Self-Training Based Few-Shot Node Classification by Knowledge Distillation

Zongqian Wu[1*], **Yujie Mo**[1*], **Peng Zhou**[3], **Shangbo Yuan**[4], **Xiaofeng Zhu**[1,2†]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China
[3]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
[4]School of Engineering and Design, Technical University of Munich, Munich, Germany
wkzongqianwu@gmail.com

Abstract

Self-training based few-shot node classification (FSNC) methods have shown excellent performance in real applications, but they cannot make the full use of the information in the base set and are easily affected by the quality of pseudo-labels. To address these issues, this paper proposes a new self-training FSNC method by involving the representation distillation and the pseudo-label distillation. Specifically, the representation distillation includes two knowledge distillation methods (*i.e.*, the local representation distillation and the global representation distillation) to transfer the information in the base set to the novel set. The pseudo-label distillation is designed to conduct knowledge distillation on the pseudo-labels to improve their quality. Experimental results showed that our method achieves supreme performance, compared with state-of-the-art methods. Our code and a comprehensive theoretical version are available at https://github.com/zongqianwu/KD-FSNC.

Introduction

Node classification usually needs enough label information to predict unlabeled nodes on the graph data (Tu et al. 2022; Liang et al. 2023). However, obtaining label information is generally expensive and time-consuming. To address this issue, few-shot node classification (FSNC) has widely been proposed to train machine learning models (*e.g.*, graph convolutional network (GCN)) on the datasets with limited label nodes. As a result, FSNC has recently attracted increasing attention in real applications.

Previous FSNC methods can be divided into two categories, *i.e.*, meta-learning based method (meta FSNC for short) and self-training based method (self-training FSNC for short) (Zhang et al. 2022; Zhou et al. 2019; Wu et al. 2022). The meta FSNC aims to learn the initialization model with prior knowledge from the base set, by either model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017) or prototypical network (ProNet) (Snell, Swersky, and Zemel 2017). For instance, G-META (Huang and Zitnik 2020) applies MAML to preserve both the structure information and the feature information of the graph, and GPN (Ding et al. 2020) computes prototypes with ProNet to obtain expressive node representations. However, the meta FSNC needs to construct multiple tasks to obtain generalization ability but it is usually time-consuming. Moreover, previous methods ignore the information in unlabeled nodes. To alleviate these issues, the self-training FSNC is proposed to use prior knowledge in the base set with inexpensive time cost and to assign a part of unlabeled nodes in the novel set with pseudo-labels. For instance, IA-FSNC (Wu et al. 2022) first reduces the time cost by constructing only one GCN in the base set, and then assigns pseudo-labels to unlabeled nodes with low information entropy in the novel set for achieving information augmentation.

However, there are still some limitations to be solved for the self-training FSNC. First, the self-training FSNC cannot generally make the full use of the information in the base set. For example, IA-FSNC only passes the parameters in the first layer of the graph convolution to initialize the parameters in the novel set, but it ignores the information in other graph convolutional layers. This possibly results in the information lack for the novel set. Second, the self-training FSNC conducts self-training to generate pseudo-labels for providing new supervisory information, but its effectiveness will degrade if the pseudo-labels are incorrect or over-confident. For example, the classification accuracy of IA-FSNC will greatly reduce if it selects either incorrect pseudo-labels or over-confident pseudo-labels too much.

Addressing the above issues of the self-training FSNC is challenging. First, the information in the base set does not have strong generalization ability (as the meta FSNC does) since it only builds one task (Wu et al. 2022). Therefore, it is essential for the novel set to fully use the information in the base set. Second, the effectiveness of the self-training FSNC depends on the quality of pseudo-labels (Miyato et al. 2018; Wei et al. 2020). Incorrect pseudo-labels will mislead the training model while over-confident pseudo-labels will lead to the over-fitting issue. Therefore, it is challenging for the self-training FSNC to properly handle either incorrect pseudo-labels or over-confident pseudo-labels.

In this paper, we propose a new self-training FSNC with knowledge distillation (KD-FSNC for short) to address the above issues. To do this, we first pre-train a GCN (Hinton et al. 2015; Romero et al. 2014; Joshi et al. 2021) as the teacher model on the base set, and then randomly initialize

*These authors contributed equally.
†Corresponding author (seanzhuxf@gmail.com).

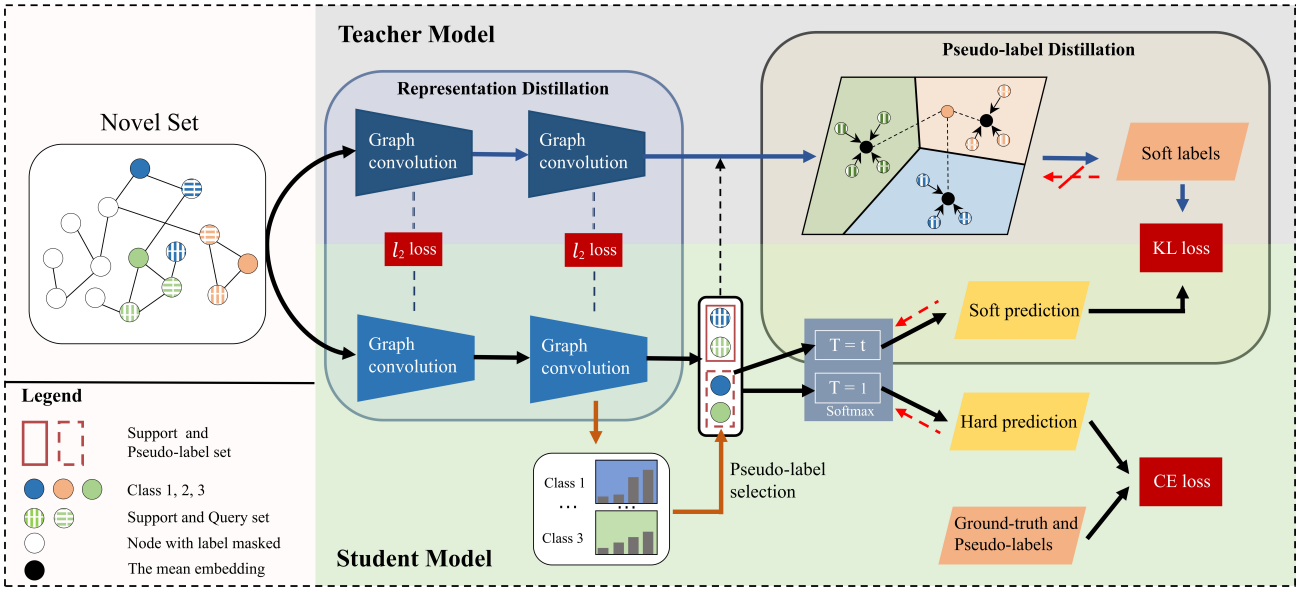


Figure 1: The proposed KD-FSNC includes two modules, *i.e.*, Representation distillation and Pseudo-label distillation. Representation distillation involves two process, *i.e.*, the local representation distillation and the global representation distillation. Specifically, given the pre-trained teacher model on the base set, all nodes of the novel set are input into two GCNs (*i.e.*, the teacher model and the student model) to output two sets of embeddings (*i.e.*, the teacher embedding and the student embedding) at each graph convolutional layer. Moreover, the teacher model is pre-trained and fixed. To learn the student model, we first conduct the local representation distillation by minimizing the l_2 distance between the student embeddings of two adjacent nodes, and conduct the global representation distillation by minimizing the l_2 distance between the teacher embedding and the student embedding for the same node. After that, we conduct Pseudo-label distillation by minimizing the KL divergence between the soft labels predicted by the teacher model and the soft prediction by the student model.

another GCN (*i.e.*, the student model) with the same architecture as the teacher model. We expect to learn an effective student model by transferring the information in the base set produced by the teacher model as much as possible. To do this, the representation distillation is designed to transfer information in all graph convolutional layers of the teacher model to learn the student model. This explores the first issue of previous FSNC methods by making the full use of the information in the base set. The pseudo-label distillation is then proposed to use the soft labels predicted by the teacher model to supervise the soft prediction of the pseudo-labels in the student model. As a result, the pseudo-label distillation improves the quality of pseudo-labels to explore the second issue in previous methods.

Related Work

Few-shot Learning

Few-shot learning (FSL) is a paradigm within deep learning designed to address the challenges associated with training models when there is only a limited amount of labeled data available for each class. In traditional deep learning, models are typically trained on large datasets with abundant examples for each category. However, in real-world scenarios, acquiring labeled data for every conceivable class may be impractical or cost-prohibitive. FSL aims to empower models to generalize and make accurate predictions even when provided with only a small number of instances per class during the training process. Previous FSL methods can be divided into two types, *i.e.*, data-based methods and model-based methods. Data-based methods involve learning an augmentation mapping, which maps the training data to new data, and then utilizes the newly generated data to expand the training set in few-shot tasks. For example, (Feyjie et al. 2020) employs generative adversarial networks to generate new samples. In the latest research on few-shot learning based on data augmentation, (Yang, Liu, and Xu 2021) assumes that each dimension of the data follows a Gaussian distribution, and similar classes exhibit similar distributions. Hence, the variance and mean of the base set are used to correct the variance and mean of the novel set.

On the other hand, model-based methods address the FSL problem by constraining the size of the hypothesis space. For instance, (Ma et al. 2020b) utilizes intra-class commonality and interclass uniqueness between support samples to estimate the relationship and adjacency relationship between different support-query pairs.

Few-shot Learning on Graph Data

Few-shot learning has gained increasing attention in the graph field (Wei et al. 2022). In the context of few-shot graph classification, (Chauhan, Nathani, and Kaul 2020) employs an innovative approach, incorporating graph probability measures and super-class clustering to enhance performance. Additionally, (Ma et al. 2020a) utilizes a meta-

learning method to acquire prior knowledge, enhancing the generalization of learned parameters to unseen categories.

In the few-shot node classification task, (Zhou et al. 2019) extended MAML to graph neural networks for the first time, utilizing the idea of meta-learning to acquire prior knowledge on non-task category nodes. Building on this work, (Ding et al. 2020) obtains a more expressive node representation by computing the prototype of each category, and they also propose a node evaluator to refine category prototypes. In addition, (Huang and Zitnik 2020) applies meta-leaning to subgraphs to solve both the node classification task and the edge prediction task since subgraphs preserve the structural information and feature information of the original graph.

Knowledge Distillation

Knowledge distillation (Hinton et al. 2015) initially emerged for model compression, aiming to guide a comparatively simple student model using a well-trained teacher model characterized by a more complex structure and a greater number of parameters. Subsequently, (Romero et al. 2014) proposed leveraging both the predictions of the teacher model and the intermediate layer features to instruct the student model. Additionally, (Gao et al. 2018) emphasized that the features from intermediate layers represent the most valuable knowledge imparted by the teacher model, asserting that the student model's expressiveness is optimal when it learns these features from the teacher.

Recently, several knowledge distillation methods have been proposed for graph neural networks (GNNs). For instance, (Joshi et al. 2021) introduced a novel graph contrastive representation distillation for GNNs, employing contrastive learning to align student node embeddings with teacher node embeddings in a shared representation space. (Deng and Zhang 2021) presented a method for knowledge distillation with GNNs that doesn't involve any training data. Meanwhile, (Zhang et al. 2021) introduced a high-accuracy, low-delay distillation model by using the teacher model as a GNN and the student model as a multi-layer perceptron.

Methodology

Notations Denoting $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a graph where $\mathcal{V}$ and $\mathcal{E}$ represent the node set and the edge set, respectively, we denote $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{A} \in \{0,1\}^{n \times n}$, respectively, as the feature matrix and the adjacency matrix of the graph, where n and d denote the number of nodes and the dimension of node features. We decompose a machine learning model $g(\mathbf{A}, \mathbf{X})$ (*e.g.*, a GCN) into the feature extractor $f(\mathbf{A}, \mathbf{X})$ by the linear classifier $\mathbf{W}$, *i.e.*, $g(\mathbf{A}, \mathbf{X}) = \mathbf{W}^T f(\mathbf{A}, \mathbf{X})$.

For FSNC, given a graph with node-label pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, where $\mathbf{y}_i \in \mathbf{Y}$ and $\mathbf{Y}$ denotes the label set. The FSNC model first learns prior knowledge from the base set, and then constructs the trained model on the *support set* $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N \times K}$ of the novel set by the information in the base set. The FSNC further evaluate the training model on the *query set* $\mathcal{Q} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=N \times K+1}^{N \times b}$ of the novel set.

Pre-training

As mentioned before, the meta FSNC usually results in expensive computation cost by constructing multiple tasks and ignores the rich information of unlabeled nodes (Yao et al. 2020; Lan et al. 2020). To alleviate these issues, the self-training FSNC is proposed to reduce time cost by only building one task on the base set and to introduce pseudo-labels for exploring rich information of unlabeled nodes. To do this, the self-training FSNC first builds only one task on the base set to learn the initialization parameters of the novel set, and then performs a pre-training process on the novel set to select credible pseudo-labels (Yang et al. 2021; Chu et al. 2022). Specifically, given the feature matrix $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$, we first define the node embedding $\mathbf{Z}$ of all nodes and its probability matrix $\mathbf{P}$ as follows:

$$\begin{cases} \mathbf{Z} = g(\mathbf{A}, \mathbf{X}), \\ \mathbf{P} = \mathrm{softmax}(\mathbf{Z}). \end{cases} \tag{1}$$

Based on Eq. (1), the cross-entropy loss of the support set in the novel set can be calculated as follows:

$$\mathcal{L}_{\mathrm{spt}} = -\sum_{i \in \mathcal{S}} \sum_{j=1}^{N} y_{i,j} \ln p_{i,j}, \tag{2}$$

where N represents the number of classes. Obviously, the feature extractor by the pre-training process contains generalization ability to output pseudo-labels with high confidence. However, previous self-training FSNC cannot make the full use of the information in the base set, and the pseudo-labels learned from a small number of the support set are often incorrect or over-confident. These issues will degrade the effectiveness of self-training.

To address the above issues, in this paper, we first follow IA-FSNC to perform only one task to train a GCN as the teacher model on the base set, and then follow traditional knowledge distillation methods to randomly initialize a GCN (*i.e.*, the student model) with the same architecture as the teacher model (Gao et al. 2018; Zhang et al. 2024). After performing the pre-training process in the student model with the support set in the novel set, we propose a new self-training FSNC with knowledge distillation (KD-FSNC for short) to learn a generalizable student model by transferring the information in the base set learned by the teacher model. Our KD-FSNC involves two knowledge distillations (*i.e.*, the representation distillation and the pseudo-label distillation) shown in Figure 1. Specifically, the representation distillation is designed to make the full use of the information in the base set, while the pseudo-label distillation is designed to improve the quality of pseudo-labels by using the information in the teacher model, thereby improving the effectiveness of the student model with limited nodes.

Local Representation Distillation It has been demonstrated that the nodes in the graph data have a high probability of belonging to the same class as their adjacent nodes (Luo et al. 2022). Therefore, it is intuitive to minimize the embedding distance (*i.e.*, maximizing the embedding similarity) of two adjacent nodes in the student model. In this paper, we use the information in the teacher model

to guarantee the above intuition. To do this, we push the local structure of every node in the teacher model to be preserved in the student model. Specifically, we first calculate the embedding similarity between the i-th node and the j-th node in the teacher model by the heat kernel, *i.e.*, $\exp\left(-\frac{\left\|f_T^i(\mathbf{A},\mathbf{X})-f_T^j(\mathbf{A},\mathbf{X})\right\|_2^2}{\sigma^2}\right)$, where σ is a normalized constant and $f_T^i(\mathbf{A},\mathbf{X})$ represents the embedding of the i-th node outputted by the teacher model. The local representation distillation loss is thus defined as:

$$\mathcal{L}_{\text{local}} = \sum_{i\in\mathcal{C}} \sum_{j\in\mathcal{N}(i)} \alpha_{i,j} \left\| f_S^i(\mathbf{A},\mathbf{X}) - f_S^j(\mathbf{A},\mathbf{X}) \right\|_2^2, \quad (3)$$

where $\mathcal{C}$ denotes the novel set, $\mathcal{N}(i)$ denotes the first-order neighborhood set of the i-th node, and $f_S^i(\mathbf{A},\mathbf{X})$ represents the embedding of the i-th node outputted by the student.

Eq. (3) with our selection of $\alpha_{i,j}$ pushes heavy penalty if the embeddings of two adjacent nodes (*i.e.*, the i-th node and the j-th node in the teacher model) are projected near in the student model. That is, minimizing Eq. (3) ensures that if the i-th node and the j-th node are neighbors in the teacher model then they are neighbors in the student model as well. Eq. (3) transfers the information (*i.e.*, the local structure of the embedding of every node on all graph convolutional layers of the teacher model) to ensure the embeddings of two adjacent nodes in the student model are similar. As a result, the local representation distillation preserves the local structure of every node in the teacher model to learn the student.

Global Representation Distillation Previous literature has demonstrated that both the local structure preservation and the global structure preservation are crucial for representation learning because they provide complementary information to each other (Wang et al. 2014). After conducting the local structure preservation by the local representation distillation, we propose to preserve the global structure by the global representation distillation with the help of the information in the teacher model. Specifically, we first extract the embeddings of the nodes in the novel set with the teacher model, and then employ the mean square error between the embedding in the teacher model and the embedding in the student model for every node to preserve the global structure, so the global representation distillation loss is:

$$\mathcal{L}_{\text{global}} = \left\| f_S(\mathbf{A},\mathbf{X}) - f_T(\mathbf{A},\mathbf{X}) \right\|_2^2. \quad (4)$$

Eq. (4) uses the information on every graph convolutional layer of the teacher model to preserve the global structure of the nodes in the novel set. As a result, the embedding in the student model for every node is similar to the embedding of the same node in the teacher model. Such global structure also solves the issue of insufficient training samples (*i.e.*, the support set in the novel set) in the student model.

We integrate the local representation distillation loss in Eq. (3) with the global representation distillation loss in Eq. (4) to define the loss function of the proposed representation distillation as follows:

$$\mathcal{L}_{\text{RD}} = (1-\beta)\mathcal{L}_{\text{local}} + \beta\mathcal{L}_{\text{global}}, \quad (5)$$

where β is a trade-off parameter to adaptively obtain the complementary information from the local structure preservation and the global structure preservation. Eq. (5) designs two representation distillation methods to preserve both the local structure and the global structure of nodes in the novel set, thus improving the robustness of the student model by lessening the issue of inefficient training nodes and making the best use of the information in the base set.

In our proposed representation distillation, the local representation distillation uses the embedding similarity on every graph convolutional layer in the teacher model to preserve the local structure of all nodes in the novel set, while the global representation distillation uses the embeddings of all nodes on every graph convolutional layer in the teacher model to preserve the global structure of all nodes in the novel set. It is noteworthy that previous self-training FSNC uses the parameters of the first graph convolutional layer in the teacher model to initialize the student model. Obviously, our method makes better use of the information of the teacher model than previous methods. Moreover, we use complementary information in the base set. Hence, our method could output a robust student model compared with previous FSNC methods.

After performing the representation distillation by Eq (5), we obtain the embeddings of the support set, the query set and the unlabeled set in the novel set. We further use these embeddings for the pseudo-label distillation.

Pseudo-label Distillation

The student model is influenced by either the use of information in the teacher model or the quality of pseudo-labels. For example, incorrect pseudo-labels introduce noise to mislead the model learning, while over-confident pseudo-labels result in the over-fitting issue rather than bringing new information. To address these issues, in this paper, we propose the pseudo-label distillation to perform knowledge distillation on pseudo-labels to improve their quality. To do this, given the embedding of all nodes in the novel set by the representation distillation, we first assign pseudo-labels to unlabeled nodes, and then perform hard prediction and soft prediction on the pseudo-labels. We then propose the pseudo-label distillation to output the soft labels of the nodes in the pseudo-label set, and further minimize the Kullback-Leibler divergence between the soft labels produced by the teacher model and the soft prediction outputted by the student model.

Pseudo-label Selection from Unlabeled Nodes After performing the representation distillation, we get the embeddings of the support set, the query set and the unlabeled set outputted by the student model. These embeddings are first used to calculate the probability matrix $\mathbf{P}$ by Eq. (1). We then use $\mathbf{P}$ to obtain the information entropy of all unlabeled nodes, *i.e.*, $\mathbf{h} = \{h_1, h_2, \cdots, h_Q\}$, where Q is the number of unlabeled nodes.

Next, we assign pseudo-labels to m nodes with the lowest information entropy for every class by:

$$\sum_{q=1}^{Q} \mathbb{1}\left(h_q < \tau_k\right) \cdot \mathbb{1}(\arg\max\left(\mathbf{p}_q\right) = k), \quad (6)$$

where k ($k \leq N$) represents the k-th class and τ_k is the information entropy of the m-th node in the k-th class. $\mathrm{argmax}(\cdot)$ is used to select the class with the highest predicted probability.

Denoting $\mathbf{Y}'$ as the new label set including original labels and new pseudo-labels, as well as the *pseudo-label set* as $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{y}'_i)\}_{i=1}^{N\times m}$, we further perform soft prediction and hard prediction on the pseudo-label set. The hard prediction is used to calculate the cross-entropy loss:

$$\mathcal{L}_{\mathrm{pse}} = -\sum_{i\in\mathcal{P}}\sum_{j=1}^{N} y'_{i,j} \ln p_{i,j}. \tag{7}$$

The soft prediction on the pseudo-label set is calculated by:

$$p_{i,k} = \frac{\exp(z_{i,k}/T)}{\sum_{k'} \exp(z_{i,k'}/T)}, \tag{8}$$

where T is a non-negative temperature. Moreover, the larger the value of T is, the smoother the output. In this paper, due to the limited number of training nodes in the support set, we propose the pseudo-label distillation to use the information of the teacher model to supervise the soft prediction outputted by the student model.

Pseudo-label Distillation Process As mentioned before, low-quality pseudo-labels (*i.e.*, incorrect pseudo-labels or over-confident pseudo-labels) will influence the effectiveness of the student model in the self-training FSNC, in this paper, we propose the pseudo-label distillation to improve the quality of pseudo-labels.

Specifically, we first input both the support set and pseudo-label set into the teacher model to obtain two types of embeddings, *i.e.*, the support embedding and the pseudo-label embedding. We then calculate the mean embedding for every class (*i.e.*, the embedding of the class center) of the support embedding. That is, the mean embedding of the k-th class is calculated by:

$$\mathbf{v}_k = \frac{1}{|\mathcal{S}_k|} \sum_{i\in\mathcal{S}_k} f_T^i\left(\mathbf{A}, \mathbf{X}\right), \tag{9}$$

where $\mathcal{S}_k$ indicates the support set in k-th class. The mean embedding for every class represents the common characteristics of the nodes within the same class. Moreover, a node is highly likely to be assigned to the class with the closest distance to the node.

We further assign every node in the pseudo-label embedding to the class with the minimal distance to its mean embedding. To do this, the predicted probability of the i-th pseudo-label node belonging to the k-th class in the teacher model is:

$$p'_{i,k} = \frac{\exp\left(-d\left(f_T^i(\mathbf{A}, \mathbf{X}), \mathbf{v}_k\right)/T\right)}{\sum_{k'} \exp\left(-d\left(f_T^i(\mathbf{A}, \mathbf{X}), \mathbf{v}_{k'}\right)/T\right)}, \tag{10}$$

where $d(\cdot,\cdot)$ is a measurement metric (*e.g.*, l_2 distance in this paper) to measure the distance between two embeddings.

Both the soft labels in Eq. (10) by the teacher model and the soft prediction in Eq. (8) by the student model construct different data distributions, but both of them predict the same nodes in the pseudo-label set. Hence, they should have connection, which can be used to improve the quality of pseudo-labels. In particular, the soft labels are obtained with sufficient nodes while the soft predictions are obtained with limited nodes, so it is obvious to use the information in the teacher model to supervise the learning of the student model. Based on the perspective of data distribution, Kullback-Leibler divergence of two different data distributions is used to learn the student model by using the information from the teacher model. To do this, Kullback-Leibler divergence is used to calculate the distillation loss for the pseudo-label set by:

$$\mathcal{L}_{\mathrm{pseKD}} = \sum_{i\in\mathcal{P}} KL(\mathbf{p}'_i, \mathbf{p}_i). \tag{11}$$

Eq. (11) alleviates the misguidance issue by smoothing incorrect pseudo-labels with the supervision of the teacher model, and explores the over-fitting issue on the novel set with limited nodes by transferring the information from the teacher model. As a result, the pseudo-label distillation is able to improve the quality of the pseudo-labels.

Finally, integrating the cross-entropy loss in Eq. (7) with the Kullback-Leibler divergence in Eq. (11), the loss function of the pseudo-label distillation is formulated as follows:

$$\mathcal{L}_{\mathrm{PD}} = (1-\lambda)\mathcal{L}_{\mathrm{pse}} + \lambda\mathcal{L}_{\mathrm{pseKD}}, \tag{12}$$

where λ is a non-negative trade-off parameter.

Overall Objective Function

The final objective function is formulated as follows:

$$\mathcal{L}_{\mathrm{KD-FSNC}} = \mathcal{L}_{\mathrm{spt}} + \mu_1\mathcal{L}_{\mathrm{RD}} + \mu_2\mathcal{L}_{\mathrm{PD}}, \tag{13}$$

where μ_1 and μ_2 are non-negative parameter weights. Obviously, Eq. (13) simultaneously conducts the representation distillation and the pseudo-label distillation to achieve effective FSNC. As a result, the student model is trained twice. Specifically, the first training is conducted on the support set by Eq. (2) while the second training is conducted on both the support set and the pseudo-label set by Eq. (13). In particular, the second training is iterative, as known as self-training in the literature.

Experiments

Experimental Settings

Datasets and Comparison Methods We assess the effectiveness of our proposed method on six benchmark datasets, including three citation datasets (*i.e.*, Cora, CiteSeer and CoraFull) (Kipf and Welling 2016), two business datasets (*i.e.*, Computers and Photo) (Shchur et al. 2018), and one co-authorship dataset *i.e.*, Coauthor (CS) (Shchur et al. 2018).

The comparison methods include the baseline method GCN (Kipf and Welling 2016), two meta FSNC methods (*i.e.*, Meta-GNN (Zhou et al. 2019) and G-META (Huang and Zitnik 2020)), one self-training FSNC method, *i.e.*, IA-FSNC (Wu et al. 2022), and two knowledge distillation methods, *i.e.*, GraphAKD (He et al. 2022) and SimKD (Chen et al. 2022).

Datasets	Shot	GCN	Meta-GNN	G-META	GraphAKD	SimKD	IA-FSNC	KD-FSNC
Cora	1	73.3(2.2)	75.8(1.9)	69.7(0.1)	79.8(1.8)	80.3(2.2)	85.5(1.8)	**87.7(1.7)**
	3	86.2(1.0)	88.0(1.7)	83.0(0.1)	85.7(1.1)	87.7(0.1)	92.0(0.7)	**92.8(0.8)**
	5	90.1(0.7)	90.8(0.3)	85.2(0.1)	87.8(0.8)	89.3(0.7)	93.4(0.5)	**93.9(0.6)**
Citeseer	1	65.1(1.9)	67.4(1.6)	65.5(0.1)	71.2(2.2)	75.0(2.8)	78.4(2.6)	**80.1(2.9)**
	3	77.3(1.6)	79.0(1.3)	78.0(0.1)	76.4(1.5)	82.1(1.4)	85.3(1.5)	**86.2(1.4)**
	5	80.6(1.5)	83.0(1.8)	82.8(0.1)	77.4(1.3)	83.2(1.4)	86.9(1.2)	**87.0(1.3)**
Computers	1	81.7(3.0)	84.1(2.1)	87.5(0.1)	83.9(2.9)	85.2(3.0)	89.9(2.4)	**90.5(2.5)**
	3	93.1(0.8)	93.9(1.0)	92.9(0.1)	91.3(1.3)	92.9(1.1)	95.8(0.4)	**95.8(0.8)**
	5	95.2(0.5)	94.4(0.5)	92.6(0.1)	93.3(0.9)	95.0(0.7)	96.8(0.3)	**96.9(0.5)**
Photo	1	84.6(2.8)	86.5(3.0)	82.3(0.1)	83.9(2.9)	83.2(3.5)	90.9(1.8)	**91.6(2.3)**
	3	94.0(0.8)	92.9(0.8)	88.6(0.1)	91.8(1.1)	93.0(0.9)	95.5(0.5)	**96.3(0.7)**
	5	94.8(0.5)	93.7(0.4)	89.5(0.1)	94.1(0.5)	95.2(0.5)	96.5(0.3)	**96.8(0.4)**
Coauthor (CS)	1	86.5(0.7)	87.4(0.3)	86.8(0.2)	85.3(0.9)	88.0(0.8)	88.2(0.7)	**89.2(0.7)**
	3	92.9(0.1)	93.3(0.3)	91.7(0.2)	91.6(0.2)	93.6(0.2)	94.4(0.1)	**94.5(0.2)**
	5	93.0(0.3)	94.0(0.2)	93.9(0.2)	92.5(0.2)	94.8(0.1)	94.8(0.1)	**94.9(0.2)**
CoraFull	1	58.8(1.8)	61.7(2.0)	64.6(0.3)	63.7(1.6)	67.2(1.7)	72.1(1.6)	**73.5(1.5)**
	3	77.4(0.9)	79.8(0.4)	73.8(0.2)	78.3(0.8)	83.9(0.6)	81.9(0.8)	**84.1(0.6)**
	5	83.5(0.6)	85.7(0.7)	77.4(0.4)	82.0(0.6)	87.2(0.4)	85.8(0.6)	**87.4(0.5)**

Table 1: Classification accuracy (mean and standard deviation) of all methods with different shot numbers on all datasets and the bold number represents the best results in the whole row.

C1	C2	C3	Cora	CiteSeer	Computer	Photo	Coauthor (CS)	CoraFull
✓			89.4(0.8)	80.8(1.4)	93.4(0.9)	93.4(0.8)	92.5(0.1)	81.3(0.6)
	✓		90.2(0.6)	84.4(1.5)	95.1(0.7)	94.4(0.8)	94.1(0.2)	83.3(0.6)
		✓	91.2(0.7)	84.0(1.3)	95.2(0.6)	95.6(0.7)	94.3(0.1)	82.0(0.7)
✓	✓		90.6(0.7)	84.9(1.5)	95.1(0.6)	95.1(0.9)	94.3(0.2)	83.5(0.5)
✓		✓	90.7(0.9)	81.5(1.4)	93.2(0.9)	93.9(0.8)	92.7(0.2)	82.3(0.6)
	✓	✓	90.5(0.6)	83.8(1.3)	94.9(0.7)	94.1(0.7)	94.0(0.2)	83.9(0.5)
✓	✓	✓	**92.8(0.8)**	**86.2(1.4)**	**95.8(0.8)**	**96.3(0.7)**	**94.5(0.2)**	**84.1(0.6)**

Table 2: Classification accuracy (mean and standard deviation) of KD-FSNC with different components at 3-shot on all datasets and the bold number represents the best results in the whole column.

Setting-up We construct the N-way K-shot task for all FSNC methods, where N is the number of novel set and each class has K samples. Specifically, we set N as 2 on Datasets Cora, CiteSeer, Computers, and Photo, and 5 on Coauthor and CoraFull. In addition, K is set as 1, 3, and 5.

All experiments are conducted on a server with NVIDIA Tesla V100S (32GB memory each). We follow (Zhou et al. 2019) to randomly partition each dataset to conduct 30 few-shot learning experiments. In each experiments, we randomly generate the support set 30 times. As a result, we repeat the experiments 900 times and report the average results and the corresponding standard deviation in this paper.

In our proposed method, we optimize all parameters by Adam optimization algorithm (Kingma and Ba 2014) and set the learning rate in the range of $\{0.01, 0.02, \cdots, 0.05\}$. We set the self-training cycle times t in the range of $\{1, 2, \cdots, 10\}$, and set the number m of pseudo-labels in the range of $\{10, 20, \cdots, 100\}$. In addition, we set the parameters of all comparison methods according to the original literature so that they output the best performance.

Result Analysis

Effectiveness on the FSNC We evaluate the effectiveness of the proposed KD-FSNC by reporting the results (*i.e.*, classification accuracy) of all methods on six datasets with different shot numbers in Table 1. Obviously, the proposed KD-FSNC consistently achieves the best results on all datasets, followed by IA-FSNC, SimKD, GraphAKD, Meta-GNN, GCN, and G-META.

First, our method outperforms meta FSNC methods (*i.e.*, Meta-GNN and G-META) by a large margin. For example, the proposed KD-FSNC averagely improves by 8.29%,

3.78% and 2.58% respectively, compared with the best meta FSNC method Meta-GNN, in terms of 1-shot, 3-shot, and 5-shot on all datasets. The reason is that the proposed method is available to provide new supervisory information, compared with the meta FSNC methods.

Second, our method achieves promising improvements, compared with the self-training FSNC method (*i.e.*, IA-FSNC). Our KD-FSNC averagely improves by 1.25%, 0.82% and 0.49% respectively, compared with IA-FSNC on all datasets. Moreover, our method has a statistically significant difference from IA-FSNC based on the paired-sample t-tests at a 95% significance level. The reason is that our KD-FSNC fully uses the information in the teacher model via the representation distillation and improves the quality of pseudo-labels through the pseudo-label distillation.

Third, our method significantly outperforms two knowledge distillation methods. For example, our KD-FSNC averagely improves by 5.63%, 2.75% and 2.05% respectively, compared with the best knowledge distillation method SimKD, in terms of 1-shot, 3-shot, and 5-shot on all datasets. This indicates that the knowledge distillation methods do not work well for the FSNC problem.

Ablation Study

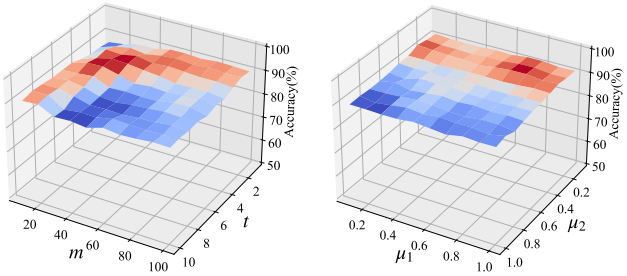
The proposed KD-FSNC conducts the local representation distillation (C1 for short) and the global representation distillation (C2 for short) to fully use the information in the base set. Moreover, it also conducts the pseudo-label distillation (C3 for short) to improve the quality of pseudo-labels.

To demonstrate the effectiveness of individual distillation methods, we investigate the node classification performance with different combinations of components on all datasets and report the results in Table 2. First, our method with all components averagely improves by 1.86%, compared with the methods with one component only. This indicates that both the representation distillation and the pseudo-label distillation are essential for our method, which verifies the feasibility of our proposed method. Second, the method with the pseudo-label distillation outperforms the method with the representation distillation. The reason is that the pseudo-label distillation improves the quality of the pseudo-label, making the learning of the student model more generalizable. Third, the local representation distillation only showed poor results, but combined with the global representation distillation gave good results. The reason is that the feature extractor of the student model has not learned enough information from the teacher model. Hence, it is reasonable to consider both the local representation distillation and the global representation distillation.

Parameter Sensitivity Analysis

Our method has four important hyper-parameters, *i.e.*, the number of self-training cycles t, the number m of pseudo-labels in each class, and parameter weights μ_1 and μ_2 in Eq. (13). We investigate the sensitivity of our method to these hyper-parameters at 3-shot on Cora dataset, and then report the results in Figure 2.

First, we vary the number of self-training cycles t in the range of $\{1, 2, \cdots, 10\}$, and the number m of pseudo-labels in each class in the range of $\{10, 20, \cdots, 100\}$. Our experimental findings reveal the sensitivity of our method to the settings of the values of t and m. Specifically, we observe an increase in accuracy as the values t rise from 1 to 4 and m increases from 10 to 60. However, a decline in accuracy is observed when both t and m assume large values. This decline can be attributed to the inevitable introduction of numerous incorrect pseudo-labels, leading the model to learn wrong information. It underscores the importance of carefully selecting these parameters to strike a balance between obtaining sufficient information and avoiding the incorporation of inaccurate labels.



(a) The effect of t and m (b) The effect of μ_1 and μ_2

Figure 2: Parameter sensitivity study of self-training t, the number m of pseudo-labels in each class, and parameter weights μ_1, and μ_2 at 3-shot on the Cora dataset.

Second, we vary the values of μ_1 and μ_2 in the range of $\{0.1, 0.2, \cdots, 1.0\}$. Our method is sensitive to the settings of μ_1 and μ_2. Specifically, the accuracy increases with increasing values of μ_1 from 0.1 to 0.8 and increasing values of μ_2 from 0.1 to 0.3. However, the accuracy starts to decrease when two values are large. The reason is that the student model over-imitates the teacher model with large μ_1 and the student model over-relies on pseudo-label information with large μ_2. Both scenarios will influence the student model learning, whose generalization ability is reduced.

Conclusion

In this paper, we introduced a new self-training method for FSNC that incorporates both representation distillation and pseudo-label distillation to enhance the effectiveness of FSNC. Specifically, we conducted representation distillation, encompassing both local representation distillation and global representation distillation. This process involves transferring information from all graph convolutional layers in the teacher model to facilitate learning by the student model. Meanwhile, we conducted the pseudo-label distillation to improve the quality of pseudo-labels. Extensive experimental results showed that our method achieved supreme performance, compared to SOTA methods.

Acknowledgments

This work was supported in part by National Key Research and Development Program of China under Grant 2022YFA1004100.

References

Chauhan, J.; Nathani, D.; and Kaul, M. 2020. Few-shot learning on graphs via super-classes based on graph spectral measures. *arXiv preprint arXiv:2002.12815*.

Chen, D.; Mei, J.-P.; Zhang, H.; Wang, C.; Feng, Y.; and Chen, C. 2022. Knowledge Distillation with the Reused Teacher Classifier. In *CVPR*, 11933–11942.

Chu, R.; Ye, X.; Liu, Z.; Tan, X.; Qi, X.; Fu, C.-W.; and Jia, J. 2022. TWIST: Two-Way Inter-label Self-Training for Semi-supervised 3D Instance Segmentation. In *CVPR*, 1100–1109.

Deng, X.; and Zhang, Z. 2021. Graph-free knowledge distillation for graph neural networks. *arXiv preprint arXiv:2105.07519*.

Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; and Liu, H. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *CIKM*, 295–304.

Feyjie, A. R.; Azad, R.; Pedersoli, M.; Kauffman, C.; Ayed, I. B.; and Dolz, J. 2020. Semi-supervised few-shot learning for medical image segmentation. *arXiv preprint arXiv:2003.08462*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Gao, M.; Shen, Y.; Li, Q.; Wan, L.; and Tang, X. 2018. Feature matters: A stage-by-stage approach for task independent knowledge transfer.

He, H.; Wang, J.; Zhang, Z.; and Wu, F. 2022. Compressing Deep Graph Neural Networks via Adversarial Knowledge Distillation. *arXiv preprint arXiv:2205.11678*.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Huang, K.; and Zitnik, M. 2020. Graph meta learning via local subgraphs. In *NeurIPS*, 5862–5874.

Joshi, C. K.; Liu, F.; Xun, X.; Lin, J.; and Foo, C.-S. 2021. On Representation Knowledge Distillation for Graph Neural Networks. *arXiv preprint arXiv:2111.04964*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lan, L.; Wang, P.; Du, X.; Song, K.; Tao, J.; and Guan, X. 2020. Node classification on graphs with few-shot novel labels via meta transformed network embedding. In *NeurIPS*, 16520–16531.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2023. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *SIGIR*, 1559–1568.

Luo, Y.; Luo, G.; Yan, K.; and Chen, A. 2022. Inferring from References with Differences for Semi-Supervised Node Classification on Graphs. *Mathematics*, 10(8): 1262.

Ma, N.; Bu, J.; Yang, J.; Zhang, Z.; Yao, C.; Yu, Z.; Zhou, S.; and Yan, X. 2020a. Adaptive-Step Graph Meta-Learner for Few-Shot Graph Classification. In *CIKM*, 1055–1064.

Ma, Y.; Bai, S.; An, S.; Liu, W.; Liu, A.; Zhen, X.; and Liu, X. 2020b. Transductive Relation-Propagation Network for Few-shot Learning. In *IJCAI*, 804–810.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30.

Tu, W.; Zhou, S.; Liu, X.; Liu, Y.; Cai, Z.; Zhu, E.; Zhang, C.; and Cheng, J. 2022. Initializing Then Refining: A Simple Graph Attribute Imputation Network. In *IJCAI*, 3494–3500.

Wang, H.; Nie, F.; Huang, H.; Wang, H.; Nie, F.; and Huang, H. 2014. Globally and locally consistent unsupervised projection. In *AAAI*.

Wei, C.; Shen, K.; Chen, Y.; and Ma, T. 2020. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*.

Wei, Y.; Fu, X.; Sun, Q.; Peng, H.; Wu, J.; Wang, J.; and Li, X. 2022. Heterogeneous graph neural network for privacy-preserving recommendation. In *ICDM*, 528–537.

Wu, Z.; Zhou, P.; Wen, G.; Wan, Y.; Ma, J.; Cheng, D.; and Zhu, X. 2022. Information Augmentation for Few-shot Node Classification. In *IJCAI*, 3601–3607.

Yang, Q.; Wei, X.; Wang, B.; Hua, X.-S.; and Zhang, L. 2021. Interactive self-training with mean teachers for semi-supervised object detection. In *CVPR*, 5941–5950.

Yang, S.; Liu, L.; and Xu, M. 2021. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*.

Yao, H.; Zhang, C.; Wei, Y.; Jiang, M.; Wang, S.; Huang, J.; Chawla, N.; and Li, Z. 2020. Graph few-shot learning via knowledge transfer. In *AAAI*, 6656–6663.

Zhang, C.; Ding, K.; Li, J.; Zhang, X.; Ye, Y.; Chawla, N. V.; and Liu, H. 2022. Few-Shot Learning on Graphs: A Survey. *arXiv preprint arXiv:2203.09308*.

Zhang, G.; Cheng, D.; Yuan, G.; and Zhang, S. 2024. Learning fair representations via rebalancing graph structure. *Information Processing & Management*, 61(1): 103570.

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2021. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *CIKM*, 2357–2360.