Transductive Propagation Network for Few-shot Learning

Yanbin Liu*
University of Technology Sydney
csyanbin@gmail.com

Juho Lee
University of Oxford / AITRICS
juho.lee@stats.ox.ac.uk

Minseop Park
AITRICS
mike_seop@aitrics.com

Saehoon Kim
AITRICS
shkim@aitrics.com

Yi Yang
University of Technology Sydney
Yi.Yang@uts.edu.au

Abstract

Few-shot learning aims to build a learner that quickly generalizes to novel classes even when a limited number of labeled examples (so-called low-data problem) are available. Meta-learning is commonly deployed to mimic the test environment in a training phase for good generalization, where episodes (i.e., learning problems) are manually constructed from the training set. This framework gains a lot of attention to few-shot learning with impressive performance, though the low-data problem is not fully addressed. In this paper, we propose *Transductive Propagation Network* (TPN), a transductive method that classifies the entire test set at once to alleviate the low-data problem. Specifically, our proposed network explicitly learns an underlying manifold space that is appropriate to propagate labels from few-shot examples, where all parameters of feature embedding, manifold structure, and label propagation are estimated in an end-to-end way on episodes. We evaluate the proposed method on the commonly used *mini*ImageNet and *tiered*ImageNet benchmarks and achieve the state-of-the-art or promising results on these datasets.

1 Introduction

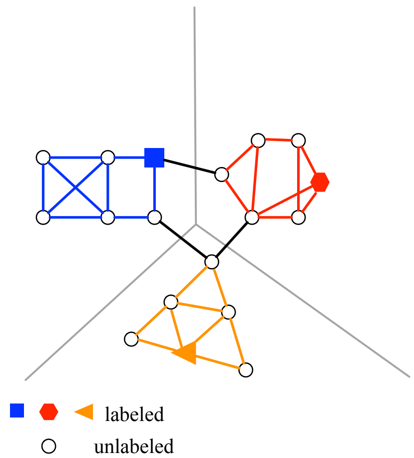
Recent breakthroughs of deep learning methods [9, 19, 4] highly rely on large amounts of labeled data. However, it increases the burden of data collection, which hinders its potential applications to the low-data regime where the labeled data are rare and difficult to gather. In contrary, humans have the ability to recognize new objects with only one or few samples [10]. For example, children can generalize the concept of "apple" after given a single sample of the apple. This significant gap between human and deep learning reawakens the research interest of few-shot learning [24, 20, 2, 14, 11, 28, 26].

Few-shot learning aims to learn a classifier that can be adapted to accommodate new classes (i.e., novel classes) not seen in the training set with a few examples of each of these classes. Traditional techniques such as fine-tuning [6] that work well for deep learning methods would severely overfit on this task [24, 2]. The overfitting problem mainly comes from two aspects. On one hand, since the training classes have no overlap with the novel classes, their data distributions have a big difference and the model trained on the training classes is usually inapplicable to the novel classes. On the other hand, only a few examples in each class may be biased in the novel class space, which makes the final classifier unreliable.

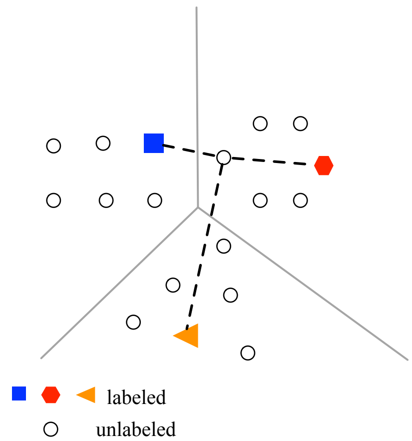
In order to solve the first problem, Vinyals et al. [24] proposed an episodic training strategy. In each episode, the algorithm learns the embedding of the few labeled examples (the *support set*) to predict

*Work was done while interning at AITRICS.

Preprint. Work in progress.



(a) Transduction



(b) Induction

Figure 1: Transduction versus Induction. **Left**: For transduction, the model classifies the entire test set at once using manifold structure. **Right**: For induction, each test example is predicted independently (e.g., using nearest neighbor classifier). Moreover, labeled examples are biased in this case.

classes for the unlabeled points (the *query set*). The purpose of episodic training is to mimic the real test environment containing few-shot support set and unlabeled query set. The consistency between training and test environment alleviates the distribution gap and improves generalization. Subsequent algorithms follow this episodic training routine. With episodic training, Finn et al. [2] learned a good initialization that can adapt quickly to the target tasks. Snell et al. [20] used episodes to train a good representation and predict classes by computing Euclidean distance with respect to class prototypes.

Although episodic strategy obtains satisfying results in several datasets, the bias issue in novel class space remains unsolved (see Figure 1 (b)). In previous works [7, 30, 23], transductive inference shows better performance than inductive methods, especially for small training sets. Motivated by this, we propose *Transductive Propagation Network* (TPN) to deal with the bias problem. Instead of predicting labels of query examples one by one, we utilize the entire query set for transductive inference (see Figure 1). Specifically, we first map the input to an embedding space using a deep neural network. Then a graph construction module is proposed to exploit the manifold structure of the novel class space using the union of support set and query set. According to the graph structure, iterative label propagation is applied to propagate labels from the support set to the query set and finally leads to a closed-form solution. With the propagated scores and ground truth labels of the query set, we compute the cross-entropy loss with respect to the feature embedding, manifold structure, and label propagation parameters. Finally, all parameters can be updated end-to-end using backpropagation.

The main contribution of this work is threefold. First, to the best of our knowledge, we are the first to model transductive inference explicitly in few-shot learning. Although [13] experimented with a transductive setting, they only share information between test examples by batch normalization rather than directly propose a transductive model. Second, we improve naive label propagation [30] by proposing a graph construction module which produces episodic-wise and example-wise parameters to exploit the manifold structure in each episode. Third, we evaluate our approach on two benchmarks, namely *mini*ImageNet and *tiered*ImageNet. The experimental results show that *Transductive Propagation Network* gains substantial improvements over state-of-the-art on 1-shot classification and promising results on 5-shot classification. Also, transductive inference obtains comparable results with semi-supervised methods without additional unlabeled data.

2 Related work

Meta-learning In recent works, few-shot learning often follows the idea of meta-learning [18, 22]. Meta-learning tries to optimize over batches of tasks rather than batches of data points. Each task corresponds to a learning problem, obtaining good performance on these tasks helps to learn quickly and generalize well to the target few-shot problem without suffering from overfitting. The well-known MAML approach [2] aims to find more transferable representations with sensitive parameters. When applied to the target few-shot problem, small changes to these parameters will produce large improvements. A first-order meta-learning approach named Reptile is proposed in [13] which is closely related to first-order MAML but doesn't need a training-test split for each task. Compared with the above methods, our algorithm has a closed-form solution for label propagation on the query points, thus avoiding gradient computation in the inner update [2, 14, 13] and usually performs more efficiently.

Embedding and metric learning approaches Another category of few-shot learning approach aims to optimize the transferable embedding using metric learning approaches. Matching networks [24] produce a weighted nearest neighbor classifier given the support set and adjust feature embeddings according to the performance on the query set. Prototypical networks [20] first compute a class's prototype to be the mean of its support set in the embedding space. Then the transferability of feature embedding is evaluated by finding the nearest class prototype for embedded query points. An extension of prototypical networks is proposed in [15] to deal with semi-supervised few-shot learning. Rather than using a fixed metric such as Euclidean distance or cosine distance, Relation Network [21] learns to learn a deep distance metric to compare a small number of images within episodes. Our proposed method is similar to these approaches in the sense that we all focus on learning deep embeddings with good generalization ability. However, our algorithm assumes a transductive setting, in which we utilize the union of support set and query set to exploit the manifold structure of novel class space by using episodic-wise parameters.

Transduction The setting of transductive inference was first introduced by Vapnik [23]. Transductive Support Vector Machines (TSVMs) [7] is a margin-based classification method that considers a particular test set and minimizes errors of these particular set of examples. In text classification, TSVMs show substantial improvements over inductive methods, especially for small training sets. Another category of transduction methods involves graph-based methods [30, 25, 27, 16, 3]. Label propagation is used in [30] to transfer labels from labeled to unlabeled data guided by the weighted graph. Label propagation is sensitive to variance parameter σ, so Linear Neighborhood Propagation (LNP) [25] constructs approximated Laplacian matrix to avoid this issue. In [31], minimum spanning tree heuristic and entropy minimization are used to learn the parameter σ. In our algorithm, we propose a graph construction module to produce episodic-wise and example-wise σ. This is consistent with the episodic setting of few-shot learning and it performs well. In transfer learning, [16] performs transductive inference by exploiting the manifold structure of novel classes using mid-level semantic attributes. By contrast, our algorithm enjoys the merits of deep embedding and jointly updates the parameters of embedding and graph construction in an end-to-end way. In few-shot learning, [13] experiments with a transductive setting and shows improvements. However, they only share information between test examples via batch normalization [5] rather than explicitly model the transductive setting as in our algorithm.

3 Main approach

3.1 Problem definition

In few-shot learning, we follow the episodic paradigm proposed by Vinyals et al. [24], which is widely-used in recent few-shot studies [20, 2, 13, 21, 12]. In this setting, we have a relatively large labeled dataset with a set of classes $\mathcal{C}_{train}$. Our ultimate goal is to train classifiers for an unseen set of novel classes $\mathcal{C}_{test}$, for which only a few labeled examples are available. The key idea of episodic paradigm is to mimic the real test environment by constructing classification tasks with the labeled data in $\mathcal{C}_{train}$.

Specifically, in each episode, a small subset of N classes are sampled from $\mathcal{C}_{train}$ to construct a *support set* and a *query set*. The *support set* contains K examples from each of the N classes (i.e., N-way K-shot setting) denoted as $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_{N\times K}, y_{N\times K})\}$, while the *query set* $\mathcal{Q} = \{(\mathbf{x}_1^*, y_1^*), (\mathbf{x}_2^*, y_2^*), \dots, (\mathbf{x}_T^*, y_T^*)\}$ includes different examples from the same N classes. Here, the support set $\mathcal{S}$ serves as the training set on which the model is trained to update the parameters to minimize the loss of its predictions for the examples in the query set $\mathcal{Q}$. This procedure goes episode by episode until convergence.

In a deeper insight, the model tries to learn a learning algorithm that can produce good predictions on the query set based on the support set. In this sense, the episodic paradigm is also referred to as learning to learn or meta-learning. However, the lack of labeled examples (K is usually very small) in support set makes it difficult to learn a reliable classifier. So in this paper, we propose a transductive setting by utilizing the whole query set for the prediction rather than predicting each example independently. Taking the entire query set into account, we can alleviate the low-data problem and reduce the test error.

3.2 Transductive Propagation Network (TPN)

In this section, we describe the proposed *Transductive Propagation Network* (TPN) in detail. Our method is composed of four distinct components: 1) we use feature embedding module to map the original input to the deep metric space (Section 3.2.1); 2) we propose a graph construction module to produce episodic-wise and example-wise parameters to exploit the manifold structure (Section 3.2.2); 3) an iterative label propagation procedure is applied, and the closed-form solution is obtained to get predictions for the query set $\mathcal{Q}$ (Section 3.2.3); 4) a cross-entropy loss function is computed on $\mathcal{Q}$ to jointly train the parameters of feature embedding, graph construction, and label propagation (Section 3.2.4).

3.2.1 Feature embedding

We first apply deep feature embedding using a convolutional neural network to extract features for the inputs (e.g., images). We use the same convolutional architecture which is adopted in several recent works [20, 21, 24]. Using the same feature embedding, we can fairly compare our method with related work and pay more attention to the performance of our model and the transductive setting.

The network is made up of four convolutional blocks where each block begins with a 2D convolutional layer with a 3×3 kernel and filter size of 64. Each convolutional layer is followed by a batch-normalization layer [5], a ReLU nonlinearity and a 2×2 max-pooling layer. We use the same parameter for both the support set $\mathcal{S}$ and the query set $\mathcal{Q}$. For convenience, we denote feature embedding as a function f with parameter φ. An input $\mathbf{x}_i$ to the network produces the feature map $f_\varphi(\mathbf{x}_i)$.

3.2.2 Graph construction

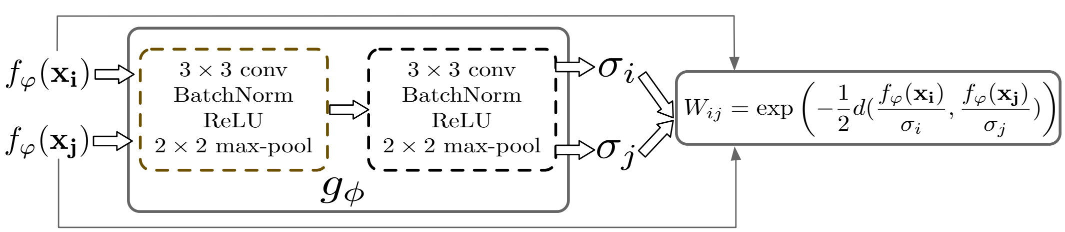


Figure 2: Structure of the graph construction module.

In manifold learning algorithms [1, 30, 29], the manifold graph is usually constructed using Gaussian similarity function:

$$W_{ij} = \exp(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}), \tag{1}$$

where $d(\cdot, \cdot)$ is a distance measure (e.g., Euclidean distance) and σ is the length scale parameter. However, it is difficult to choose the optimal σ in label propagation [25, 31].

In this paper, we propose a graph construction module built on the union set of support set and query set: $\mathcal{S} \cup \mathcal{Q}$. The structure of the module is shown in Figure 2, which consists of two convolutional blocks each with the same structure as feature embedding. We denote it as a function g with parameter ϕ. In each episode, this module produces a parameter $\sigma_i = g_\phi(f_\varphi(\mathbf{x}_i))$ for every example i of the union set. This means that our module is episodic-wise and example-wise, which make it especially suitable for few-shot learning. Alternatively, another way is to compute the pairwise σ_{ij} on the union set. But this needs a computation cost of $\frac{n(n-1)}{2}$ compared with n in our method, where

$n = N \times K + T$. The graph weight is computed as follows:

$$W_{ij} = \exp\left(-\frac{1}{2}d(\frac{f_\varphi(\mathbf{x_i})}{\sigma_i}, \frac{f_\varphi(\mathbf{x_j})}{\sigma_j})\right) . \tag{2}$$

We then apply a normalization technique similar to normalized graph Laplacians [1] on graph weight W. The normalized weight $S = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with its (i,i)-value to be the sum of the i-th row of W. Since W is constructed on $\mathcal{S} \cup \mathcal{Q}$, $W \in R^{(N\times K+T)\times(N\times K+T)}$.

3.2.3 Label propagation

In this section, we describe how to get the predictions for the query set $\mathcal{Q}$ using label propagation. Let $\mathcal{F}$ denote the set of $(N \times K + T) \times N$ matrix with nonnegative entries. We define a label matrix $Y \in \mathcal{F}$ with $Y_{ij} = 1$ if $\mathbf{x}_i$ from the support set is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise. We define $F \in \mathcal{F}$ as the classification score produced by our algorithm. F is initialized to Y and only the support set contains labeled examples. By running label propagation, labels are propagated on the union set $\mathcal{S} \cup \mathcal{Q}$ according to the graph structure using the following formulation:

$$F_{t+1} = \alpha S F_t + (1-\alpha)Y , \tag{3}$$

where $\alpha \in (0,1)$ controls the amount of propagated information. By iteratively executing Equation (3), F_t will converge to the optimal F^*. And the label of $\mathbf{x}_i$ is predicted as $\tilde{y}_i = \arg\max_{j\leq N} F^*_{ij}$.

The label propagation algorithm is equivalent to the following optimization problem [30]:

$$F^* = \arg\min_{F\in\mathcal{F}} \frac{1}{2}\left(\sum_{i,j=1}^{N\times K+T} W_{ij}\|\frac{1}{\sqrt{D_{ii}}}F_i - \frac{1}{\sqrt{D_{jj}}}F_j\|^2 + \mu \sum_{i=1}^{N\times K+T} \|F_i - Y_i\|^2\right) , \tag{4}$$

where $\mu > 0$ is the regularization parameter. Differentiating the objective function with respect to F and making some transformation, we have the final closed-form solution:

$$F^* = (I - \alpha S)^{-1} Y , \tag{5}$$

where I is the identity matrix and $\alpha = \frac{1}{1+\mu}$. The closed-form solution is an appealing result. At each training episode, we can directly compute F^* rather than iterate using Equation (3). In this way, the computation cost can be reduced, thus making our algorithm efficient in practice.

3.2.4 Loss function

In the episodic training stage, we perform transductive label propagation on labeled support set $\mathcal{S}$ and unlabled query set $\mathcal{Q}$ to get predicted labels $\tilde{\mathbf{y}}$ of the query set. The parameter learning proceeds by computing the cross-entropy loss using predicted scores F^* and ground truth labels $\mathbf{y}$ from $\mathcal{Q}$. At first, F^* is converted to probabilistic score using softmax:

$$P(\tilde{y}_i = j|\mathbf{x}_i) = \frac{\exp(F^*_{ij})}{\sum_{j=1}^{N}\exp(F^*_{ij})} . \tag{6}$$

Then the loss function is denoted as:

$$J(\varphi, \phi, \alpha) = \sum_{i=1}^{T}\sum_{j=1}^{N} -\mathbb{I}(y_i == j)\log(P(\tilde{y}_i = j|\mathbf{x}_i)) , \tag{7}$$

where $\mathbb{I}(b)$ is an indicator function, $\mathbb{I}(b) = 1$ if b is true and 0 otherwise.

Note that in Equation (7), loss is computed with respect to three parameters φ, ϕ and α. For φ and ϕ, they are highly related to $\mathcal{S}$ and $\mathcal{Q}$ while α is regarded as a global free parameter. All these parameters are jointly updated by the episodic training in an end-to-end manner.

4 Experiments

For transductive few-shot learning, we first performed experiments on two datasets: the commonly used *mini*ImageNet dataset with the splits proposed by Ravi and Larochelle [14] and the newly released large-scale *tiered*ImageNet dataset proposed by Ren et al. [15]. Then we compared our transductive setting with semi-supervised few-shot learning method on an adapted version of these two datasets [15].

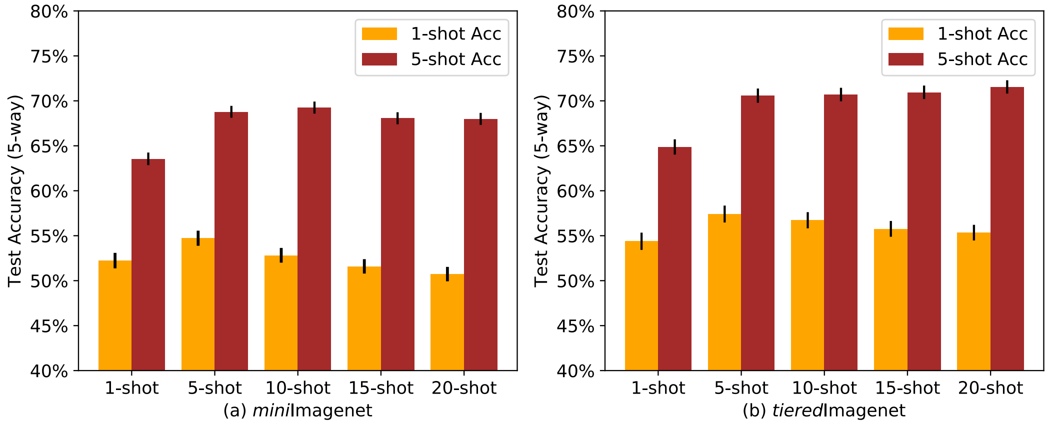


Figure 3: Model performance with different training shots. The x-axis indicates the number of shots in training, and the y-axis indicates 5-way test accuracy for 1-shot and 5-shot. Error bars indicate 95% confidence intervals as computed over 600 test episodes.

4.1 *mini*ImageNet few-shot classification

The *mini*ImageNet dataset, originally proposed by Vinyals et al. [24], is a modified version of ILSVRC-12 dataset [17]. The splits used by Vinyals et al. [24] consists of 100 classes with 600 examples each. In our experiments, in order to directly compare with state-of-the-art algorithms for few-shot learning, we rely on the class splits used by Ravi and Larochelle [14], which has a different set of 100 classes including 64 for training, 16 for validation, and 20 for test. All images are of size 84×84 pixels. We follow the standard procedure by training on 64 classes and validate on 16 classes.

In our experiments, we used the embedding structure described in Section 3.2.1 and the graph construction module described in Section 3.2.2. With an input image of size 84×84, we got a 1600-dimensional feature embedding and a 1-dimensional σ. All of our models were trained with Adam [8]. We used an initial learning rate of 10^{-3} and cut the learning rate in half every 10,000 episodes. Training was stopped when the validation loss reached a plateau.

In the episodes construction, Prototypical networks [20] show that training with a higher way (e.g., 30-way 1-shot training for the 5-way 1-shot task) can significantly improve classification results (see also Table 1). However, in our transductive setting, we found that it is beneficial to use the higher shot for training while aligning the training way to test way. To investigate the proper number of training shot, we performed 5-way training with various number of shots (1, 5, 10, 15, and 20) in constructing the episodes to test the 5-way 1-shot and 5-way 5-shot accuracies. Experimental results are shown in Figure 3 (a). It can be seen that 5-shot training achieves best results for the 5-way 1-shot task while 10-shot training achieves best results for the 5-way 5-shot task. We conjecture that the increased shot helps the discovery of manifold distribution and the propagation process from labeled data, thus leading to better generalization ability. But if we continue to increase the training shot, the training episode violates too much from the test episode, thus causing the performance drop.

We compare *Transductive Propagation Network* with several recent approaches in Table 1. Among them, MAML [2], transductive Reptile [13] and our method are transduction methods while others are induction methods. We achieved state-of-the-art results for both aligned episode setting and "Higher Shot" setting with substantial improvements for 1-shot task. It can be seen that the gain of "Higher Shot" training for 1-shot task (+2.49%) is larger than that of 5-shot task (+0.47%). We explain this by the fact that 1-shot task is more data-hungry than 5-shot task.

4.2 *tiered*ImageNet few-shot classification

Similar to *mini*ImageNet , *tiered*ImageNet [15] is also a subset of ILSVRC-12 [17]. Compared with *mini*ImageNet , it has a larger number of classes from ILSVRC-12 (608 classes rather than 100 for *mini*ImageNet). However, different from *mini*ImageNet, *tiered*ImageNet has a hierarchical structure of broader categories corresponding to high-level nodes in the Imagenet [9]. The top hierarchy has

Table 1: Few-shot classification accuracies on *mini*ImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals.

		5-way Acc	
Model	Transduction	1-shot	5-shot
MAML [2]	Y	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
Reptile [13]	N	$47.07 \pm 0.26\%$	$62.74 \pm 0.37\%$
Reptile + Transduction [13]	Y	$49.97 \pm 0.32\%$	$65.99 \pm 0.58\%$
Prototypical Nets [20]	N	$46.14 \pm 0.77\%$	$65.77 \pm 0.70\%$
Prototypical Nets (Higher Way) [20]	N	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
Relation Net [21]	N	$51.38 \pm 0.82\%$	$67.07 \pm 0.69\%$
Transductive Propagation Network	Y	$\mathbf{52.23 \pm 0.86\%}$	$\mathbf{68.78 \pm 0.66\%}$
Transductive Propagation Network (Higher Shot)	Y	$\mathbf{54.72 \pm 0.84\%}$	$\mathbf{69.25 \pm 0.67\%}$

* "Higher Way" means using more classes in training episodes and "Higher Shot" means using more shots in training episodes. Top results are highlighted.

Table 2: Few-shot classification accuracies on *tiered*ImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals.

		5-way Acc	
Model	Transduction	1-shot	5-shot
MAML [2]	Y	$51.67 \pm 1.81\%$	$70.30 \pm 1.75\%$
Reptile [13]	N	$48.97 \pm 0.21\%$	$66.47 \pm 0.21\%$
Reptile + Transduction [13]	Y	$52.36 \pm 0.23\%$	$71.03 \pm 0.22\%$
Prototypical Nets [20]	N	$48.58 \pm 0.87\%$	$69.57 \pm 0.75\%$
Prototypical Nets (Higher Way) [20]	N	$53.31 \pm 0.89\%$	$\mathbf{72.69 \pm 0.74\%}$
Relation Net [21]	N	$\mathbf{54.48 \pm 0.93\%}$	$71.31 \pm 0.78\%$
Transductive Propagation Network	Y	$54.38 \pm 0.96\%$	$70.58 \pm 0.79\%$
Transductive Propagation Network (Higher Shot)	Y	$\mathbf{57.41 \pm 0.94\%}$	$\mathbf{71.55 \pm 0.74\%}$

* "Higher Way" means using more classes in training episodes and "Higher Shot" means using more shots in training episodes. Top results are highlighted.

34 categories, each including 10 to 30 subclasses. The top categories are divided into 20 training (351 classes), 6 validation (97 classes) and 5 test (160 classes) categories (see [15] for detail). The average number of examples in each class is 1281. This high-level split strategy ensures that the training classes are distinct from the test classes semantically. This is a more challenging and realistic few-shot setting since there is no assumption that training classes should be similar to test classes. Similar to *mini*ImageNet, we follow standard training and validation procedure.

The network structure and training hyper-parameters are the same as in *mini*ImageNet experments except that we cut the learning rate in half every 25,000 episodes. The reason for larger decay step is that *tiered*ImageNet has more classes and more examples in each class which needs larger training iterations. For episodic construction, we exploited various training shots in Figure 3 (b) and selected 5-shot for 5-way 1-shot task as well as 20-shot for 5-way 5-shot task.

Since *tiered*ImageNet is a new dataset and the comparing algorithms did not report results on it, we produced the *tiered*ImageNet results (Table 2) by running the corresponding codes released by the original authors. We achieved state-of-the-art results for 1-shot classification and promising results for 5-shot task. One reason for the inferior performance of 5-shot task may be that the *tiered*ImageNet contains more classes with more examples each class compared with *mini*ImageNet, which makes the low-data issue not so severe. Also, it is consistent with *mini*ImageNet that 1-shot classification gains larger improvements (+3.03%) compared with 5-shot classification (+0.97%) when "Higher Shot" training is adopted.

Table 3: Comparison with semi-supervised prototypical networks [15] on *mini*ImageNet using the labeled/unlabeled splits proposed by [15]. * denotes the results reported by [15].

Model	Use Unlabeled Set	5-way Acc 1-shot	5-way Acc 5-shot
Prototypical Nets [20]*	N	43.61 ± 0.27	59.08 ± 0.22
Soft k-Means [15]*	Y	50.09 ± 0.45	$\mathbf{64.59 \pm 0.28}$
Soft k-Means+Cluster [15]*	Y	49.03 ± 0.24	63.08 ± 0.18
Masked Soft k-Means [15]*	Y	$\mathbf{50.41 \pm 0.31}$	$\mathbf{64.39 \pm 0.24}$
Transductive Propagation Network	N	$\mathbf{50.15 \pm 0.86}$	64.16 ± 0.68

Table 4: Comparison with semi-supervised prototypical networks [15] on *tiered*ImageNet using the labeled/unlabeled splits proposed by [15]. * denotes the results reported by [15].

Model	Use Unlabeled Set	5-way Acc 1-shot	5-way Acc 5-shot
Prototypical Nets [20]*	N	46.52 ± 0.52	66.15 ± 0.22
Soft k-Means [15]*	Y	51.52 ± 0.36	$\mathbf{70.25 \pm 0.31}$
Soft k-Means+Cluster [15]*	Y	51.85 ± 0.25	69.42 ± 0.17
Masked Soft k-Means [15]*	Y	$\mathbf{52.39 \pm 0.44}$	$\mathbf{69.88 \pm 0.20}$
Transductive Propagation Network	N	$\mathbf{53.41 \pm 0.94}$	68.10 ± 0.78

4.3 Comparison with semi-supervised few-shot learning

Our setting is similar to semi-supervised setting [29, 32] in the sense that we both learn from labeled and unlabeled data. However, for transductive inference, we regard the entire test set as unlabeled set while semi-supervised learning usually needs a separate unlabeled set, such as in [15]. In this section, we compare our method with three variations of the semi-supervised prototypical networks [15].

For semi-supervised prototypical networks [15], they create an additional split to separate the images of each class into disjoint labeled and unlabeled sets. Specifically, for *tiered*ImageNet, 10% of the images of each class are sampled to form the labeled split. The remaining 90% can only be used as the unlabeled portion of episodes. Likewise, the labeled ratio for *mini*ImageNet is 40%.

For our method, we only use the labeled split in training and test episodes. Experimental results are shown in Table 3 and Table 4. All methods construct the 5-way 5-shot training episodes for a fair comparison. It can be seen that our method achieves comparable or higher performance compared to [15] with only 10% or 40% of the training data. This result demonstrates that if the test data is available to access, transductive inference can make good predictions without collecting the large set of unlabeled data from training classes. Also, we can loosen our burden to get the unlabeled data from test classes, since it is even harder to obtain data from novel classes.

5 Conclusion

In this work, we proposed the transductive setting for few-shot learning. Our proposed approach, namely *Transductive Propagation Network* (TPN), utilizes the entire test set for transductive inference. Specifically, our approach is composed of four steps: feature embedding, graph construction, label propagation and loss computation. Graph construction is a key step that produces episodic-wise and example-wise parameters to exploit the manifold structure in each episode. In our method, all parameters are learned end-to-end using cross-entropy loss with respect to the ground truth labels and the predicted scores in the query set. We obtain state-of-the-art results on *mini*ImageNet and promising results on *tiered*ImageNet. Also, without using the additional unlabeled set, we achieve comparable results with semi-supervised method. In future work, we are going to explore the episodic-wise distance metric rather than only using episodic-wise parameters for the Euclidean distance.

References

[1] Fan RK Chung. Spectral graph theory (cbms regional conference series in mathematics, no. 92). 1996.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[3] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015.

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

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

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

[7] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning*, volume 99, pages 200–209, 1999.

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

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[10] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Conference of the Cognitive Science Society*, volume 33, 2011.

[11] Yoonho Lee and Seungjin Choi. Meta-learning with adaptive layerwise metric and subspace. *arXiv preprint arXiv:1801.05558*, 2018.

[12] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

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

[14] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2017.

[15] 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. *International Conference on Learning Representations*, 2018.

[16] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *Advances in Neural Information Processing Systems*, pages 46–54, 2013.

[17] 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.

[18] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[20] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017.

[21] 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 *Computer Vision and Pattern Recognition*, 2018.

[22] Sebastian Thrun and Lorien Pratt. Learning to learn. 1998.

[23] Vladimir Vapnik. *Statistical learning theory. 1998*. Wiley, New York, 1998.

[24] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[25] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. In *International Conference on Machine Learning*, pages 985–992. ACM, 2006.

[26] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Computer Vision and Pattern Recognition*, 2018.

[27] Mingrui Wu and Bernhard Scholkopf. Transductive classification via local learning regularization. In *Artificial Intelligence and Statistics*, pages 628–635, 2007.

[28] Zhongwen Xu, Linchao Zhu, and Yi Yang. Few-shot object recognition from machine-labeled web images. In *Computer Vision and Pattern Recognition*, 2017.

[29] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pages 40–48, 2016.

[30] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pages 321–328, 2004.

[31] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation.

[32] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.