Transductive Relation-Propagation Network for Few-shot Learning

Yuqing Ma[1] , **Shihao Bai**[1] , **Shan An**[3] , **Wei Liu**[1] , **Aishan Liu**[1] ,
Xiantong Zhen[4] and **Xianglong Liu**[*1,2]
[1]State Key Lab of Software Development Environment, Beihang University, China
[2]Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, China
[3]Department of Augmented Reality and Virtual Reality, JD
[4]Inception Institute of Artificial Intelligence
{mayuqing, 16061167, by1906073@, 16061175, liuaishan}@buaa.edu.cn,
zhenxt@gmail.com, xlliu@nlsde.buaa.edu.cn

Abstract

Few-shot learning, aiming to learn novel concepts from few labeled examples, is an interesting and very challenging problem with many practical advantages. To accomplish this task, one should concentrate on revealing the accurate relations of the support-query pairs. We propose a transductive relation-propagation graph neural network (TRPN) to explicitly model and propagate such relations across support-query pairs. Our TRPN treats the relation of each support-query pair as a graph node, named relational node, and resorts to the known relations between support samples, including both intra-class commonality and inter-class uniqueness, to guide the relation propagation in the graph, generating the discriminative relation embeddings for support-query pairs. A pseudo relational node is further introduced to propagate the query characteristics, and a fast, yet effective transductive learning strategy is devised to fully exploit the relation information among different queries. To the best of our knowledge, this is the first work that explicitly takes the relations of support-query pairs into consideration in few-shot learning, which might offer a new way to solve the few-shot learning problem. Extensive experiments conducted on several benchmark datasets demonstrate that our method can significantly outperform a variety of state-of-the-art few-shot learning methods.

1 Introduction

Learning novel concepts from only one or a few examples is an interesting and very challenging problem with many practical advantages, *e.g.*, developing real-time interactive vision applications for portable devices, transferring knowledge from existing models to novel categories without re-training, *etc*. In contrast to the standard deep learning models containing millions of parameters, few-shot learning has been defined for this purpose. In the past years, a variety of few-shot learning methods have been proposed, which made great attempts to utilize the information contained in the limited labeled data.

Optimization-based solution, as one of the most popular few-shot learning paradigms, tries to capture the relation information among the tasks, leveraging the previous learning experience as a prior over tasks. For instance, in [Mishra *et al.*, 2017], the authors combined temporal convolutions with soft attention, which enables the meta-learner to aggregate contextual information from past experience and pinpoint specific pieces of information within that context. [Nichol *et al.*, 2018] introduced a first-order gradient-based meta-learning algorithm named Reptile, whose training process is similar to joint training. However, as pointed by [Rusu *et al.*, 2018], while these approaches iterate over samples from all classes in their updates, they lack the capability of learning effective embeddings.

Generation-based methods adopt a meta-learner for few-shot data augmentation or learn to predict classification weights for novel classes. [Gidaris and Komodakis, 2018] proposed a few-shot object recognition system capable of dynamically learning novel categories from only a few training data while does not forget the base categories, leading to feature representations that generalize better on unseen categories. [Gidaris and Komodakis, 2019] employed a Denoising Autoencoder network which takes a set of classification weights corrupted with Gaussian noise as input and learns to reconstruct the target-discriminative classification weights.

Metric-based solution serves as another promising few-shot learning paradigm, which exploits the feature similarity information by embedding both support and query samples into a shared feature space. For instance, Matching Net [Vinyals *et al.*, 2016] introduced the episodic training mechanism into few-shot learning and proposed the model by combining attention and memory together. In [Snell *et al.*, 2017], a Prototypical Network was proposed by taking the mean of each class as its corresponding prototype representation, which helps reduce the intra-class variations and thus learns a discriminative metric space. In [Li *et al.*, 2019c], a Covariance Metric Network was proposed to exploit both the covariance representation and covariance metric-based on the distribution consistency for the few-shot classification tasks.

[*]Corresponding Author

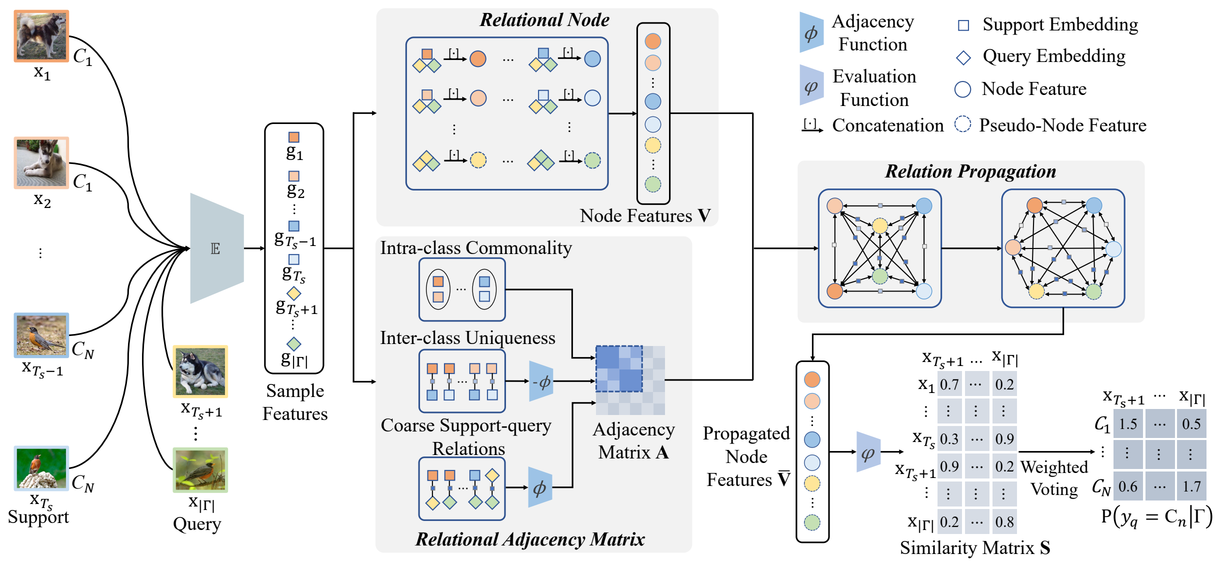


Figure 1: The Relation-Propagation Graph Network.

In fact, in order to efficiently determine the class of the queries based on few labeled samples, one should concentrate on revealing the accurate relations between the support and the query examples, especially in metric-based methods. Recently, a few approaches [Garcia and Bruna, 2017; Liu *et al.*, 2018; Kim *et al.*, 2019] adopted graph networks to implicitly model the sample relations, by treating each sample as the graph node and predicting the support-query relations based on the updated node features. Although they have shown great potential to solve the few-shot learning task, without directly modelling the relations of the support-query pairs, the underlying information shared across different support-query pairs suffers from the severe underutilization, inevitably leading to the inferior performance.

To address this problem, in this paper we propose a transductive relation-propagation graph neural network (TRPN) to explicitly model and propagate the sample relations across the support-query pairs. TRPN treats the sample relation of each support-query pair as a graph node, named relational node. Intuitively, similar support samples from the same class often share similar relations with the same query. Therefore, our TRPN resorts to the known relations between support samples, including both intra-class commonality and inter-class uniqueness, to estimate the relational adjacency among the different support-query pairs. With the relational graph, both sample similarities and the support-query relations can be propagated and aggregated to generate more discriminative relational embeddings for support-query pairs. We further introduce a pseudo relational node (*i.e.*, the query-query pair) to consider the query characteristics, which can bring the coarse relations of support-query pairs into the relation propagation. With the learnt relation representations of support-query pairs, the few-shot learning task can be cast into a simple node classification task. In practice, to further exploit the relation information among different queries, we also provide a fast, yet effective transductive learning strategy.

To the best of our knowledge, this is the first work that explicitly takes the relations of support-query pairs into consideration in few-shot learning, which might offer a new way to solve the few-shot learning problem. We conduct extensive experiments on several benchmark datasets such as *mini*ImageNet [Vinyals *et al.*, 2016] and *tiered*ImageNet [Ren *et al.*, 2018], and the results demonstrate that our method can significantly outperform a variety of the state-of-the-art few-shot learning methods. For instance, it can improve the 1-shot and 5-shot accuracy on *mini*ImageNet from 62.05% to 68.25% and from 78.63% to 85.40%, respectively compared to the second best method. More surprisingly, it can even achieve better performance with a shallow backbone network than most state-of-the-arts with a deeper one.

2 Approach

In this section, we will present our transductive relation-propagation graph neural network (TRPN) including both the non-transductive setting and the transductive setting in Section 2.2 and 2.3, respectively. Before that, we will first introduce the preliminary of few-shot setting.

2.1 Preliminary

Let $\mathcal{S}$ denote a support set, which contains N different image classes $(\mathcal{C}_1, \ldots, \mathcal{C}_N)$ and K (K is small, *e.g.*, $K = 5$) labeled samples per class. Thus the total number of support samples is $T_S = N \times K$. Given a query set $\mathcal{Q}$ where the total number of it is T_Q, few-shot learning aims to determine the class of each unlabeled sample in $\mathcal{Q}$ based on the set $\mathcal{S}$. This setting is also called N-way K-shot classification.

We follow the episodic training which efficiently learns the transferable knowledge from a relatively large labeled dataset with a set of classes $\mathcal{C}_{train}$. The objective is to train classifiers for an unseen set of novel classes $\mathcal{C}_{test}$, for which only a few labeled examples are available. In each episode, a small subset of N classes are sampled from $\mathcal{C}_{train}$ to construct an N-way K-shot problem as follows: $\Gamma = \mathcal{S} \cup \mathcal{Q}$ where $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{T_\mathcal{S}}$ and $\mathcal{Q} = \{(\mathbf{x}_i, y_i)\}_{i=T_\mathcal{S}+1}^{T_\mathcal{S}+T_\mathcal{Q}}$. $\mathbf{x}_i, y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ are the i-th sample and its label, respectively. Also we define $|\Gamma| = T_\mathcal{S} + T_\mathcal{Q}$ as the total number of the samples in the task Γ. In each episode, the model is trained to minimize the loss of its predictions of $\mathcal{Q}$ through learning the labeled support set $\mathcal{S}$.

2.2 Relation-Propagation Graph Network

In this section, we elaborate on the details of our relation-propagation model. Unlike previous graph neural network based few-shot learning models where each node represents a data sample, ignoring the valuable information shared among different support-query pairs, in the proposed TRPN, each node in the graph represents the relation of a support-query pair. As to the adjacency matrix among the nodes, since similar support samples usually have similar relations with the same query sample, we can naturally estimate node adjacency according to the known relations (labels) of support set.

With the graph representations, we can update the relational node features via a relation propagation, and thus learn the discriminative relation embeddings for support-query pairs. The relation-propagation model further utilizes a similarity function to evaluate the possibility that the consisting samples come from the same class. In this way, the few-shot learning can be cast into a node classification problem. The support-query relations can be inferred based on the similarity evaluation over their relation embeddings.

Specifically, for each sample $\mathbf{x}_i$ of the task Γ, a convolutional embedding network is first employed to extract the feature representation $\mathbf{g}_i$. Subsequently, a fully-connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}; \Gamma)$ is initially constructed to characterize the relations of the task, where $\mathcal{V}$ and $\mathcal{A}$ denote the set of relational nodes and relational adjacency matrix of the graph, respectively. Our graph contains only one layer in avoid of bringing potential concerns of over-smoothing [Li *et al.*, 2018].

Relational Node

To explicitly model the sample relations between support-query pairs, we introduce the relational nodes indicating the relation embeddings of support-query pairs. Specifically, we denote the relation node embedding matrix as $\mathbf{V}$, where $\mathbf{V}_i$ is the embedding of the i-th node. Given a query sample $\mathbf{x}_q$ where $q \in \{T_\mathcal{S}+1, \cdots, |\Gamma|\}$, $\mathbf{V}_i$ is initialized by the concatenation of the support feature $\mathbf{g}_i$ and query feature $\mathbf{g}_q$:

$$\mathbf{V}_i = [\mathbf{g}_i, \mathbf{g}_q], \tag{1}$$

where $i = \{1, \cdots, T_\mathcal{S}\}$, and $[\cdot, \cdot]$ denotes the concatenation operation.

Since the prediction of query samples mainly depends on the labeled support samples, we name the first concatenated support sample of a relational node as the dominant sample.

Though the support sample dominates the learning of relations of support-query pairs, however, without considering the characteristic of the query sample, the relational node embeddings will discard the raw, yet important relational information contained between support and query samples, and thus inevitably lead to the inferior performance. To avoid this issue, we further introduce pseudo relational nodes so as to actively involve the query samples in the graph. A pseudo relational node is defined as:

$$\mathbf{V}_q = [\mathbf{g}_q, \mathbf{g}_q], \tag{2}$$

Without loss of generality, the first concatenated sample is also named as the dominant sample of the pseudo relational node.

Relational Adjacency Matrix

The relational adjacency $\mathbf{A}_{ij}$ is the element of adjacency matrix $\mathbf{A}$ on the graph, representing the commonality between dominant samples i and j of relational nodes $\mathbf{V}_i$ and $\mathbf{V}_j$. The adjacency matrix should reflect the known information (label) of the support set, including both intra-class commonality and inter-class uniqueness, and thus weightedly propagate the relations to guide the learning of discriminative relation embeddings, with the consideration of query characteristics.

Hence, we define the adjacency between relational nodes according to the relation of their dominant samples as:

$$\mathbf{A}_{ij} = \begin{cases} 1, & if \quad i,j \leqslant T_\mathcal{S},\ l_{ij} = 1 \\ -\phi([\mathbf{g}_i, \mathbf{g}_j]), & if \quad i,j \leqslant T_\mathcal{S},\ l_{ij} = 0 \\ \phi([\mathbf{g}_i, \mathbf{g}_j]), & if \quad i = q \text{ or } j = q \end{cases} \tag{3}$$

where l_{ij} is the associate-label for each node, defined according to ground truth labels of consisting samples:

$$l_{ij} = \begin{cases} 1, & if \quad y_i = y_j \\ 0, & otherwise \end{cases} \tag{4}$$

and $\phi(\cdot)$ is the adjacency function. There are a number of candidate implementations for adjacency function, such as cosine similarity, negative squared Euclidean distance, or neural network, *etc*.

From Formula 3, we can see that, for two relational nodes whose dominant support samples come from the same class, we simply use their associate-label as the adjacency to preserve the intra-class commonality. In contrast, for two nodes whose dominant support samples come from different classes, we use the adjacency function to evaluate the adjacency between the two samples, and invert the results. Therefore, the more similar two different classes are, the more commonality of the relation each class will subtract during the feature aggregation. Through diminishing the commonality according to the negative adjacency score, it maximizes the uniqueness of the two classes and helps the network to learn the discriminative representations which assemble the similar samples and disperse the different ones. Thus, the known information including both intra-class commonality and inter-class uniqueness is considered in the adjacency matrix. Meanwhile, the coarse predicted relations between support-query pairs are also computed by the adjacency function, with the query characteristics involved during relation propagation.

Relation Propagation

Once we have the weighted graph, we can update the node features to pursue the discriminative relational embeddings for support-query pairs, using the following propagation rule:

$$\overline{\mathbf{V}} = \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{V}\mathbf{W}). \tag{5}$$

Here, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the relational adjacency matrix of the graph added with self-connections. $\mathbf{I}$ is the identity matrix, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ and $\mathbf{W}$ is a trainable weight matrix. σ denotes an activation function. After the relation propagation, the updated relational embedding $\overline{\mathbf{V}}$ becomes more discriminative and could better reflect the relations between support-query pair.

After relation propagation and feature aggregation, we use a similarity function φ operating on the $\overline{\mathbf{V}}$ to measure the similarity level between support-query pair, and thus obtain the similarity score vector for the q-th query sample $\mathbf{s}_q = \varphi(\overline{\mathbf{V}})$. We conduct above operation for each query sample and construct the similarity matrix $\mathbf{S}$ composed of their similarity vector, where $\mathbf{S}$ is a $T_{\mathcal{Q}} \times |\Gamma|$ matrix, with $\mathbf{S}_{qi}$ indicating the similarity score of the i-th support samples and the query sample $\mathbf{x}_q$. Here the similarity score can be considered as a probability that the two samples are from the same class. Therefore, each query samples can be classified by simple weighted voting.

2.3 Transductive Learning

Although the proposed model could exploit the valuable information shared between different support-query pairs and generate discriminative relation embeddings, the fundamental difficulty of learning with scarce data remains for a novel classification task. One way to achieve further improvements with a limited amount of training data is to consider relations between samples in the query set and thus predict them as a whole, which is referred to as transduction, or transductive inference [Kim *et al.*, 2019; Liu *et al.*, 2018]. A simple way to implement transductive learning, as previous graph models did, is to involve more relational nodes consisting of support-query pairs and pseudo relational nodes, which could easily propagate the relations between query samples on the graph due to feature aggregation.

However, we think such implementation is computation exhaustive and will cause performance degradation, since the relations between dominate samples, whatever the concatenated query sample is, are the same and indistinguishable. Therefore, for each relational node, we combine a support sample with the whole query set, and thus the relations between query samples could be simultaneously processed both inside the relational node and in the propagation process on the graph. Specifically, the relational node features under the transductive setting could be initialized as:

$$\mathbf{V}_i = [\mathbf{g}_i, \mathbf{g}_{T_{\mathcal{S}}+1}, \mathbf{g}_{T_{\mathcal{S}}+2}, \cdots, \mathbf{g}_{|\Gamma|}], \tag{6}$$

Similarly, the pseudo node could be defined as follows:

$$\mathbf{V}_q = [\mathbf{g}_q, \mathbf{g}_{T_{\mathcal{S}}+1}, \mathbf{g}_{T_{\mathcal{S}}+2}, \cdots, \mathbf{g}_{|\Gamma|}], \tag{7}$$

where $i = \{1, \cdots, T_{\mathcal{S}}\}$, and $q = \{T_{\mathcal{S}} + 1, \cdots, |\Gamma|\}$. The relational adjacency matrix $\mathbf{A}$ is constructed similarly to the one under the non-transductive setting, indicating pairwise adjacency:

$$\mathbf{A}_{ij} = \begin{cases} 1, & if\ i, j \leqslant T_{\mathcal{S}}, l_{ij} = 1 \\ -\phi([\mathbf{g}_i, \mathbf{g}_j]), & if\ i, j \leqslant T_{\mathcal{S}}, l_{ij} = 0 \\ \phi([\mathbf{g}_i, \mathbf{g}_j]), & \begin{array}{l} if\ T_{\mathcal{S}} + 1 \leqslant i \leqslant |\Gamma| \\ \text{or } T_{\mathcal{S}} + 1 \leqslant j \leqslant |\Gamma| \end{array} \end{cases} \tag{8}$$

Hence, we can follow the same relation-propagation procedure as which under the non-transductive setting.

Similarly, we utilize a similarity function φ operating on the propagated relational nodes $\overline{\mathbf{V}}$ which directly outputs a $T_{\mathcal{Q}} \times |\Gamma|$ similarity matrix $\mathbf{S}$.

2.4 Learning and Inference

Loss Function

To guide the learning of the graph network, we first introduce the relational node classification loss:

$$\ell_n = \sum_{i=1}^{|\Gamma|} \sum_{q=T_{\mathcal{S}}+1}^{|\Gamma|} l_{qi} \log \mathbf{S}_{qi} + (1 - l_{qi}) \log(1 - \mathbf{S}_{qi}).$$

The node loss is to promise the node feature could represent the relationship between two samples and further help the framework to figure out whether the query sample belongs to the same category as the other does.

Besides, if the adjacency function ϕ is implemented by neural networks, we could further take advantage of the label information of support samples to enhance the capability of ϕ as follows:

$$\ell_s = \sum_{i=1}^{|\Gamma|} \sum_{j=1}^{|\Gamma|} l_{ij} \log \phi([\mathbf{g}_i, \mathbf{g}_j]) + (1 - l_{ij}) \log(1 - \phi([\mathbf{g}_i, \mathbf{g}_j]))$$

As a result, the final loss is as follows:

$$\ell = \ell_n + \lambda \ell_s,$$

where λ is the hyper-parameter to balance the weights between the two loss functions.

Inference on Novel Class

In the test phase, we evaluate the performance of our TRPN on novel categories that are not seen in episodic training. We perform the relation propagation procedure of the proposed graph network and compute the probability that a query sample $\mathbf{x}_q$ belongs to the n-th category through weighted voting:

$$\mathrm{P}(y_q = \mathcal{C}_n | \Gamma) = \sum_{\{i | (\mathbf{x}_i, y_i) \in \mathcal{S} \wedge y_i = \mathcal{C}_n\}} \mathbf{S}_{qi}. \tag{9}$$

The category with the highest probability is regarded as the final prediction.

3 Experiments

In this section, we evaluate our TRPN on two widely used datasets, compared with a number of state-of-the-art few-shot approaches.

3.1 Experimental Setup

Datasets

We employ the widely used datasets in prior studies, including *mini*ImageNet dataset [Vinyals *et al.*, 2016] and *tiered*ImageNet dataset [Ren *et al.*, 2018]. Both of them are subsets of the ILSVRC-12 dataset [Russakovsky *et al.*, 2015]. The *mini*ImageNet dataset consists of 100 classes, each of which contains 600 images of size 84 × 84, while the *tiered*ImageNet contains 608 classes with 77915 images

Models	Backbone	1-shot	5-shot
Optimization-based			
mLSTM [Ravi and Larochelle, 2017]	Conv4	43.44 ± 0.77	60.60 ± 0.71
MAML [Finn *et al.*, 2017]	Conv4	48.70 ± 1.84	63.10 ± 0.92
Meta-SGD [Li *et al.*, 2017]	Conv4	50.47 ± 1.87	64.03 ± 0.94
SNAIL [Mishra *et al.*, 2017]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92
REPTILE [Nichol *et al.*, 2018]	Conv4	49.97 ± 0.32	65.99 ± 0.58
MTL [Sun *et al.*, 2019]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80
LEO [Rusu *et al.*, 2018]	WRN-28	61.76 ± 0.08	77.59 ± 0.12
Generation-based			
PLATIPUS [Finn *et al.*, 2018]	Conv4	50.13 ± 1.86	-
VERSA [Gordon *et al.*, 2018]	Conv4	53.40 ± 1.82	67.37 ± 0.86
LwoF [Gidaris and Komodakis, 2018]	ResNet	55.45 ± 0.89	70.13 ± 0.68
Param_Predict [Qiao *et al.*, 2018]	WRN-28	59.60 ± 0.41	73.74 ± 0.19
wDAE [Gidaris and Komodakis, 2019]	WRN-28	61.07 ± 0.15	76.75 ± 0.11
Metric-based			
Matching Net [Vinyals *et al.*, 2016]	Conv4	43.56 ± 0.84	55.31 ± 0.73
Prototypical Net [Snell *et al.*, 2017]	Conv4	49.42 ± 0.78	68.20 ± 0.66
Relation Net [Sung *et al.*, 2018]	Conv4	50.40 ± 0.80	65.30 ± 0.70
TADAM [Oreshkin *et al.*, 2018]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
CTM [Li *et al.*, 2019a]	ResNet-18	62.05 ± 0.55	78.63 ± 0.06
CovaMNet [Li *et al.*, 2019c]	Conv4	51.19 ± 0.76	67.65 ± 0.63
DN4 [Li *et al.*, 2019b]	Conv4	51.24 ± 0.74	71.02 ± 0.64
TapNet [Yoon *et al.*, 2019]	ResNet-12	61.65 ± 0.15	76.36 ± 0.10
Graph-based			
GNN [Garcia and Bruna, 2017]	Conv4	50.33 ± 0.36	66.41 ± 0.63
TPN [Liu *et al.*, 2018]	ResNet	59.46	75.65
EGNN [Kim *et al.*, 2019]	Conv4	-	76.37 ± 0.30
TRPN	Conv4	$\mathbf{57.84 \pm 0.51}$	$\mathbf{78.57 \pm 0.44}$
TRPN	WRN-28	$\mathbf{68.25 \pm 0.50}$	$\mathbf{85.40 \pm 0.39}$

Table 1: Few-shot image classification accuracies of 5-way 1-shot and 5-way 5-shot tasks on *mini*ImageNet .

in total. The classes of *tieredImageNet* are grouped into 34 higher-level nodes based on WordNet hierarchy [Deng *et al.*, 2009], and further partitioned into disjoint sets of training, testing, and validation nodes, ensuring a distinct distance between training and testing classes thus making the classification more challenging. For both datasets, we adopt the common splits as previous work.

Network Architectures

We use two widely-used feature embedding architectures as our backbone: Conv-4 [Kim *et al.*, 2019] and WRN-28 [Zagoruyko and Komodakis, 2016]. For Conv-4 consisting of 4 convolutional blocks, we adopt the same embedding network architecture used in [Kim *et al.*, 2019] and train the embedding network and graph network in an end-to-end manner. WRN-28 is a 28-layer wide residual network with width factor 10 whose output is a 640-dimensional feature vector. We pre-train the WRN-28 network by optimizing the accuracy of the multi-classes classification on the whole training set of *mini*ImageNet or *tiered*ImageNet, and then freeze the parameters during relation propagation. The adjacency function ϕ and evaluation function φ are implemented with Multi-Layer Perceptions (MLPs) consisting of 3 fully-connected layers. Specially, under non-transductive setting, φ shares the same parameters with ϕ.

Implementation Details

Following [Vinyals *et al.*, 2016], we adopt the episodic training procedure. Standard data augmentation including random crop, left-right flip, and color jitter are applied in the training stage. The number of training iterations on *mini*ImageNet and *tiered*ImageNet are 100K and 200K, respectively. We use Adam optimizer [Kingma and Ba, 2014] with an initial learning rate of 0.001, and reduce the learning rate by half every 15K and 30K iterations, respectively on *mini*ImageNet and *tiered*ImageNet. The weight decay is set to $1e^{-6}$. The mini-batch size for all experiments is 20. We use the validation set to select the training episodes with the best accuracy.

Models	Backbone	1-shot	5-shot
Optimization-based			
MAML [Finn *et al.*, 2017]	Conv4	51.67 ± 1.81	70.30 ± 0.08
REPTILE [Nichol *et al.*, 2018]	Conv4	52.36 ± 0.23	71.03 ± 0.22
Meta-SGD [Li *et al.*, 2017]	Conv4	$\mathbf{62.95 \pm 0.03}$	79.34 ± 0.06
LEO [Rusu *et al.*, 2018]	WRN-28	66.33 ± 0.05	81.44 ± 0.09
Generation-based			
LwoF [Gidaris and Komodakis, 2018]	Conv4	50.90 ± 0.46	66.69 ± 0.36
wDAE [Gidaris and Komodakis, 2019]	WRN-28	68.18 ± 0.16	83.09 ± 0.12
Metric-based			
Matching Net [Vinyals *et al.*, 2016]	Conv4	54.02 ± 0.00	70.11 ± 0.00
Prototypical Net [Snell *et al.*, 2017]	Conv4	53.31 ± 0.89	72.69 ± 0.74
Relation Net [Sung *et al.*, 2018]	Conv4	54.48 ± 0.93	71.32 ± 0.70
CTM [Li *et al.*, 2019a]	ResNet-18	64.78 ± 0.11	81.05 ± 0.13
TapNet [Yoon *et al.*, 2019]	ResNet-12	63.08 ± 0.15	80.26 ± 0.12
Graph-based			
GNN [Garcia and Bruna, 2017]	Conv4	43.56 ± 0.84	55.31 ± 0.73
TPN [Liu *et al.*, 2018]	Conv4	57.53 ± 0.96	72.85 ± 0.74
EGNN [Kim *et al.*, 2019]	Conv4	-	$\mathbf{80.15 \pm 0.30}$
TRPN	Conv4	59.26 ± 0.50	79.66 ± 0.45
TRPN	WRN-28	$\mathbf{70.25 \pm 0.50}$	$\mathbf{85.21 \pm 0.37}$

Table 2: Few-shot image classification accuracies of 5-way 1-shot and 5-way 5-shot tasks on *tiered*Imagenet.

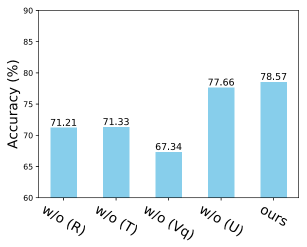
Evaluation Protocols

On both datasets, we conduct 5-way 1-shot and 5-shot experiments which are standard few-shot learning settings. For evaluation, each episode is formed by randomly sampling 1 query for each of 5 classes. We report the mean accuracy $(\%)$ of 10K randomly generated episodes as well as the 95% intervals on test set.

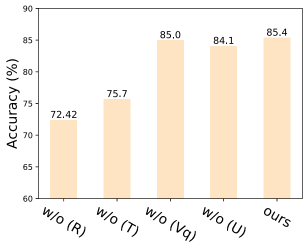
3.2 Comparison with State-of-the-Arts

We first investigate the performance of our model, compared with state-of-the-art few-shot approaches, respectively on *mini*ImageNet and *tiered*ImageNet. These approaches are particularly divided into optimization-based, generation-based and metric-based. Since our TRPN is a graph model, we further split the graph models from metric models, namely the graph-based methods. Table 1 and Table 2 list the few-shot classification accuracies of the 5-way 1-shot and 5-shot tasks along with the specifications of the backbone embedding models for feature extraction.

From Table 1, we can observe that as the shots increase, all the methods perform better, which is adhere to our intuition. Moreover, a deeper embedding network will lead to a better classification performance compared to methods equipped with Conv4, achieving above 70% accuracy on 5-shot setting. In most cases, our TRPN model significantly outperforms others under the same experimental setting, achieving 68.25% and 85.40% accuracy on 5-way 1-shot and 5-shot setting, respectively. Though as presented in [Chen *et al.*, 2019], as the backbone gets deeper, the gap among different methods drastically reduces, our TRPN model can consistently gain nearly 7% improvements on 5-shot setting over the second best approach CTM, confirming the superiority of TRPN mainly owning to the relation propagation. Meanwhile, TRPN with "Conv-4" backbone achieves 57.84% and 78.57% respectively on 1-shot and 5-shot setting, even outperforming most of the state-of-the-art models equipped with

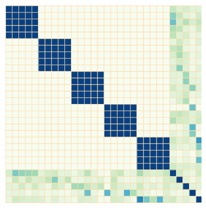


(a) Conv-4

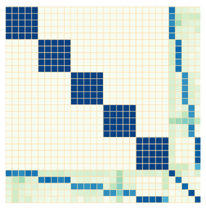


(b) WRN-28

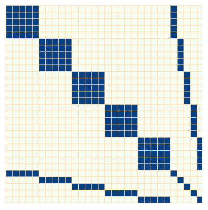
Figure 2: Ablation study on *mini*ImageNet. It respectively shows the few-shot classification results without relational nodes, under non-transductive setting, without pseudo relational nodes, without inter-class uniqueness in adjacency matrix and that obtained by the full TRPN framework from right to left.



(a) initial adjacency



(b) similarity score



(c) ground-truth

Figure 3: Visualization of the relation propagation.

deeper embedding networks. Similar trend can also be observed in Table 2. The proposed TRPN shows comparable results with the state-of-the-arts, achieving 70.25% and 85.21% accuracy on 5-way 1-shot and 5-shot setting, respectively.

3.3 Ablation Study

As aforementioned, our method mainly gains from the whole relational graph framework. Here, we study the effects of different parts in our TRPN model. Figure 2 respectively shows the few-shot classification results without relational nodes, under non-transductive setting, without pseudo relational nodes, without inter-class uniqueness in adjacency matrix and that obtained by the full TRPN framework. From the results, we can see that both the proposed framework without relational nodes and that under non-transductive setting cannot utilize sufficient information existing in the task, and thus suffer from a significant performance drop. Without pseudo relational nodes, the proposed model with "Conv-4" backbone witnesses a large drop from 78.57% to 67.34%, while performance on TRPN model with "WRN-28" only declines by nearly 0.4%. We consider the difference lies in the representative ability of different backbones. Without considering inter-class uniqueness, the results decrease by 1% around, due to neglecting the commonality of different categories.

Figure 3 further shows how our relation-propagation graph network propagates the relations of the support-query pairs. Since we mainly focus on support-query relations, we simply use the given label in visualization matrices, and respectively visualize the relations including the initial coarse adjacency based on the raw node features, the predicted similarities

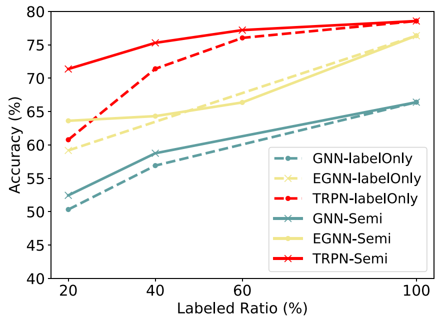


Figure 4: Semi-supervised few-shot classification accuracies of the 5-way 5-shot tasks on *mini*ImageNet.

based on the learnt relational embeddings and ground-truth. It is easy to conclude that starting from the coarse prediction between support-query pairs where relations are uncertain with some randomness, our model can precisely predict the true relations using the relation propagation.

3.4 Semi-supervised Few-shot Classification

For more efficient relation propagation on the graph, we extend our graph based architecture to learn from a mixture of labeled and unlabeled examples. We followed the same semi-supervised settings proposed in [Garcia and Bruna, 2017; Kim *et al.*, 2019] for fair comparison. Two settings are considered in this experiment, namely "LabeledOnly" and "Semi". "LabeledOnly" denotes learning with only labeled support samples, and "Semi" means a 5-way 5-shot setting with partially labeled samples in the support set where labeled samples are balanced among classes. We compare our performance with GNN and EGNN equipped with the "Conv-4" backbone. As shown in Figure 4, our TRPN significantly surpasses other graph models under both semi-supervised settings. Generally speaking, semi-supervised learning achieves superior performance compared to labeled-only training. Apart from the higher accuracy with every label ratio, TRPN also shows a faster learning speed as labeled ratio increases especially on "LabeledOnly" setting, verifying that it could learn the discriminative relation embeddings from the given labels.

4 Conclusion

In this paper, we presented a novel transductive relation-propagation graph neural network (TRPN), the first work which explicitly models and propagates the relations across support-query pairs. We also introduced a pseudo relational node and devised an effective transductive learning strategy to further exploit the relation information among different sample pairs. Extensive experiments conducted on several benchmark datasets demonstrate the superiority of TRPN compared with state-of-the-art few-shot learning methods.

Acknowledgements

This work was supported by National Natural Science Foundation of China (61872021, 61690202), and Beijing Nova Program of Science and Technology (Z191100001119050).

References

[Chen *et al.*, 2019] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Finn *et al.*, 2017] 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-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[Finn *et al.*, 2018] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[Garcia and Bruna, 2017] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

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

[Gidaris and Komodakis, 2019] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. *arXiv preprint arXiv:1905.01102*, 2019.

[Gordon *et al.*, 2018] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E Turner. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv:1805.09921*, 2018.

[Kim *et al.*, 2019] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2017] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

[Li *et al.*, 2019b] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[Li *et al.*, 2019c] Wenbin Li, Jinglin Xu, Jing Huo, Lei Wang, Yang Gao, and Jiebo Luo. Distribution consistency based covariance metric networks for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8642–8649, 2019.

[Liu *et al.*, 2018] 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. *arXiv preprint arXiv:1805.10002*, 2018.

[Mishra *et al.*, 2017] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[Nichol *et al.*, 2018] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[Oreshkin *et al.*, 2018] Boris 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 721–731, 2018.

[Qiao *et al.*, 2018] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

[Ravi and Larochelle, 2017] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017.

[Ren *et al.*, 2018] 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. *arXiv preprint arXiv:1803.00676*, 2018.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[Rusu *et al.*, 2018] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

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

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[Yoon *et al.*, 2019] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. *arXiv preprint arXiv:1905.06549*, 2019.

[Zagoruyko and Komodakis, 2016] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.