Towards Few-Shot Self-explaining Graph Neural Networks

Jingyu Peng[1], Qi Liu[1,2], Linan Yue[1], Zaixi Zhang[1], Kai Zhang[1](✉), and Yunhao Sha[1]

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China
{jypeng28,lnyue,zaixi,percy}@mail.ustc.edu.cn,
{qiliuql,kkzhang08}@ustc.edu.cn

[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China

Abstract. Recent advancements in Graph Neural Networks (GNNs) have spurred an upsurge of research dedicated to enhancing the explainability of GNNs, particularly in critical domains such as medicine. A promising approach is the self-explaining method, which outputs explanations along with predictions. However, existing self-explaining models require a large amount of training data, rendering them unavailable in few-shot scenarios. To address this challenge, in this paper, we propose a **M**eta-learned **S**elf-**E**xplaining GNN (MSE-GNN), a novel framework that generates explanations to support predictions in few-shot settings. MSE-GNN adopts a two-stage self-explaining structure, consisting of an *explainer* and a *predictor*. Specifically, the *explainer* first imitates the attention mechanism of humans to select the explanation subgraph, whereby attention is naturally paid to regions containing important characteristics. Subsequently, the *predictor* mimics the decision-making process, which makes predictions based on the generated explanation. Moreover, with a novel meta-training process and a designed mechanism that exploits task information, MSE-GNN can achieve remarkable performance on new few-shot tasks. Extensive experimental results on four datasets demonstrate that MSE-GNN can achieve superior performance on prediction tasks while generating high-quality explanations compared with existing methods. The code is publicly available at https://github.com/jypeng28/MSE-GNN.

Keywords: Explainability · Graph Neural Network · Meta Learning

1 Introduction

Due to the widespread presence of graph data in diverse domains [48,49], Graph Neural Networks (GNNs) [6,14,36] are attracting increasing attention from the research community. Leveraging the message passing paradigm, GNNs have exhibited remarkable efficacy across multiple scenes, including molecule property prediction [35], social network analysis [2,45], and recommender system

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14946, pp. 109–126, 2024.
https://doi.org/10.1007/978-3-031-70365-2_7

[4]. Despite these successes, a significant drawback of GNN models is their lack of explainability, making it unavailable for humans to understand the basis of predictions. This limitation undermines the complete trust in GNN predictions, consequently restricting their application in high-stake scenarios including medical [50] and finance [24] fields. Furthermore, the European Union has explicitly emphasized the necessity of explainability for trustworthy AI in [28] and any studies focusing on explainability have been conducted on interpretability in other fields [41,43]. Therefore, there is an immediate and pressing need for research into the explainability of GNNs.

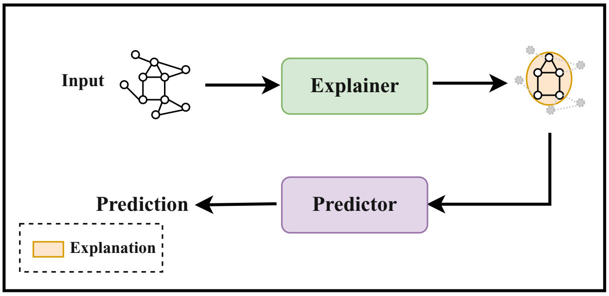


Fig. 1. Paradigm of "*explainer-predictor*" two-stage self-explaining models. The first part is composed of a *explainer* which selects an explanation subgraph for each input graph. The second part is a *predictor* which makes predictions based on the explanation subgraph. Given an input example from Synthetic dataset [39], *explainer* select as explanation, then *predictor* predicts $\hat{y} = house$ based on .

The field of GNN explainability has witnessed substantial scholarly attention [16,17,21,30]. Generally, research on the explainability of GNN can be divided into two main categories: post-hoc explanations and self-explaining methods [40]. Among them, the post-hoc explanation strives to elucidate the predictions made by a trained GNN model. Typically, this is achieved by leveraging another explanatory model to select a subset of input as the explanation for GNN prediction. Despite their utility, these post-hoc explainers often fall short of revealing the actual reasoning process of the model [25] and require optimization for each input graph, which is time-consuming. Therefore, in this paper, we focus on self-explaining methods.

The self-explaining method refers to intrinsically explainable GNN models that offer predictions and explanations concurrently, with the prediction being rooted in the explanation. One prevalent type of self-explaining model typically adopts a "*explainer-predictor*" two-stage paradigm, as illustrated in Fig. 1. This paradigm contains two stages, one is called the *explainer*, which generates an explanation for each input graph, and the other is the *predictor* making predictions based on the generated explanation [17,37].

Although the self-explaining methods in GNN are promising, they still suffer from heavily relying on extensive training data, which restricts their applicability in situations with limited data sizes. For instance, during new drug discovery processes, clinical trials are conducted to assess various drug attributes such as toxicity and side effects. Due to safety concerns, the number of participants in these trials is restricted, resulting in limited experimental data. In such few-shot scenarios, existing self-explaining models fail to achieve satisfactory performance, while existing few-shot learning methods are lack of explainability. Hence, there is a pressing need to design a self-explaining GNN for few-shot scenarios.

Drawing on the fundamental human intelligence traits of rapid learning and self-explainability [7,23,29], we develop **M**eta-learned **S**elf-**E**xplaining GNN (MSE-GNN) for few-shot scenarios:

I. During classification tasks, humans initially concentrate on regions that contain crucial features, and subsequently perform classification based on these features, adhering to a two-stage paradigm [23].
II. When learning new concepts, humans tend to seek representative instances or prototypes and compare new instances with these prototypes to categorize them [29].
III. Humans can learn meta-knowledge from a multitude of tasks, which enables them to achieve impressive performance on new tasks with limited data, which is called *"learn to learn"* [7].

By incorporating these attributes into our MSE-GNN, we aim to solve the explainability of GNNs in few-shot scenarios, and then enhance the performance of both explanation and prediction tasks.

Specifically, the MSE-GNN model follows the two-stage paradigm as depicted in Fig. 1, which naturally mimics the human's two-stage recognition process as mentioned in I. Among them, the *explainer*, which is composed of a GNN encoder and a MLP, predicts the probability of each node being selected as an explanation. Then, node representations encoded by another GNN encoder are separated into explanation and non-explanation based on the prediction of the *explainer*. Subsequently, the *predictor* mimics the decision-making process, which makes predictions based on the explanation with a MLP.

Furthermore, the MSE-GNN model incorporates a novel mechanism that exploits task information to help with selecting explanations and making predictions. Prototype, as stated in II, has been proven to be effective to generate representative representations for each category [31,46]. Therefore, in MSE-GNN, the concept of prototype is utilized in generating task information. The training framework of optimization-based meta-learning imitates the paradigm of "*learning to learn*" in III, where models can acquire meta-knowledge by learning from a vast array of tasks. One of the most popular and effective methods is MAML [7] (Model-Agnostic Meta-Learning). Therefore, we design a new meta-training framework based on MAML to train MSE-GNN.

We conduct extensive experiments on one synthetic dataset [39] and three real datasets of graph classification tasks [11,15], which show excellent performance on both prediction and explanation generated.

2 Problem Definition

In this section, we will elaborate on the problem definition of our research. Following [20], we form the few-shot graph classification problem as N-way K-shot graph classification. Given the dataset $\mathcal{G} = \{(G_1, y_1), (G_2, y_2), ..., (G_n, y_n)\}$, where G_i denotes a graph with a node set V_i and a edge set E_i. n_i denotes the number of nodes in the graph. The structure feature is represented by an adjacency matrix $A_i \in \mathbb{R}^{n_i \times n_i}$. The node attribute matrix is represented as $X_i \in \mathbb{R}^{n_i \times d}$, where d is the dimension of the node attribute.

Then, the dataset is splitted into $\{G^{train}, y^{train}\}$ and $\{G^{test}, y^{test}\}$ as training set and test set respectively according to label y. Where $y^{train} \bigcap y^{test} = \varnothing$. When training, a task $\mathcal{T}$ is sampled each time and each task contains support set $D^{train}_{sup} = (G^{train}_i, y^{train}_i)^s_{i=1}$ and query set $D^{train}_{que} = (G^{train}_i, y^{train}_i)^q_{i=1}$, where s and q stands for the size of support set and query set respectively. It is noteworthy that the same class space is shared in the same task.

In each task, our goal is to optimize our model on the support set D_{sup} and make predictions on the query set D_{que}. If a support set contains N classes and K data for each class, then we name the problem as N-way K-shot. When testing, we firstly finetune the learned model on support set $D^{test}_{sup} = (G^{test}_i, y^{test}_i)^s_{i=1}$ and then report the classification performance of finetuned model on $D^{test}_{que} = (G^{test}_i, y^{test}_i)^q_{i=1}$. Our goal of the few-shot graph classification problem is to develop a model that can obtain meta-knowledge across $\{G^{train}, y^{train}\}$ and predicts labels for graphs in the query set in test stage D^{test}_{que}.

In the explanation generation task, for each graph G_i, a node mask vector $m_i \in [0, 1]^{n_i \times 1}$ is the explanation subgraph selected, a higher value means that the corresponding node is more important for making prediction and vice versa. Although selecting edges for explanation is a viable approach, in this paper we focus on node selection due to its computational complexity.

3 The Proposed MSE-GNN

3.1 Architecture of MSE-GNN

In Fig. 2, we show the overall architecture of the MSE-GNN, which contains three components: an *explainer* g that outputs the explanation selected, a *predictor* p making the final prediction, and a graph encoder f.

Before we present the details of MSE-GNN, we first clarify several concepts. Specifically, existing works often combine self-explaining methods with the concept of rationale [17,37]. The rationale in graph data refers to the subsets of nodes or subsets of edges, which form subgraphs that determine the prediction. Hence, we posit that explanation and rationale are equivalent, as they share the same concept.

In MSE-GNN, the input graph is encoded by f and each node v is encoded into a node embedding $h_{(v)} \in \mathbb{R}^d$, where d stands for the dimension of hidden size. The encoder can be any kind of GNN, e.g. GCN [14], GIN [38], and

GraphSAGE [10]. The selector outputs a mask vector m for each graph as an explanation, which divides the graph into rationale (explanation) G_r and non-rationale G_n. Then the *predictor* makes predictions based on the graph embedding rationale subgraph. Meanwhile, augmented graphs that combine rationale and non-rationale from different graphs are fed into the *predictor* to ensure the robustness of the *predictor.* We categorize the parameters into fast parameters and slow parameters according to the timing of updating, which will be described in detail in Sect. 3.3.

Task Information. MSE-GNN generates task information for the *explainer* and the *predictor* to facilitate explanation generation and prediction within each task, which is composed of prototypes representing each class.

In each task, a support set is provided, which contains data from multiple classes. We aim to extract prototypes from these data that capture the characteristics of each class in the task, in order to help with task-specific selection of explanations and the classification task. Encoded by encoder f, each graph is represented by a matrix containing embedding of each node:

$$H_i = [..., h_{(v)}, ...]^T_{v \in V_i} = f(G_i) \in \mathbb{R}^{|V_i| \times d}. \tag{1}$$

To obtain representation for each graph h_i, the readout function, e.g. mean pooling is employed, to aggregate node embeddings. By leveraging the concept of prototype learning, we further fuse the graph representations of each class with another readout function. Thus, we can obtain a prototype embedding for each class:

$$TI_c = f_{readout}([..., f_{readout}(H_i), ...]_{y_i=c}) \in \mathbb{R}^d. \tag{2}$$

For an N-way K-shot classification problem, the task information is formed by concatenating prototypes of N classes. It is worth noting that, task information for each input graph of both D_{sup} and D_{que} is composed solely of graphs in D_{sup} to prevent label leakage.

Explainer. The *explainer* is responsible for choosing the explanation subgraph corresponding to each input graph. Specifically, given an input graph G_i, the *explainer* firstly uses another GNN encoder to map each node to another node embedding $h'_{(v)}$ for each node in V_i for selection. Then, a MLP is utilized to transform the node embeddings into a soft mask vector $m_i \in [0,1]^{n_i \times 1}$, with task information TI_c and node embedding $h'_{(v)}$ concatenated as input:

$$m_i = \sigma(MLP([..., [h'_{(v)}, TI], ...]^T_{v \in V_i})), \tag{3}$$

where σ denotes the sigmoid function. Hence, we can decompose the input graph G_i into a rationale subgraph and non-rationale subgraph according to m_i respectively:

$$G_i^r = \{A_i, X_i \odot m_i\} \qquad G_i^n = \{A_i, X_i \odot \overline{m_i}\}, \tag{4}$$

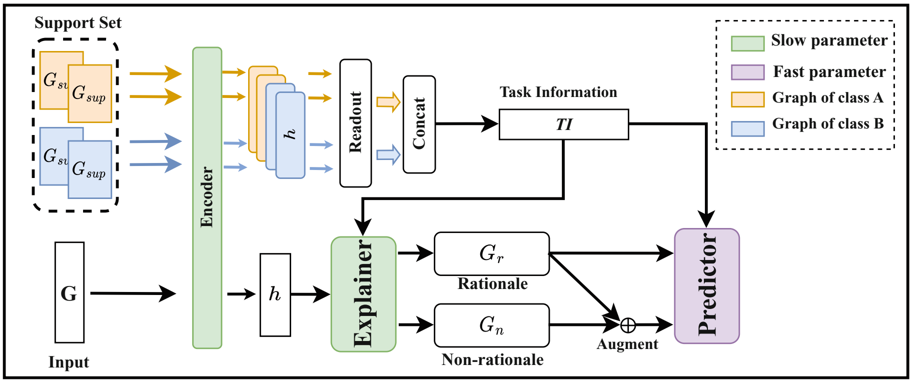


Fig. 2. Overall architecture of MSE-GNN. The model employs a "*explainer-predictor*" 2-stage self-explaining structure. The *explainer* selects explanation subgraphs for each input graph. The *predictor* mimics the decision-making process, which makes predictions solely based on the generated explanation.

where $\overline{m_i} = \mathbf{1} - m_i$. Meanwhile, given the node embedding $h_{(v)}$ from encoder f, we can obtain the graph embedding for G_i^r and G_i^n:

$$h_i^r = f_{readout}(H_i \odot m_i) \qquad h_i^n = f_{readout}(H_i \odot \overline{m_i}). \tag{5}$$

Predictor and Graph Augmentation. The *predictor* takes the graph embedding h as input and makes the final prediction $\hat{y} = p(h)$ with a MLP. Moreover, we enhance the robustness of the *predictor* through graph augmentation. Specifically, within the input graph, the rationale component represents the crucial part that determines the category, while the non-rationale component represents the noisy part. By combining the rationale and non-rationale from different graphs in the same task, additional data with noise are generated. Then we assign the label based on rationale. This approach allows us to increase the amount of noisy data, thereby improving the robustness of the *predictor*. We do the combination operation by adding subgraph embeddings:

$$h_{(i,j)} = h_i^r + h_j^n \qquad y_{(i,j)} = y_i, \tag{6}$$

where h_i^r denotes rationale from G_i and h_j^n means the non-rationale from G_j.

Therefore, in addition to task information TI, the *predictor* p receives the embeddings of both the rationale subgraphs $h_i^r$and the artificially augmented graphs $h_{(i,j)}$ for optimization, and the output are denoted as $\hat{y}_i$ and $y_{(\hat{i,j})}$ respectively.

3.2 Optimization Objective

The optimization objective of MSE-GNN is to achieve both high accuracy in predictions and generate precise explanations, which reveal the underlying reasons behind the predictions. Therefore, we design several types of loss functions and constraints. For the sake of simplicity, we consider a binary classification task without loss of generality.

Algorithm 1 . Meta-training of MSE-GNN.

Input: Distribution over meta-training tasks: $p(\mathcal{T})$; Local learning rate: η_1; Global learning rate: η_2; Local update times: T.

Output: Meta-trained parameters for encoder and explanation selector:θ_f, θ_g, and initialization of parameters for *predictor* θ_p

```
Initialize $\theta = \{\theta_f, \theta_g, \theta_p\}$ randomly;
while not converged do
    Sample task $\mathcal{T}$ with support graphs $D_{sup}^{train}$ and query graphs $D_{que}^{train}$.
    Set fast adaptation parameters: $\theta'_p = \theta_p$
    for t = 0 → T do
        Evaluate $\nabla_{\theta_p}\mathcal{L}_{sup}(\theta_f, \theta_g, \theta'_p)$ by calculating loss via Eq. 10
        Update $\theta'_p : \theta'_p \leftarrow \theta'_p - \eta_1 \cdot \nabla_{\theta'_p}\mathcal{L}_{sup}(\theta_f, \theta_g, \theta'_p)$
    end for
    Evaluate $\nabla_{\theta}\mathcal{L}_{que}(\theta_f, \theta_g, \theta'_p)$ by calculating loss via Eq. 10
    Update $\theta : \theta \leftarrow \theta - \eta_2 \cdot \nabla_{\theta}\mathcal{L}_{que}(\theta_f, \theta_g, \theta'_p)$
end while
```

With the prediction of each rationale graph embedding $p(h_i)$ and corresponding ground-truth label y_i, the loss function is defined as:

$$L_i^r = y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}). \tag{7}$$

For the artificially augmented graph, our aim is to minimize the prediction values for instances of the same category while maximizing the prediction values for instances of different categories. To achieve this, we employ a contrastive loss function. For example, for a 2-way K-shot classification task, we can obtain $4K^2$ augmented graphs, where each rationale graph is combined with other $2K - 1$ non-rationales, then the loss is computed as:

$$L_i^a = -\frac{1}{2k-1} \sum_{j=1}^{j=2K} 1_{i \neq j} \cdot 1_{y_i = y_j} \log \frac{\exp(\hat{y_i} \cdot \hat{y_j})/\tau}{\sum_{k=1}^{k=K} 1_{i \neq k} \exp(\hat{y_i} \cdot \hat{y_j})/\tau}, \tag{8}$$

where τ is a scalar temperature hyperparameter.

Besides, to address the deviation in the size of rationales, we introduce a penalty based on the number of rationale nodes, the following regularization term is utilized:

$$L^{reg} = |\frac{1_N^\top \cdot m_i}{n_i} - \gamma|, \tag{9}$$

where γ is manually set to control the rationale size. Finally, the total loss function can be formulated as:

$$L = \alpha_r \cdot L^r + \alpha_a \cdot L^a + \alpha_{reg} \cdot L^{reg}, \tag{10}$$

where α_r, α_a, and α_{reg} are hypermeters controlling the weight of each loss.

Table 1. Statistics of four datasets.

	Synthetic	MNIST-sp	Molsider	Moltox21
# Graphs	10,000	70,000	1,427	7,831
Avg # nodes	74.5	75.0	33.6	18.6
Avg # edges	237.8	777.0	70.7	38.6
# Train tasks/classes	5	5	19	7
# Validate tasks/classes	2	2	3	2
# Test tasks/classes	3	3	5	3

3.3 Meta Training

Inspired by the concept of *"learn to learn"* [7], we propose a new meta-training framework based on MAML to obtain meta knowledge from various tasks. We denote θ_f, θ_g, and θ_p as the parameters of encoder, explanation selector, and the *predictor*. Specifically, MSE-GNN is trained from two procedures. One is global update, which aims to learn the parameters of encoder θ_f, explanation generator θ_g, and initialization of the *predictor* θ_p from different tasks, the other is called local update, which performs fast adaption on new tasks and locally update only parameters of the *predictor* θ'_p within each task. According to the timing of updating, we categorize the parameters into fast parameters (θ_p) and slow parameters (θ_f and θ_g), as shown in Fig. 2.

The meta-training process is demonstrated in Algorithm 1. Firstly, we sample a task composed of support D^{train}_{sup} and query data D^{train}_{que} for each episode. Then adaption is operated by updating θ_p for T times on D^{train}_{sup}, where T is a hyperparameter controlling the number of local updates, which is shown in lines 5–8. With updated θ'_p, we utilize the loss on D^{train}_{que} to update θ_f, θ_g and θ_p.

It is important to highlight that, the *explainer* is trained from a variety of tasks and frozen when optimizing each task, which ensures the stability of the explanation selected across different tasks and prevents over-fitting. Therefore, θ_f and θ_g are only updated in the global update and fixed in the local update. While the *predictor* needs to learn the relationship between features and categories in different tasks based on the generated explanations. As a result, the θ_p is optimized in the local update to learn the association between features and categories. Hyperparameters of loss computation in line 6 and line 9 can be differently set according to the goal of local and global optimization.

4 Experiments

4.1 Datasets and Experimental Setup

Dataset. We conduct extensive experiments on four datasets to validate the performance of MSE-GNN: (i) **Synthetic**: Due to the lack of graph datasets with explanation ground-truth, following [39], we create a synthetic dataset for classification, which contains 10 classes and 500 samples for each class. Each graph is composed of two parts: the rationale part and the non-rationale part. The label of each graph is determined by the rationale part. Therefore, the ground-truth of the explanation subgraph is the rationale part of each graph. (ii) **MNIST-sp** [15]: MNIST-sp takes the MNIST images and transforms them into 70,000 superpixel graphs. Each graph consists of 75 nodes and is assigned one of 10 class labels. The subgraphs that represent the digits can be interpreted as ground truth explanations. (iii) **OGBG-Molsider** and **OGBG-Moltox21** [11]: These two datasets are molecule datasets from the graph property prediction task on Open Graph Benchmark (OGBG), they contain 27 and 12 binary labels for each graph, which transformed into 27 and 12 binary classification tasks respectively. The dataset statistics are available in Table 1.

Table 2. 2-way 5-shot Classification Performance with a standard deviation of baseline methods and MSE-GNN.

	Accuracy				AUC-ROC			
	Synthetic		MNIST-sp		OGBG-molsider		OGBG-moltox21	
	GIN	GraphSAGE	GIN	GraphSAGE	GIN	GraphSAGE	GIN	GraphSAGE
ProtoNet	$0.8284_{\pm 0.058}$	$0.8327_{\pm 0.027}$	$0.5736_{\pm 0.008}$	$0.6575_{\pm 0.034}$	$0.5540_{\pm 0.006}$	$0.5468_{\pm 0.006}$	$0.6614_{\pm 0.009}$	$0.6495_{\pm 0.008}$
MAML	$0.8259_{\pm 0.007}$	$0.6409_{\pm 0.327}$	$0.6283_{\pm 0.012}$	$0.6722_{\pm 0.009}$	$0.6219_{\pm 0.005}$	$0.6538_{\pm 0.016}$	$0.7217_{\pm 0.030}$	$0.6965_{\pm 0.014}$
ASMAML	$0.8911_{\pm 0.010}$	$0.7849_{\pm 0.014}$	$0.6526_{\pm 0.004}$	$0.6699_{\pm 0.023}$	$0.6288_{\pm 0.007}$	$\mathbf{0.6818_{\pm 0.008}}$	$0.7432_{\pm 0.030}$	$0.7181_{\pm 0.017}$
GREA_Raw	$0.6970_{\pm 0.005}$	$0.6970_{\pm 0.020}$	$0.6405_{\pm 0.009}$	$0.6667_{\pm 0.009}$	$0.5210_{\pm 0.009}$	$0.5180_{\pm 0.007}$	$0.5654_{\pm 0.015}$	$0.5479_{\pm 0.006}$
CAL_Raw	$0.7248_{\pm 0.006}$	$0.7488_{\pm 0.007}$	$0.6498_{\pm 0.006}$	$0.6670_{\pm 0.010}$	$0.5978_{\pm 0.044}$	$0.6230_{\pm 0.008}$	$0.6161_{\pm 0.064}$	$0.6814_{\pm 0.014}$
GREA_Meta	$0.8728_{\pm 0.013}$	$0.9180_{\pm 0.002}$	$0.6537_{\pm 0.009}$	$0.7430_{\pm 0.008}$	$0.6542_{\pm 0.005}$	$0.6303_{\pm 0.008}$	$0.7650_{\pm 0.004}$	$0.7582_{\pm 0.007}$
CAL_Meta	$0.8451_{\pm 0.021}$	$0.9096_{\pm 0.003}$	$\mathbf{0.6888_{\pm 0.007}}$	$\mathbf{0.7445_{\pm 0.019}}$	$0.6580_{\pm 0.012}$	$0.6553_{\pm 0.018}$	$0.7442_{\pm 0.012}$	$0.7652_{\pm 0.005}$
MSE-GNN	$\mathbf{0.9103_{\pm 0.004}}$	$\mathbf{0.9200_{\pm 0.004}}$	$0.6515_{\pm 0.008}$	$0.7309_{\pm 0.009}$	$\mathbf{0.6673_{\pm 0.007}}$	$0.6587_{\pm 0.002}$	$\mathbf{0.7735_{\pm 0.006}}$	$\mathbf{0.7728_{\pm 0.011}}$

Experimental Setup. To investigate whether generating explanations can help with the classification task, we chose three few-shot learning methods: ProtoNet [29], MAML [7], ASMAML [20]. To compare with existing self-explaining methods, we selected two state-of-the-art self-explaining models: GREA [17] and CAL [30] as baselines to compare the performance of classification and quality of generated explanations. Moreover, for fairness, we adapt meta-training to GREA [17] and CAL [30], enabling them to adapt to few-shot scenarios, which are denoted as GREA_Meta and CAL_Meta respectively.

We use GIN and GraphSAGE as GNN backbones for all methods. The performance of all models is evaluated on D_{que}^{test}. For the Synthetic and MNIST-sp with explanation ground-truth, we use Accuracy to evaluate the classification performance and AUC-ROC to evaluate the quality of the explanation selected. For the two molecule datasets, due to the absence of explanation ground-truth, we only evaluate the classification performance using Area under the ROC curve (AUC) following [17]. For meta-training, we utilize Adam optimizer for local and global updates and set local update times T to 5. Local learning rate η_1 is set to 0.001 and global learning rate η_1 is tuned over {1e-5, 1e-4, 1e-3}. γ in Eq. 9 is tuned over {0.1, 0.2, 0.3, 0.4, 0.5}, number of GNN layers is tuned over {2,3}. We select hyperparameters based on related works and grid searches. All our experiments are conducted with one Tesla V100 GPU.

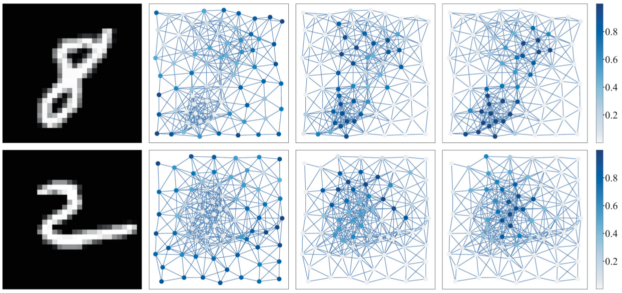
Table 3. For the Synthetic and MNIST-sp with explanation ground-truth, AUC-ROC is utilized to evaluate the quality of the explanation selected.

		Synthetic	MNIST-sp
GIN	GREA_Raw	0.4934$_{\pm 0.006}$	0.4789$_{\pm 0.044}$
	CAL_Raw	0.4741$_{\pm 0.0250}$	0.4395$_{\pm 0.039}$
	GREA_Meta	0.6745$_{\pm 0.0265}$	0.7855$_{\pm 0.013}$
	CAL_Meta	0.6201$_{\pm 0.0550}$	0.1707$_{\pm 0.0243}$
	MSE-GNN	**0.7000**$_{\pm \mathbf{0.006}}$	**0.8222**$_{\pm \mathbf{0.030}}$
Graghsage	GREA_Raw	0.4929$_{\pm 0.023}$	0.5496$_{\pm 0.064}$
	CAL_Raw	0.5080$_{\pm 0.054}$	0.4906$_{\pm 0.116}$
	GREA_Meta	0.7099$_{\pm 0.014}$	0.6513$_{\pm 0.040}$
	CAL_Meta	0.6858$_{\pm 0.015}$	0.6613$_{\pm 0.229}$
	MSE-GNN	**0.7189**$_{\pm \mathbf{0.012}}$	**0.7077**$_{\pm \mathbf{0.038}}$

Performance on Synthetic Graphs and MNIST-sp. To explore whether MSE-GNN can achieve high performance on classification and generate high-quality explanation, we conduct 2-way 5-shot experiments on Synthetic and MNIST-sp datasets which contain ground-truth explanations for each graph. The experimental results are summarized in Table 2 and Table 3. We first compare meta-trained self-explaining baseline models (GREA_Meta, CAL_Meta) with themselves (GREA_Raw, CAL_Raw). We can observe that significant performance boosts are brought by meta-training on both classification and explanation, which indicates that meta-training can leverage the meta-knowledge learned across training tasks effectively on new tasks.

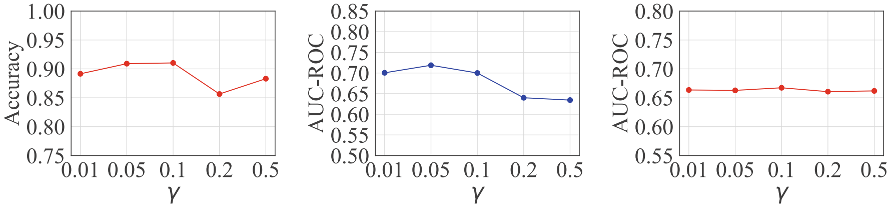
On Synthetic, MSE-GNN shows superiority to other baseline methods on the performance of classification and explanation quality. Compared to meta-trained self-explaining baselines, MSE-GNN performs better on both classification and explanation as MSE-GNN utilizes task information and effectively leverages the augmented graph through the introduction of supervised contrastive loss. Moreover, the inherent denoising capability of self-explaining models contributes to the superior classification performance of MSE-GNN compared to ProtoNet, MAML, and ASMAML.

Unexpectedly, CAL achieves the best classification performance on MNIST-sp, especially when using GIN as the backbone, surpassing MSE-GNN by over 5%. Meanwhile, the quality of explanations is significantly lower compared to GREA and MSE-GNN. By visualization in Fig. 3, which reveals the internal reasoning process of models, we can find that CAL generated explanations that were opposite to our expectations, indicating that CAL infers the digit based on the shape of the background. It is also easy to understand that the digital in a picture can be inferred from the background since the number part and the background part are complementary sets. Therefore, despite the generated explanations being contrary to our expectations, CAL's performance demonstrated that utilizing background information for digit prediction is more effective on MNIST-sp. The reason for CAL generating opposite explanations is that it lacks constraints on the size of the explanation. As a result, it tends to favor subgraphs that contain more useful information and overlook the size of the explanation subgraph. Furtherly comparing the visualization of explanations of MSE-GNN and GREA, we can find that explanations of MSE-GNN are more compact and focus more on the digital part, which is in line with the result in Table 3.



(a) Raw (b) CAL (c) GREA (d) MSE-GNN

Fig. 3. Raw figure of MNIST-sp and visualization of explanations generated by CAL(a), GREA(b) and MSE-GNN(c). Darker nodes indicate higher importance scores.

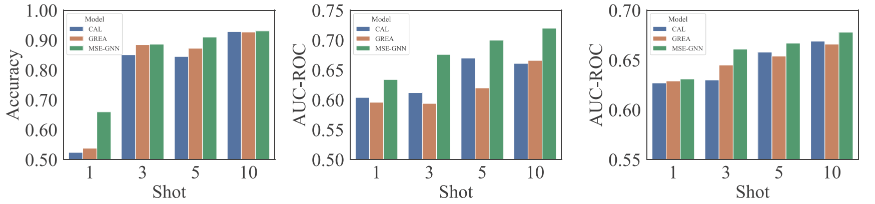


(a) Classification Performance on Synthetic (b) Quality of explanation on Synthetic (c) Classification Performance on OGBG-Molsider

Fig. 4. Classification Performance and quality of explanation selected on Synthetic and OGBG-Molsider with different γ.

Performance on OGBG. MSE-GNN achieves comparable classification performance on these two molecule datasets, demonstrating the effectiveness of its structure. Furthermore, we can observe that the self-explaining models with meta-training outperform all meta-learning models except on OGBG-molsider using GraphSAGE. This is because the process of generating explanations can potentially improve the classification task by eliminating irrelevant noise.

Performance with Different Size of Support Set. Intuitively, for a classification task, the size of the training set has a significant impact on the model's performance. Therefore, in the scenario of few-shot learning, we evaluate the performance of MSE-GNN and other self-explaining models under different support set sizes. Experimental results are shown in Fig. 5. First, comparing different methods, we observe that MSE-GNN consistently outperforms other baselines across different support set sizes, which further validates the performance of MSE-GNN on both classification and explaining. Next, comparing the performance of MSE-GNN across different support set sizes, we observe that as the support set size increases, both the classification accuracy and the quality of generated explanations improve. This also demonstrates the importance of training set size on model performance.



(a) Classification Performance on Synthetic

(b) Quality of explanation on Synthetic

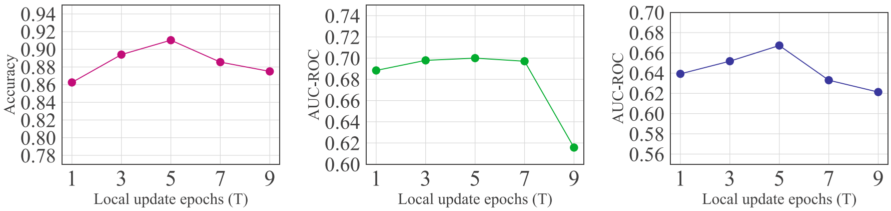
(c) Classification Performance on OGBG-Molsider

Fig. 5. Classification Performance and quality of explanation selected on Synthetic and OGBG-Molsider with different size of support sets.

Ablation Study. Table 4 demonstrates the impact of contrastive loss and task information utilized in MSE-GNN on Synthetic with GIN. When applying Contrastive Loss (CL), both the classification accuracy and the quality of generated explanations of the model are improved. This indicates that introducing contrastive loss can enhance the model's performance and lead to better results in prediction and explanation tasks. On the other hand, when applying Task Information (TI), the model's performance is also improved across all datasets. This suggests that incorporating task information into the model can provide additional context and guidance, thereby enhancing the model's ability. Moreover, when both CL and TI are used together, the model excels significantly across all datasets, indicating that the combination of CL and TI can synergistically contribute to better performance on both classification and explanation tasks.

Table 4. Impact of contrastive loss and task information.

CL	TI	Synthetic		OGBG-molsider
		Classif.	Explain.	Classif.
		$0.8728_{\pm 0.013}$	$0.6745_{\pm 0.027}$	$0.6542_{\pm 0.005}$
✓		$0.8809_{\pm 0.037}$	$0.6860_{\pm 0.028}$	$0.6623_{\pm 0.011}$
	✓	$0.8800_{\pm 0.011}$	$0.6766_{\pm 0.014}$	$0.6616_{\pm 0.001}$
✓	✓	$\mathbf{0.9103_{\pm 0.004}}$	$\mathbf{0.7000_{\pm 0.006}}$	$\mathbf{0.6673_{\pm 0.007}}$



(a) Classification Performance on Synthetic

(b) Quality of explanation on Synthetic

(c) Classification Performance on OGBG-Molsider

Fig. 6. Classification Performance and quality of explanation selected on Synthetic and OGBG-Molsider with different T.

Sensitivity Analysis. In MSE-GNN, the parameter γ is crucial in controlling the size of the selected explanation. To examine the sensitivity of the model to different values of γ, we conduct a sensitivity analysis on the Synthetic and OGBG-Molsider datasets with GIN. As illustrated in Fig. 4, the results demonstrate that MSE-GNN achieves the best classification performance when γ is set to 0.1 on both datasets, while the explaining performance achieves best when γ equals 0.05 on Synthetic. We observe that as the value of γ deviates from these two optimal points, the classification performance or the quality of generated explanations decreases. We also notice that the impact of γ is less pronounced on the OGBG-Molsider dataset, indicating that the model is less sensitive to γ on OGBG-Molsider.

Furthermore, T, which stands for the number of local update epochs, affects both the effectiveness and efficiency of the MSE-GNN. We compared the performance of MSE-GNN with different local update epochs on the Synthetic and OGBG-Molsider datasets. The experimental results shown in Fig. 6 indicate that when T is set to 5, MSE-GNN achieves the best classification and explaining performance on both Synthetic and OGBG-molsider. A too-small (too-large) T may result in underfitting (overfitting) of the model for new tasks.

5 Related Works

Few-Shot Learning and Meta Learning on Graph Classification. Few-shot learning aims to learn a model with only a few samples. A promising kind of method is meta learning. Meta learning is also known as "learning to learn", which attempts to learn meta-knowledge from a variety of tasks. There two categories for meta-learning [44]: metric-based models [3,8,22,29,32] and optimization-based models [7,9,20,34,51]. The former focuses on computing the distance between query data and class prototypes [29]. The latter aims to learn an effective initialization of parameters, which enables rapid adaption [7]. [51] firstly applied meta learning framework to the node classification task. [20] utilize a step controller for the robustness and generalization of meta-learner. Notwithstanding the remarkable accuracy improvement achieved by these methods on few-shot learning tasks, their lack of explainability hinders their applicability in certain scenarios such as the medical and finance area.

Explainability in Graph Neural Network. With more attention paid to the applications of GNNs, the explainability of GNNs is more crucial. The explanation increases the models' transparency and enhances practitioners' trust in GNN models by enriching their understanding of why the decision is made by GNNs. Explainability of GNNs can be categorized into two classes [40,42]: post-hoc explanations and self-explainable GNNs. Post-hoc explanations attempt to give explanations for trained GNNs with additional *explainer* model [1,5,12,13,18,19,33,39]. However, these post-hoc explainers often fail to unveil the true reasoning process of the model due to the non-convexity and complexity of the underlying GNN models [25]. Self-explaining GNNs design specific GNN models which are interpretable intrinsically [1,17,21,30,37,50]. They output the prediction and corresponding explanation simultaneously. DIR [37] aims to extract causal rationales that remain consistent across various distributions while eliminating unstable spurious patterns. GREA [17] is another self-explainable model that introduces a new augmentation operation called environment replacement that automatically creates virtual data examples to improve rationale identification. Another category of self-explaining models leverages the concept of prototype learning [1,26,27,47,50]. ProtGNN [50] provides explanations by selecting subgraphs that are the most relevant to graph patterns for identifying graphs of each class. However, existing self-explainable GNNs overlook the scarcity of labeled graph data in many applications. Thus, it's important to build few-shot learning models with self-explainability.

6 Conclusion

In this paper, we proposed MSE-GNN to address the explainability of GNN in few-shot scenarios. To be specific, MSE-GNN adopted a "*explainer-predictor*" 2-stage self-explaining structure and a meta-training framework based on meta-learning, which improved performance in few-shot scenarios. MSE-GNN also introduced a mechanism to leverage task information to assist explanation generation and result prediction. Additionally, MSE-GNN employed graph augmentation to enhance model robustness. Extensive experimental results demonstrated that MSE-GNN achieves strong performance in classification tasks while selecting high-quality explanations in few-shot scenarios.

Acknowledgements. This research was partially supported by supported by Anhui Provincial Natural Science Foundation (No. 2308085QF229), the Fundamental Research Funds for the Central Universities (No. WK2150110034), Technology Innovation Community in Yangtze River Delta (No. 2023CSJZN0200), and the Fundamental Research Funds for the Central Universities.

References

1. Azzolin, S., Longa, A., Barbiero, P., Lio, P., Passerini, A.: Global explainability of GNNs via logic combination of learned concepts. In: The Eleventh International Conference on Learning Representations (2022)
2. Bian, T., et al.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 549–556 (2020)
3. Chauhan, J., Nathani, D., Kaul, M.: Few-shot learning on graphs via super-classes based on graph spectral measures. In: International Conference on Learning Representations (2019)
4. Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M.: Revisiting graph based collaborative filtering: a linear residual graph convolutional network approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 27–34 (2020)
5. Duval, A., Malliaros, F.D.: GraphSVX: Shapley value explanations for graph neural networks. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, 13–17 September 2021, Proceedings, Part II 21, pp. 302–318. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86520-7_19
6. Dwivedi, V.P., Joshi, C.K., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Benchmarking graph neural networks. J. Mach. Learn. Res. **24**, 1–48 (2023)
7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
8. Gao, W., et al.: Leveraging transferable knowledge concept graph embedding for cold-start cognitive diagnosis. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 983–992 (2023)
9. Guo, Z., et al.: Few-shot graph learning for molecular property prediction. In: Proceedings of the Web Conference 2021, pp. 2559–2567 (2021)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
11. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. Adv. Neural. Inf. Process. Syst. **33**, 22118–22133 (2020)

12. Huang, Q., Yamada, M., Tian, Y., Singh, D., Chang, Y.: GraphLIME: local interpretable model explanations for graph neural networks. IEEE Trans. Knowl. Data Eng. **35**(7), 6968–6972 (2022)
13. Kamal, A., Vincent, E., Plantevit, M., Robardet, C.: Improving the quality of rule-based GNN explanations. In: Koprinska, I., et al. (eds.) Joint European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2022. CCIS, vol. 1752, pp. 467–482. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-23618-1_31
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2016)
15. Knyazev, B., Taylor, G.W., Amer, M.: Understanding attention and generalization in graph neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
16. Lin, W., Lan, H., Wang, H., Li, B.: OrphicX: a causality-inspired latent variable model for interpreting graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13729–13738 (2022)
17. Liu, G., Zhao, T., Xu, J., Luo, T., Jiang, M.: Graph rationalization with environment-based augmentations. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1069–1078 (2022)
18. Lucic, A., Ter Hoeve, M.A., Tolomei, G., De Rijke, M., Silvestri, F.: CF-GNNExplainer: counterfactual explanations for graph neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 4499–4511. PMLR (2022)
19. Luo, D., et al.: Parameterized explainer for graph neural network. Adv. Neural. Inf. Process. Syst. **33**, 19620–19631 (2020)
20. Ma, N., et al.: Adaptive-step graph meta-learner for few-shot graph classification. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1055–1064 (2020)
21. Müller, P., Faber, L., Martinkus, K., Wattenhofer, R.: DT+GNN: a fully explainable graph neural network using decision trees. arXiv preprint arXiv:2205.13234 (2022)
22. Niu, G., et al.: Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 213–222 (2021)
23. Posner, M.I., Petersen, S.E.: The attention system of the human brain. Annu. Rev. Neurosci. **13**, 25–42 (1990)
24. Pourhabibi, T., Ong, K.L., Kam, B.H., Boo, Y.L.: Fraud detection: a systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. **133**, 113303 (2020)
25. Rudin, C.: Please stop explaining black box models for high stakes decisions. Stat **1050**, 26 (2018)
26. Seo, S., Kim, S., Park, C.: Interpretable prototype-based graph information bottleneck. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
27. Shin, Y.M., Kim, S.W., Yoon, E.B., Shin, W.Y.: Prototype-based explanations for graph neural networks (student abstract). In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 13047–13048 (2022)
28. Smuha, N.A.: The EU approach to ethics guidelines for trustworthy artificial intelligence. Comput. Law Rev. Int. **20**, 97–106 (2019)
29. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

30. Sui, Y., Wang, X., Wu, J., Lin, M., He, X., Chua, T.S.: Causal attention for interpretable and generalizable graph classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1696–1705 (2022)
31. Vuorio, R., Sun, S.H., Hu, H., Lim, J.J.: Multimodal model-agnostic meta-learning via task-aware modulation. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
32. Wang, S., Huang, X., Chen, C., Wu, L., Li, J.: Reform: error-aware few-shot knowledge graph completion. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1979–1988 (2021)
33. Wang, X., Shen, H.W.: GNNInterpreter: a probabilistic generative model-level explanation for graph neural networks. In: The Eleventh International Conference on Learning Representations (2022)
34. Wang, Y., Abuduweili, A., Yao, Q., Dou, D.: Property-aware relation networks for few-shot molecular property prediction. Adv. Neural. Inf. Process. Syst. **34**, 17441–17454 (2021)
35. Wieder, O., et al.: A compact review of molecular property prediction with graph neural networks. Drug Discov. Today Technol. **37**, 1–12 (2020)
36. Wu, L., Cui, P., Pei, J., Zhao, L., Guo, X.: Graph neural networks: foundation, frontiers and applications. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4840–4841 (2022)
37. Wu, Y., Wang, X., Zhang, A., He, X., Chua, T.S.: Discovering invariant rationales for graph neural networks. In: International Conference on Learning Representations (2021)
38. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2018)
39. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
40. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: a taxonomic survey. IEEE Trans. Pattern Anal. Mach. Intell. **45**, 5782–5799 (2022)
41. Yue, L., Liu, Q., Du, Y., An, Y., Wang, L., Chen, E.: DARE: disentanglement-augmented rationale extraction. Adv. Neural. Inf. Process. Syst. **35**, 26603–26617 (2022)
42. Yue, L., Liu, Q., Liu, Y., Gao, W., Yao, F., Li, W.: Cooperative classification and rationalization for graph generalization. In: Proceedings of the ACM Web Conference, vol. 2024 (2024)
43. Yue, L., Liu, Q., Wang, L., An, Y., Du, Y., Huang, Z.: Interventional rationalization. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 11404–11418 (2023)
44. Zhang, C., et al.: Few-shot learning on graphs: a survey. In: The 31st International Joint Conference on Artificial Intelligence (IJCAI) (2022)
45. Zhang, K., et al.: EATN: an efficient adaptive transfer network for aspect-level sentiment analysis. IEEE Trans. Knowl. Data Eng. **35**(1), 377–389 (2021)
46. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5773–5780 (2019)
47. Zhang, K., et al.: Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. arXiv preprint arXiv:2203.16369 (2022)
48. Zhang, Z., Hu, Q., Yu, Y., Gao, W., Liu, Q.: FedGT: federated node classification with scalable graph transformer. arXiv preprint arXiv:2401.15203 (2024)

49. Zhang, Z., Liu, Q., Hu, Q., Lee, C.K.: Hierarchical graph transformer with adaptive node sampling. Adv. Neural Inf. Process. Syst. **35**, 21171–21183 (2022)
50. Zhang, Z., Liu, Q., Wang, H., Lu, C., Lee, C.: ProtGNN: towards self-explaining graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 9127–9135 (2022)
51. Zhou, F., Cao, C., Zhang, K., Trajcevski, G., Zhong, T., Geng, J.: Meta-GNN: on few-shot node classification in graph meta-learning. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2357–2360 (2019)