PTN: A Poisson Transfer Network for Semi-supervised Few-shot Learning

Huaxi Huang¹, Junjie Zhang², Jian Zhang¹, Qiang Wu¹, Chang Xu³

¹University of Technology Sydney, Sydney NSW 2007, Australia
²Shanghai University, Shanghai, China
³The University of Sydney, Sydney NSW 2006, Australia

{Huaxi.Huang.student., Jian.Zhang, Qiang.Wu}@uts.edu.au, junjie.zhang.avalon@gmail.com, c.xu@sydney.edu.au

Abstract

The predicament in semi-supervised few-shot learning (SS-FSL) is to maximize the value of the extra unlabeled data to boost the few-shot learner. In this paper, we propose a Poisson Transfer Network (PTN) to mine the unlabeled information for SSFSL from two aspects. First, the Poisson Merriman–Bence–Osher (MBO) model builds a bridge for the communications between labeled and unlabeled examples. This model serves as a more stable and informative classifier than traditional graph-based SSFSL methods in the message-passing process of the labels. Second, the extra unlabeled samples are employed to transfer the knowledge from base classes to novel classes through contrastive learning. Specifically, we force the augmented positive pairs close while push the negative ones distant. Our contrastive transfer scheme implicitly learns the novel-class embeddings to alleviate the over-fitting problem on the few labeled data. Thus, we can mitigate the degeneration of embedding generality in novel classes. Extensive experiments indicate that PTN outperforms the state-of-the-art few-shot and SSFSL models on miniImageNet and tieredImageNet benchmark datasets.

Introduction

Few-shot learning (Miller, Matsakis, and Viola 2000; Fei-Fei, Fergus, and Pietro 2006; Vinyals et al. 2016) aims to learn a model that generalizes well with a few instances of each novel class. In general, a few-shot learner is firstly trained on a substantial annotated dataset, also noted as the base-class set, and then adapted to unseen novel classes with a few labeled instances. During the evaluation, a set of few-shot tasks are fed to the learner, where each task consists of a few support (labeled) samples and a certain number of query (unlabeled) data. This research topic has been proved immensely appealing in the past few years, as a large number of few-shot learning methods are proposed from various perspectives. Mainstream methods can be roughly grouped into two categories. The first one is learning from episodes (Vinyals et al. 2016), also known as meta-learning, which adopts the base-class data to create a set of episodes. Each episode is a few-shot learning task, with support and query samples that simulate the evaluation procedure. The second type is the transfer-learning based method, which focuses on learning a decent classifier by transferring the domain knowledge from a model pre-trained on the large base-class set (Chen et al. 2018; Qiao et al. 2018). This paradigm decouples the few-shot learning progress into representation learning and classification, and has shown favorable performance against meta-learning methods in recent works (Tian et al. 2020; Ziko et al. 2020). Our method shares somewhat similar motivation with transfer-learning based methods and proposes to utilize the extra unlabeled novel-class data and a pre-trained embedding to tackle the few-shot problem.

Compared with collecting labeled novel-class data, it is much easier to obtain abundant unlabeled data from these classes. Therefore, semi-supervised few-shot learning (SS-FSL) (Ren et al. 2018; Liu et al. 2018; Li et al. 2019b; Yu et al. 2020) is proposed to combine the auxiliary information from labeled base-class data and extra unlabeled novel-class data to enhance the performance of few-shot learners. The core challenge in SSFSL is how to fully explore the auxiliary information from these unlabeled. Previous SSFSL works indicate that graph-based models (Liu et al. 2018; Ziko et al. 2020) can learn a better classifier than inductive ones (Ren et al. 2018; Li et al. 2019b; Yu et al. 2020), since these methods directly model the relationship between the labeled and unlabeled samples during the inference. However, current graph-based models adopt the Laplace learning (Zhu, Ghahramani, and Lafferty 2003) to conduct label propagation, the solutions of Laplace learning develop localized spikes near the labeled samples but are almost constant far from the labeled samples, i.e., label values are not propagated well, especially with few labeled samples. Therefore, these models suffer from the underdeveloped message-passing capacity for the labels. On the other hand, most SSFSL methods adapt the feature embedding pre-trained on base-class data (meta- or transfer-pre-trained) as the novel-class embedding. This may lead to the embedding degeneration problem, as the pre-trained model is designed for the base-class recognition, it tends to learn the embedding that represents only base-class information, and lose information that might be useful outside base classes.

To address the above issues, we propose a novel transfer-learning based SSFSL method, named Poisson Transfer Network (PTN). Specifically, to improve the capacity of graph-based SSFSL models in message passing, we propose to revise the Poisson model tailored for few-shot problems by...
incorporating the query feature calibration and the Poisson MBO model. Poisson learning (Calder et al. 2020) has been provably more stable and informative than traditional Laplace learning in low label rate semi-supervised problems. However, directly employing Poisson MBO for SSFSL may suffer from the cross-class bias due to the data distribution drift between the support and query data. Therefore, we improve the Poisson MBO model by explicitly eliminating the cross-class bias before label inference. To tackle the novel-class embedding degeneration problem, we propose to transfer the pre-trained base-class embedding to the novel-class embedding by adopting unsupervised contrastive training (He et al. 2020; Chen et al. 2020) on the extra unlabeled novel-class data. Constraining the distances between the augmented positive pairs, while pushing the negative ones distant, the proposed transfer scheme captures the novel-class distribution implicitly. This strategy effectively avoids the possible overfitting of retraining feature embedding on the few labeled instances.

By integrating the Poisson learning and the novel-class specific embedding, the proposed PTN model can fully explore the auxiliary information of extra unlabeled data for SSFSL tasks. The contributions are summarized as follows:

- We propose a Poisson learning based model to improve the capacity of mining the relations between the labeled and unlabeled data for graph-based SSFSL.
- We propose to adapt unsupervised contrastive learning in the representation learning with extra unlabeled data to improve the generality of the pre-trained base-class embedding for novel-class recognition.
- Extensive experiments are conducted on two benchmark datasets to investigate the effectiveness of PTN, and PTN achieves state-of-the-art performance.

**Related Work**

**Few-Shot Learning**

As a representative of the learning methods with limited samples, e.g., weakly supervised learning (Lan, Yuen, and Chellappa 2017; Zhang et al. 2018), semi-supervised learning (Zhu, Ghahramani, and Lafferty 2003; Calder and Slepčev 2019), few-shot learning can be roughly grouped into two categories: meta-learning models and transfer-learning models. Meta-learning models adopt the episodic training mechanism (Vinyals et al. 2016), of which metric-based models optimize the transferable embedding of both auxiliary and target data, and queries are identified according to the embedding distances (Sung et al. 2018; Li et al. 2019a; Simon et al. 2020; Zhang et al. 2020). Meanwhile, meta-optimization models (Finn, Abbeel, and Levine 2017; Rusu et al. 2018) target at designing optimization-centered algorithms to adapt the knowledge from meta-training to meta-testing. Instead of separating base classes into a set of few-shot tasks, transfer-learning methods (Qiao et al. 2018; Gidaris and Komodakis 2018; Chen et al. 2018; Qi, Brown, and Lowe 2018) utilize all base classes to pre-train the few-shot model, which is then adapted to novel-class recognition. Most recently, Tian et al. (Tian et al. 2020) decouple the learning procedure into the base-class embedding pre-training and novel-class classifier learning. By adopting multivariate logistic regression and knowledge distillation, the proposed model outperforms the meta-learning approaches. Our proposed method is inspired by the transfer-learning framework, where we adapt this framework to the semi-supervised few-shot learning by exploring both unlabeled novel-class data and base-class data to boost the performance of few-shot tasks.

**Semi-Supervised Few-shot Learning (SSFSL)**

SSFSL aims to leverage the extra unlabeled novel-class data to improve the few-shot learning. Ren et al. (Ren et al. 2018) propose a meta-learning based framework by extending the prototypical network (Snell, Swersky, and Zemel 2017) with unlabeled data to refine class prototypes. LST (Li et al. 2019b) re-trains the base model using the unlabeled data with generated pseudo labels. During the evaluation, it dynamically adds the unlabeled sample with high prediction confidence into testing. In (Yu et al. 2020), TransMatch proposes to initialize the novel-class classifier with the pre-trained feature imprinting, and then employs MixMatch (Berthelot et al. 2019) to fine-tune the whole model with both labeled and unlabeled data. As closely related research to SSFSL, the transductive few-shot approaches (Liu et al. 2018; Kim et al. 2019; Ziko et al. 2020) also attempt to utilize unlabeled data to improve the performance of the few-shot learning. These methods adopt the entire query set as the unlabeled data and perform inference on all query samples together. For instance, TPN (Liu et al. 2018) employs graph-based transductive inference to address the few-shot problem, and a semi-supervised extension model is also presented in their work.

Unlike the above approaches, in this paper, we adopt the transfer-learning framework and propose to fully explore the extra unlabeled information in both classifier learning and embedding learning with different learning strategies.

**Methodology**

**Problem Definition**

In the standard few-shot learning, there exists a labeled support set $S$ of $C$ different classes, $S = \{ (x_s, y_s) \}_{s=1}^{K \times C}$, where $x_s$ is the labeled sample and $y_s$ denote its label. We use the standard basis vector $e_i \in \mathbb{R}^C$ represent the $i$-th class, i.e., $y_s \in \{ e_1, e_2, \ldots, e_C \}$. Given an unlabeled query sample $x_q$ from the query set $Q = \{ x_q \}_{q=1}^{V}$, the goal is to assign the query to one of the $C$ support classes. The labeled support set and unlabeled query set share the same label space, and the novel-class dataset $D_{novel}$ is thus defined as $D_{novel} = S \cup Q$. If $S$ contains $K$ labeled samples for each of $C$ categories, the task is noted as a $C$-way-$K$-shot problem. It is far from obtaining an ideal classifier with the limited annotated $S$. Therefore, few-shot models usually utilize a fully annotated dataset, which has similar data distribution but disjoint label space with $D_{novel}$ as an auxiliary dataset $D_{base}$, noted as the base-class set.

For the semi-supervised few-shot learning (SSFSL), we have an extra unlabeled support set $U = \{ x_u \}_{u=1}^{N}$. These
1) Pre-train Stage

- Base-Class Set
- Feature Embedding Pre-training
- Cross Entropy

2) Unsupervised Transfer Stage

- Novel-Class Unlabeled Set
- Unsupervised Embedding Transfer

3) Label Inference Stage

- Poisson Inference
- Labeled Test Sample
- Extra Unlabeled

Figure 1: The overview of the proposed PTN. We first pre-train a feature embedding $f_{b_0}$ from the base-class set using standard cross-entropy loss. This embedding is then fine-tuned with the external novel-class unlabeled data by adopting unsupervised transferring loss $\ell_{UT}$ to generate $f_0$. Finally, we revise a graph model named PoissonMBO to conduct the query label inference.

Representation Learning

The representation learning aims to learn a well-generalized novel-class embedding through Feature Embedding Pre-training and Unsupervised Embedding Transfer.

Feature Embedding Pre-training

On the left side of Figure 1, the first part of PTN is the feature embedding pre-training. By employing the cross-entropy loss between predictions and ground-truth labels in $D_{base}$, we train the base encoder $f_{b_0}$ in a fully-supervised way, which is the same as (Chen et al. 2018; Yu et al. 2020; Tian et al. 2020). This stage can generate powerful embedding for the downstream few-shot learner.

Unsupervised Embedding Transfer

Directly employ the pre-trained base-class embedding for the novel-class may suffer from the degeneration problem. However, retraining the base-class embedding with the limited labeled instances is easy to lead to overfitting. How can we train a novel-class embedding to represent things beyond labels when our only supervision is the limited labels? Our solution is unsupervised contrastive learning. Unsupervised learning, especially Contrastive learning (He et al. 2020; Chen et al. 2020), recently has shown great potential in representation learning for various downstream vision tasks, and most of these works training a model from scratch. However, unsupervised pre-trained models perform worse than fully-supervised pre-trained models. Unlike previous works, we propose to adopt contrastive learning to retrain the pre-trained embedding with the unlabeled novel data. In this way, we can learn a decent novel-class embedding by integrating the fully-supervised pre-trained scheme with unsupervised contrastive fine-tuning.

Specifically, for a minibatch of $n$ examples from the unlabeled novel-class subset $U_l = \{x_u\}_{u=1}^n$, randomly sampling two data augmentation operators $t, t' \in T$, we can generate a new feature set $Z = \{Z_t = \{f_{b_0}(t(x_u))\}_{u=1}^n\} \cup \{Z_{t'} = \{f_{b_0}(t'(x_u))\}_{u=1}^n\}$, resulting in $n$ pairs of feature points. We treat each feature pair from the same raw data input as the positive pair, and the other $2(n-1)$ feature points as negative samples. Then the contrastive loss for the minibatch is defined as

$$\ell_{cont} = -\frac{1}{n} \sum_{i,j=1, i \neq j} \log \frac{\exp(\cosine(z_i, z_j)/\tau)}{\sum_{k \neq i} \exp(\cosine(z_i, z_k)/\tau)},$$

where $z_i, z_j$ denote a positive feature pair from $Z$, $\tau$ is a temperature parameter, and $\cosine(\cdot)$ represents the cosine similarity. Then, we adopt a Kullback-Leibler divergence ($\ell_{KL}$) between two feature subset $Z_t$ and $Z_{t'}$ as the regulation term. Therefore, the final unsupervised embedding transfer loss $\ell_{UT}$ is defined as

$$\ell_{UT} = \ell_{cont} + \lambda \ell_{KL}(Z_t || Z_{t'}).$$

By training the extra unlabeled data with this loss, we can learn a robust novel-class embedding $f_0$ from $f_{b_0}$.

Poisson Label Inference

Previous studies (Zhu, Ghahramani, and Lafferty 2003; Zhou et al. 2004; Zhu, Lafferty, and Rosenfeld 2005; Liu...
et al. 2018; Ziko et al. 2020) indicate that the graph-based few-shot classifier has shown superior performance against inductive ones. Therefore, we propose constructing the classifier with a graph-based Poisson model, which adopts different optimizing strategy with representation learning. Poisson model (Calder et al. 2020) has been proved superior over traditional Laplace-based graph models (Zhu, Grahmani, and Lafferty 2003; Zhou et al. 2004) both theoretically and experimentally, especially for the low label rate semi-supervised problem. However, directly applying this model to the few-shot task will suffer from a cross-class bias challenge, caused by the data distribution bias between support data (including labeled support and unlabeled support data) and query data.

Therefore, we revise this powerful model by eliminating the support-query bias as the classifier. We explicitly propose a query feature calibration strategy before the final Poisson label inference. It is worth noticing that the proposed graph-based classifier can be directly appended to the pre-trained embedding without adopting the unsupervised embedding transfer training. We dob this baseline model as Decoupled Poisson Network (DPN).

**Query Feature Calibration** The support-query data distribution bias, also referred to as the cross-class bias (Liu, Song, and Qin 2020), is one of the reasons for the degeneracy of the few-shot learner. In this paper, we propose a simple but effective method to eliminate this distribution bias for Poisson graph inference. For a SSFS task, we fuse the labeled support set \( S \) and the extra unlabeled set \( U \) as the final support set \( B = S \cup U \). We denote the normalized embedded support feature set and query feature set as \( Z_b = \{ z_b \} \) and \( Z_q = \{ z_q \} \), the cross-class bias is defined as

\[
\Delta_{\text{cross}} = \mathbb{E}_{z_b \sim p_B} [z_b] - \mathbb{E}_{z_q \sim p_Q} [z_q] = \frac{1}{|B|} \sum_{b=1}^{|B|} z_b - \frac{1}{|Q|} \sum_{q=1}^{|Q|} z_q.
\]

(3)

We then add the bias \( \Delta_{\text{cross}} \) to query features. To such a degree, support-query bias is somewhat eliminated. After that, a Poisson MBO model is adopted to infer the query label.

**The Poisson Merriman–Bence–Osher Model** We denote the embedded feature set as \( Z_{\text{novel}} = Z_b \cup Z_q = \{ z_1, z_2, \ldots, z_m \} \) \((m = K \times C + N + V)\), where the first \( K \times C \) feature points belong to the labeled support set, the last \( V \) feature points belong to the query set, and the remaining \( N \) points denote the unlabeled support set. We build a graph with the feature points as the vertices, and the edge weight \( w_{ij} \) is the similarity between feature point \( z_i \) and \( z_j \) defined as \( w_{ij} = \exp \left( -4 |z_i - z_j|^2 / d_K(z_i)^2 \right) \), where \( d_K(z_i) \) is the distance between \( z_i \) and its \( K \)-th nearest neighbor. We set \( w_{ij} \geq 0 \) and \( w_{ij} = w_{ji} \). Correspondingly, we define the weight matrix \( W_z = [w_{ij}] \), the degree matrix as \( D = \text{diag}(d_i) = \sum_{j=1}^m w_{ij} \), and the normalized Laplacian as \( L = D - W \). As the first \( K \times C \) feature points have the ground-truth label, we use \( \bar{y} = \frac{1}{K \times C} \sum_{s=1}^{K \times C} y_s \) to denote the average label vector, and we let indicator \( \mathbb{I}_{ij} = 1 \) if \( i = j \), else \( \mathbb{I}_{ij} = 0 \). The goal of this model is to learn a classifier \( g : z \rightarrow \mathbb{R}^C \). By solving the Poisson equation:

\[
L g(z_i) = \sum_{j=1}^{K \times C} (y_j - \bar{y}) \mathbb{I}_{ij} \quad \text{for} \quad i = 1, \ldots, m,
\]

(4)

satisfying \( \sum_{j=1}^m \sum_{i=1}^m w_{ik} g(z_i) = 0 \), we can then result in the label prediction function \( g(z_i) = (g_1(z_i), g_2(z_i), \ldots, g_C(z_i)) \). The predict label \( \hat{y}_i \) of vertex \( z_i \) is then determined as \( \hat{y}_i = \arg \max_{y_j \in \{1, \ldots, C\}} \{ g_j(x_i) \} \).

Let \( G \) denote the set of \( m \times C \) matrix, which is the prediction label matrix of the all data. We concatenate the support label to form a label matrix \( Y = [y_a] \in \mathbb{R}^{C \times (K \times C)} \). Let \( A = [Y - \bar{y}, 0^{C \times (m-K \times C)}] \) denotes the initial label of all the data, in which all unlabeled data’s label is zero. The query label of Eq. (4) can be determined by:

\[
G^{tp+1} = G^{tp} + D^{-1}(A T - L G^{tp}),
\]

(5)

where \( G^{tp} \) denotes the predicted labels of all data at the timestamp \( tp \). We can get a stable classifier \( g \) with a certain number of iteration using Eq. (5). After that, we adopt a graph-cut method to improve the inference performance by incrementally adjusting the classifier’s decision boundary. The graph-cut problem is defined as

\[
\min_{g : Z 
rightarrow H, (g_i, z_i) \in o} \left\{ g^T L g - \mu \sum_{i=1}^{K \times C} (y_i - \bar{y}) \cdot g(z_i) \right\},
\]

(6)

where \( H = \{ e_1, e_2, \ldots, e_C \} \) denotes the annotated samples’ label set, \( (g_z) = \frac{1}{m} \sum_{i=1}^m g(z_i) \) is the fraction of vertices to each of \( C \) classes, and \( o = \{ o_1, o_2, \ldots, o_C \} T \in \mathbb{R}^C \) is the prior knowledge of the class size distribution that \( o_i \) is the fraction of data belonging to class \( i \). With the constraint \( (g_z) = o \), we can encode the prior knowledge into the Poisson Model. \( g^T L g = \frac{1}{2} \sum_{i,j=1}^m w_{ij}(g(i) - g(j))^2 \), this term is the graph-cut energy of the classification given by \( g = [g(z_1), g(z_2), \ldots, g(z_m)]^T \), widely used in semi-supervised graph models (Zhu, Grahmani, and Lafferty 2003; Zhu, Lafferty, and Rosenfeld 2004).

In Eq. (6), the solution will get discrete values, which is hard to solve. To relax this problem, we use the Merriman-Bence-Osher (MBO) scheme (Garcia-Cardona et al. 2014) by replacing the graph-cut energy with the Ginzburg-Landau approximation:

\[
\min_{g : Z \rightarrow \mathbb{R}^C} \left\{ \text{GL}_T(g) - \mu \sum_{i=1}^{K \times C} (y_i - \bar{y}) \cdot g(z_i) \right\},
\]

(7)

\[
\text{GL}_T(g) = g^T L g + \frac{1}{\tau} \sum_{i=1}^m \prod_{j=1}^C |g(z_i) - e_j|^2.
\]

In Eq. (7), \( \text{SP}(Z \rightarrow \mathbb{R}^C) \) represents the space of projections \( g : Z \rightarrow \mathbb{R}^C \), which allow the classifier \( g \) to take on any real values, instead of the discrete value from \( H \) in Eq. (6). More importantly, this leads to a more efficiently computation of the Poisson model. The Eq. (7) can be efficiently solved with alternates gradient decent strategy, as shown in lines 9-20 of Algorithm 1.
Proposed Algorithm

The overall proposed algorithm is summarized in Algorithm 1. Inputting the base-class set $D_{base}$, novel-class set $D_{novel}$, prior classes’ distribution $o$, and other parameters, PTN will predict the query samples’ label $G \in \mathbb{R}^{m \times C}$. The query label $y_q$ is then determined as $y_q = \arg \max_{1 \leq j \leq C} G_{qj}$. More specifically, once the encoder $f_0$ is learned using the base set $D_{base}$, we employ the proposed unsupervised embedding transfer method in Step 2 in Algorithm 1. After that, we build the graph with the feature set $Z_{novel}$ and compute the related matrices $W, D, L, A$ according to $Z_{novel}$.

**Algorithm 1: PTN for SSFSL**

<table>
<thead>
<tr>
<th>Input</th>
<th>$D_{base}, D_{novel} = S \cup U \cup Q$, $o, \mu, M_1, M_2, M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Query samples’ label prediction $G$</td>
</tr>
<tr>
<td>Train a base model $W_0 \circ f_0(x)$ with all samples and labels from $D_{novel}$.</td>
<td></td>
</tr>
<tr>
<td>Apply unsupervised embedding transfer method to fine-tune the $f_0$ with novel unlabeled data $U$ by using $\ell_{UT}$ in Eq. (2), and result in $f_0$.</td>
<td></td>
</tr>
<tr>
<td>Extract features on $D_{novel}$ as $Z_{novel}$.</td>
<td></td>
</tr>
<tr>
<td>Apply query feature calibration using Eq. (3); Compute $W, D, L, A$ according to $Z_{novel}$.</td>
<td></td>
</tr>
<tr>
<td>$G \leftarrow 0_{m \times C}$</td>
<td></td>
</tr>
<tr>
<td>PoissonMBO</td>
<td></td>
</tr>
<tr>
<td>Update $G$ using Eq. (5) with given steps</td>
<td></td>
</tr>
<tr>
<td>for $i = 1$ to $M_1$ do</td>
<td></td>
</tr>
<tr>
<td>for $j = 1$ to $M_2$ do</td>
<td></td>
</tr>
<tr>
<td>$G \leftarrow G - d_{mix} \begin{bmatrix} LG - \mu A^T \end{bmatrix}$</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>$r \leftarrow \text{ones}(1, C)$</td>
<td></td>
</tr>
<tr>
<td>for $j = 1$ to $M_3$ do</td>
<td></td>
</tr>
<tr>
<td>$\hat{o} \leftarrow 1/\lambda \text{Proj}_H(G \cdot \text{diag}(r))$</td>
<td></td>
</tr>
<tr>
<td>$r \leftarrow \max(\min(r + \phi \cdot (o - \hat{o}), v_{\alpha}), v_{\sigma})$</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>$G \leftarrow \text{Proj}_H(G \cdot \text{diag}(r))$</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>$G \leftarrow G[m - V : m, :]$</td>
<td></td>
</tr>
</tbody>
</table>

Experiments

Datasets

We evaluate the proposed PTN on two few-shot benchmark datasets: miniImageNet and tieredImageNet. The miniImageNet dataset (Vinyals et al. 2016) is a subset of the ImageNet, consisting of 100 classes, and each class contains 600 images of size $84 \times 84$. We follow the standard split of 64 base, 16 validation, and 20 test classes (Vinyals et al. 2016; Tian et al. 2020). The tieredImageNet (Ren et al. 2018) is another subset but with 608 classes instead. We follow the standard split of 351 base, 97 validation, and 160 test classes for the experiments (Ren et al. 2018; Liu et al. 2018). We resize the images from tieredImageNet to $84 \times 84$ pixels, and randomly select $C$ classes from the novel class to construct the few-shot task. Within each class, $K$ examples are selected as the labeled data, and $V$ examples from the rest as queries. The extra $N$ unlabeled samples are selected from the $C$ classes or rest novel classes. We set $C = 5, K = \{1, 5\}, V = 15$ and study different sizes of $N$. We run 600 few-shot tasks and report the mean accuracy with the 95% confidence interval.

Implementation Details

Same as previous works (Rusu et al. 2018; Dhillion et al. 2019; Liu, Song, and Qin 2020; Tian et al. 2020; Yu et al. 2020), we adopt the wide residual network (WRN-28-10) (Zagoruyko and Komodakis 2016) as the backbone of our base model $W_0 \circ f_0$, and we follow the protocols in (Tian et al. 2020; Yu et al. 2020) fusing the base and validation classes to train the base model from scratch. We set the batch size to 64 with SGD learning rate as 0.05 and weight decay as $5e^{-4}$. We reduce the learning rate by 0.1 after 60 and 80 epochs. The base model is trained for 100 epochs.

In unsupervised embedding transfer, the data augmentation $T$ is defined same as (Lee et al. 2019; Tian et al. 2020). For fair comparisons against TransMatch (Yu et al. 2020), we also augment each labeled image 10 times by random transformations and generate the prototypes of each class as labeled samples. We apply SGD optimizer with a momentum of 0.9. The learning rate is initialized as $1e^{-5}$, and
the cosine learning rate scheduler is used for 10 epochs. We set the batch size to 80 with \( \lambda = 1 \) in Eq. (2). For Poisson inference, we construct the graph by connecting each sample to its \( K \)-nearest neighbors with Gaussian weights. We set \( K = 30 \) and the weight matrix \( W \) is summarized with \( w_{ij} = 0 \), which accelerates the convergence of the iteration in Algorithm 1 without change the solution of the Equation 4. We set the max \( t_p = 100 \) in step 7 of Algorithm 1 by referring to the stop constraint discussed in the Proposed Algorithm section. We set hyper-parameters \( \mu = 1.5, M_1 = 20, M_2 = 40 \) and \( M_3 = 100 \) empirically. Moreover, we set \( \varphi = 10 \), \( v_\alpha = 0.5 \), \( v_\eta = 1.0 \).

**Experimental Results**

**Comparison with the State-Of-The-Art** In our experiments, we group the compared methods into five categories, and the experimental results on two datasets are summarized in Table 1. With the auxiliary unlabeled data available, our proposed PTN outperforms the metric-based and optimization-based few-shot models by large margins, indicating that the proposed model effectively utilizes the unlabeled information for assisting few-shot recognition. By integrating the unsupervised embedding transfer and Poisson-MBO classifier, PTN achieves superior performance over both transductive and existing SSFSL approaches. Specifically, under the 5-way-1-shot setting, the classification accuracies are 81.57% vs. 63.02% TransMatch (Yu et al. 2020), 84.70% vs. 80.30% LaplacianShot (Ziko et al. 2020) on miniImageNet and tieredImageNet, respectively; under the 5-way-5-shot setting, the classification accuracies are 89.14% vs. 78.70% LST (Li et al. 2019b), 88.43% vs. 84.70% BD-CSPN (Liu, Song, and Qin 2020) on miniImageNet and tieredImageNet, respectively. These results demonstrate the superiority of PTN for SSFSL tasks.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Type</th>
<th>Backbone</th>
<th>minImageNet</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototypical-Net (Snell et al.)</td>
<td>Metric, Meta</td>
<td>ConvNet-256</td>
<td>49.42</td>
<td>68.20</td>
<td>0.66</td>
</tr>
<tr>
<td>Relation Network (Sung et al.)</td>
<td>Metric, Meta</td>
<td>ConvNet-64</td>
<td>50.44</td>
<td>65.32</td>
<td>0.70</td>
</tr>
<tr>
<td>TADAM (Oreshkin et al.)</td>
<td>Metric, Meta</td>
<td>ResNet-12</td>
<td>58.50</td>
<td>76.70</td>
<td>0.30</td>
</tr>
<tr>
<td>DPGN (Yang et al.)</td>
<td>Metric, Meta</td>
<td>ResNet-12</td>
<td>67.77</td>
<td>84.60</td>
<td>0.43</td>
</tr>
<tr>
<td>RFS (Tian et al.)</td>
<td>Metric, Transfer</td>
<td>ResNet-12</td>
<td>64.82</td>
<td>80.14</td>
<td>0.43</td>
</tr>
<tr>
<td>MAML (Finn et al.)</td>
<td>Optimization, Meta</td>
<td>ConvNet-64</td>
<td>48.70</td>
<td>63.11</td>
<td>0.92</td>
</tr>
<tr>
<td>SNAIL (Mishra et al.)</td>
<td>Optimization, Meta</td>
<td>ResNet-12</td>
<td>55.71</td>
<td>68.88</td>
<td>0.92</td>
</tr>
<tr>
<td>LEO (Rusu et al.)</td>
<td>Optimization, Meta</td>
<td>WRN-28-10</td>
<td>61.76</td>
<td>77.59</td>
<td>0.12</td>
</tr>
<tr>
<td>MetaOptNet (Lee et al.)</td>
<td>Optimization, Meta</td>
<td>ResNet-12</td>
<td>64.09</td>
<td>80.00</td>
<td>0.45</td>
</tr>
<tr>
<td>TPN (Liu et al.)</td>
<td>Transductive, Meta</td>
<td>ConvNet-64</td>
<td>55.51</td>
<td>69.86</td>
<td>0.65</td>
</tr>
<tr>
<td>BD-CSPN (Liu et al.)</td>
<td>Transductive, Meta</td>
<td>WRN-28-10</td>
<td>70.31</td>
<td>81.90</td>
<td>0.60</td>
</tr>
<tr>
<td>Transductive Fine-tuning (Dhillon et al.)</td>
<td>Transductive, Transfer</td>
<td>WRN-28-10</td>
<td>65.73</td>
<td>78.40</td>
<td>0.52</td>
</tr>
<tr>
<td>LaplacianShot (Ziko et al.)</td>
<td>Transductive, Transfer</td>
<td>DenseNet</td>
<td>75.57</td>
<td>84.72</td>
<td>0.13</td>
</tr>
<tr>
<td>Masked Soft k-Means (Ren et al.)</td>
<td>Semi, Meta</td>
<td>ConvNet-128</td>
<td>50.41</td>
<td>64.39</td>
<td>0.24</td>
</tr>
<tr>
<td>TPN-semi (Liu et al.)</td>
<td>Semi, Meta</td>
<td>ConvNet-64</td>
<td>52.78</td>
<td>66.41</td>
<td>0.21</td>
</tr>
<tr>
<td>LST (Li et al.)</td>
<td>Semi, Meta</td>
<td>ResNet-12</td>
<td>70.10</td>
<td>78.70</td>
<td>0.80</td>
</tr>
<tr>
<td>TransMatch (Yu et al.)</td>
<td>Semi, Transfer</td>
<td>WRN-28-10</td>
<td>62.93</td>
<td>82.24</td>
<td>0.59</td>
</tr>
<tr>
<td>DPN (Ours)</td>
<td>Semi, Transfer</td>
<td>WRN-28-10</td>
<td>79.67</td>
<td>86.30</td>
<td>0.95</td>
</tr>
<tr>
<td>PTN (Ours)</td>
<td>Semi, Transfer</td>
<td>WRN-28-10</td>
<td>82.66</td>
<td>88.43</td>
<td>0.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Type</th>
<th>Backbone</th>
<th>tieredImageNet</th>
<th>1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototypical-Net (Snell et al.)</td>
<td>Metric, Meta</td>
<td>ConvNet-256</td>
<td>53.31</td>
<td>72.69</td>
<td>0.74</td>
</tr>
<tr>
<td>Relation Network (Sung et al.)</td>
<td>Metric, Meta</td>
<td>ConvNet-64</td>
<td>54.48</td>
<td>71.32</td>
<td>0.78</td>
</tr>
<tr>
<td>DPGN (Yang et al.)</td>
<td>Metric, Meta</td>
<td>ResNet-12</td>
<td>72.45</td>
<td>87.24</td>
<td>0.39</td>
</tr>
<tr>
<td>RFS (Tian et al.)</td>
<td>Metric, Transfer</td>
<td>ResNet-12</td>
<td>71.52</td>
<td>86.03</td>
<td>0.49</td>
</tr>
<tr>
<td>MAML (Finn et al.)</td>
<td>Optimization, Meta</td>
<td>ConvNet-64</td>
<td>51.67</td>
<td>70.30</td>
<td>0.75</td>
</tr>
<tr>
<td>LEO (Rusu et al.)</td>
<td>Optimization, Meta</td>
<td>WRN-28-10</td>
<td>66.33</td>
<td>81.44</td>
<td>0.09</td>
</tr>
<tr>
<td>MetaOptNet (Lee et al.)</td>
<td>Optimization, Meta</td>
<td>ResNet-12</td>
<td>65.81</td>
<td>81.75</td>
<td>0.53</td>
</tr>
<tr>
<td>TPN (Liu et al.)</td>
<td>Transductive, Meta</td>
<td>ConvNet-64</td>
<td>59.91</td>
<td>73.30</td>
<td>0.75</td>
</tr>
<tr>
<td>BD-CSPN (Liu et al.)</td>
<td>Transductive, Meta</td>
<td>WRN-28-10</td>
<td>78.74</td>
<td>86.92</td>
<td>0.63</td>
</tr>
<tr>
<td>Transductive Fine-tuning (Dhillon et al.)</td>
<td>Transductive, Transfer</td>
<td>WRN-28-10</td>
<td>73.34</td>
<td>85.50</td>
<td>0.50</td>
</tr>
<tr>
<td>LaplacianShot (Ziko et al.)</td>
<td>Transductive, Transfer</td>
<td>DenseNet</td>
<td>80.30</td>
<td>87.93</td>
<td>0.15</td>
</tr>
<tr>
<td>Masked Soft k-Means (Ren et al.)</td>
<td>Semi, Meta</td>
<td>ConvNet-128</td>
<td>52.39</td>
<td>69.88</td>
<td>0.20</td>
</tr>
<tr>
<td>TPN-semi (Liu et al.)</td>
<td>Semi, Meta</td>
<td>ConvNet-64</td>
<td>55.74</td>
<td>71.01</td>
<td>0.23</td>
</tr>
<tr>
<td>LST (Li et al.)</td>
<td>Semi, Meta</td>
<td>ResNet-12</td>
<td>77.70</td>
<td>85.20</td>
<td>0.80</td>
</tr>
<tr>
<td>DPN (Ours)</td>
<td>Semi, Transfer</td>
<td>WRN-28-10</td>
<td>82.18</td>
<td>88.02</td>
<td>0.72</td>
</tr>
<tr>
<td>PTN (Ours)</td>
<td>Semi, Transfer</td>
<td>WRN-28-10</td>
<td>84.70</td>
<td>89.14</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 1: The 5-way, 1-shot and 5-shot classification accuracy (%) on the two datasets with 95% confidence interval. The best results are in bold. The upper and lower parts of the table show the results on miniImageNet and tieredImageNet, respectively.
Ablation Study

We analyze different components of the PTN and summarize the results in Table 3. All compared approaches are based on the pre-trained WRN-28-10 embedding.

Different Extra Unlabeled Samples  We show the results of using different numbers of extra unlabeled samples in Table 2. For Num_U = 0, PTN* can be viewed as the transductive model without extra unlabeled data, where we treat query samples as the unlabeled data, and we do not fine-tune the embedding with query labels for fair comparisons. Contrary to PTN*, the proposed PTN model utilizes the query samples to fine-tune the embedding when Num_U=0. It can be observed that our PTN model achieves better performances with more extra unlabeled samples, which indicates the effectiveness of PTN in mining the unlabeled auxiliary information for the few-shot problem.

Results with Distractor Classes  Inspired by (Ren et al. 2018; Liu et al. 2018; Yu et al. 2020), we further investigate the influence of distractor classes, where the extra unlabeled data are collected from classes with no overlaps to labeled support samples. We follow the settings in (Ren et al. 2018; Liu et al. 2018). As shown in Figure 2, even with distractor class data, the proposed PTN still outperforms other SSFSL methods by a large margin, which indicates the robustness of the proposed PTN in dealing with distracted unlabeled data.

Acknowledgments

The authors greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organizations and the Australian Federal Government’s Business-Cooperative Research Centres Program) through Project R3.7.3 - Rail infrastructure defect detection through video analytics.
References


Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A Baseline for Few-Shot Image Classification. In ICLR.


