of 3×3. Each convolutional layer is followed by a 2×2 max-pooling layer. A global average-pooling is applied in the end of the fourth block. Before feeding the images to the backbones, we follow the recent few-shot learning approaches [5, 48, 47] to perform data augmentation, i.e., color jittering, random crop and horizontal flip. Note that the mapping and transformation functions f_i, f_g, f_l and T_r are single-layer convolutional networks with batch normalization and LeakyReLU.

Training. We train our model on miniImageNet, tieredImageNet, CIFAR-FS and CUB-200-2011 for 200K, 200K, 100K and 100K iterations respectively. In each iteration, we construct 28 episodes for meta-training. Adam optimizer [13] is used in all experiments with the initial learning rate 0.001. We set the weight decay to 1e-5 and decay the learning rate by 0.1 every 15K iterations.

Evaluation. We conduct the 5-way 1-shot and 5-shot experiments on the four benchmarks for few-shot classification task. We follow [47, 29] to construct 10K episodes in the meta-test and report the mean prediction accuracy of them to measure the effectiveness of the proposed ECKPN.

4.3. Classification Results

We compare the classification results of the proposed ECKPN with the recent state-of-the-art few-shot methods and report the classification results of the 5-way 1-shot and 5-shot under different backbones (Conv-4 and ResNet-12) in Table 2, 3 and 4. From these experimental results, we have the following observations. (1) The proposed ECKPN achieves the state-of-the-art classification results compared with the recent methods on all four benchmarks for both the 5-shot and 1-shot setting, which demonstrates the effectiveness of our model. Especially for the 1-shot setting on the miniImageNet dataset, the proposed method equipped with the Conv-4 and ResNet-12 achieves improvement of 2.88% and 2.71% respectively compared with the second-best approach DPGN. These results demonstrate the necessity of modeling the class-level knowledge in the few-shot classification task. (2) The proposed method achieves more improvements in the 1-shot setting than in the 5-shot setting. Since the number of samples in the 5-shot setting is larger than in the 1-shot setting, with more samples, the recent graph-based methods can adapt to the novel classes better based on only the sample relations, which reduces the performance gain of our ECKPN. However, our ECKPN can still achieve the improvements of 0.7%-0.8% under the 5-shot setting on all four benchmarks.

4.4. Semi-supervised Classification Results

In this part, we apply the proposed ECKPN in the semi-supervised classification task to further evaluate its generalization ability. Specifically, we follow [12, 28] to partially label the support samples with different ratios (i.e., 20%, 40%, 60% and 100%). The label ratio 20% means that 20% labeled and 80% unlabeled support samples are used

Method	Backbone	5way-1shot	5way-5shot
MatchingNet [43]	Conv-4	43.56 ±0.84	55.31 ±0.73
ProtoNet [39]	Conv-4	49.42 ±0.78	68.20 ±0.66
RelationNet [41]	Conv-4	50.44 ±0.82	65.32 ±0.70
Dynamic [5]	Conv-4	56.20 ±0.86	71.94 ±0.57
Reptile [31]	Conv-4	49.97 ±0.32	65.99 ±0.58
MAML [3]	Conv-4	48.70 ±1.84	55.31 ±0.73
Meta-SGD [26]	Conv-4	50.47 ±1.87	64.03 ±0.94
GNN-FSL [38]	Conv-4	50.33 ±0.36	66.41 ±0.63
TPN [28]	Conv-4	55.51 ±0.86	69.86 ±0.65
EGNN [12]	Conv-4	-	76.34 ±0.48
TRPN [29]	Conv-4	57.84 ±0.51	78.57 ±0.44
DPGN [47]	Conv-4	66.01 ±0.36	82.83 ±0.41
ECKPN	**Conv-4**	**68.89**±0.34	**83.59**±0.44
LEO [37]	Others	61.76 ±0.08	77.59 ±0.12
CloserLook [15]	Others	51.75 ±0.80	74.27 ±0.63
CTM [22]	Others	62.05 ±0.55	78.63 ±0.06
wDAE [6]	Others	61.07 ±0.15	76.75 ±0.11
AWGIM [7]	Ohers	63.12 ±0.08	78.40 ±0.11
AFHN [23]	Others	62.38 ±0.72	78.16 ±0.56
FEAT [48]	ResNet-12	62.96 ±0.02	78.49 ±0.02
TADAM [32]	ResNet-12	58.50 ±0.30	76.70 ±0.30
TapNet [50]	ResNet-12	61.65 ±0.15	76.36 ±0.10
MataGAN [53]	ResNet-12	52.71 ±0.64	68.63 ±0.67
Shot-Free [34]	ResNet-12	59.04 ±0.43	77.64 ±0.39
SNAIL [30]	ResNet-12	55.71 ±0.99	68.88 ±0.92
MTL [40]	ResNet-12	61.20 ±1.80	75.53 ±0.80
MetaOptNet [19]	ResNet-12	62.64 ±0.61	78.63 ±0.46
DeepEMD [52]	ResNet-12	65.91 ±0.82	82.41 ±0.56
DPGN [47]	ResNet-12	67.77 ±0.32	84.60 ±0.43
ECKPN	**ResNet-12**	**70.48**±0.38	**85.42**±0.46

Table 2. Few-shot classification accuracies (%) on miniImageNet.

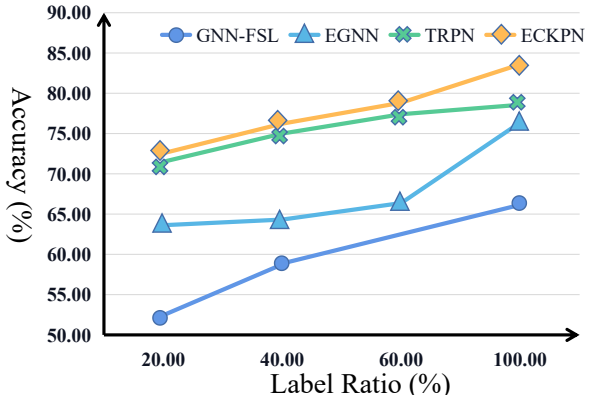

Figure 3. Semi-supervised few-shot classification accuracies (%) in 5-way 5-shot on miniImageNet.

to train the model in each episode. We compare the proposed ECKPN with the recent GNN-FSL [38], EGNN [12] and TRPN [29] equipped with Conv-4. We show the 5-way 5-shot classification results in Figure 3. As shown, the proposed ECPKN achieves better performances than existing methods under all label ratios, which demonstrates the effectiveness of capturing the class-level knowledge to guide the inference of the query samples.

Method	Backbone	5way-1shot	5way-5shot
MatchingNet [43]	Conv-4	54.02 ±0.00	70.11 ±0.00
ProtoNet [39]	Conv-4	53.31 ±0.89	72.69 ±0.74
RelationNet [41]	Conv-4	54.48 ±0.93	71.32 ±0.70
Reptile [31]	Conv-4	52.36 ±0.23	71.03 ±0.22
MAML [3]	Conv-4	51.67 ±1.81	70.30 ±0.08
Meta-SGD [26]	Conv-4	62.95 ±0.03	79.34 ±0.06
GNN-FSL [38]	Conv-4	43.56 ±0.84	55.31 ±0.73
TPN [28]	Conv-4	57.53 ±0.96	72.85 ±0.74
EGNN [12]	Conv-4	-	80.15 ±0.30
TRPN [29]	Conv-4	59.26 ±0.50	79.66 ±0.45
DPGN [47]	Conv-4	69.43 ±0.49	85.92 ±0.42
ECKPN	**Conv-4**	**70.45** ±0.48	**86.74**±0.42
wDAE [6]	Others	68.18 ±0.16	83.09 ±0.12
CTM [22]	Others	64.78 ±0.11	81.05 ±0.13
LEO [37]	Others	66.33 ±0.05	81.44 ±0.09
AWGIM [7]	Others	67.69 ±0.11	82.82 ±0.13
MetaOptNet [19]	ResNet-12	65.81 ±0.74	81.75 ±0.53
TapNet [50]	ResNet-12	63.08 ±0.15	80.26 ±0.12
DeepEMD [52]	ResNet-12	71.16 ±0.87	86.03 ±0.58
Shot-Free [34]	ResNet-12	66.87 ±0.43	82.64 ±0.39
DPGN [47]	ResNet-12	72.45 ±0.51	87.24 ±0.39
ECKPN	**ResNet-12**	**73.59** ±0.45	**88.13** ±0.28

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

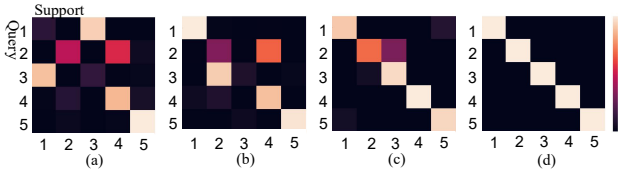

Figure 4. The visualization of the support-query similarities in 5-way 1-shot setting. (a), (b) and (c) represent the similarities of support and query samples in the first, third and last layers of the comparison module. (d) denotes the ground truth support-query similarities. The white denotes the high confidence and the black denotes the low confidence.

4.5. Ablation Studies

In this part, we conduct more experiments to analyze the impacts of the designed comparison module, squeeze module, calibration module, multi-head relations, multi-modal class representations. All experiments are conducted on the miniImageNet for the 5-way 1-shot classification task.

Impact of the comparison module. In the comparison module, we exploit L message passing layers to update the sample representations. In this part, we perform experiments to show the impact of the layer numbers. As shown in Figure 5 (b), with the increase of the layer number, the classification accuracy increases at first and then keeps stable. Therefore, we set the number of the layers to 6 (i.e., $L = 6$) in this paper. Furthermore, we visualize the similarities of support and query samples in the first, third and last layers of the comparison module in Figure 4. As shown, the proposed ECKPN can characterize the support-query similarities better with more message passing layers used in the comparison module, which qualitatively illustrates the ef-

CUB-200-2011			
Method	Backbone	5way-1shot	5way-5shot
ProtoNet [39]	Conv-4	51.31 ±0.91	70.77 ±0.69
RelationNet [41]	Conv-4	62.45 ±0.98	76.11 ±0.69
MatchingNet [43]	Conv-4	61.16 ±0.89	72.86 ±0.70
MAML [3]	Conv-4	55.92 ±0.95	72.09 ±0.76
DN4 [24]	Conv-4	53.15 ±0.84	81.90 ±0.60
CloserLook [15]	Conv-4	60.53 ±0.83	79.34 ±0.61
DPGN [47]	Conv-4	76.05 ±0.51	89.08 ±0.38
ECKPN	**Conv-4**	**77.20** ±0.36	**89.72** ±0.31
DeepEMD [52]	ResNet-12	75.65 ±0.83	88.69 ±0.50
TADAM	ResNet-12	72.00 ±0.70	84.20 ±0.50
FEAT [48]	ResNet-12	68.87 ±0.22	82.90 ±0.15
DPGN [47]	ResNet-12	75.71 ±0.47	91.48 ±0.33
ECKPN	**ResNet-12**	**77.43**±0.54	**92.21** ±0.41
CIFAR-FS			
Method	Backbone	5way-1shot	5way-5shot
ProtoNet [39]	Conv-4	55.5 ±0.7	72.0 ±0.6
RelationNet [41]	Conv-4	55.0 ±1.0	69.3 ±0.8
MAML [3]	Conv-4	58.9 ±1.9	71.5 ±1.0
R2D2 [1]	Conv-4	65.3 ±0.2	79.4 ±0.1
DPGN [47]	Conv-4	76.4 ±0.5	88.4 ±0.4
ECKPN	**Conv-4**	**77.5**±0.4	**89.1** ±0.5
DeepEMD [52]	ResNet-12	46.47 ±0.8	63.22 ±0.7
MetaOpNet [19]	ResNet-12	72.0 ±0.7	84.2 ±0.5
Shot-Free [34]	ResNet-12	69.2 ±0.4	84.7 ±0.4
DPGN [47]	ResNet-12	77.9 ±0.5	90.2 ±0.4
ECKPN	**ResNet-12**	**79.2**±0.4	**91.0** ±0.5

Table 4. Few-shot classification accuracies (%) on CUB-200-2011 and CIFAR-FS.

fectiveness of the designed comparison module.

Impact of the squeeze and calibration modules. In this paper, we design the squeeze and calibration modules to explicitly learn the class-level knowledge to guide the inference of the query samples. Therefore, it is necessary for us to quantitatively evaluate the effectiveness of these two modules in improving the classification accuracy. We list the classification results of None-Calibrate and None-Class in Table 5, where the None-Calibrate denotes the variant of our model without the calibration module, i.e., directly using the class-level knowledge generated in the squeeze module to guide the inference, and the None-Class denotes the variant of our model without the squeeze and calibration modules, i.e., directly using the pairwise relations in the comparison module for inference. Compared with the proposed ECKPN, the classification accuracy of the None-Calibrate decreases by 0.65% and 0.72% when using the backbone of Conv-4 and ResNet-12, respectively. Similarly, the classification accuracy of the None-class decreases by 1.57% and 1.36% when using the backbone of Conv-4 and ResNet-12, respectively. These results show the effectiveness of the designed squeeze and calibration modules.

Impact of the multi-head relations. To study the effects of multi-head relations, we show the classification results of the proposed ECKPN with different head numbers

Method	Conv-4	ResNet-12
None-Calibrate	68.24	69.76
Non-Class	67.32	69.12
Non-Z	68.53	69.97
Non-V	68.16	69.61
ECKPN	**68.89**	**70.48**

Table 5. The impacts of the squeeze module, the calibration module and the multi-modal class-knowledge in the proposed ECKPN.

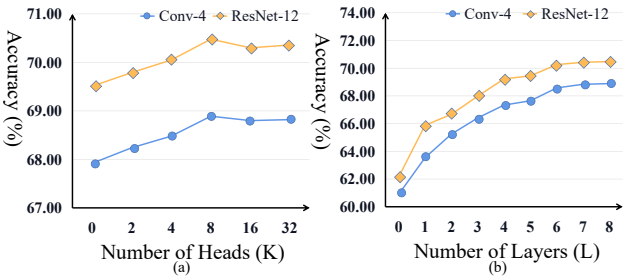

Figure 5. The classification results under different head and layer numbers in 5-way 1-shot on miniImageNet.

(i.e., K denotes the number of chunks used to separate the visual features.) in Figure 5 (a). As shown, the head number influences the classification results obviously. To trade-off between the accuracy and complexity, we build the 8-head relations for message passing in the comparison module.

Impact of the multi-modal class knowledge. To study the effect of the multi-modal class knowledge, we design two variants of our model, None-Z and None-V. The former denotes the model without using the semantic knowledge Z (i.e., $V_c^{'}$ in (6) is equal to V_c) and the latter denotes the model without using the visual knowledge V_c (i.e., $V_c^{'}$ is equal to Z). As shown in Table 5, the proposed ECKPN achieves performance gains of 0.3%-0.5% and 0.7-0.9% compared with the None-Z and None-V, which demonstrates the importance of the constructed multi-modal class-level knowledge.

5. Conclusion

In this work, we propose a novel Explicit Class Knowledge Propagation Network (ECKPN) for the transductive few-shot classification task. Our ECKPN stacks three elaborately designed modules of comparison, squeeze and calibration to explicitly explore the class-level knowledge. We leverage the generated class-level knowledge representations to guide the inference of the query samples and achieve the state-of-the-art classification performances on four benchmarks, which illustrates the effectiveness of the proposed ECKPN. In the future, we would like to extend our model for incremental few-shot learning.

Acknowledgements. This work was supported by National Key Research and Development Program of China (No. 2018AAA0100604), National Natural Science Foundation of China (No. 61832002, 61720106006, 62072455, 61721004, U1836220, U1705262, 61872424).

References

[1] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations*, 2019. 6, 8

[2] Thomas Elsken, Benedikt Staffler, Jan Hendrik Metzen, and Frank Hutter. Meta-learning of neural architectures for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12362–12372, 2020. 1

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017. 1, 2, 6, 7, 8

[4] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9537–9548, 2018. 2

[5] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 6, 7

[6] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2019. 7

[7] Yiluan Guo and Ngai-Man Cheung. Attentive weights generation for few shot learning via information maximization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13496–13505, 2020. 1, 2, 7

[8] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017. 2, 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678, 2014. 1

[11] Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. Learning to learn with conditional class dependencies. In *7th International Conference on Learning Representations*, 2019. 1, 2, 3

[12] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019. 1, 2, 3, 6, 7

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015. 6

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017. 2, 3, 5

[15] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. A closer look at feature space data augmentation for few-shot intent classification. In Colin Cherry, Greg Durrett, George F. Foster, Reza Haffari, Shahram Khadivi, Nanyun Peng, Xiang Ren, and Swabha Swayamdipta, editors, *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 1–10, 2019. 7, 8

[16] Alexandre Lacoste, Thomas Boquet, Negar Rostamzadeh, Boris N. Oreshkin, Wonchang Chung, and David Krueger. Deep prior. *CoRR*, abs/1712.05016, 2017. 2

[17] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems.*, pages 2526–2534, 2013. 1

[18] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[19] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665. Computer Vision Foundation / IEEE, 2019. 2, 7, 8

[20] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130, 2015. 1, 3

[21] Fei-Fei Li, Robert Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *IEEE International Conference on Computer Vision*, pages 1134–1141, 2003. 1

[22] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2019. 7

[23] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13467–13476, 2020. 1, 2, 7

[24] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019. 8

[25] Wenbin Li, Jinglin Xu, Jing Huo, Lei Wang, Yang Gao, and Jiebo Luo. Distribution consistency based covariance metric networks for few-shot learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 8642–8649, 2019. 2

[26] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. 2, 7

[27] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning to propagate for graph meta-learning. In *Advances in Neural Information Processing Systems*, pages 1037–1048, 2019. 2, 3

[28] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *7th International Conference on Learning Representations*, 2019. 1, 2, 3, 6, 7

[29] Yuqing Ma, Shihao Bai, Shan An, Wei Liu, Aishan Liu, Xiantong Zhen, and Xianglong Liu. Transductive relation-propagation network for few-shot learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 804–810, 2020. 1, 2, 3, 4, 6, 7

[30] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations, ICLR*, 2018. 1, 2, 6, 7

[31] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 1, 2, 3, 7

[32] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 719–729, 2018. 2, 3, 7

[33] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014. 5

[34] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 331–339, 2019. 7, 8

[35] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *6th International Conference on Learning Representations*, 2018. 6

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 211–252, 2015. 6

[37] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations*, 2019. 1, 2, 3, 7

[38] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *6th International Conference on Learning Representationss*, 2018. 2, 3, 7

[39] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 3, 7, 8

[40] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019. 2, 7

[41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2, 3, 7, 8

[42] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations*, 2018. 2, 3

[43] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. 1, 2, 3, 6, 7, 8

[44] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 6

[45] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018. 1

[46] Yuxin Wu and Kaiming He. Group normalization. *International Journal of Computer Vision*, 128(3):742–755, 2020. 2, 4

[47] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. DPGN: distribution propagation graph network for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13387–13396, 2020. 1, 2, 3, 6, 7, 8

[48] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *CoRR*, abs/1812.03664, 2018. 6, 7, 8

[49] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7343–7353, 2018. 2

[50] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7115–7123, 2019. 2, 3, 7

[51] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014. 2, 4

[52] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12200–12210, 2020. 1, 2, 7, 8

[53] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pages 2371–2380, 2018. 2, 7