Supervised Graph Contrastive Learning for Few-shot Node Classification

Zhen Tan ✉[1], Kaize Ding[1], Ruocheng Guo[2], and Huan Liu[1]

[1] Arizona State University, Tempe AZ, USA
[2] Bytedance AI Lab, London, UK
{ztan36,kding9,huanliu}@asu.edu, rguo.asu@gmail.com

Abstract. Graphs present in many real-world applications, such as financial fraud detection, commercial recommendation, and social network analysis. But given the high cost of graph annotation or labeling, we face a severe graph label-scarcity problem, i.e., a graph might have a few labeled nodes. One example of such a problem is the so-called *few-shot node classification*. A predominant approach to this problem resorts to *episodic meta-learning*. In this work, we challenge the status quo by asking a fundamental question whether meta-learning is a must for few-shot node classification tasks. We propose a new and simple framework under the standard few-shot node classification setting as an alternative to meta-learning to learn an effective graph encoder. The framework consists of supervised graph contrastive learning with novel mechanisms for data augmentation, subgraph encoding, and multi-scale contrast on graphs. Extensive experiments on three benchmark datasets (CoraFull, Reddit, Ogbn) show that the new framework significantly outperforms state-of-the-art meta-learning based methods.

Keywords: Few-shot Learning · Graph Neural Networks · Graph Contrastive Learning

1 Introduction

Graphs are ubiquitous in many real-world applications. Graph Neural Networks (GNNs) [20,30,35] have been applied to model a myriad of network-based systems in various domains, such as social networks [13], citation networks [18], and knowledge graphs [24]. Despite these breakthroughs, it has been noticed that conventional GNNs fail to make accurate predictions when the labels are scarcely available [7,40,6]. One such problem is so-called *few-shot node classification*. It consists of two disjoint phases: In the first phase (train), classes with substantial labeled nodes (i.e., *base classes*) are available to learn a GNN model; and in the second phase (test), the GNN classifies nodes of unseen or *novel classes* with few labeled nodes. A few-shot node classification task is called *N-way K-shot node classification* if a node is classified into N classes and each class contains only a few K labeled nodes in the test phase.

This shortage of labeled training data for the novel classes poses a great challenge to learning effective GNNs. A prevailing paradigm to tackle this problem is *episodic meta-learning* [41,7,33,12,23]; its representative algorithms are Matching Network [31], MAML [10], and Prototypical Network [25]. Episodic meta-learning is inspired by how humans learn unseen classes with few samples via utilizing previously learned prior knowledge. During the training phase, it generates numerous meta-train tasks (or episodes) by emulating the test tasks, following the same N-way K-shot node classification structure. An example of episodic meta-learning is shown in Figure 1. In each episode, K labeled nodes are randomly sampled from N base classes, forming a *support set*, to train the GNN model while emulating the N-way K-shot node classification in the test phase. The GNN then predicts labels for an emulated *query set* of nodes randomly sampled from the same classes as the support set. A Cross-Entropy Loss is used for backpropagation to update the GNN. Current research [41,7,33,12,9] has shown that via numerous such episodic emulations across different sampled meta-tasks on bases classes, the trained encoder can extract transferable meta-knowledge to fast adapt to unseen novel classes.

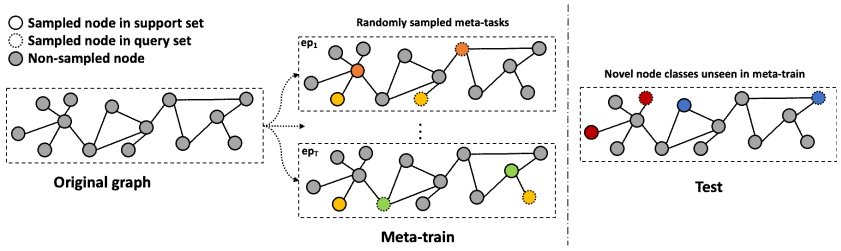

Fig. 1: Episodic meta-learning for few-shot node classification (ep_i is the ith episode). Colors indicate different classes. Specially, grey nodes mean those nodes are not sampled. Different types of nodes indicate if nodes are from a support set or a query set.

These episodic meta-learning based methods entail the following steps: (1) random sampling in each episode for meta-train in order to acquire the topological knowledge crucial for learning representative node embeddings. Since only a small portion of the nodes and classes are randomly selected per episode, the topological knowledge learned with those nodes is piecemeal and insufficient to train expressive GNN encoder, especially if the selected nodes share little correlation. (2) To boost accuracy, those meta-learning based methods have to rely on a large number of episodes. In other words, a large number of samples from the original graph is required for meta-train to capture the topological knowledge that can be transferred for use in the test phase. Consequently, these meta-learning algorithms can take time to converge with their emulation-based meta-train. The two problems are closely related. The *piecemeal knowledge* from emulation-based meta-learning unique for few-shot node classification entails the need for *large numbers of episodes* to acquire better topological knowledge. In this paper, we investigate if an alternative approach to episodic meta-learning can be developed so that the two problems can be addressed from their root

causes for better performance of few-shot node classification. We posit that the key to few-shot node classification is to learn a generalizable GNN encoder that can produce discriminative representation on novel classes by learning transferable topological patterns implied in bases classes. If we can address the piecemeal knowledge problem, we can better use the few labeled nodes from novel classes to fine-tune another simple classifier (e.g., Logistic Regression, SVM, shallow MLP, etc.) to predict labels for other unlabeled nodes.

As an alternative to episodic meta-learning, we propose a novel approach for few-shot node classification, supervised graph contrastive learning [19,11]. Graph contrastive learning is proven effective in training powerful GNNs [38,14,42]. Multiple views of the original graph are first created through predefined transformations [38,14,42,8] (e.g., randomly dropping edges, randomly perturbing node attributes). Then a contrastive loss is applied to maximize feature consistency under differently augmented views. However, none of those existing methods accommodate the unique characteristics of few-shot node classification. In this paper, we propose a novel graph contrastive learning method especially designed for few-shot node classification. We will present technical details on how the new supervised contrastive learning can avoid the two problems with the episodic meta-learning after a formal problem statement is given.

Contributions. Our contributions include: (1) we are the first to investigate an alternative to the prevailing meta-learning paradigm for graph few-shot learning; (2) we propose a supervised graph contrastive learning method tailored for few-shot node classification by developing novel mechanisms for data augmentation, subgraph encoding, and multi-scale contrast on a graph; and (3) we conduct systematic experiments to assess the proposed framework in comparison with the existing meta-learning based graph few-shot learning methods and representative graph contrastive learning methods in terms of accuracy and efficiency.

2 Problem Formulation

In this paper, we focus on few-shot node classification on a single graph. Formally, given an attributed network $G = (\mathcal{V}, \mathcal{E}, \mathbf{X}) = (\mathbf{A}, \mathbf{X})$, where $\mathcal{V}$, $\mathcal{E}$, $\mathbf{A}$ and $\mathbf{X}$ denote the nodes, edges, adjacency matrix and node attributes, respectively. The few-shot node classification problem assumes the existence of a series of homogeneous node classification tasks $\mathcal{T} = \{\mathcal{D}^i\}_{i=1}^I = \{(\mathbf{A}_{\mathcal{C}^i}, \mathbf{X}_{\mathcal{C}^i})\}_{i=1}^I$, where $\mathcal{D}^i$ denotes the given dataset of a task, I denotes the number of such tasks, $\mathbf{X}_{\mathcal{C}^i}$ denotes the attributes of nodes whose labels belong to the label space $\mathcal{C}^i$, and $\mathbf{A}_{\mathcal{C}^i}$ similarly. Following the literature [41,7,33,12,23], we call the classes available during training as base classes, $\mathcal{C}_{base}$, and the classes for target test phase as novel classes, $\mathcal{C}_{novel}$, $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \varnothing$. Conventionally, there are substantial gold-labeled nodes for $\mathcal{C}_{base}$, but few labeled nodes for novel classes $\mathcal{C}_{novel}$. We can formally define the problem of few-shot node classification as follows:

Definition 1. ***Few-shot Node Classification:*** *Given an attributed graph* $G = (\boldsymbol{A}, \boldsymbol{X})$ *with a divided node label space* $\mathcal{C} = \{\mathcal{C}_{base}, \mathcal{C}_{novel}\}$*, we have substantial*

labeled nodes from $\mathcal{C}_{base}$, and few-shot labeled nodes (support set $\mathcal{S}$) for $\mathcal{C}_{novel}$. The task is to predict the labels for unlabeled nodes (query set $\mathcal{Q}$) from $\mathcal{C}_{novel}$.

3 Methodology

In graph representation learning, usually a GNN encoder g_θ is employed to model the high dimensional graph knowledge and project the node attributes to a low dimensional latent space. The classifier f_ψ is then applied to the latent node representations for node classification. The essence of few-shot node classification is to learn a encoder g_θ that can transfer the topological and semantic knowledge learned from substantial data of base classes to generate discriminative embeddings for nodes from novel classes with limited supervisory information.

As a prevailing paradigm, meta-learning is adopted [41,7,33] to jointly learn g_θ and f_ψ by episodically optimizing the Cross-Entropy Loss (CEL) on sampled meta-tasks. However, optimizing CEL over sampled piecemeal graph structures will engender node embeddings excessively discriminative against the current sampled nodes, rendering them sub-optimal for nodes classification in the test phase where the nodes are sampled arbitrarily from unseen novel classes. To mitigate such issues, those meta-learning based methods rely on a large number of episodes to train the model on numerous differently sampled meta-tasks to learn more transferable node embeddings, which makes the training process highly unscalable, especially for large graphs.

As a remedy, in this paper, we propose a decoupled method to learn the graph encoder g_θ and the final node classifier f_ψ separately. We put forward a supervised graph contrastive learning to firstly pretrain the GNN encoder g_θ to generate more transferable node embeddings. With such high-quality node embeddings, we can fine-tune a simple linear classifier to perform the final few-shot node classification. As shown in Figure 2, our framework consists of the following key components:

- An augmentation function $T(\cdot)$ that transforms a sampled centric node into a correlated view. We propose a node connectivity based augmentation mechanism to sample nodes that are highly correlated to the centric nodes to form a subgraph as its augmented view (Section 3.1).
- A GNN encoder g_θ that encodes the subgraphs rather than the whole graph like all the existing meta-learning methods do. In such a manner, our model will consume much less time to converge (Section 3.2).
- A contrastive mechanism that enables the encoder g_θ to discriminates embeddings between differently augmented views by capturing structural patterns across base classes and extrapolating such knowledge onto unseen novel classes (Section 3.3).
- A linear classifier f_ψ fine-tuned by a few labeled nodes from novel classes and is tasked to predict labels for those unlabeled nodes (Section 3.4).

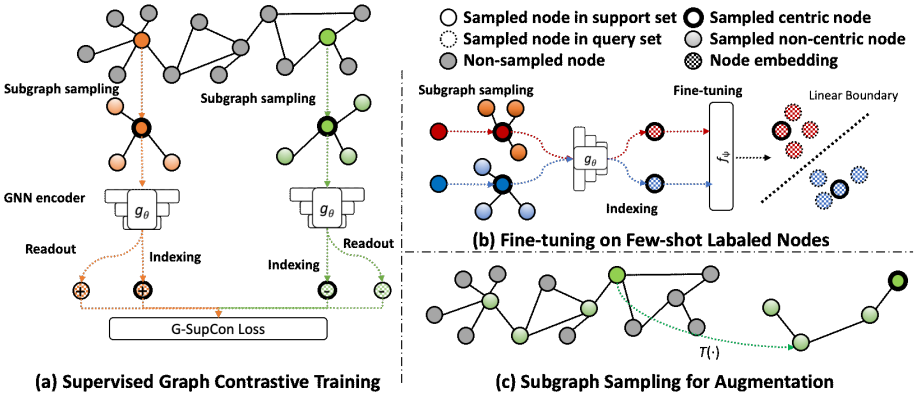

Fig. 2: Colors indicate different classes. Specially, grey nodes mean those nodes are not sampled. Different types of nodes indicate if nodes are from a support set or a query set, or sampled as centric nodes. Ombré nodes indicate the nodes are sampled non-centric nodes for the subgraphs. Crisscross nodes are node embeddings from the GNN encoder. (a) Supervised graph contrastive training framework. (b) Fine-tuning on few-shot labeled nodes from novel classes. (c) Node connectivity based subgraph sampling strategy samples nodes that are strongly connected to the centric nodes.

3.1 Data Augmentation

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}) = (\mathbf{A}, \mathbf{X})$, following the conventions in contrastive learning methods [4,15], a transformation $T(\cdot)$ is used to generate a new view $\mathbf{x}'_j$ for a node j ($\forall j \in \{1, ..., M\}$, M is the number of nodes) with node attributes $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$:

$$\mathbf{x}'_j = T(\mathbf{A}, \mathbf{X}, j) \tag{1}$$

There are multiple transformations for graph data, such as node masking or feature perturbation. However, such methods can introduce extra noise that impairs the learned node representations. In this paper, we propose a node connectivity based subgraph sampling strategy as the data augmentation mechanism. Connectivity score is a family of metrics that measures the connection strength between a pair of nodes in a graph without using the node attributes [3,18]. Notably, since only the adjacency matrix is needed for calculation, such augmentation can be pre-computed before training:

$$\mathbf{S} = Connect(\mathcal{V}, \mathcal{E}) = Connect(\mathbf{A}), \tag{2}$$

where each column $\mathbf{s}_j$ of $\mathbf{S}$ contains the scores between node j and all nodes in the graph. We set the score between a node and itself as a constant value $\gamma = 0.3$ for better performance. Intuitively, nodes that share more semantic similarities tend to have more correlations. However, such nodes may not be geologically close to each other. Node connectivity can capture the correlation between nodes

by considering both the global and local graph structures [3,17]. As shown in Figure 2 (c), for any given node, we treat it as the centric node and sample other nodes that have the highest connectivity scores with the centric node to build a contextualized subgraph as its augmented view. Now the transformation for data augmentation can be defined as:

$$\begin{aligned} \mathbf{x}'_j &= T(\mathbf{A}, \mathbf{X}, j) \\ &= \mathbf{X}[top\_rank(\mathbf{S}[j,:], \alpha)] \\ &= \mathbf{X}[top\_rank(Connect(\mathbf{A})[j,:], \alpha)] \\ &= (\mathbf{A}'_j, \mathbf{X}'_j) = \mathcal{G}_s(j) \end{aligned} \tag{3}$$

where *top_rank* is a function that returns the indices of the top α values, α is a hyperparamter that controls the augmented subgraph size $(\alpha + 1)$, and $\mathcal{G}_s(j)$ is the augmented subgraph for node j with adjcency matrix $\mathbf{A}'_j$ and sampled node attributes $\mathbf{X}'_j$. In this paper, we present two methods to calculate the node connectivity scores: Node Algebraic Distance (NAD) [3] and Personalized PageRank (PPR) [17].

Node Algebraic Distance (NAD). Following [3], we first randomly assign a random value u_i $(0 < u_i < 1)$ to any node i in the graph to form a vector $\mathbf{u} \in \mathbb{R}^M$. Then, we iteratively updates the value of a node by aggregating its neighboring weighted values: At the t-th iteration, for node i we have:

$$\hat{u}_i^t = \sum_j \mathbf{A}(i,j) u_j^{t-1} / \sum_j \mathbf{A}(i,j), \tag{4}$$

$$\mathbf{u}^t = (1-\eta)\mathbf{u}^{t-1} + \eta \hat{\mathbf{u}}^t, \tag{5}$$

where η is a parameter set to 0.5. After a few iterations, the difference between the values of node i and j indicates the coupling between them. The smaller difference stands for a stronger connection. The final score matrix is:

$$\mathbf{S} = \{\mathbf{S}(i,j)\}_{i,j=1}^M = \frac{1}{|u_i - u_j| + \epsilon} \tag{6}$$

where ϵ is a parameter set to 0.01. We column-normalize $\mathbf{S}$ and then set the value of $\mathbf{S}(i,i), (\forall i \in M)$ to γ.

Personalized PageRank (PPR). For PPR we have:

$$\mathbf{S} = \phi \cdot (\mathbf{I} - (1-\phi) \cdot \mathbf{A}\mathbf{D}^{-1}) \tag{7}$$

where ϕ is a parameter usually set as 0.15, and $\mathbf{D}$ is a diagonal matrix with: $\mathbf{D}(i,i) = \sum_j \mathbf{A}(i,j)$. We column-normalize the scores $\mathbf{S}$ to make the scores on the diagonal equal to $\gamma = 0.3$ as in NAD.

3.2 Subgraph Encoder

With the pre-computed connectivity score matrix, for each node j, we sample the top α nodes with the highest connectivity scores to construct a subgraph $\mathcal{G}_s(j) = (\mathbf{A}'_j, \mathbf{X}'_j)$. The larger α is, the richer context is given in the subgraph (We set $\alpha = 19$ for memory limitation). Then, as shown in Figure 2 (a), we feed the resulting subgraph into a GNN encoder g_θ. In particular, as the size of the subgraphs is a fixed small number (20 in our case) as opposed to the original graph, compared to the existing methods where the encoder encodes the whole graph, our method reduces the dimension for neighborhood aggregation to a much smaller magnitude and makes it highly scalable and faster to converge:

$$\mathbf{Z}'_j = g_\theta(\mathbf{x}'_j) = g_\theta(\mathbf{A}'_j, \mathbf{X}'_j), \tag{8}$$

where $\mathbf{Z}'_j \in \mathbb{R}^{(\alpha+1)\times F}$ and F is the embedding size. We normalize the latent features for better performance. To enable the contrastive learning across different scales (e.g. nodes and subgraphs), we propose a readout function $R(\cdot)$ which maps the representation of all nodes in the subgraph, $\mathbf{Z}'_j$, to a vector $\mathbf{z}'_j$ as the summarized subgraph representation. Intuitively, $\mathbf{z}'_j$ is a weighted average of the representations of all nodes in the subgraph. The weights are their normalized connectivity scores $\hat{\mathbf{s}}_j \in \mathbb{R}^\alpha$ to node j. We set the connectivity score of a node j to itself as a constant value γ (as defined in Section 3.1) before normalization:

$$\mathbf{z}'_j = R(\mathbf{Z}', \hat{\mathbf{s}}_j) = \sigma(\hat{\mathbf{s}}_j^T \cdot \mathbf{Z}') \tag{9}$$

where σ is the sigmoid function, and $\mathbf{z}'_j \in \mathbb{R}^F$ is the subgraph representation. The centric node embedding $\mathbf{z}_j \in \mathbb{R}^F$ can be directly indexed from $\mathbf{Z}'_j$.

3.3 Multi-scale Graph Contrastive Learning with Augmented Views

Since the augmented subgraphs contain topological information crucial for learning expressive encoder, we propose to use multi-scale contrastive learning that combines three categories of contrastive pairs to enforce the model to learn from both individual attributes and contextualized knowledge from sampled views: the contrast between (1) nodes and nodes ($\mathbf{z}_i \,\&\, \mathbf{z}_j$), (2) subgraphs and subgraphs ($\mathbf{z}'_i \,\&\, \mathbf{z}'_j$), and (3) nodes and subgraphs ($\mathbf{z}_i \,\&\, \mathbf{z}'_j$), where i, j are arbitrary node indices. For any node, the nodes in the same class together with their corresponding subgraphs are viewed as positives, and all the rest are viewed as negatives. Now, given the batch size B, we define a *duo-viewed* batch, consisting of the representations of both nodes and their corresponding augmented subgraphs:

$$\{(\mathbf{h}_b, y_b)\}_b^{2B} = \{(\mathbf{z}'_b, y_b), (\mathbf{z}_b, y_b)\}_b^B. \tag{10}$$

Loss Function To better suit the setting of few-shot node classification, we adapt the Supervised Contrastive loss (SupCon) from [19] to pretrain the GNN encoders. We term the proposed loss G-SupCon for convenience. Compared to

unsupervised contrastive loss (e.g. Deep InfoMax (DIM) [1], SimCLR [4], Margin Loss [18] etc.), and Cross-Entropy Loss (CEL), G-SupCon utilizes the ground-truth label to sample mini-batches to help to better align the representation of nodes in the same class more closely and push nodes from different classes further apart. Such learning patterns can be easier to transfer to unseen novel classes to generate highly discriminative representations. To ensure the balance in training data, we sample $B/|C_{tr}|$ nodes per class as centric nodes in each mini-batch for training, where $|C_{tr}|$ is the number of classes for pretraining (i.e., base classes). We term it Balanced Sampling (BS) for convenience. Then, the loss function is defined as:

$$\mathcal{L} = \sum_{b \in B} \frac{-1}{|P(b)|} \sum_{p \in P(b)} \log \frac{\exp\left(\mathbf{h}_b \cdot \mathbf{h}_p / \tau\right)}{\sum_{a \in A(b)} \exp\left(\mathbf{h}_b \cdot \mathbf{h}_a / \tau\right)}, \tag{11}$$

where $A(b)$ is the set of indices from 1 to $2B$ excluding b, and $P(b)$ is the set of indices of all positives in a duo-viewed batch excluding b. This contrastive loss includes all the three categories of contrastive pairs mentioned before. Additionally, $\tau \in \mathbb{R}^+$ is a scalar temperature parameter defined as $\tau = \beta/\sqrt{degree(\mathcal{G})}$, where $degree(\mathcal{G})$ is the average degree of the graph, and β is a hyperparameter. The temperature parameter controls the sensitivity of the trained model to the hard negative samples. As nodes in different classes also share some correlation with the centric nodes, especially for more complex graphs with higher average degrees, we do not want them to be fully separated apart in the representation space. So, we add the degree as a penalty for the temperature parameter to guide the separateness.

Under this fully-supervised setting, we pretrain our GNN encoder with all the base node labels and fix it during fine-tuning.

3.4 Linear Classifier Fine-tuning

As indicated in Figure 2 (b), with a pretrained GNN encoder, when fine-tuning on a target few-shot node classification dataset $\mathcal{D}^i$ on novel classes, we tune a separate linear classifier f_ψ, (e.g. logistic regression, SVM, a linear layer, etc.) with the few labeled nodes in the support set $\mathcal{S}^i$, and task it to predict the labels for nodes in the query set $\mathcal{Q}^i$. The representations of nodes in $\mathcal{D}^i$ are obtained through the same procedure: treat each node as centric node to retrieve a contextualized subgraph, and then feed the sampled subgraphs to the pretrained GNN encoder, and the centric node embedding can be directly indexed and used to fine-tune the classifier f_ψ by optimizing a naive Cross-Entropy Loss.

4 Experiments

In this section, we design experiments to evaluate the proposed framework by comparing with three categories of methods for few-shot node classification: (1) naive supervised pretraining using GNNs, (2) state-of-the-art meta-learning

based methods, and (3) by going one step further to compare with contrastive learning methods. For the third category, to our best knowledge, we are the first to implement state-of-the-art self-supervised graph contrastive methods to few-shot node classification and compare them with the proposed graph supervised contrastive learning.

4.1 Experimental Settings

Table 1: Statistics of the commonly used datasets

	# Nodes	# Edges	# Features	# Labels	Base	Novel
CoraFull	19,793	126,842	8,710	70	42	28
Reddit	232,965	11,606,919	602	41	24	17
Ogbn-arxiv	169,343	1,166,243	128	40	24	16

Evaluation Datasets We conduct our experiments on three widely used graph few-shot learning benchmark datasets where a sufficient number of node classes are available for sampling few-shot node classification tasks: CoraFull [2], Reddit [13], and Ogbn-arxiv [35]. Their statistics are given in Table 1.

Baseline Methods In this work, we compare our framework with the following 3 categories of methods.

- Naive supervised pretraining. We use GCN [20] as a naive encoder and pretrain it with all nodes from the base classes, and following convention, we fine-tune a single linear layer as the classifier for each few-shot node classification task. Also, we implement an initialization strategy, TFT [5] for the classifier by setting its weight as a matrix consisting of concatenated prototype vectors of novel classes.
- State-of-the-art meta-learning methods for few-shot node classification on a single graph: MAML based Meta-GNN [41], Matching Network [31] based AMM-GNN [33], and Prototypical Network based GPN [7]. We do not include methods like [36,34] in the baselines because they require extra auxiliary graph data, nor methods like [21,23] because they have similar performance with the chosen baselines according to their original papers.
- State-of-the-art self-supervised pretraining methods on a single graph: MVGRL [14], SUBG-Con [18], GraphCL [38], and GCA [42]. These methods pretrain a GNN encoder with nodes from base classes without using the labels. The classifiers are then fine-tuned on novel support nodes and their accuracy is reported on predicting labels for novel query nodes.

Evaluation Protocol: To make fair comparison, all the scores reported are accuracy values averaged over 10 random seeds and all the baselines share the same splits of base classes and novel classes as shown in Table 1.

Implementation Details: In Table 1, the specific data splits are listed for each dataset. For a fair comparison, we adopt the same encoder for all compared methods. Specifically, the graph encoder g_θ consists of one GCN layer [20] with

PReLU activation. The effect of the encoder architecture are further explored in Section 4.4. We choose logistic regression as the linear classifier for fine-tuning. The encoder is trained with Adam optimizers whose learning rates are set to be 0.001 initially with a weight decay of 0.0005. And the coefficients for computing running averages of gradient and square are set to be $\beta_1 = 0.9$, $\beta_2 = 0.999$. The default values of batch size B and graph temperature parameter β are set to 500 and 1.0, respectively. For the baseline methods, we use the default parameters provided in their implementations.

4.2 Overall Evaluation

We present the comparative results between our framework and three categories of baseline methods described earlier. It is worth mentioning that, this work is the first to investigate the necessity of episodic meta-learning for Few-shot Node Classification (FNC) problems. For a fair comparison, all methods share the same GCN encoder architecture as the proposed framework. Also, when experimented on each dataset, they share the same random seeds for data split, leading to identical evaluation data. The results are shown in Table 2. We summarize our findings next.

Table 2: Comparative Results: three datasets under different N-way K-shot settings

Methods	**Loss**	**CoraFull (%)**			**Reddit (%)**			**Ogbn-arxiv (%)**		
		10-way 5-shot	5-way 5-shot	3-way 1-shot	10-way 5-shot	5-way 5-shot	3-way 1-shot	10-way 5-shot	5-way 5-shot	3-way 1-shot
GCN	CEL	37.25	45.68	43.23	36.28	44.34	39.62	30.83	38.40	35.41
TFT	CEL	63.50	70.18	66.41	59.75	69.80	58.32	47.68	62.25	60.48
Meta-GNN	CEL	55.23	66.25	60.25	48.62	65.50	55.78	41.20	58.67	55.68
AMM-GNN	CEL	60.80	70.52	65.27	53.28	67.20	55.84	44.33	61.02	58.64
GPN	CEL	62.02	73.40	67.07	59.20	69.31	60.20	50.58	64.12	62.20
GraphCL	SimCLR	79.68	87.35	84.80	82.51	87.34	84.16	57.30	67.34	62.27
MVGRL	DIM	81.34	88.40	85.03	84.78	89.65	86.29	57.82	68.24	63.36
SUBG-Con	Margin Loss	80.71	88.02	86.26	83.70	89.55	85.57	56.29	66.36	62.10
GCA	SimCLR	82.43	90.52	87.89	85.61	90.96	87.26	59.03	69.53	65.49
Ours (NAD)	**G-SupCon**	**86.13**	**93.65**	89.42	88.52	93.36	91.70	**62.24**	73.25	71.63
Ours (PPR)	**G-SupCon**	85.34	93.62	**90.56**	**89.10**	**94.12**	**92.31**	61.85	**73.65**	**72.90**

Necessity of employing episodic meta-learning style methods for graph few-shot learning. First and foremost, we find that almost all the contrastive pretraining based methods outperform the existing meta-learning based FNC algorithms. Even the most straightforward one, TFT, which only leverages a simple initialization to the separate classifier, can produce comparable scores to the best meta-learning based FNC method, GPN. We have shown that through appropriate pretraining, including self-supervised and supervised training, adding a simple linear classifier can outperform the existing meta-learning based framework by a significant margin.

Effectiveness of our framework to learn discriminative node embeddings for FNC problems. Compared with other existing meta-learning based methods, our framework outperforms them by a large margin under different settings. We attribute this to the following facets: (1) The effectiveness of the proposed node-connectivity-based sampling strategy that can provide highly correlated context-specific information for the centric node by considering both global and local information. Besides, NAD and PPR can provide similar outcomes. NAD

can perform better on graphs with a higher average degree; and (2) The supervised contrastive learning loss function G-SupCon can utilize label information to further enforce the GNN encoder to generate more discriminative representations by minimizing the distances among nodes from the same classes while segregating nodes from different classes in the representation space.

Robustness to various N-way K-shot settings. Similar to the meta-learning based FNC methods, the performance of contrastive pretraining based methods also degrades when the K decreases or N increases. However, from the results shown in Table 2, we find that those methods are more robust to settings with decreasing K and increasing N. This means that the encoder can better extrapolate to novel classes by generating more discriminative node representations. To a large extent, the degradation lies in the less accurate classifier due to fewer training nodes. Learning to better measure the classifier under scenarios with extremely scarce support nodes (very small K) is also worth further research.

4.3 Further Experiments

To further evaluate our framework, we conduct more experiments next. The default setting for the following experiments is 5-way 5-shot, where we set $\beta = 1.0$, $B = 500$, and use a single GCN layer as the encoder.

Efficiency Study. As discussed in Section 3.2, our method is scalable and has much less convergence time because we construct a GNN encoder for the sampled subgraph rather than the whole graph. Also, Section 4.2 shows that performance is not sacrificed for achieving scalability. On the contrary, due to the effective sampling strategy, only fine-grained data are fed into the encoder, resulting in a considerable boost in accuracy. We show the actual time consumption for a single run of training of our model and typical meta-learning and pretraining baselines. The experiment is conducted on a single RTX 3090 GPU. In Table 3, we show that our method can achieve excellent performance and consume much less training time. Note that here we only consider the training time, excluding the time for computing the node connectivity scores $\mathbf{S}$ which can be pre-calculated.

Table 3: Training time of GPN, GCA, and Ours (proposed method)

Methods	**CoraFull**	**Reddit**	**Ogbn-arxiv**
GPN	1352s	3248s	2562s
GCA	584s	3651s	3237s
Ours	**10s**	**21s**	**54s**

Representation Clustering. This experiment is designed to demonstrate the high-quality representation generated by the encoder in the proposed framework. We show the clustering results of the representation of the query nodes in 5 randomly sampled novel classes in Table 4 and visualize the embedding in Figure 3 (without fine-tuning on support set). It can be observed that an encoder trained by the proposed pretraining strategy possesses a stronger extrapolation ability to generate highly discriminative boundaries for unseen novel classes.

Table 4: Performance on novel classes query node embedding clustering, reported in Normalized Mutual Information (**NMI**) and Adjusted Rand Index (**ARI**)

Methods	**CoraFull**		**Reddit**		**Ogbn-arxiv**	
	NMI	ARI	NMI	ARI	NMI	ARI
GPN	0.5134	0.4327	0.3690	0.3115	0.2905	0.2235
GCA	0.7531	0.7351	0.7824	0.7756	0.3786	0.3219
Ours	**0.8567**	**0.8229**	**0.8890**	**0.8720**	**0.5280**	**0.4485**

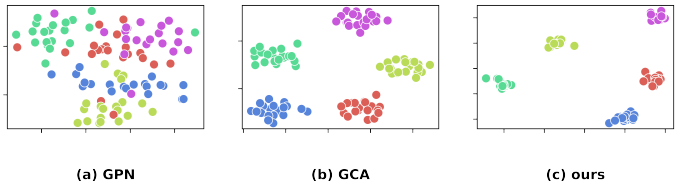

Fig. 3: t-SNE embedding visualization on CoraFull data: (a) GPN (b) GCA (C) Ours

4.4 Ablation & Parameter Analysis

In this section we study how sensitive the proposed framework is to the design choice of its components. In particular, we consider the architecture of encoder g_θ, graph temperature parameter β, and batch size B.

Analysis on Encoder Architecture. Our framework is independent of the encoder architecture. We evaluate our framework with three widely-used GNNs: GCN [20], GAT [30], and GIN [35], as shown in Table 5. The difference between encoder architectures is insignificant, so we choose GCN as the default encoder.

Analysis on Loss Function. To better demonstrate the effectiveness of our G-SubCon loss, we design an experiment to compare different loss functions. The SimCLR loss does not consider the label information, so the Balance Sampling (BS) strategy we proposed in Section 3.3 is not feasible for it. But in order to explicitly show the influence from the loss function, we also show the result of SimCLR with the same sampled data splits from BS as our G-SubCon. We list the results from all the candidate loss functions under both settings in Table 6. It can be observable that all the losses suffer from the class imbalance issue, and the simple BS scheme can improve performance. Also, our proposed G-SupCon outperforms all others even with identical data splits. In short, both the BS scheme and the G-SupCon loss function are effective in terms of accuracy on few-shot node classification tasks.

Table 5: Results of our model with different encoders

Encoder	**CoraFull (%)**	**Reddit (%)**	**Ogbn-arxiv (%)**
GCN	**93.62**	**94.12**	73.65
GAT	92.05	94.04	73.46
GIN	93.25	93.86	**74.10**

Analysis on Graph Temperature Parameter and Batch Size. As presented in Figure 4 (a), we test the sensitivity of our framework regarding the graph temperature parameter β and batch size B. Observably, our framework is

not that sensitive to these two hyperparameters. The best value of β is 1.0 and we set it as default. Generally speaking, the larger batch size can produce higher scores because more contrast can be made in each batch. We choose 500 as the default batch size for computational efficiency and its decent performance.

Table 6: Results of our model trained with different loss functions

Loss Function	BS	**CoraFull (%)**	**Reddit (%)**	**Ogbn-arxiv (%)**
Cross Entropy	No	70.18	69.80	62.25
Cross Entropy	Yes	73.86	74.08	63.67
SimCLR	No	87.54	89.60	66.34
SimCLR	Yes	89.48	92.88	69.98
G-SupCon	**Yes**	**93.62**	**94.12**	**73.65**

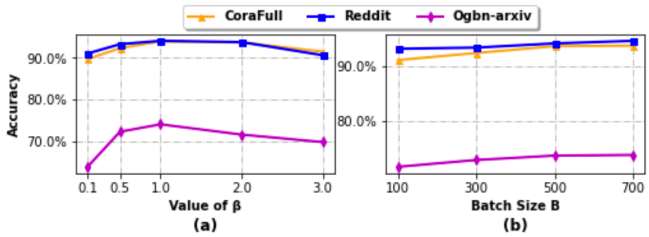

Fig. 4: (a) Accuracy vs Graph temperature parameter β (b) Accuracy vs Batch size B on the three datasets

5 Related Work

Few-shot Node Classification. Graph Neural Networks (GNNs) [20,30,35] are a family of deep neural network models for graph-structured data, which exploits recurrent neighborhood aggregation to preserve the graph structure information and transform the node attributes simultaneously. Recently, increasing attention has been paid to few-shot node classification problems, episodic meta-learning [10] has become the most dominant paradigm. It trains the GNN encoders by explicitly emulating the test environment for few-shot learning [10], where the encoders are expected to gain the adaptability to extrapolate onto new domains. Meta-GNN [41] applies MAML [10] to learn directions for optimization with limited labels. AMM-GNN [33] deploys Matching Network [31] to learn transferable metric among different meta-tasks. GPN [7] adopts Prototypical Networks [25] to make the classification based on the distance between the node feature and the prototypes. MetaTNE [21] and RALE [23] also use episodic meta-learning to enhance the adaptability of the learned GNN encoder and achieve similar results. Furthermore, HAG-Meta [27] extends the problem to incremental learning setting. Recently, people in the image domain argue that the reason for the fast adaptation in the existing works lies in feature reuse rather than those complicated mate-learning algorithms [5,28]. In other words, with a carefully pretrained encoder, decent performance can be obtained through direct fine-tuning a simple classifier on the target domain. Since then, various pretraining strategies [28,22] have been put forward to tackle the few-shot image classification problem. However, no research has been done in the graph domain

with its crucial distinction from images that nodes in a graph are not i.i.d. data. Their interactive relationships are reflected by both the topological and semantic information. Our work here is the first attempt to bridge the gap by developing a novel graph supervised contrastive learning for few-shot node classification.

Graph Contrastive Learning. Contrastive learning has become popular representation learning paradigm in image [4], text [32], and graph [14,42] domains. Starting from the self-supervised setting, contrastive learning methods are proved to learn discriminative representation by contrasting a predefined distance between positive and negative samples. Usually, those samples are augmented through some heuristic transformations from original data. Specifically, in the graph domain, the transformations can be categorized into the following types: (1) graph structure based augmentation, e.g., randomly drop edges or nodes [14,42], randomly sample subgraphs [18]. (2) graph feature based augmentation, e.g., randomly mask or perturb attributes of nodes or edges [29]. [42] further improves those augmentations by adding masks according to feature importance. Some works [37,26,39] try to explore different ways to automatically generate augmented views. Recently, for image [19] and text [11] domains, people notice that by injecting label information to the contrastive loss to compact or enlarge the distances of augmentations of instances within the same or different classes, supervised contrastive loss outperforms the original unsupervised version and even cross-entropy loss in the setting of transfer learning, by providing highly discriminative representation learned from texts or images. However, no existing work has focused on its extrapolation ability for graphs, especially, under an extremer few-shot situation.

6 Conclusion, Limitations and Outlook

In this work, we question the fundamental question whether episodic meta-learning is necessary for few-shot node classification. To answer the question, we propose a graph supervised contrastive learning tailored for the few-shot node classification problem and demonstrate its superb adaptability to extrapolate onto novel classes by fine-tuning a simple linear classifier. Through extensive experiments on benchmark datasets, we demonstrate that our framework can surpass episodic meta-learning methods for few-shot node classification in terms of both accuracy and efficiency. Therefore, this work offers the answer: episodic meta-learning is not a must for few-shot node classification.

Due to limited space, limitations of our work need to be acknowledged. (1) *Limited strategy consideration.* To pretrain the GNN encoder on base classes, we only consider naive supervised training and graph contrastive training. There are many other pretraining strategy that worth further investigation (e.g. [16]). (2) *Lack of theoretical justification.* Our work mainly presents empirical studies which may throw up many questions in need of further theoretical justification, for instance, to what magnitude contrastive pretraining surpasses meta-learning and the reason behind it.

We hope our work will shed new light on few-shot node classification tasks. There are many promising directions worth further research. For example, when

base classes also have very limited labeled nodes or even no label at all, from Table 4.2, we can see that self-supervised pretraining can also help improve the performance. So it would be interesting to investigate the pretraining strategy under semi-supervised or unsupervised settings. In addition, since the pretraining phase may involve extra noise through data sampling or augmentation, methods to calibrate the learned embedding or refine the obtained prototypes are also potential directions for research.

7 Acknowledgments

This work is partially supported by Army Research Office (ARO) W911NF2110030 and Army Research Lab (ARL) W911NF2020124.

References

1. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910 (2019)
2. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815 (2017)
3. Chen, J., Safro, I.: A measure of the connection strengths between graph vertices with applications. arXiv preprint arXiv:0909.4275 (2009)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
5. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. In: ICLR (2019)
6. Ding, K., Wang, J., Caverlee, J., Liu, H.: Meta propagation networks for graph few-shot semi-supervised learning. AAAI (2022)
7. Ding, K., Wang, J., Li, J., Shu, K., Liu, C., Liu, H.: Graph prototypical networks for few-shot learning on attributed networks. In: CIKM (2020)
8. Ding, K., Xu, Z., Tong, H., Liu, H.: Data augmentation for deep graph learning: A survey. arXiv preprint arXiv:2202.08235 (2022)
9. Ding, K., Zhou, Q., Tong, H., Liu, H.: Few-shot network anomaly detection via cross-network meta-learning. In: TheWebConf (2021)
10. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)
11. Gunel, B., Du, J., Conneau, A., Stoyanov, V.: Supervised contrastive learning for pre-trained language model fine-tuning. In: ICLR (2020)
12. Guo, Z., Zhang, C., Yu, W., Herr, J., Wiest, O., Jiang, M., Chawla, N.V.: Few-shot graph learning for molecular property prediction. In: WWW (2021)
13. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS (2017)
14. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: ICML (2020)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
16. Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. In: ICLR (2020)
17. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW (2003)

18. Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., Zhu, Y.: Sub-graph contrast for scalable self-supervised graph representation learning. In: ICDM (2020)
19. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. NeurIPS (2020)
20. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
21. Lan, L., Wang, P., Du, X., Song, K., Tao, J., Guan, X.: Node classification on graphs with few-shot novel labels via meta transformed network embedding. NeurIPS (2020)
22. Liu, C., Fu, Y., Xu, C., Yang, S., Li, J., Wang, C., Zhang, L.: Learning a few-shot embedding model with contrastive learning. In: AAAI (2021)
23. Liu, Z., Fang, Y., Liu, C., Hoi, S.C.: Relative and absolute location embedding for few-shot node classification on graph. In: AAAI (2021)
24. Park, N., Kan, A., Dong, X.L., Zhao, T., Faloutsos, C.: Estimating node importance in knowledge graphs using graph neural networks. In: KDD (2019)
25. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning (2017)
26. Suresh, S., Li, P., Hao, C., Neville, J.: Adversarial graph augmentation to improve graph contrastive learning. NeurIPS (2021)
27. Tan, Z., Ding, K., Guo, R., Liu, H.: Graph few-shot class-incremental learning. In: WSDM (2022)
28. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: ECCV (2020)
29. Tong, Z., Liang, Y., Ding, H., Dai, Y., Li, X., Wang, C.: Directed graph contrastive learning. NeurIPS (2021)
30. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
31. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. NeurIPS (2016)
32. Wang, D., Ding, N., Li, P., Zheng, H.: Cline: Contrastive learning with semantic negative examples for natural language understanding. In: ACL (2021)
33. Wang, N., Luo, M., Ding, K., Zhang, L., Li, J., Zheng, Q.: Graph few-shot learning with attribute matching. In: CIKM (2020)
34. Wen, Z., Fang, Y., Liu, Z.: Meta-inductive node classification across graphs. In: SIGIR (2021)
35. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
36. Yao, H., Zhang, C., Wei, Y., Jiang, M., Wang, S., Huang, J., Chawla, N., Li, Z.: Graph few-shot learning via knowledge transfer. In: AAAI (2020)
37. You, Y., Chen, T., Shen, Y., Wang, Z.: Graph contrastive learning automated. In: ICML (2021)
38. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. NeurIPS (2020)
39. You, Y., Chen, T., Wang, Z., Shen, Y.: Bringing your own view: Graph contrastive learning without prefabricated data augmentations. WSDM (2022)
40. Zhang, C., Ding, K., Li, J., Zhang, X., Ye, Y., Chawla, N.V., Liu, H.: Few-shot learning on graphs: A survey. In: IJCAI (2022)
41. Zhou, F., Cao, C., Zhang, K., Trajcevski, G., Zhong, T., Geng, J.: Meta-gnn: On few-shot node classification in graph meta-learning. In: CIKM (2019)
42. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: WWW (2021)