Learning to Propagate Labels: Transductive Propagation Network for Few-shot Learning

Yanbin Liu[1], **Juho Lee**[2,3], **Minseop Park**[3], **Saehoon Kim**[3], **Eunho Yang**[4],
Sungju Hwang[4] **& Yi Yang**[1]
[1]CAI, University of Technology Sydney, [2]University of Oxford, [3]AITRICS, [4]KAIST
csyanbin@gmail.com, juho.lee@stats.ox.ac.uk,
{mike_seop, shkim}@aitrics.com, {eunhoy, sjhwang82}@kaist.ac.kr,
Yi.Yang@uts.edu.au

Abstract

The goal of few-shot learning is to learn a classifier that generalizes well even when trained with a limited number of training instances per class. The recently introduced meta-learning approaches tackle this problem by learning a generic classifier across a large number of multiclass classification tasks and generalizing the model to a new task. Yet, even with such meta-learning, the low-data problem in the novel classification task still remains. In this paper, we propose *Transductive Propagation Network* (TPN), a novel meta-learning framework for transductive inference that classifies the entire test set at once to alleviate the low-data problem. Specifically, we propose to *learn to propagate labels* from labeled instances to unlabeled test instances, by learning a graph construction module that exploits the manifold structure in the data. TPN jointly learns both the parameters of feature embedding and the graph construction in an end-to-end manner. We validate TPN on multiple benchmark datasets, on which it largely outperforms existing few-shot learning approaches and achieves the state-of-the-art results.

1 Introduction

Recent breakthroughs in deep learning (Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2015; He *et al.*, 2016) highly rely on the availability of large amounts of labeled data. However, this reliance on large data increases the burden of data collection, which hinders its potential applications to the low-data regime where the labeled data is rare and difficult to gather. On the contrary, humans have the ability to recognize new objects after observing only one or few instances (Lake *et al.*, 2011). For example, children can generalize the concept of "apple" after given a single instance of it. This significant gap between human and deep learning has reawakened the research interest on few-shot learning (Vinyals *et al.*, 2016; Snell *et al.*, 2017; Finn *et al.*, 2017; Ravi and Larochelle, 2017; Lee and Choi, 2018; Xu *et al.*, 2017; Wang *et al.*, 2018).

Few-shot learning aims to learn a classifier that generalizes well with a few examples of each of these classes. Traditional techniques such as fine-tuning (Jia *et al.*, 2014) that work well with deep learning models would severely overfit on this task (Vinyals *et al.*, 2016; Finn *et al.*, 2017), since a single or only a few labeled instances would not accurately represent the true data distribution and will result in learning classifiers with high variance, which will not generalize well to new data.

In order to solve this overfitting problem, Vinyals *et al.* (2016) proposed a meta-learning strategy which learns over diverse classification tasks over large number of episodes rather than only on the target classification task. In each episode, the algorithm learns the embedding of the few labeled examples (the *support set*), which can be used to predict classes for the unlabeled points (the *query set*) by distance in the embedding space. 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. This episodic meta-learning strategy, due to its generalization performance, has been adapted by many follow-up work on few-shot learning. Finn *et al.* (2017) learned a good initialization that can

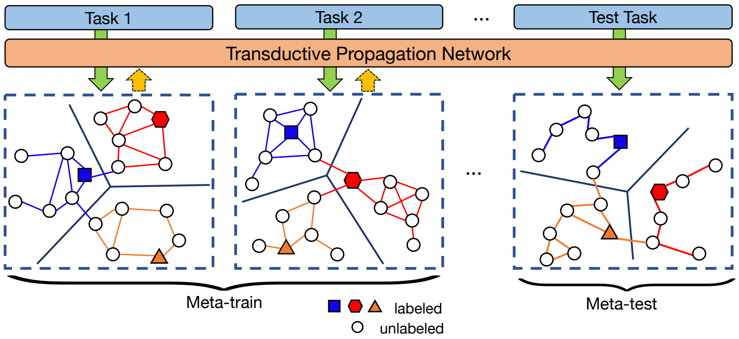


Figure 1: A conceptual illustration of our transductive meta-learning framework, where lines between nodes represent graph connections and their colors represent the potential direction of label propagation. The neighborhood graph is episodic-wisely trained for transductive inference.

adapt quickly to the target tasks. Snell *et al.* (2017) used episodes to train a good representation and predict classes by computing Euclidean distance with respect to class prototypes.

Although episodic strategy is an effective approach for few-shot learning as it aims at generalizing to unseen classification tasks, the fundamental difficulty with learning with scarce data remains for a novel classification task. One way to achieve larger improvements with limited amount of training data is to consider relationships between instances in the test set and thus predicting them as a whole, which is referred to as transduction, or transductive inference. In previous work (Joachims, 1999; Zhou *et al.*, 2004; Vapnik, 1999), transductive inference has shown to outperform inductive methods which predict test examples one by one, especially in small training sets. One popular approach for transduction is to construct a network on both the labeled and unlabeled data, and propagate labels between them for joint prediction. However, the main challenge with such label propagation (and transduction) is that the label propagation network is often obtained without consideration of the main task, since it is not possible to learn them at the test time.

Yet, with the meta-learning by episodic training, we can learn the label propagation network as the query examples sampled from the training set can be used to simulate the real test set for transductive inference. Motivated by this finding, we propose *Transductive Propagation Network* (TPN) to deal with the low-data problem. Instead of applying the inductive inference, 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 and graph construction parameters. Finally, all parameters can be updated end-to-end using backpropagation.

The main contribution of this work is threefold.

- To the best of our knowledge, we are the first to model transductive inference explicitly in few-shot learning. Although Nichol *et al.* (2018) experimented with a transductive setting, they only share information between test examples by batch normalization rather than directly proposing a transductive model.
- In transductive inference, we propose to *learn to propagate labels* between data instances for unseen classes via episodic meta-learning. This learned label propagation graph is shown to significantly outperform naive heuristic-based label propagation methods (Zhou *et al.*, 2004).
- We evaluate our approach on two benchmark datasets for few-shot learning, namely *mini*ImageNet and *tiered*ImageNet. The experimental results show that our *Transductive Propagation Network* outperforms the state-of-the-art methods on both datasets. Also, with semi-supervised learning, our algorithm achieves even higher performance, outperforming all semi-supervised few-shot learning baselines.

2 Related work

Meta-learning In recent works, few-shot learning often follows the idea of meta-learning (Schmidhuber, 1987; Thrun and Pratt, 2012). 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 (Finn *et al.*, 2017) aims to find more transferable representations with sensitive parameters. A first-order meta-learning approach named Reptile is proposed by Nichol *et al.* (2018). It is closely related to first-order MAML but does not 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 updateand 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 (Vinyals *et al.*, 2016) produce a weighted nearest neighbor classifier given the support set and adjust feature embedding according to the performance on the query set. Prototypical networks (Snell *et al.*, 2017) 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 Ren *et al.* (2018) to deal with semi-supervised few-shot learning. Relation Network (Sung *et al.*, 2018) 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 (Vapnik, 1999). Transductive Support Vector Machines (TSVMs) (Joachims, 1999) is a margin-based classification method that minimizes errors of a particular test set. It shows substantial improvements over inductive methods, especially for small training sets. Another category of transduction methods involves graph-based methods (Zhou *et al.*, 2004; Wang and Zhang, 2006; Rohrbach *et al.*, 2013; Fu *et al.*, 2015). Label propagation is used in Zhou *et al.* (2004) to transfer labels from labeled to unlabeled data instances guided by the weighted graph. Label propagation is sensitive to variance parameter σ, so Linear Neighborhood Propagation (LNP) (Wang and Zhang, 2006) constructs approximated Laplacian matrix to avoid this issue. In Zhu and Ghahramani (2002), minimum spanning tree heuristic and entropy minimization are used to learn the parameter σ. In all these prior work, the graph construction is done on a pre-defined feature space using manually selected hyperparamters since it is not possible to learn them at test time. Our approach, on the other hand, is able to learn the graph construction network since it is a meta-learning framework with episodic training, where at each episode we simulate the test set with a subset of the training set.

In few-shot learning, Nichol *et al.* (2018) experiments with a transductive setting and shows improvements. However, they only share information between test examples via batch normalization (Ioffe and Szegedy, 2015) rather than explicitly model the transductive setting as in our algorithm.

3 Main approach

In this section, we introduce the proposed algorithm that utilizes the manifold structure of the given few-shot classification task to improve the performance.

3.1 Problem definition

We follow the episodic paradigm (Vinyals *et al.*, 2016) that effectively trains a meta-learner for few-shot classification tasks, which is commonly employed in various literature (Snell *et al.*, 2017; Finn *et al.*, 2017; Nichol *et al.*, 2018; Sung *et al.*, 2018; Mishra *et al.*, 2018). Given a relatively large

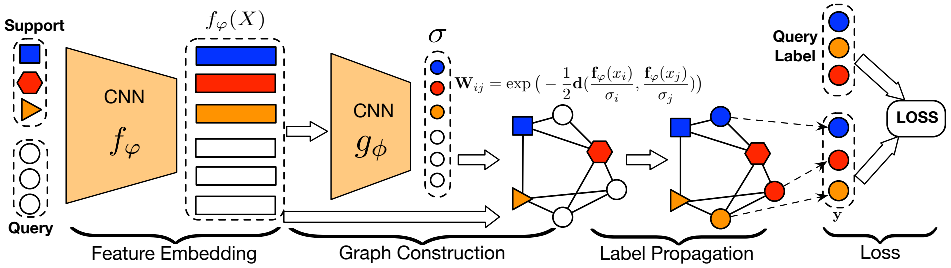

Support
Query
CNN
f_φ
$f_\varphi(X)$
CNN
g_ϕ
σ
$\mathbf{W}_{ij} = \exp\big(-\frac{1}{2}\mathbf{d}(\frac{\mathbf{f}_\varphi(x_i)}{\sigma_i}, \frac{\mathbf{f}_\varphi(x_j)}{\sigma_j})\big)$
Query Label
LOSS
y
Feature Embedding
Graph Construction
Label Propagation
Loss

Figure 2: The overall framework of our algorithm in which the manifold structure of the entire query set helps to learn better decision boundary. The proposed algorithm is composed of four components: feature embedding, graph construction, label propagation, and loss generation.

labeled dataset with a set of classes $\mathcal{C}_{train}$, the objective of this setting is to train classifiers for an unseen set of novel classes $\mathcal{C}_{test}$, for which only a few labeled examples are available.

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 $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{N\times K}, y_{N\times K})\}$, while the *query set* $\mathcal{Q} = \{(\mathbf{x}_1^*, y_1^*), (\mathbf{x}_2^*, y_2^*), \ldots, (\mathbf{x}_T^*, y_T^*)\}$ includes different examples from the same N classes. Here, the support set $\mathcal{S}$ in each episode serves as the labeled training set on which the model is trained to minimize the loss of its predictions for the query set $\mathcal{Q}$. This procedure mimics training classifiers for $\mathcal{C}_{test}$ and goes episode by episode until convergence.

Meta-learning implemented by the episodic training reasonably performs well to few-shot classification tasks. Yet, due to the lack of labeled instances (K is usually very small) in the support set, we observe that a reliable classifier is still difficult to be obtained. This motivates us to consider a transductive setting that utilizes 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 provide more reliable generalization property.

3.2 Transductive Propagation Network (TPN)

We introduce *Transductive Propagation Network* (TPN) illustrated in Figure 2, which consists of four components: feature embedding with a convolutional neural network; graph construction that produces example-wise parameters to exploit the manifold structure; label propagation that spreads labels from the support set $\mathcal{S}$ to the query set $\mathcal{Q}$; a loss generation step that computes a cross-entropy loss between propagated labels and the ground-truths on $\mathcal{Q}$ to jointly train all parameters in the framework.

3.2.1 Feature embedding

We employ a convolutional neural network f_φ to extract features of an input $\mathbf{x}_i$, where $f_\varphi(\mathbf{x}_i; \varphi)$ refers to the feature map and φ indicates a parameter of the network. Despite the generality, we adopt the same architecture used in several recent works (Snell *et al.*, 2017; Sung *et al.*, 2018; Vinyals *et al.*, 2016). By doing so, we can provide more fair comparisons in the experiments, highlighting the effects of transductive approach. 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 (Ioffe and Szegedy, 2015), a ReLU nonlinearity and a 2×2 max-pooling layer. We use the same embedding function f_φ for both the support set $\mathcal{S}$ and the query set $\mathcal{Q}$.

3.2.2 Graph construction

Manifold learning (Chung and Graham, 1997; Zhou *et al.*, 2004; Yang *et al.*, 2016) discovers the embedded low-dimensional subspace in the data, where it is critical to choose an appropriate neighborhood graph. A common choice is Gaussian similarity function:

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

where $d(\cdot,\cdot)$ is a distance measure (e.g., Euclidean distance) and σ is the length scale parameter. The neighborhood structure behaves differently with respect to various σ, which means that it needs to carefully select the optimal σ for the best performance of label propagation (Wang and Zhang, 2006; Zhu and Ghahramani, 2002). In addition, we observe that there is no principled way to tune the scale parameter in meta-learning framework, though there exist some heuristics for dimensionalty reduction methods (Zelnik-Manor and Perona, 2004; Sugiyama, 2007).

To obtain a proper neighborhood graph in meta-learning, we propose a graph construction module built on the union set of support set and query set: $\mathcal{S} \cup \mathcal{Q}$. This module is composed of a convolutional neural network g_ϕ which takes the feature map $f_\varphi(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathcal{S} \cup \mathcal{Q}$ to produce an *example-wise* length-scale parameter $\sigma_i = g_\phi(f_\varphi(\mathbf{x}_i))$. Note that the scale parameter is determined example-wisely and learned in an episodic training procedure, which adapts well to different tasks and makes it suitable for few-shot learning. The graph construction module consists of two convolutional blocks (64 and 1 filters) and two fully-connected layers (8 and 1 neurons), as shown in Appendix A. Our similarity function is then defined as follows:

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

where $W \in R^{(N\times K+T)\times(N\times K+T)}$ for all instances in $\mathcal{S} \cup \mathcal{Q}$. We only keep the k-max values in each row of W to construct a k-nearest neighbour graph. Then we apply the normalized graph Laplacians (Chung and Graham, 1997) on W, that is, $S = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with its (i,i)-value to be the sum of the i-th row of W.

3.2.3 Label propagation

We now describe how to get predictions for the query set $\mathcal{Q}$ using label propagation, before the last cross-entropy loss step. 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 $\mathbf{x}_i$ is from the support set and labeled as $y_i = j$, otherwise $Y_{ij} = 0$. Starting from Y, label propagation iteratively determines the unknown labels of instances in 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 $F_t \in \mathcal{F}$ denotes the predicted labels at the timestamp t, S denotes the normalized weight, and $\alpha \in (0,1)$ controls the amount of propagated information. It is well known that the sequence $\{F_t\}$ converges to $F^* = (I - \alpha S)^{-1}Y$, where I is the identity matrix (Zhou *et al.*, 2004). We directly utilize this result for the label propagation, making a whole episodic meta-learning procedure more efficient in practice.

3.2.4 Classification loss generation

The objective of this step is to compute the classification loss between the predictions of the query set via label propagation and the ground-truths. We compute the cross-entropy loss between predicted scores F^* and ground-truth labels from $\mathcal{Q}$ to learn all parameters in an end-to-end fashion, where F^* is converted to probabilistic score using softmax:

$$P(\tilde{y_i} = j|\mathbf{x}_i) = \frac{\exp(F^*_{ij})}{\sum_{j=1}^{N}\exp(F^*_{ij})}. \tag{4}$$

Here, $\tilde{y_i}$ denotes the final predicted label for ith instance in the query set and F^*_{ij} denotes the jth component of predicted label from label propagation. Then the loss function is computed as:

$$J(\varphi,\phi) = \sum_{i=1}^{T}\sum_{j=1}^{N} -\mathbb{I}(y_i == j)\log(P(\tilde{y_i} = j|\mathbf{x}_i)), \tag{5}$$

where y_i means the ground-truth label of $\mathbf{x}_i$ and $\mathbb{I}(b)$ is an indicator function, $\mathbb{I}(b) = 1$ if b is true and 0 otherwise.

Note that in Equation (5), the loss is dependent on two set of parameters φ, ϕ (even though the dependency is implicit through F^*_{ij}). All these parameters are jointly updated by the episodic training in an end-to-end manner.

4 Experiments

We evaluate and compare our TPN with state-of-the-art approaches on two datasets, i.e., *mini*ImageNet (Ravi and Larochelle, 2017) and *tiered*ImageNet (Ren *et al.*, 2018). The former is the most popular few-shot learning benchmark and the latter is a much larger dataset released recently for few-shot learning.

4.1 Datasets

***mini*ImageNet**. The *mini*ImageNet dataset is a collection of Imagenet (Krizhevsky *et al.*, 2012) for few-shot image recognition. It is composed of 100 classes randomly selected from Imagenet with each class containing 600 examples. 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 (2017), which includes 64 classes for training, 16 for validation, and 20 for test. All images are resized to 84×84 pixels.

***tiered*ImageNet**. Similar to *mini*ImageNet , *tiered*ImageNet (Ren *et al.*, 2018) is also a subset of Imagenet (Krizhevsky *et al.*, 2012), but it has a larger number of classes from ILSVRC-12 (608 classes rather than 100 for *mini*ImageNet). Different from *mini*ImageNet, it has a hierarchical structure of broader categories corresponding to high-level nodes in Imagenet. The top hierarchy has 34 categories, which are divided into 20 training (351 classes), 6 validation (97 classes) and 8 test (160 classes) categories. 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. Similarly, all images are resized to 84×84 pixels.

4.2 Experimental Setup

For fair comparison with other methods, we adopt a widely-used CNN (Finn *et al.*, 2017; Snell *et al.*, 2017) as the feature embedding function f_φ (Section 3.2.1). The hyper-parameter k of k-nearest neighbour graph (Section 3.2.2) is set to 20 and α of label propagation is set to 0.99, as suggested in Zhou *et al.* (2004).

Following Snell *et al.* (2017), we adopt the episodic training procedure, i.e, we sample a set of N-way K-shot training tasks to mimic the N-way K-shot test problems. Moreover, Snell *et al.* (2017) proposed a "Higher Way " training strategy which used more training classes in each episode than test case. However, we find that it is beneficial to train with more examples than test phase (Appendix B.1). This is denoted as "Higher Shot" in our experiments. For 1-shot and 5-shot test problem, we adopt 5-shot and 10-shot training respectively. In all settings, the query number is set to 15 and the performance are averaged over 600 randomly generated episodes from the test set.

All our models were trained with Adam (Kingma and Ba, 2015) and an initial learning rate of 10^{-3}. For *mini*ImageNet, we cut the learning rate in half every $10,000$ episodes and for *tiered*ImageNet, we cut the learning rate 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. We ran the training process until the validation loss reached a plateau.

4.3 Few-shot Learning Results

We compare our method with several state-of-the-art approaches in various settings. Even though the transductive method has never been used explicitly, batch normalization layer was used transductively to share information between test examples. For example, in Finn *et al.* (2017); Nichol *et al.* (2018), they use the query batch statistics rather than global BN parameters for the prediction, which leads to performance gain in the query set. Besides, we propose two simple transductive methods as baselines that explicitly utilize the query set. First, we propose the MAML+Transduction with slight modification of loss function to: $\mathcal{J}(\theta) = \sum_{i=1}^{T} \mathbf{y}_i \log \mathbb{P}(\widehat{\mathbf{y}}_i|\mathbf{x}_i) + \sum_{i,j=1}^{N\times K+T} W_{ij}\|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j\|_2^2$

Table 1: Few-shot classification accuracies on *mini*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

		5-way Acc		10-way Acc	
Model	Transduction	1-shot	5-shot	1-shot	5-shot
MAML (Finn *et al.*, 2017)	BN	48.70	63.11	31.27	46.92
MAML+Transduction	Yes	50.83	66.19	31.83	48.23
Reptile (Nichol *et al.*, 2018)	No	47.07	62.74	31.10	44.66
Reptile + BN (Nichol *et al.*, 2018)	BN	49.97	65.99	32.00	47.60
PROTO NET (Snell *et al.*, 2017)	No	46.14	65.77	32.88	49.29
PROTO NET (Higher Way) (Snell *et al.*, 2017)	No	49.42	68.20	34.61	50.09
RELATION NET (Sung *et al.*, 2018)	BN	51.38	67.07	34.86	47.94
Label Propagation	Yes	52.31	68.18	35.23	51.24
TPN	Yes	**53.75**	**69.43**	**36.62**	**52.32**
TPN (Higher Shot)	Yes	**55.51**	**69.86**	**38.44**	**52.77**

* "Higher Way" means using more classes in training episodes. "Higher Shot" means using more shots in training episodes. "BN" means information is shared among test examples using batch normalization.

Table 2: Few-shot classification accuracies on *tiered*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

		5-way Acc		10-way Acc	
Model	Transduction	1-shot	5-shot	1-shot	5-shot
MAML (Finn *et al.*, 2017)	BN	51.67	70.30	34.44	53.32
MAML + Transduction	Yes	53.23	70.83	34.78	54.67
Reptile (Nichol *et al.*, 2018)	No	48.97	66.47	33.67	48.04
Reptile + BN (Nichol *et al.*, 2018)	BN	52.36	71.03	35.32	51.98
PROTO NET (Snell *et al.*, 2017)	No	48.58	69.57	37.35	57.83
PROTO NET (Higher Way) (Snell *et al.*, 2017)	No	53.31	72.69	38.62	58.32
RELATION NET (Sung *et al.*, 2018)	BN	54.48	71.31	36.32	58.05
Label Propagation	Yes	55.23	70.43	39.39	57.89
TPN	Yes	**57.53**	**72.85**	**40.93**	**59.17**
TPN (Higher Shot)	Yes	**59.91**	**73.30**	**44.80**	**59.44**

* "Higher Way" means using more classes in training episodes. "Higher Shot" means using more shots in training episodes. "BN" means information is shared among test examples using batch normalization.

for transductive inference. The additional term serves as transductive regularization. Second, the naive heuristic-based label propagation methods (Zhou *et al.*, 2004) is proposed to explicitly model the transductive inference.

Experimental results are shown in Table 1 and Table2. Transductive batch normalization methods tend to perform better than pure inductive methods except for the "Higher Way" PROTO NET. Label propagation without learning to propagate outperforms other baseline methods in most cases, which verifies the necessity of transduction. The proposed TPN achieves the state-of-the-art results and surpasses all the others with a large margin even when the model is trained with regular shots. When "Higher Shot" is applied, the performance of TPN continues to improve especially for 1-shot case. This confirms that our model effectively finds the episodic-wise manifold structure of test examples through learning to construct the graph for label propagation.

Another observation is that the advantages of 5-shot classification is less significant than that of 1-shot case. For example, in 5-way *mini*ImageNet , the absolute improvement of TPN over published state-of-the-art is 4.13% for 1-shot and 1.66% for 5-shot. To further investigate this, we experimented 5-way k-shot ($k = 1, 2, \cdots, 10$) experiments. The results are shown in Figure 3. Our TPN performs consistently better than other methods with varying shots. Moreover, it can be seen that TPN outperforms other methods with a large margin in lower shots. With the shot increase, the advantage of transduction narrows since more labelled data are used. This finding agrees with the results in TSVM (Joachims, 1999): when more training data are available, the bonus of transductive inference will be decreased.

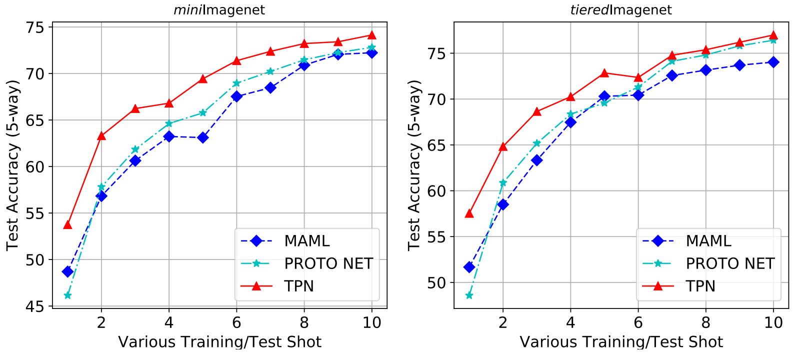


Figure 3: 5-way performance with various training/test shots.

Table 3: Semi-supervised comparison on *mini*ImageNet and *tiered*ImageNet.

	5-way *mini*ImageNet		5-way *tiered*ImageNet	
Model	1-shot	5-shot	1-shot	5-shot
Soft k-Means (Ren *et al.*, 2018)	50.09 ± 0.45	64.59 ± 0.28	51.52 ± 0.36	70.25 ± 0.31
Soft k-Means+Cluster (Ren *et al.*, 2018)	49.03 ± 0.24	63.08 ± 0.18	51.85 ± 0.25	69.42 ± 0.17
Masked Soft k-Means (Ren *et al.*, 2018)	50.41 ± 0.31	64.39 ± 0.24	52.39 ± 0.44	69.88 ± 0.20
TPN-semi	**52.78 ± 0.27**	**66.42 ± 0.21**	**55.74 ± 0.29**	**71.01 ± 0.23**

4.4 Comparison with semi-supervised few-shot learning

The main difference of traditional semi-supervised learning and transduction is the source of unlabeled data. Transductive methods directly use test set as unlabeled data while semi-supervised learning usually has an extra unlabeled set. In order to compare with semi-supervised methods, we propose a semi-supervised version of TPN, named TPN-semi, which classifies one test example each time by propagating labels from the labeled set and extra unlabeled set.

We use *mini*ImageNet and *tiered*ImageNet with the labeled/unlabeled data split proposed by Ren *et al.* (2018). Specifically, they split the images of each class into disjoint labeled and unlabeled sets. For *mini*ImageNet, the ratio of labeled/unlabeled data is 40% and 60% in each class. Likewise, the ratio is 10% and 90% for *tiered*ImageNet. All semi-supervised methods (including TPN-semi) sample support/query data from the labeled set (e.g, 40% from *mini*ImageNet) and sample unlabeled data from the unlabeled sets (e.g, 60% from *mini*ImageNet).

Following Ren *et al.* (2018), we report the average accuracy over 10 random labeled/unlabeled splits and the uncertainty computed in standard error. Results are shown in Table 3. It can be seen that TPN-semi outperforms all other algorithms with a large margin, especially for 1-shot case. Although TPN is originally designed to perform transductive inference, we show that it can be successfully adapted to semi-supervised learning tasks with little modification. In certain cases where we can not get all test data, the TPN-semi can be used as an effective alternative algorithm.

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 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 prediction scores in the query set. We obtained the state-of-the-art results on *mini*ImageNet and *tiered*ImageNet. Also, the semi-supervised adaptation of our algorithm achieved higher results than other semi-supervised methods. In future work, we are going to explore the episodic-wise distance metric rather than only using example-wise parameters for the Euclidean distance.

References

Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

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.

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.

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.

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.

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.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, 2015.

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.

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.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pages 2933–2942, 2018.

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

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

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

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.

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

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

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

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

Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

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.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

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.

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

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

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

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

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, 2004.

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.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002.

A Graph Construction Structure

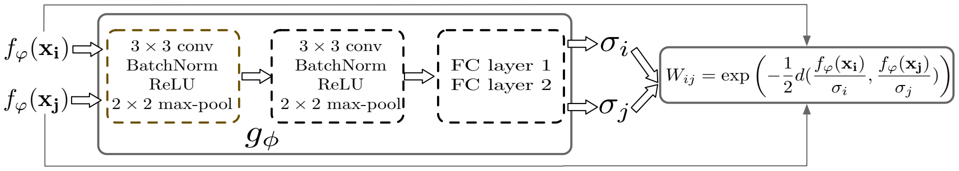

$f_\varphi(\mathbf{x_i})$
$f_\varphi(\mathbf{x_j})$
3×3 conv
BatchNorm
ReLU
2×2 max-pool
3×3 conv
BatchNorm
ReLU
2×2 max-pool
FC layer 1
FC layer 2
g_ϕ
σ_i
σ_j
$W_{ij} = \exp\left(-\frac{1}{2}d(\frac{f_\varphi(\mathbf{x_i})}{\sigma_i}, \frac{f_\varphi(\mathbf{x_j})}{\sigma_j})\right)$

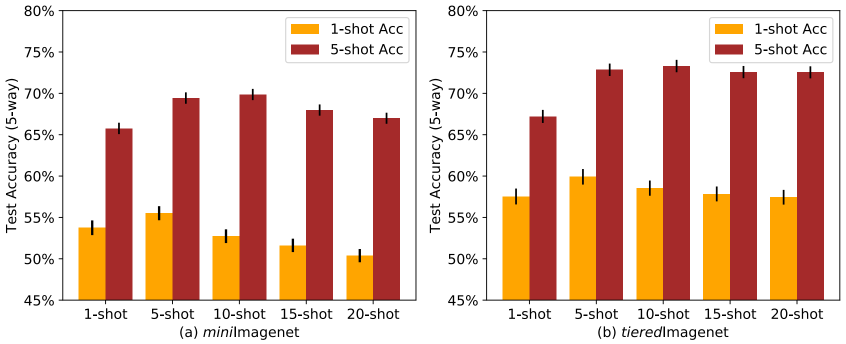
Figure 4: Detailed architecture of the graph construction module, in which the length-scale parameter is example-wisely determined.

The graph construction module described in Section 3.2.2 consists of two convolutional blocks (64 and 1 filters) and two fully-connected layers (8 and 1 neurons), as shown in Figure 4.

B Ablation Study

In this section, we performed several ablation studies with respect to training shots and query number.

B.1 Training Shots



(a) *mini*Imagenet

(b) *tiered*Imagenet

Figure 5: 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.

B.2 Query Number

Table 4: Accuracy with various query numbers

	*mini*ImageNet 1-shot					
	5	10	15	20	25	30
Train=15	52.29	52.95	53.75	53.92	54.57	54.47
Test=15	53.53	53.72	53.75	52.79	52.84	52.47
Train=Test	51.94	53.47	53.75	54.00	53.59	53.32
	*mini*ImageNet 5-shot					
	5	10	15	20	25	30
Train=15	66.97	69.30	69.43	69.92	70.54	70.36
Test=15	68.50	68.85	69.43	69.26	69.12	68.89
Train=Test	67.55	69.22	69.43	69.85	70.11	69.94

At first, we designed three experiments to study the influence of the query number in both training and test phase: (1) fix training query to 15; (2) fix test query to 15; (3) training query equals test query. The results are shown in Table 4. Some conclusions can be drawn from this experiment: (1) When training query is fixed, increasing the test query will lead to the performance gain. Moreover, even a small test query (e.g., 5) can yield good performance; (2) When test query is fixed, the performance is relatively stable with various training query numbers; (3) If the query number of training matches test, the performance can also be improved with increasing number.