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

Huaxi Huang[1], **Junjie Zhang**[2], **Jian Zhang**[1], **Qiang Wu**[1], **Chang Xu** [3]

[1]University of Technology Sydney, Sydney NSW 2007, Australia
[2]Shanghai University, Shanghai, China
[3]The 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 (SSFSL) 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 (SSFSL) (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

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

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 episode 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 $\mathbf{e}_i \in \mathbb{R}^C$ represent the i-th class, *i.e.*, $y_s \in \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_C\}$. 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 $\mathcal{D}_{novel}$ is thus defined as $\mathcal{D}_{novel} = S \cup Q$. 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 $\mathcal{D}_{novel}$ as an auxiliary dataset $\mathcal{D}_{base}$, 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

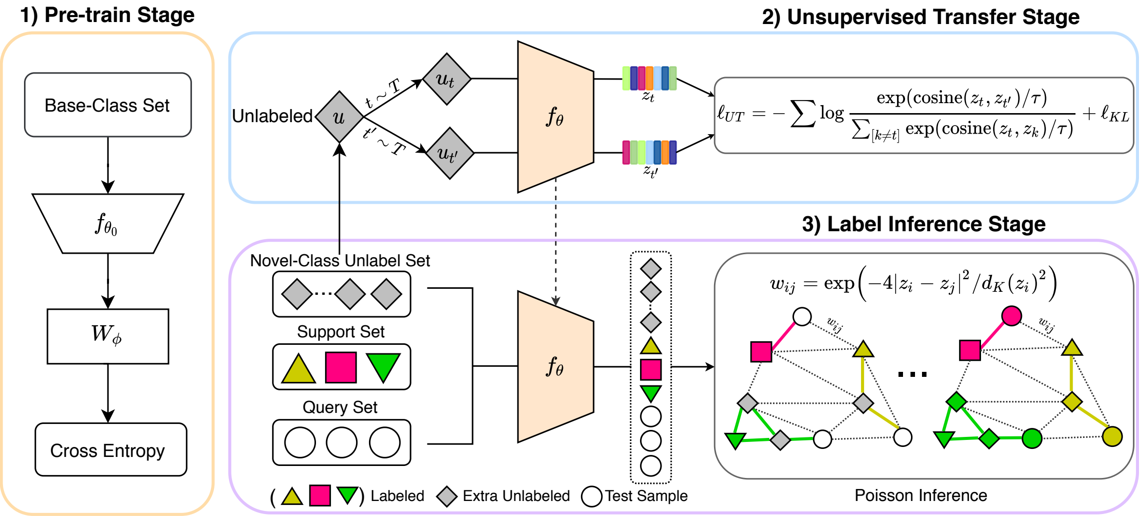

Figure 1: The overview of the proposed PTN. We first pre-train a feature embedding f_{θ_0} from the base-class set using standard cross-entropy loss. This embedding is then fine-tuned with the external novel-class unlabeled data by adopting unsupervised transferring loss ℓ_{UT} to generate f_θ. Finally, we revise a graph model named PoissonMBO to conduct the query label inference.

additional N unlabeled samples are usually from each of the C support classes in standard-setting, or other novel-class under distractor classification settings. Then the new novel-class dataset $\mathcal{D}_{novel}$ is defined as $\mathcal{D}_{novel} = S \cup Q \cup U$. The goal of SSFSL is maximizing the value of the extra unlabeled data to improve the few-shot methods.

For a clear understanding, the details of proposed PTN are introduced as follows: we first introduce the proposed Representation Learning, and then we illustrate the proposed Poisson learning model for label inference.

Representation Leaning

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 $\mathcal{D}_{base}$, we train the base encoder f_{θ_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 $U_i = \{x_u\}_{u=1}^n$, randomly sampling two data augmentation operators $t, t' \in T$, we can generate a new feature set $Z = \{Z_t = \{f_{\theta_0} \circ t(x_u)\}_{u=1}^n\} \cup \{Z_{t'} = \{f_{\theta_0} \circ t'(x_u)\}_{u=1}^n\}$, 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

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

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

$$\ell_{UT} = \ell_{cont} + \lambda \ell_{KL}(Z_t \parallel Z_{t'}). \quad (2)$$

By training the extra unlabeled data with this loss, we can learn a robust novel-class embedding f_θ from f_{θ_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, Ghahramani, 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 SSFSL 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

$$\begin{aligned}\Delta_{\text{cross}} &= \mathbb{E}_{z_b \sim p_{\mathcal{B}}}[z_b] - \mathbb{E}_{z_q \sim p_{\mathcal{Q}}}[z_q] \\ &= \frac{1}{|\mathcal{B}|}\sum_{b=1}^{|\mathcal{B}|} z_b - \frac{1}{|\mathcal{Q}|}\sum_{q=1}^{|\mathcal{Q}|} z_q.\end{aligned} \tag{3}$$

We then add the bias Δ_{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 $Z_{novel} = Z_b \cup Z_q = \{z_1, z_2, \ldots, z_m\}$ $(m = K \times C + N + V)$, 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 $w_{ij} = \exp\left(-4\left|z_i - z_j\right|^2 / d_K\left(z_i\right)^2\right)$, where $d_K\left(z_i\right)^2$ 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 as $W = [w_{ij}]$, the degree matrix as $D = \operatorname{diag}([d_i = \sum_{j=1}^{m} w_{ij}])$, and the unnormalized 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 $g : z \to \mathbb{R}^C$. By solving the Poisson equation:

$$Lg\left(z_i\right) = \sum_{j=1}^{K\times C}\left(y_j - \bar{y}\right)\mathbb{I}_{ij} \quad \text{for } i = 1, \ldots, m, \tag{4}$$

satisfying $\sum_{i=1}^{m}\sum_{k=1}^{m} w_{ik} g\left(z_i\right) = 0$, 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 $\hat{y}_i = \arg\max_{j\in\{1,\ldots,C\}}\{g_j(x_i)\}$. 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 $Y = [y_s] \in \mathbb{R}^{C\times(K\times C)}$. Let $A = [Y - \bar{y}, \mathbf{0}^{C\times(m-K\times C)}]$ 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:

$$G^{tp+1} = G^{tp} + D^{-1}(A^T - LG^{tp}), \tag{5}$$

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

$$\min_{\substack{g:Z\to H\\(g)_z = o}}\left\{g^T L g - \mu\sum_{i=1}^{K\times C}\left(y_i - \bar{y}\right)\cdot g\left(z_i\right)\right\}, \tag{6}$$

where $H = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_C\}$ 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 piror 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, Ghahramani, and Lafferty 2003; Zhu, Lafferty, and Rosenfeld 2005; Zhou et al. 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:

$$\begin{aligned}&\min_{\substack{g\in\mathrm{SP}\{Z\to\mathbb{R}^C\}\\(g)_z = o}}\left\{\mathrm{GL}_{\tau'}(g) - \mu\sum_{i=1}^{K\times C}\left(y_i - \bar{y}\right)\cdot g\left(z_i\right)\right\}, \\ &\mathrm{GL}_{\tau'}(g) = g^T L g + \frac{1}{\tau'}\sum_{i=1}^{m}\prod_{j=1}^{C}\left|g\left(z_i\right) - \mathbf{e}_j\right|^2.\end{aligned} \tag{7}$$

In Eq. (7), $\mathrm{SP}\{Z \to \mathbb{R}^C\}$ represents the space of projections $g : Z \to \mathbb{R}^C$, 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.

```
Algorithm 1: PTN for SSFSL
  Input  : $\mathcal{D}_{base}, \mathcal{D}_{novel} = S \cup U \cup Q$,
           $o$, $\mu$, $M_1, M_2, M_3$
  Output: Query samples' label prediction $G$
1  Train a base model $\mathbf{W}_\phi \circ f_{\theta_0}(x)$ with all samples
     and labels from $\mathcal{D}_{novel}$;
2  Apply unsupervised embedding transfer method to
     fine-tune the $f_{\theta_0}$ with novel unlabeled data $U$ by
     using $\ell_{UT}$ in Eq. (2), and result in $f_\theta$;
3  Apply $f_\theta$ to extract features on $D_{novel}$ as $Z_{novel}$;
4  Apply query feature calibration using Eq. (3);
5  Compute $W, D, L, A$ according to $Z_{novel}$,
     $G \leftarrow \mathbf{0}^{m \times C}$
6  $PoissonMBO$
7      Update $G$ uisng Eq. (5) with given steps
8      $\mathrm{d}_{mx} \leftarrow 1/\max_{1 \leq i \leq m} D_{ii}, G \leftarrow \mu G$
9      for $i = 1$ to $M_1$ do
10         for $j = 1$ to $M_2$ do
11             $G \leftarrow G - \mathrm{d}_{mx}\left(LG - \mu A^T\right)$
12         end
13         $r \leftarrow \mathbf{ones}(1, C)$
14         for $j = 1$ to $M_3$ do
15             $\hat{o} \leftarrow \frac{1}{n}\mathbf{1}^T\mathbf{Proj}_H(G \cdot \mathrm{diag}(r))$
16             $r \leftarrow \max\left(\min\left(r + \varphi \cdot (o - \hat{o}), \upsilon_\alpha\right), \upsilon_\sigma\right)$
17         end
18         $G \leftarrow \mathbf{Proj}_H(G \cdot \mathrm{diag}(r))$
19     end
20 $G \leftarrow G[m - V : m, :]$;
```

Proposed Algorithm

The overall proposed algorithm is summarized in Algorithm 1. Inputting the base-class set $\mathcal{D}_{base}$, novel-class set $\mathcal{D}_{novel}$, prior classes' distribution o, and other parameters, PTN will predict the query samples' label $G \in \mathbb{R}^{V \times C}$. The query label $\hat{y}_q$ is then determined as $\hat{y}_q = \arg\max_{1 \leq j \leq C} G_{qj}$. More specifically, once the encoder f_θ is learned using the base set $\mathcal{D}_{base}$, 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 in step 3-5. In the label inference stage in steps 6-20, we first apply Poisson model to robust propagate the labels in step 7, and then solve the graph-cut problem by using MBO scheme in several steps of gradient-descent to boost the classification performance. The stop condition in step 7 follow the constraint: $\left\|\mathbf{sp}_{tp} - W\mathbf{1}/\left(\mathbf{1}^T W \mathbf{1}\right)\right\|_\infty \leq 1/m$, where $\mathbf{1}$ is a all-ones column vector, $\mathbf{sp}_{tp} = WD^{-1}\mathbf{sp}_{tp-1}$, $\mathbf{sp_0}$ is a m-column vector with ones in the first $K \times C$ positions and zeros elsewhere. Steps 9-19 are aimed to solve the graph-cut problem in Eq. (7), To solve the problem, we first divide the Eq. (7) into $E_1 = g^T Lg - \mu\sum_{i=1}^{K \times C}\left(y_i - \bar{y}\right) \cdot g\left(z_i\right)$ and $E_2 = \frac{1}{\tau'}\sum_{i=1}^{m}\prod_{j=1}^{C}\left|g\left(z_i\right) - \mathbf{e}_j\right|^2$, and then employing the gradient decent alternative on these two energy functions. Steps 10-12 are used to optimize the E_1. We optimize the E_2 in steps 14-17, $\mathbf{Proj}_H : \mathbb{R}^C \to H$ is the closet point projection, $r = [r_1, \ldots, r_C]^T$ ($r_i > 0$), φ is the time step, and $\upsilon_\alpha, \upsilon_\sigma$ are the clipping values, By adopting the gradient descent scheme in steps 14-17, the vector r is generated that also satisfies the constraint $(g)_z = o$ in Eq.(7). After obtaining the PoissonMBO's solution G, the query samples' label prediction matrix is resolved by step 20.

The main inference complexity of PTN is $\mathcal{O}(M_1 M_2 E)$, where E is the number of edges in the graph. As a graphed-based model, PTN's inference complexity is heavier than inductive models. However, previous studies (Liu et al. 2018; Calder et al. 2020) indicate that this complexity is affordable for few-shot tasks since the data scale is not very big. Moreover, we do not claim that our model is the final solution for SSFSL. We aim to design a new method to make full use of the extra unlabeled information. We report inference time comparison experiments in Table.1 in the supplemental materials. The average inference time of PTN is 13.68s.

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 84×84. 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 84×84 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; Dhillon 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 $W_\phi \circ f_{\theta_0}$, and we follow the protocals 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 $1e^{-3}$, and

Methods	Type	Backbone	miniImageNet	
			1-shot	5-shot
Prototypical-Net (Snell, Swersky, and Zemel 2017)	Metric, Meta	ConvNet-256	49.42±0.78	68.20±0.66
Relation Network (Sung et al. 2018)	Metric, Meta	ConvNet-64	50.44±0.82	65.32±0.70
TADAM (Oreshkin, López, and Lacoste 2018)	Metric, Meta	ResNet-12	58.50±0.30	76.70±0.30
DPGN (Yang et al. 2020)	Metric, Meta	ResNet-12	67.77±0.32	84.60±0.43
RFS (Tian et al. 2020)	Metric, Transfer	ResNet-12	64.82±0.60	82.14±0.43
MAML (Finn, Abbeel, and Levine 2017)	Optimization, Meta	ConvNet-64	48.70±1.84	63.11±0.92
SNAIL (Mishra et al. 2018)	Optimization, Meta	ResNet-12	55.71±0.99	68.88±0.92
LEO (Rusu et al. 2018)	Optimization, Meta	WRN-28-10	61.76±0.08	77.59±0.12
MetaOptNet (Lee et al. 2019)	Optimization, Meta	ResNet-12	64.09±0.62	80.00±0.45
TPN (Liu et al. 2018)	Transductive, Meta	ConvNet-64	55.51±0.86	69.86±0.65
BD-CSPN (Liu, Song, and Qin 2020)	Transductive, Meta	WRN-28-10	70.31±0.93	81.89±0.60
Transductive Fine-tuning (Dhillon et al. 2019)	Transductive, Transfer	WRN-28-10	65.73±0.68	78.40±0.52
LaplacianShot (Ziko et al. 2020)	Transductive, Transfer	DenseNet	75.57±0.19	84.72±0.13
Masked Soft k-Means (Ren et al. 2018)	Semi, Meta	ConvNet-128	50.41±0.31	64.39±0.24
TPN-semi (Liu et al. 2018)	Semi, Meta	ConvNet-64	52.78±0.27	66.42±0.21
LST (Li et al. 2019b)	Semi, Meta	ResNet-12	70.10±1.90	78.70±0.80
TransMatch (Yu et al. 2020)	Semi, Transfer	WRN-28-10	62.93±1.11	82.24±0.59
DPN (Ours)	Semi, Transfer	WRN-28-10	79.67±1.06	86.30±0.95
PTN (Ours)	Semi, Transfer	WRN-28-10	**82.66±0.97**	**88.43±0.67**

Methods	Type	Backbone	tieredImageNet	
			1-shot	5-shot
Prototypical-Net (Snell, Swersky, and Zemel 2017)	Metric, Meta	ConvNet-256	53.31±0.89	72.69±0.74
Relation Network (Sung et al. 2018)	Metric, Meta	ConvNet-64	54.48±0.93	71.32±0.78
DPGN (Yang et al. 2020)	Metric, Meta	ResNet-12	72.45±0.51	87.24±0.39
RFS (Tian et al. 2020)	Metric, Transfer	ResNet-12	71.52±0.69	86.03±0.49
MAML (Finn, Abbeel, and Levine 2017)	Optimization, Meta	ConvNet-64	51.67±1.81	70.30±1.75
LEO (Rusu et al. 2018)	Optimization, Meta	WRN-28-10	66.33±0.05	81.44±0.09
MetaOptNet (Lee et al. 2019)	Optimization, Meta	ResNet-12	65.81±0.74	81.75±0.53
TPN (Liu et al. 2018)	Transductive, Meta	ConvNet-64	59.91±0.94	73.30±0.75
BD-CSPN (Liu, Song, and Qin 2020)	Transductive, Meta	WRN-28-10	78.74±0.95	86.92±0.63
Transductive Fine-tuning (Dhillon et al. 2019)	Transductive, Transfer	WRN-28-10	73.34±0.71	85.50±0.50
LaplacianShot (Ziko et al. 2020)	Transductive, Transfer	DenseNet	80.30±0.22	87.93±0.15
Masked Soft k-Means (Ren et al. 2018)	Semi, Meta	ConvNet-128	52.39±0.44	69.88±0.20
TPN-semi (Liu et al. 2018)	Semi, Meta	ConvNet-64	55.74±0.29	71.01±0.23
LST (Li et al. 2019b)	Semi, Meta	ResNet-12	77.70±1.60	85.20±0.80
DPN (Ours)	Semi, Transfer	WRN-28-10	82.18±1.06	88.02±0.72
PTN (Ours)	Semi, Transfer	WRN-28-10	**84.70±1.14**	**89.14±0.71**

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

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 $w_{ii} = 0$, which accelerates the convergence of the iteration in Algorithm 1 without change the solution of the Equation 4. We set the max $tp = 100$ 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 $\varphi = 10, \upsilon_\alpha = 0.5, \upsilon_\sigma = 1.0$.

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 88.43% vs. 78.70% LST (Li et al. 2019b), 89.14% vs. 81.89% BD-CSPN (Liu, Song, and Qin 2020) on miniImageNet and tieredImageNet, respectively. These results demonstrate the superiority of PTN for SSFSL tasks.

Methods	Num_U	1-shot	5-shot
PTN*	0	76.20±0.82	84.25±0.61
PTN	0	77.01±0.94	85.32±0.68
PTN	20	77.20±0.92	85.93±0.82
PTN	50	79.92±1.06	86.09±0.75
PTN	100	81.57±0.94	87.17±0.58
PTN	200	**82.66±0.97**	**88.43±0.76**

Table 2: The 5-way, 1-shot and 5-shot classification accuracy (%) with different number of extra unlabeled samples on miniImageNet. PTN* denotes that we adopt PTN as the transductive model without fine-tune embedding. Best results are in bold.

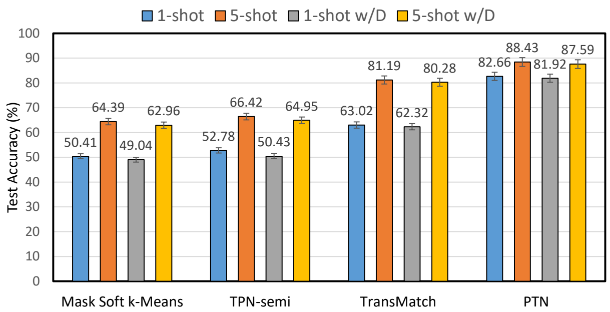

Figure 2: The 5-way, 1-shot and 5-shot classification accuracy (%) with different number of extra unlabeled samples on miniImageNet. w/D means with distractor classes.

Different Extra Unlabeled Samples We show the results of using different numbers of extra unlabeled instances 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 utilize 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.

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.

Methods	1-shot	5-shot
TransMatch	62.93±1.11	82.24±0.59
Label Propagation (LP)	74.04±1.00	82.60±0.68
PoissonMBO	79.67±1.02	86.30±0.65
DPN	80.00±0.83	87.17±0.51
Unsup Trans+LP [a]	75.65±1.06	84.46±0.68
Unsup Trans+PoissonMBO	80.73±1.11	87.41±0.63
Unsup Trans+PTN [b]	**82.66±0.97**	**88.43±0.76**

[a] Unsup Trans means Unsupervised Embedding Transfer.

[b] PTN consists of Unsup Trans and DPN.

Table 3: Ablation studies about the proposed PTN, all methods are based on a pretrained embedding with 200 extra unlabeled samples each class on miniImageNet for 5-way, 1-shot and 5-shot classification (%). Best results are in bold.

First of all, we investigate the graph propagation component (classifier). It can be observed that graph-based models such as Label Propagation (Zhou et al. 2004) and PoissonMBO (Calder et al. 2020) outperform the inductive model TransMatch (Yu et al. 2020), which is consistent with previous researches (Zhu, Lafferty, and Rosenfeld 2005; Liu et al. 2018; Ziko et al. 2020). Compared to directly applying PoissonMBO on few-shot tasks, the proposed DPN ***(without Unsupervised Embedding Transfer)*** achieves better performance, which indicates it is necessary to perform the feature calibration to eliminate the cross-class biases between support and query data distributions before label inference.

For investigating the proposed unsupervised embedding transfer in representation learning, we observe that all the graph-based models achieve clear improvement after incorporating the proposed transfer module. For instance, the Label Propagation obtains 1.61%, 1.86% performance gains on 5-way-1-shot, and 5-way-5-shot minImageNet classification. These results indicate the effectiveness of the proposed unsupervised embedding transfer. Finally, by integrating the unsupervised embedding transfer and graph propagation classifier, the PTN model achieves the best performances compared against all other approaches in Table 3.

Conclusion

We propose a Poisson Transfer Network (PTN) to tackle the semi-supervised few-shot problem, aiming to explore the value of unlabeled data from two aspects: Poisson module as a more robust classifier and unsupervised transfer module to improve the generality of the embedding on novel classes. Extensive experiments indicate that PTN outperforms state-of-the-art few-shot and semi-supervised few-shot methods.

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

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 5049–5059.

Calder, J.; Cook, B.; Thorpe, M.; and Slepcev, D. 2020. Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates. In *ICML*, 1306–1316. PMLR.

Calder, J.; and Slepčev, D. 2019. Properly-weighted graph Laplacian for semi-supervised learning. *Applied Mathematics & Optimization* 1–49.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *ICLR* .

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2018. A Closer Look at Few-shot Classification. In *ICLR*.

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

Fei-Fei, L.; Fergus, Rob, P.; and Pietro. 2006. One-shot learning of object categories. *TPAMI* 28(4): 594–611.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 1126–1135.

Garcia-Cardona, C.; Merkurjev, E.; Bertozzi, A. L.; Flenner, A.; and Percus, A. G. 2014. Multiclass data segmentation using diffuse interface methods on graphs. *TPAMI* 36(8): 1600–1613.

Gidaris, S.; and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *CVPR*, 4367–4375.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Kim, J.; Kim, T.; Kim, S.; and Yoo, C. D. 2019. Edge-labeling graph neural network for few-shot learning. In *CVPR*, 11–20.

Lan, X.; Yuen, P. C.; and Chellappa, R. 2017. Robust mil-based feature template learning for object tracking. In *AAAI*, 4118–4125.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *CVPR*, 10657–10665.

Li, W.; Xu, J.; Huo, J.; Wang, L.; Gao, Y.; and Luo, J. 2019a. Distribution consistency based covariance metric networks for few-shot learning. In *AAAI*, volume 33, 8642–8649.

Li, X.; Sun, Q.; Liu, Y.; Zhou, Q.; Zheng, S.; Chua, T.-S.; and Schiele, B. 2019b. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, 10276–10286.

Liu, J.; Song, L.; and Qin, Y. 2020. Prototype Rectification for Few-Shot Learning. In *ECCV*, 741–756.

Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S. J.; and Yang, Y. 2018. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*.

Miller, E. G.; Matsakis, N. E.; and Viola, P. A. 2000. Learning from one example through shared densities on transforms. In *CVPR*, volume 1, 464–471.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A Simple Neural Attentive Meta-Learner. In *ICLR*.

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 721–731.

Qi, H.; Brown, M.; and Lowe, D. G. 2018. Low-shot learning with imprinted weights. In *CVPR*, 5822–5830.

Qiao, S.; Liu, C.; Shen, W.; and Yuille, A. L. 2018. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 7229–7238.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. In *ICLR*.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-Learning with Latent Embedding Optimization. In *ICLR*.

Simon, C.; Koniusz, P.; Nock, R.; and Harandi, M. 2020. Adaptive Subspaces for Few-Shot Learning. In *CVPR*, 4136–4145.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NeurIPS*, 4077–4087.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, 1199–1208.

Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J. B.; and Isola, P. 2020. Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need? In *ECCV*, 266–282.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NeurIPS*, 3630–3638.

Yang, L.; Li, L.; Zhang, Z.; Zhou, X.; Zhou, E.; and Liu, Y. 2020. DPGN: Distribution Propagation Graph Network for Few-shot Learning. In *CVPR*, 13390–13399.

Yu, Z.; Chen, L.; Cheng, Z.; and Luo, J. 2020. TransMatch: A Transfer-Learning Scheme for Semi-Supervised Few-Shot Learning. In *CVPR*, 12856–12864.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. In *BMVC*.

Zhang, X.; Wei, Y.; Feng, J.; Yang, Y.; and Huang, T. 2018. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 1325–1334.

Zhang, X.; Wei, Y.; Yang, Y.; and Huang, T. S. 2020. Sg-one: Similarity guidance network for one-shot semantic segmentation. *TCYB* 50(9): 3855–3865.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *NIPS*, 321–328.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 912–919.

Zhu, X.; Lafferty, J.; and Rosenfeld, R. 2005. *Semi-supervised learning with graphs*. Ph.D. thesis, Carnegie Mellon University, language technologies institute, school of computer science.

Ziko, I. M.; Dolz, J.; Granger, E.; and Ayed, I. B. 2020. Laplacian Regularized Few-Shot Learning. In *ICML*, 11660–11670.