Pseudo-Labeling with Graph Active Learning for Few-shot Node Classification

Quan Li[1], Lingwei Chen[2], Shixiong Jing[1], Dinghao Wu[1]
[1]College of Information Sciences and Technology, Pennsylvania State University, University Park PA, USA
[2]Department of Computer Science and Engineering, Wright State University, Dayton OH, USA

***Abstract*—Graphs have emerged as one of the most important and powerful data structures to perform content analysis in many fields. In this line of work, node classification is a classic task, which is generally performed using graph neural networks (GNNs). Unfortunately, regular GNNs cannot be well generalized into the real-world application scenario when the labeled nodes are few. To address this challenge, we propose a novel few-shot node classification model that leverages pseudo-labeling with graph active learning. We first provide a theoretical analysis to argue that extra unlabeled data benefit few-shot classification. Inspired by this, our model proceeds by performing multi-level data augmentation with consistency and contrastive regularizations for better semi-supervised pseudo-labeling, and further devising graph active learning to facilitate pseudo-label selection and improve model effectiveness. Extensive experiments on four public citation networks have demonstrated that our model can effectively improve node classification accuracy with considerably few labeled data, which significantly outperforms all state-of-the-art baselines by large margins.**

***Index Terms*—node classification, graph neural networks, data augmentation, active learning, pseudo-labeling**

I. Introduction

Graphs have recently emerged as one of the most important and powerful data structures to perform real-world content analysis [1]–[4]. In this line of work, node classification is a classic task, which is generally performed using graph neural networks (GNNs) [5]–[7] through neighborhood information aggregation. However, GNNs cannot be well generalized into the real-world application scenario when the labeled nodes are few. For example, it is generally expensive and time-consuming to obtain the relation, location, or theme labels for a large number of texts [8], [9]; when performing social network analysis, due to privacy concerns, most social media websites and apps limit the access to some personal information, where attribute labels may only be available on few users [10], [11]. In other words, when applied to such datasets, GNNs may suffer from low generalizability to the unlabeled nodes.

To address few-shot learning challenge, meta-learning has been proposed to leverage distribution of tasks to learn a shared initialization that adapts to new task [12]–[14]; this leads to a surge of graph meta-learning models to leverage prior knowledge for few-shot node classification [15]–[17]. The classes for meta-training and meta-testing are disjoint, but the data are typically obtained from the same domain [18], which is impractical in many real-world content analysis settings. More importantly, these models overlook the benefits from unlabeled nodes to facilitate few-shot node classification. As such, self-training GNN models [19], [20] are proposed to make use of the unlabeled nodes; however, they are still unsatisfying in two aspects: (1) similar to meta-learning, their model parameters need to be initialized using prior knowledge from base classes; and (2) the labeling and selection of the unlabeled nodes are too simple to introduce new precise supervisory information for classification performance improvement.

In this paper, we take initiatives to design a few-shot node classification model via pseudo-labeling with graph active learning to address the above issues, where this model is only built upon one learning task with the target classes. We first provide a simple theoretical analysis to argue that extra unlabeled data benefit few-shot classification, especially when the pseudo-labeling strategy is better formulated in a semi-supervised manner. GNN itself is known as a semi-supervised model through message passing [21], [22], but its vanilla design suffers from over-smoothing [23] on node embedding and over-fitting the scarce label information [24]. This leads to low generalizability, especially for large graphs, which further weakens GNNs to learn from few labeled nodes.

We thus propose to enhance GNN's semi-supervised capability for pseudo-labeling by designing data augmentation through consistency regularization [25], [26] and contrastive regularization [27], [28]. Different from previous studies [24], [29], to better facilitate augmenting few labeled nodes in our learning scenario, we perform multi-level (i.e., weak and strong) random perturbations onto node features and graph structure that renders nodes less sensitive to specific neighborhoods, and use contrastive regularization to complement consistency regularization that diversifies the model predictions and enables the confident labeling information to be propagated from the labeled nodes into more unlabeled ones at higher orders during training. Further, we introduce an effective yet efficient graph active learning paradigm by maximizing m-hop propagation of information gain to not only select high-confidence and balanced pseudo-labels, but also those most valuable ones that can best contribute to label propagation and model improvement. The selected pseudo-labels are then combined with true labels to learn the final few-shot node classification model.

II. Preliminaries

A. Graph Neural Networks

We denote the given graph as $G = (V, E, \mathbf{X})$, where $V (n = |V|)$ is the set of nodes, E is the set of edges specifying relationships among nodes, and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the feature matrix. Each labeled node is associated with a ground truth $y \in Y = \{0, 1, \cdots, k-1\}$. Edges E can be encoded as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_{ij} = \{0, 1\}$, where if $(v_i, v_j) \in E$, then $\mathbf{A}_{ij} = 1$; otherwise, $\mathbf{A}_{ij} = 0$. The GNN models are designed so that each node can aggregate information from its neighbors and generate higher-level node embedding. The graph aggregation layer is defined as follows:

$$\mathbf{H}^{(l)} = \text{aggregate}\left(\mathbf{H}^{(l-1)}, \mathbf{A}, \mathbf{W}^{(l)}\right) \tag{1}$$

where $\mathbf{H}^{(l-1)}$ and $\mathbf{H}^{(l)}$ are the input and output ($l \geq 1$) for layer l, $\mathbf{W}^{(l)}$ is a learnable weight matrix, and $\mathbf{H}^{(0)} = \mathbf{X}$. The output of GNNs with L layers can be computed as $\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) = \text{softmax}\left(\mathbf{H}^{(L)}\right)$. Accordingly, the GNN models can be optimized by minimizing the following training loss:

$$\mathbf{W}^* = \underset{\mathbf{W}}{\text{argmin}}\ \mathcal{L}(\mathbf{Z}, \mathbf{y}_l) + \lambda\|\mathbf{W}\|_2^2 \tag{2}$$

We focus on transductive inferences in this paper where all node connections and features are accessible during training. Therefore, $\mathcal{L}(\cdot)$ is specifically formulated to improve the GNN's semi-supervised capability for better label propagation.

B. *Few-shot Node Classification*

Given the graph $G = (V, E, \mathbf{X})$, nodes V can be divided into labeled node set V_l and unlabeled node set V_u. Due to high cost of annotation or limited access to node information, we practically consider only few of the nodes have labels (i.e., $|V_l| \ll |V|$). The few-shot node classification problem can then be defined to use labeled and unlabeled data to train a GNN model $f_{\mathbf{W}}(\mathbf{A}, \mathbf{X})$ that can effectively predict the labels for unlabeled nodes from V_u.

III. Proposed Model

In this section, we present the technical details of our proposed model, the overview of which is illustrated in Figure 1.

A. *Theoretical Motivation*

To theoretically analyze our motivation that unlabeled data boost the data-limited classification performance, we can use a binary classifier with a data generation probability P that mixes different Gaussian distributions for different labels (i.e., $y = i \sim \mathcal{N}(\mu_i, \sigma^2)$, $i \in \{0, 1\}$, and $\mu = (\mu_0 + \mu_1)/2$), such that an optimal binary classifier would classify an input x as positive when $x > \mu$. Accordingly, when the unlabeled data from P is available, we can generate $\bar{n}$ pseudo-labels from them using a target classifier with $\bar{n}_0$ negatives $\{x_i^0\}_{i=1}^{\bar{n}_0}$ and $\bar{n}_1$ positives $\{x_i^1\}_{i=1}^{\bar{n}_1}$. As our training data is balanced, we assume that the target classifier provides the same (or similar) accuracy for different labels. To learn the decision boundary μ using the pseudo-labels from the unlabeled data, the estimate can be formulated as $\bar{\mu} = \frac{1}{2}(\sum_{i=1}^{\bar{n}_0} x_i^0/\bar{n}_0 + \sum_{i=1}^{\bar{n}_1} x_i^1/\bar{n}_1)$. Based on the aforementioned setup for classifier and data distribution, a theorem can be derived as follows:

***Theorem** 1:* The estimate $\bar{\mu}$ satisfies $|\bar{\mu} - \mu| \leq \zeta$, with probability $P \geq 1 - 2e^{-\frac{2\zeta^2}{\sigma^2}\cdot\frac{\bar{n}_0\bar{n}_1}{\bar{n}_0+\bar{n}_1}}$ for any $\zeta > 0$.

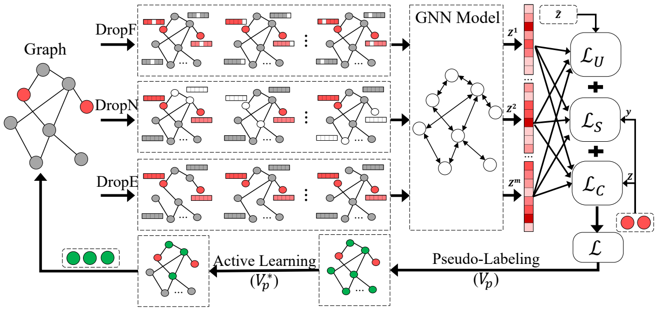


Fig. 1. The overview of our proposed model (DropF: drop features, DropN: drop nodes, DropE: drop edges).

Proof of Theorem 1. Given $\bar{n}$ pseudo-labels with $\bar{n}_0$ negatives and $\bar{n}_1$ positives, if the pseudo-label is correct, $x_i^0 \sim \mathcal{N}(\mu_0, \sigma^2)$ and $x_i^1 \sim \mathcal{N}(\mu_1, \sigma^2)$. As such, we can bound $\sum_{i=1}^{\bar{n}_0} x_i^0/\bar{n}_0 + \sum_{i=1}^{\bar{n}_1} x_i^1/\bar{n}_1$ using standard Gaussian concentration inequality. Accordingly, we can derive:

$$\mathbb{P}(|\sum_{i=1}^{\bar{n}_0} x_i^0/\bar{n}_0 + \sum_{i=1}^{\bar{n}_1} x_i^1/\bar{n}_1 - (\mu_0 + \mu_1)| > t) \leq 2e^{-\frac{t^2}{2\sigma^2}\cdot\frac{1}{1/\bar{n}_0+1/\bar{n}_1}} \tag{3}$$

Given $\zeta > 0$, considering the condition that the estimate $\bar{\mu}$ satisfies $|\bar{\mu} - \mu| \leq \zeta$, the formulation of the estimate $\bar{\mu}$ and $\mu = (\mu_0 + \mu_1)/2$, this inequality can be specified as:

$$|(\sum_{i=1}^{\bar{n}_0} x_i^0/\bar{n}_0 + \sum_{i=1}^{\bar{n}_1} x_i^1/\bar{n}_1) - (\mu_0 + \mu_1)| \leq 2\zeta \tag{4}$$

Since we already have the concentration inequality in Eq. (3), we can easily obtain the following lower bound on the probability of $|\bar{\mu} - \mu| \leq \zeta$:

$$\mathbb{P}(|\bar{\mu} - \mu| \leq \zeta) \geq 1 - 2e^{-\frac{2\zeta^2}{\sigma^2}\cdot\frac{1}{1/\bar{n}_0+1/\bar{n}_1}} = 1 - 2e^{-\frac{2\zeta^2}{\sigma^2}\cdot\frac{\bar{n}_0\bar{n}_1}{\bar{n}_0+\bar{n}_1}} \tag{5}$$

This completes the proof of Theorem 1.

We can interpret the theorem in the way that if the target classifier is reasonably well-performing such that the pseudo-labels are promisingly correct, $\bar{n}_0$ and $\bar{n}_1$ can be viewed as approximations for the number of actual positives and negatives in the unlabeled data. When $\bar{n}_0 = \bar{n}_1$ ($\bar{n} = \bar{n}_0 + \bar{n}_1$), $\frac{\bar{n}_0\bar{n}_1}{\bar{n}_0+\bar{n}_1}$ is maximized, and $e^{-\frac{2\zeta^2}{\sigma^2}\cdot\frac{\bar{n}_0\bar{n}_1}{\bar{n}_0+\bar{n}_1}}$ is minimized; accordingly, the probability P is maximized, which implies that $\bar{\mu}$ is more closely to estimate μ. In other words, more balanced pseudo-labels from the unlabeled data with a good target classifier are significantly useful to learn the optimal classification boundary. When we generalize this analysis to our application scenario that classifies nodes on graphs, the target classifier is a GNN model. To collect more balanced pseudo-labels from the unlabeled nodes, a semi-supervised strategy tends to be more powerful to leverage unlabeled nodes for better pseudo-labeling with low cost [22], [26]. This aligns with the assumption of having "a good target classifier" in Theorem 1. Inspired by these, we probe the effectiveness of pseudo-labels on few-shot node classification through adopting semi-supervised learning, which is introduced in detail as follows.

B. Semi-supervised Learning

Despite its wide semi-supervised applications, the vanilla GNN design suffers from over-smoothing [23] on node embeddings and over-fitting the scarce label information [24] that leads to low generalizability. Its straightforward leverage thus constrains the derived pseudo-labels from estimating the optimal classification boundary and fail to enhance few-shot performance. To this end, we elaborate data augmentation through consistency regularization [24]–[26] and contrastive regularization [27]–[29] to address the over-smoothing and over-fitting issues and improve GNN's semi-supervised capability for better pseudo-labeling.

Data Augmentation. Data augmentation aims to break the co-dependency of specific neighborhood for each node by adding random perturbations, such that the node labels and features can be more effectively propagated through graph structure to higher orders. Given node features and graph structure, we design a multi-level random perturbation, including weak augmentation and strong augmentation.

- *Weak augmentation* perturbs feature space of each node to generate augmentations. We randomly drop features with drop rate r by zeroing the corresponding columns in feature matrix $\mathbf{X}$ without impacting graph structures and obtain a new feature matrix $\overline{\mathbf{X}}$.
- *Strong augmentation* perturbs graph structure to induce more significant output variations [30] by randomly dropping nodes or edges with drop rate r. To drop nodes, we zero the designated rows in feature matrix $\mathbf{X}$ to derive $\widetilde{\mathbf{X}}$. To drop edges, we select the specified edges and set their values in adjacency matrix $\mathbf{A}$ to 0 to get $\widehat{\mathbf{A}}$.

Each augmentation is performed by multiplying $\mathbf{X}$ or $\mathbf{A}$ with the corresponding mask matrix to drop the specified features, nodes, or edges. After each random perturbation, the augmented data is fed to a GNN model to calculate the prediction output. Based on different data augmentations, the prediction outputs can be differently represented as follows:

$$\overline{\mathbf{Z}} = f_{\mathbf{W}}(\mathbf{A}, \overline{\mathbf{X}}) \quad \text{or} \quad \widetilde{\mathbf{Z}} = f_{\mathbf{W}}(\mathbf{A}, \widetilde{\mathbf{X}}) \quad \text{or} \quad \widehat{\mathbf{Z}} = f_{\mathbf{W}}(\widehat{\mathbf{A}}, \mathbf{X}) \tag{6}$$

where each output $\mathbf{Z} \in \{\overline{\mathbf{Z}}, \widetilde{\mathbf{Z}}, \widehat{\mathbf{Z}}\} \in \mathbb{R}^{n \times k}$ is the prediction probabilities for all nodes (labeled and unlabeled) in the graph.

Consistency Regularization. Consistency regularization [24], [26] works in a way that model predictions should be invariant to the input when masked using different perturbations [31]. We utilize it to construct the loss function to regulate our semi-supervised learning, which consists of an unsupervised loss $\mathcal{L}_U$ and a supervised loss $\mathcal{L}_S$.

- Unsupervised loss. Given outputs $\{\mathbf{Z}^b\}_{b=1}^{B}$ generated by B random perturbations with one specific augmentation, we first calculate the label distribution center as $\mathbf{Z}_i^* = \frac{1}{B}\sum_{b=1}^{B} \mathbf{Z}_i^b$, and then optimize their prediction consistency by minimizing the squared L_2 distance between each output $\mathbf{Z}_i$ corresponding node i and its label distribution center $\mathbf{Z}_i^*$:
$$\mathcal{L}_{cr} = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=0}^{n-1} \parallel \mathbf{Z}_i^b - \mathbf{Z}_i^* \parallel_2^2 \tag{7}$$
Following this formulation, we can accordingly construct the unsupervised loss as $\mathcal{L}_U = \mathcal{L}_{cr-f} + \mathcal{L}_{cr-n} + \mathcal{L}_{cr-e}$ by aggregating the losses from three types of augmentations (i.e., $\mathcal{L}_{cr-f}$ for dropping features, $\mathcal{L}_{cr-n}$ for dropping nodes, and $\mathcal{L}_{cr-e}$ for dropping edges).
- Supervised loss. With V_l denoting the set of nodes with labels, we minimize a standard cross-entropy loss between the ground truth of each labeled node and its B predictions in one specific augmentation:
$$\mathcal{L}_{ce} = -\frac{1}{B}\sum_{b=1}^{B}\sum_{i=0}^{|V_l|-1} \mathbf{y}_i \log \mathbf{Z}_i^b \tag{8}$$
leading to the supervised loss $\mathcal{L}_S = \mathcal{L}_{ce-f} + \mathcal{L}_{ce-n} + \mathcal{L}_{ce-e}$ by aggregating the losses from three types of augmentations.

Contrastive Regularization. It can be observed from Eq. (7) and Eq. (8) that consistency regularization that only considers positive node pairs pulls the augmented node features in the same label cluster but may fail to push the features in different clusters. This might homogenize the model predictions and restrict the active label propagation. To address this potential issue, we further employ contrastive regularization to diversify model predictions and enhance label propagation. Specifically, we adjust SupContrast [32] into semi-supervised learning setting. We define the set of positive pairs as the predictions between the augmentations $\mathbf{Z}_i^b$ and the corresponding labeled nodes $\mathbf{Z'}_i$, and the set of negative pairs as the predictions between the augmentations $\mathbf{Z}_i^b$ and the labeled nodes in different label clusters $\mathbf{Z'}_j$ $(i \neq j)$. Then, the contrastive regularization is to minimize the following loss:

$$\mathcal{L}_{ct} = -\frac{1}{B}\sum_{b=1}^{B}\sum_{i=0}^{|V_l|-1} \log \frac{\exp(\mathbf{Z}_i^b \cdot \mathbf{Z'}_i/\tau)}{\exp(\mathbf{Z}_i^b \cdot \mathbf{Z'}_i/\tau) + \sum_{j\neq i}^{K} \exp(\mathbf{Z}_i^b \cdot \mathbf{Z'}_j/\tau)} \tag{9}$$

where τ is the temperature parameter, K is the number of negative pairs for each augmentation, and $\cdot$ denotes the inner (dot) product. In this way, the contrastive regularization loss can be constructed as $\mathcal{L}_C = \mathcal{L}_{ct-f} + \mathcal{L}_{ct-n} + \mathcal{L}_{ct-e}$ by aggregating the losses from three types of augmentations.

Optimization. The final loss function of our semi-supervised learning model for pseudo-labeling is:

$$\mathcal{L} = \mathcal{L}_S + \alpha\mathcal{L}_U + \beta\mathcal{L}_C \tag{10}$$

where α and β are both balance parameters that are set up to adjust the relative weight of unsupervised loss and contrastive regularization loss, respectively.

C. Graph Active Learning

We can obtain the pseudo-labels by minimizing $\mathcal{L}$ using gradient descent. As discussed in the theoretical motivation, the correctness and balance of pseudo-labels affect the effectiveness of using pseudo-labels to estimate the optimal classification boundary. As such, we retain the high-confidence pseudo-labels from $|V_u|$ unlabeled nodes whose largest prediction class probability falls above a predefined threshold ν, which can use the following formula:

$$V_p = \{(\mathbf{X}_i, \text{argmax}(\mathbf{Z}_i)) | \mathbb{1}(\max(\mathbf{Z}_i) \geq \nu)\}_{i=1}^{|V_u|} \tag{11}$$

Considering the fact that some nodes can better contribute to label propagation and model improvement in the graph, we would like to further design a graph active learning to facilitate selecting the most valuable pseudo-labels.

We use a simple GNN model as an oracle, and the goal here is to select a subset of pseudo-labels V_p^*, such that the GNN model trained with the supervision of V_p^* and V_l can get the lowest loss on the test node set. Specifically, we leverage a criteria to maximize propagation of information gain (i.e., entropy reduction) on graph [33]. Each pseudo-label will propagate its label information to its m-hop neighbors and impact them. This influence score of node v_i on node v_j after m-layer propagation can be calculated as follows:

$$\hat{I}(v_i, v_j, m) = \|\mathrm{E}[\partial \mathbf{X}_\mathbf{j}^\mathbf{m} / \partial \mathbf{X}_\mathbf{i}^\mathbf{0}]\| \tag{12}$$

which is the $L1$-norm of the expected Jacobian matrix. Formally, we normalize this influence score as

$$I(v_i, v_j, m) = \hat{I}(v_i, v_j, m) / \sum_{v \in V} \hat{I}(v, v_j, m) \tag{13}$$

where $I(v_i, v_j, m)$ represents the sum across probabilities of all possible influential paths with length of m from v_j to v_i for a m-layer GNN. This implies that the larger $I(v_i, v_j, m)$, the more v_i impacts on v_j if v_i is pseudo-labeled.

As such, we extend the information gain of a single node to its m-hop neighbors in the graph as follows:

$$G(v_i, v_j, m) = H(\sum_{v \in V_l} I(v, v_j, m)\mathbf{Z}_v) - H(\sum_{v \in V_l \cup \{v_i\}} I(v, v_j, m)\mathbf{Z}_v) \tag{14}$$

where H is the entropy. In this way, to select a pseudo-label, we can proceed by maximizing the following objective function $F(V_p)$:

$$v_p = \underset{V_p}{\operatorname{argmax}}\ F(V_p) = \sum_{v_i \in V_p} \sum_{v_j \in N(v_i)} G(v_i, v_j, m) \tag{15}$$

where $N(v_i)$ is v_i and its m-hop neighbors. Considering the influence propagation, the proposed objective can be used to find a subset V_p^* that can maximize the information gain of all influenced nodes as more as possible. Accordingly, we use greedy search to select such a subset of pseudo-labels V_p^* from V_p with the size of S for each label. These selected pseudo-labels are then fed to the data-augmented semi-supervised learning to train the final GNN model.

IV. Experimental Results and Analysis

A. Experimental Setup

Datasets and parameters. In this paper, we evaluate our model with four public citation datasets: Cora, Citeseer, PubMed [5], and DBLP [34]. We use 5 labeled instances per class as training data and randomly select 500 instances from the remaining as test data. The drop rate $r = 0.5$ and the size $B = 4$ are set for all data augmentation strategies, the balanced parameters are set as $\alpha = 0.5$ and $\beta = 0.3$, the size of pseudo-labels selected for each class is $S = 5$, and the negative pair size for contrastive regularization is $K = 5$. We further evaluate the impacts of different training sizes N, the pseudo-label size S, the drop rate r, the balance parameter α and β, and the number of negative pairs K in Section IV-C.

TABLE I
Comparison results for Cora and Citeseer (accuracy %)

Models	Shots	Cora	Citeseer
GCN	1	60.33	58.44
	3	75.15	67.99
SGC	1	61.64	56.91
	3	75.67	65.67
Graph-SAGE	1	50.89	53.49
	3	53.12	55.01
META-GCN	1	63.72	61.91
	3	76.78	69.43
META-SGC	1	65.27	60.46
	3	77.19	68.65
G-META	1	64.57	61.26
	3	73.76	69.82
GPN	1	60.79	60.53
	3	76.21	68.67
Our model	1	**76.11**	**64.25**
	3	**83.60**	**71.34**

TABLE II
Comparison results for PubMed and DBLP (accuracy %)

Models	Shots	PubMed	DBLP
GCN	3	58.89	43.90
	5	65.77	51.20
SGC	3	63.37	40.20
	5	64.93	50.30
META-GCN	3	-	60.70
	5	-	63.10
G-META	3	-	63.20
	5	-	64.20
GPN	3	-	62.60
	5	-	64.40
Our model	3	**76.41**	**64.80**
	5	**83.80**	**73.42**

Baselines. In our study, we use 7 state-of-the-art GNN models as baselines, including graph convolutional network (GCN) [5], Graph-SAGE [6], and simple graph convolution (SGC) [35] for addressing over-smoothing/over-fitting; META-GCN [16], META-SGC [35], G-META [17], and graph prototypical network (GPN) [36] for few-shot learning.

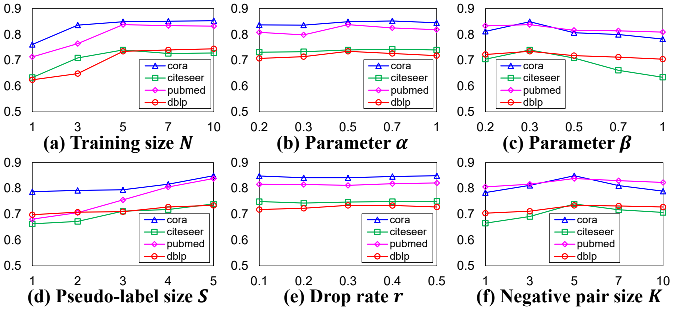
B. Comparisons with Baselines

In this section, we compare our model with the selected GNN baselines for few-shot node classification. The results for baselines are taken directly from related papers for comparisons, while "-" means that we cannot find the result for that specific model. As shown in Table I and Table II, we can observe that mate-learning based few-shot models demonstrate better performance than traditional GNNs, while our model outperforms baselines by a large margin in different shots. For example, when "1-shot" is applied, the classification accuracy is 76.11% and 64.25% for Cora and Citeseer respectively with the improvement margin ranging in $(11, 26)\%$ for Cora and $(2, 11)\%$ for Citeseer. For PubMed and DBLP, our model also delivers the better performance with only " 3-shots " are available, where the performance increases by a margin of $(13, 18)\%$ and $(1.5, 25)\%$, respectively. These comparison results in Table I and Table II also reveal another interesting observation: the performance of our model with just 1-shot or 3-shot is comparable to or better than that of most baselines with a higher number of shots. This advantage is particularly evident when the comparisons are conducted on PubMed and DBLP. These observations demonstrate that our model can achieve state-of-the-art performance with less labeled nodes, making it a promising method for few-shot node classification tasks. In summary, the comparative study confirms that (1) regular GNNs can capture the structural information of graphs but struggle to learn from few labeled nodes, (2) meta-learning paradigm can improve the few-shot performance to some extend, and (3) our model that leverages semi-supervised pseudo-labeling with consistency regularization and contrastive regularization, and graph activate learning for pseudo-label selection contributes better to few-shot learning than GNN-based meta-learning and regular GNNs.

C. Parameter Evaluation

The performance of our model can be potentially impacted by the following parameters: training size N (number of labeled nodes per class), number of pseudo-labels S selected for each class using graph active learning, drop rate r for weak/strong augmentation, number of negative pairs K for contrastive regularization, and α and β for adjusting the training loss weights. In this section, we evaluate our model using accuracy under different parameter settings: $N \in \{k \times 1, k \times 3, k \times 5, k \times 7, k \times 10\}$; $S \in [1, 5]$ with $N = k \times 5$; $r \in [0.1, 0.5]$; $\alpha \in [0, 1]$ with $\beta = 0.3$ and $\beta \in [0, 1]$ with $\alpha = 0.5$; and $K \in \{1, 3, 5, 7, 10\}$. All the experimental results regarding these parameters are illustrated in Figure 2.

- As illustrated in Figure 2(a), when we apply more shots in training, the performance of our model keeps increasing, but the increments of the performance in $[5, 10]$ are more stable than that in $[1, 5]$. With the training size increasing, the advantage of our model narrows and the performance is closer to the upper bound.
- Figure 2(b) indicates a relatively small accuracy increase when we enlarge α. However, Figure 2(c) demonstrates that when we enlarge β, the accuracy first slightly increases, rises to a high level at $\beta = 0.3$, and then drastically drops when β changes from 0.3 to 1. The reason behind this trend could be: when β is relatively small, the negative pairs can facilitate narrowing down the limitations of consistency regularization; when β is large, the negative pairs in contrastive regularization might overshadow the influence of the positive pairs enforced by consistency regularization, which may degrade the performance of the model.
- Regarding the impact of the selected pseudo-label size S from graph active learning, Figure 2(d) implies that the performance of our model generally continues to improve with the increase of S, where the model enjoys the most benefit when the size ranges from 2 to 4.
- As for the drop rate r, Figure 2(e) shows that the accuracy of our model across all four datasets remains stable when increasing r. This suggests that the drop rate does not have a significant impact on the performance of our model.



(a) Training size ***N*** (b) Parameter ***α*** (c) Parameter ***β***

(d) Pseudo-label size ***S*** (e) Drop rate ***r*** (f) Negative pair size ***K***

Fig. 2. Evaluation on the impacts of different parameters (accuracy %).

- Figure 2(f) provides an insight that as more negative pairs are incorporated, the accuracy of the model increases; the highest accuracy is achieved when $K = 5$. However, as the number of negative pairs continues to increase to a very large number, the performance of the model begins to decline. This observation further confirms that when introducing contrastive regularization to complement consistency regularization, the negative pairs need to be appropriately constructed and integrated with positive pairs to effectively enhance the model performance.

D. Ablation Study

In this section, we design the ablation study to further investigate how different components contribute to the performance of our model. Our model proceeds with pseudo-labeling using data augmentation strategies (involving consistency regularization and contrastive regularization) and graph active learning for pseudo-label selection. We add these components respectively and formulate seven GNN models: (1) GCN: use the traditional GCN directly to perform few-shot node classification; (2) GCN+DropFeature: utilize the weak augmentation strategy to get augmented features; (3) GCN+DropEdges: drop edges to perturb adjacency matrix; (4) GCN+DropNodes: drop nodes to augment node features; (5) GCN+DropAll: apply all three augmentation strategies to perform semi-supervised learning with consistency regularization; (6) GCN+DropAll+GAL: leverage graph active learning to select nodes with pseudo-labels; (7) GCN+DropAll+Contrast+GAL: the complete design of our model. The results for ablation study are shown in Table III.

The experimental results suggest that the data augmentation strategies have different effects on the performance of the model. Specifically, strong augmentation strategies tend to have a greater impact on performance than weak augmentation strategies, while both types of augmentation can lead to some improvement. Combining these data augmentation strategies with consistency regularization further improves the results. The graph active learning has the greatest contribution to our model, which significantly improves the classification performance by $(5, 8)\%$ of accuracy. Contrastive regularization is able to further advance state-of-the-art performance to a higher level, which implies that this operation yields an

TABLE III
EVALUATION ON MODEL COMPONENTS ($N = k \times 5$, ACCURACY %)

Model	Cora	Citeseer	PubMed	DBLP
GCN	75.04	65.64	65.77	51.20
GCN + DropFeatures	75.82	65.93	65.75	52.12
GCN + DropEdges	76.81	65.62	69.54	55.91
GCN + DropNodes	77.13	67.97	71.42	56.33
GCN + DropAll	79.05	68.10	74.59	60.61
GCN + DropAll + GAL	84.40	72.13	82.85	68.66
GCN + GropAll + Contrast + GAL	**84.90**	**73.96**	**83.80**	**73.42**

additional advantage for pseudo-labeling and graph-based few-shot learning. These observations reaffirm the effectiveness of our design for node classification with only few labeled data.

V. CONCLUSION

In this paper, we extend the task of node classification to a more challenging and realistic case where only few labeled data are available. To overcome this challenge, we propose a novel few-shot node classification model, which incorporates various techniques including semi-supervised pseudo-labeling with multi-level data augmentation, consisting of consistency regularization and contrastive regularization. Additionally, we introduce graph active learning to facilitate pseudo-label selection and improve the overall performance of the model. Extensive experiments have been conducted on four citation networks. The results demonstrate that our model achieves state-of-the-art performance, reaffirming its effectiveness in node classification, its superiority over baseline methods, and its practical significance in addressing the challenges of few-shot node classification.

ACKNOWLEDGMENTS

L. Chen's work is partially supported by the NSF under grant CNS-2245968.

REFERENCES

[1] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[2] Q. Li, L. Chen, Y. Cai, and D. Wu, "Hierarchical graph neural network for patient treatment preference prediction with external knowledge," in *PAKDD*, 2023, pp. 204–215.

[3] B. Ashmore and L. Chen, "Hover: Homophilic oversampling via edge removal for class-imbalanced bot detection on graphs," in *CIKM*, 2023.

[4] M. Réau, N. Renaud, L. C. Xue, and A. M. Bonvin, "Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces," *Bioinformatics*, vol. 39, no. 1, p. btac759, 2023.

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

[6] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, 2017.

[7] S. Liu, L. Chen, H. Dong, Z. Wang, D. Wu, and Z. Huang, "Higher-order weighted graph convolutional networks," *arXiv preprint arXiv:1911.04129*, 2019.

[8] J. Wei, C. Huang, S. Vosoughi, Y. Cheng, and S. Xu, "Few-shot text classification with triplet networks, data augmentation, and curriculum learning," *arXiv preprint arXiv:2103.07552*, 2021.

[9] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun, "Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," *arXiv preprint arXiv:1810.10147*, 2018.

[10] Q. Li, X. Li, L. Chen, and D. Wu, "Distilling knowledge on text graph for social media attribute inference," in *SIGIR*, 2022, pp. 2024–2028.

[11] Q. Li, L. Chen, S. Jing, and D. Wu, "Knowledge distillation on cross-modal adversarial reprogramming for data-limited attribute inference," in *Companion Proceedings of the ACM Web Conference*, 2023, pp. 65–68.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[13] R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim, "Multimodal model-agnostic meta-learning via task-aware modulation," *NeurIPS*, vol. 32, 2019.

[14] H. Yao, Y. Wei, J. Huang, and Z. Li, "Hierarchically structured meta-learning," in *ICML*, 2019, pp. 7045–7054.

[15] K. Ding, Q. Zhou, H. Tong, and H. Liu, "Few-shot network anomaly detection via cross-network meta-learning," in *Proceedings of the Web Conference*, 2021, pp. 2448–2456.

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

[17] K. Huang and M. Zitnik, "Graph meta learning via local subgraphs," *NeurIPS*, vol. 33, pp. 5862–5874, 2020.

[18] A. Islam, C.-F. R. Chen, R. Panda, L. Karlinsky, R. Feris, and R. J. Radke, "Dynamic distillation network for cross-domain few-shot recognition with unlabeled data," *NeurIPS*, vol. 34, pp. 3584–3595, 2021.

[19] Z. Wu, P. Zhou, G. Wen, Y. Wan, J. Ma, D. Cheng, and X. Zhu, "Information augmentation for few-shot node classification," in *IJCAI*, 2022, pp. 3601–3607.

[20] C. Zhang, K. Ding, J. Li, X. Zhang, Y. Ye, N. V. Chawla, and H. Liu, "Few-shot learning on graphs: A survey," *arXiv preprint arXiv:2203.09308*, 2022.

[21] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *International Conference on Learning Representations*, 2018.

[22] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.

[23] L. Zhao and L. Akoglu, "Pairnorm: Tackling oversmoothing in gnns," *International Conference on Learning Representations (ICLR)*, 2020.

[24] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, and J. Tang, "Graph random neural networks for semi-supervised learning on graphs," *Advances in neural information processing systems*, vol. 33, pp. 22 092–22 103, 2020.

[25] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *NeurIPS*, vol. 33, 2020.

[26] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, and et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *NeurIPS*, vol. 33, pp. 596–608, 2020.

[27] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, "Selfreg: Self-supervised contrastive regularization for domain generalization," in *ICCV*, 2021, pp. 9619–9628.

[28] D. Lee, S. Kim, I. Kim, Y. Cheon, M. Cho, and W.-S. Han, "Contrastive regularization for semi-supervised learning," in *CVPR*, 2022.

[29] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[30] L. Chen, X. Li, and D. Wu, "Enhancing robustness of graph convolutional networks via dropping graph connections," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, 2021, pp. 412–428.

[31] Y. Fan, A. Kukleva, and B. Schiele, "Revisiting consistency regularization for semi-supervised learning," in *DAGM German Conference on Pattern Recognition*. Springer, 2021, pp. 63–78.

[32] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[33] W. Zhang, Y. Wang, Z. You, M. Cao, P. Huang, J. Shan, Z. Yang, and B. Cui, "Information gain propagation: A new way to graph active learning with soft labels," in *ICLR*, 2022.

[34] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *SIGKDD*, 2008, pp. 990–998.

[35] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[36] K. Ding, J. Wang, J. Li, K. Shu, C. Liu, and H. Liu, "Graph prototypical networks for few-shot learning on attributed networks," in *CIKM*, 2020, pp. 295–304.