Robust Graph Meta-learning via Manifold Calibration with Proxy Subgraphs

Zhenzhong Wang[1], **Lulu Cao**[2], **Wanyu Lin**[1*], **Min Jiang** [2], **Kay Chen Tan** [1]

[1] Department of Computing, The Hong Kong Polytechnic University
[2] School of Informatics, Xiamen University
zhenzhong16.wang@connect.polyu.hk, lulucao@stu.xmu.edu.cn, wan-yu.lin@polyu.edu.hk, minjiang@xmu.edu.cn,
kctan@polyu.edu.hk

Abstract

Graph meta-learning has become a preferable paradigm for graph-based node classification with long-tail distribution, owing to its capability of capturing the intrinsic manifold of support and query nodes. Despite the remarkable success, graph meta-learning suffers from severe performance degradation when training on graph data with structural noise. In this work, we observe that the structural noise may impair the smoothness of the intrinsic manifold supporting the support and query nodes, leading to the poor transferable priori of the meta-learner. To address the issue, we propose a new approach for graph meta-learning that is robust against structural noise, called **Pro**xy subgraph based **M**anifold **C**alibration method (Pro-MC). Concretely, a subgraph generator is designed to generate proxy subgraphs that can calibrate the smoothness of the manifold. The proxy subgraph compromises two types of subgraphs with two biases, thus preventing the manifold from being rugged and straightforward. By doing so, our proposed meta-learner can obtain generalizable and transferable prior knowledge. In addition, we provide a theoretical analysis to illustrate the effectiveness of Pro-MC. Experimental results have demonstrated that our approach can achieve state-of-the-art performance under various structural noises.

INTRODUCTION

Graph neural networks (GNNs) (Kipf and Welling 2017), extracting node representations by propagating and updating the representations along the graph topology, have become an important research topic in various practical frontiers (Xie and Grossman 2018; Fout et al. 2017). Among those applications, various scenarios involve graph-based node classification tasks, e.g., fake account detection in social networks (Zhu et al. 2012).

Although GNNs have shown outstanding capability in graph-based node classification tasks, their performance significantly degrades when they are trained on graphs with *long-tail* distribution (Chauhan, Nathani, and Kaul 2020). Specifically, in canonical graph-based few-shot node classification tasks, it is non-trivial to generalize to novel classes as the novel classes typically only have one or very few labeled nodes. Besides, GNNs are vulnerable to *structural noise* which refers to noisy or perturbed edges caused by unnoticeable perturbations or attacks (Dai et al. 2022). Due to the message-passing process, noisy information can be propagated along the noisy edges, contaminating the deduced node embedding and degrading the performance.

One paradigm to handle the *long-tail* issue is graph meta-learning (Huang and Zitnik 2020; Liu et al. 2021; Zhou et al. 2019a). In graph meta-learning, the intrinsic manifold of support nodes and query nodes can be learned by propagating information along the topological structure (see Fig.1 (c)), thus obtaining an optimum classification boundary that can be used as the transferable prior knowledge for the meta-learner. On the contrary, the lack of topology information can induce a boundary with poor generalization (see Fig.1 (b)). On the other hand, many efforts have been made to alleviate the effects of structural noise. For example, some works assume edges between dissimilar nodes as noisy edges, and they prune such noisy edges to suppress the noisy information (Entezari et al. 2020; Zhu et al. 2019; Wu et al. 2019; Dai et al. 2022). However, simply pruning edges may result in information loss because even in a clean graph, the dissimilar nodes may still have inherent relationships (Tang et al. 2020). Others attempt to construct a low-rank approximation of the adjacent matrix, as in a clean graph, the rank of the adjacency matrix is usually lower than that of a noisy graph (Luo et al. 2021; Zhou, Zha, and Song 2013; Jin et al. 2020). However, low-rank approximation methods often neglect to exploit node features as complementary information for purifying embedding.

Compared with *long-tail* distribution solely appeared, *long-tail* distribution with *structural noise* is more challenging. Specifically, we observe the contaminated node embedding may impair the smoothness $S(\mathcal{M})$ [1] of the manifold supporting the support and query nodes, thus leading to a poor generalizable priori (see Fig.1 (d) and (e)). Inspired by the observation, we argue that the manifold's smoothness property may serve as guidance to resist structural noise.

In this work, we use subgraph-level embedding to dilute the noisy information and calibrate the smoothness of the manifold. Firstly, noisy subgraph embedding is extracted from the raw graph. Because the raw graph suffers from structural noises, the manifold represented by the noisy sub-

[*] Corresponding author

[1] The smoothness $S(\mathcal{M})$ is defined in Eq. (2)

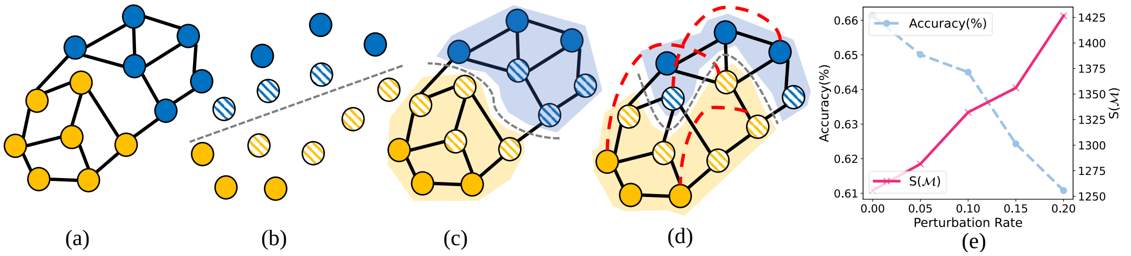


Figure 1: (a) The ground truth of a toy example. (b) A possible classification boundary (the grey dotted line) when training with labeled samples. (c) A possible classification boundary when the manifold supporting support and query nodes are considered. (d) The manifold is rugged due to the structural noise (red dotted lines), and the classification boundary leads to misclassification. (e) The relationship between manifold smoothness $S(\mathcal{M})$ and accuracy of a classic graph meta-learning method, Meta-GNN, (Zhou et al. 2019a) with respect to varying perturbation rates on the Cora dataset (Sen et al. 2008), where a targeted attack method, nettack (Zügner, Akbarnejad, and Günnemann 2018), is applied to perturb the training set.

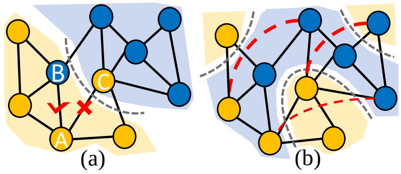


Figure 2: (a) Over-denoising manifold: Node A is more similar to node B of a different class than node C of the same class. Suppressing the dissimilar node, i.e., node C, will get a too straightforward manifold and lead to misclassification. (b) Rugged manifold: Injecting structural noise is prone to lead to a rugged manifold.

graph embedding is rugged, which could misclassify nodes affected by noisy edges (see Fig. 2 (b)). Then, smooth subgraph embedding is obtained by suppressing dissimilar neighbors. Simply suppressing neighbors may lose inherent patterns of a class and get a straightforward manifold, which results in misclassifying dissimilar nodes that belong to the same class (see Fig. 2 (a)). To avoid the straightforward manifold or rugged manifold, another subgraph representation, called proxy subgraph, is generated by a subgraph generator. The proxy subgraph calibrates the manifold by compromising the bias of the smooth subgraph (i.e., straightforward manifold) and the bias of the noisy subgraph (i.e., rugged manifold). With the help of the proxy subgraph, we expect the manifold can be calibrated with proper smoothness, thus improving the quality of transferable priori of the meta-learner. The main contributions of this work are summarized as follows,

1. We analyze the possible causes for the performance degradation of graph meta-learning under structural noise, and we propose a proxy subgraph-based robust graph meta-learning method to address *long-tail* distribution with *structural noise*.
2. A subgraph generator is designed to synthesize proxy subgraphs to calibrate the manifold with moderate smoothness. Compared with existing denoising methods, our proxy subgraph method considers both node features and graph topology and prevents the manifold from being straightforward or rugged, so that the meta-learner can obtain generalizable and transferable prior knowledge.
3. We conduct extensive experiments on three representative datasets. Experimental results show that the proposed algorithm achieves promising results under various structural noises. In addition, we provide a theoretical generalized error of the proposed proxy subgraph-based graph meta-learning on structural noise.

PRELIMINARIES

Few-shot Node Classification: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denotes an undirected graph with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, where $\mathbf{X} = (x_1, ..., x_{|\mathcal{V}|})^T \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a set of feature vectors, and $x \in \mathbb{R}^d$ is a d-dimensional feature vector for each node in $\mathcal{V}$. $\mathcal{C}_b$ and $\mathcal{C}_n$ stand for the base and novel classes, respectively, where $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. Given the base classes with a sufficient number of labeled nodes and k labeled nodes per class for m novel classes, few-shot node classification aims to get a learner $f : \mathcal{V} \to \mathcal{C}_n$ for the remaining unlabeled novel nodes. This setting is called m-way k-shot node classification.

Graph Meta-learning: Graph meta-learning has shown great promise in few-shot node classification. In graph meta-learning, the base classes $\mathcal{C}_b$ and novel classes $\mathcal{C}_n$ are used for the meta-train tasks and the meta-test task, respectively. To construct a meta-train task $\mathcal{T}_t = \{S_t, Q_t\}$, k labeled nodes per class for m classes are firstly randomly sampled from the base classes $\mathcal{C}_b$, and this set of nodes is denoted as the support set S_t, and the set of remaining unlabeled nodes among the m classes is denoted as the query set Q_t. After sampling a series of meta-train tasks and performing the learning process on these tasks, the meta-test task is used for optimizing the meta-learner with the obtained transferable priori. Similarly, to construct a meta-test task $\mathcal{T}_t^{'} = \{S_t^{'}, Q_t^{'}\}$, m-way k-shot nodes served as the support set $S_t^{'}$ are sampled from the novel classes $\mathcal{C}_n$, and the set of remaining unlabeled nodes from the novel classes $\mathcal{C}_n$ is denoted as the query set $Q_t^{'}$. Finally, the performance of the meta-learner is evaluated on the query set $Q_t^{'}$.

A number of graph meta-learning works have been proposed in recent years. Meta-GNN (Zhou et al. 2019a) follows the episodic training paradigm and uses GNNs as the meta-learner for few-shot node classification. Some works focus on augmenting representation to enhance receptive fields for efficient information propagation (Ding et al. 2022; Zhao, Wang, and Xiang 2021; Huang and Zitnik 2020; Liu et al. 2021, 2019; Lan et al. 2020; Liu et al. 2020). GFL (Yao et al. 2020) draws the support from auxiliary graphs to improve generalization ability on the target graph. RALE and CNL (Liu et al. 2021; Zhao et al. 2021) aim to capture the potential long-ranged dependencies to learn node embedding. Some works extract subgraphs to augment the representation (Zhao, Wang, and Xiang 2021; Huang and Zitnik 2020; Ma et al. 2020; Zhang et al. 2020). As discussed, richer structures of graphs, including subgraphs, long-ranged dependencies, etc., can be considered useful knowledge for meta-learning.

Robust GNNs: Structural noise arises in nature or is injected deliberately by attackers. Extensive studies have demonstrated that GNNs are vulnerable to structural noise. To defend against structural noise, a common idea is to remove noisy edges (Entezari et al. 2020). RGCN (Zhu et al. 2019) introduces Gaussian constraints on model parameters to absorb the effects of noisy changes. PA-GNN (Tang et al. 2020) leverages supervision knowledge from clean graphs and applies the meta-optimization method to learn a robust GNN. GCN-Jaccard (Wu et al. 2019) eliminates edges that connect nodes with low Jaccard similarity. RS-GNN (Dai et al. 2022) aims to learn a robust noise-resistant GNN with limited labeled nodes by densifying graphs and eliminating noisy edges.

Recent works show that constructing a low-rank adjacent matrix of a graph can defend against structural noise. Pro-GNN (Jin et al. 2020) observes that noisy edges can quickly increase the rank of the adjacency matrix. Taking the cue, Pro-GNN learns a robust GNN with the low-rank property. Similarly, GCN-SVD (Entezari et al. 2020) is proposed to resist the high-rank attack and vaccinate GCN with the low-rank approximation of the perturbed graph. PTDNet (Luo et al. 2021) imposes the low-rank constraint on the sparsified graph for better generalization.

In our work, we aim to explore important graph structures, i.e., subgraphs, to learn a robust GNN, which enables the proposed model can recover a manifold with proper smoothness from perturbed graphs under different structural noises.

PROPOSED ALGORITHM

The framework of Pro-MC is illustrated in Fig. 3. There are three types of subgraph-level embedding, i.e., noisy subgraph embedding, smooth subgraph embedding, and proxy subgraph embedding. For a node v of interest, noisy subgraph embedding $\mathbf{S}_v^{(no)}$ is synthesized from the raw graph, and the generated manifold is rugged. Taking $\mathbf{S}_v^{(no)}$ as input, the subgraph link reconstruction module suppresses dissimilar neighbors by reducing the weights of their edges to generate smooth subgraph embedding $\mathbf{S}_v^{(sm)}$. Simply suppressing dissimilar nodes results in a straightforward manifold. To find a proper manifold, a subgraph generator generates proxy subgraph embedding $\mathbf{S}_v^{(pr)}$ to compromise the biases of $\mathbf{S}_v^{(no)}$ and $\mathbf{S}_v^{(sm)}$. Finally, with the help of the proxy subgraph embedding, a robust transferable priori can be obtained.

Noisy Subgraph Embedding

Without losing generality, GNNs learn the embedding for a node $v \in \mathcal{V}$ via a model f_{θ_g} parameterized by θ_g. We write the embedding as $\mathbf{H}_v = f_{\theta_g}(v)$. GNNs enforce the learned embedding of two connected nodes to become similar so that the homophily property can be captured. To realize the property, prevailing GNNs implicitly optimize the following term (Zhu et al. 2021),

$$\arg\min_{\theta_g} tr(\mathbf{H}^T\mathbf{L}\mathbf{H}), \tag{1}$$

where $\mathbf{L}$ is the normalized symmetric positive semi-definite graph Laplacian matrix. Eq. (1) can indicate the smoothness $S(\mathcal{M})$ of the manifold, because minimizing Eq. (1) is equivalent to smooth the manifold $\mathcal{M}$ (Chapelle, Scholkopf, and Zien 2009),

$$\begin{aligned} S(\mathcal{M}) = tr(\mathbf{H}^T\mathbf{L}\mathbf{H}) &= \int_v f_{\theta_g}(v)\Delta_{\mathcal{M}} f_{\theta_g}(v) d\mathcal{P}(\mathcal{V}) \\ &= \int_v ||\nabla_{\mathcal{M}} f_{\theta_g}(v)||^2 d\mathcal{P}(\mathcal{V}), \end{aligned} \tag{2}$$

where $\Delta_{\mathcal{M}}$ is the weighted Laplace-Beltrami operator associated with the marginal probability $\mathcal{P}(\mathcal{V})$. Due to the contaminated embedding caused by structural noise, the smoothness can be impaired, as shown in Fig. 1 (d) and Fig. 1 (e).

We assume the nodes are more similar to their original neighbors than the neighbors connected by the noisy edges. Based on this assumption, we get the noisy subgraph embedding $\mathbf{S}_v^{(no)}$ to represent the embedding of node v to dilute the noisy information to some extent,

$$\mathbf{S}_v^{(no)} = Readout(\mathbf{H}_u | u \in \Omega(v) \cup v) = \frac{\sum_{u \in \Omega(v) \cup v} \mathbf{H}_u}{|\Omega(v) \cup v|}, \tag{3}$$

where $\Omega(v)$ is the neighbors of node v. Because the noisy edges still exist and propagate noise information, there is contaminated information in the noisy subgraph embedding. Next, we will introduce a subgraph link reconstruction module to reduce the effects of noisy edges to refine the embedding.

Smooth Subgraph Embedding

The subgraph link reconstruction module aims to generate smooth subgraph embedding by reducing the weights of edges connecting dissimilar nodes. To suppress the effect of dissimilar nodes, we use an information theory mechanism (Ying et al. 2019) to select a subset node embedding $\widetilde{\mathbf{H}}_v$ that is highly related to the node v, where $\widetilde{\mathbf{H}}_v \subset \mathbf{H}$. The subset $\widetilde{\mathbf{H}}_v$ can be randomly sampled from $\mathbf{H}$ by maximizing mutual information (MI),

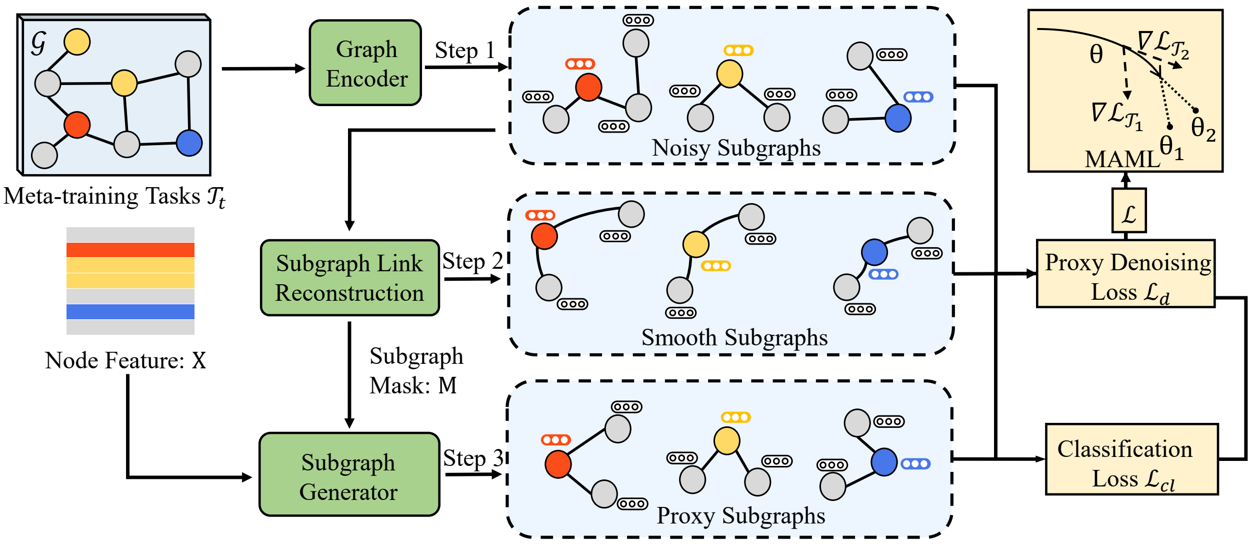

$\mathcal{G}$
Meta-training Tasks $\mathcal{T}_t$
Node Feature: X
Graph Encoder
Step 1
Noisy Subgraphs
Subgraph Link Reconstruction
Step 2
Smooth Subgraphs
Subgraph Mask: M
Subgraph Generator
Step 3
Proxy Subgraphs
θ
$\nabla\mathcal{L}_{\mathcal{T}_2}$
$\nabla\mathcal{L}_{\mathcal{T}_1}$
θ$_2$
MAML θ$_1$
$\mathcal{L}$
Proxy Denoising Loss $\mathcal{L}_d$
Classification Loss $\mathcal{L}_{cl}$

Figure 3: The pipeline of the proposed Pro-MC: Step 1. Noisy subgraph embedding is extracted from the raw graph containing structural noise. Step 2. Smooth subgraph embedding is obtained by suppressing dissimilar nodes. Step 3. A subgraph generator generates the proxy subgraph embedding to compromise the biases of noisy subgraph embedding and smooth subgraph embedding. With the help of the proxy subgraph embedding, a robust transferable prior is obtained.

$$E_{\widetilde{\mathbf{H}}_v \subset \mathbf{H}}\left[MI(\mathbf{H}_v, \widetilde{\mathbf{H}}_v)\right] = E[H(\mathbf{H}_v)] - E[H(\mathbf{H}_v|\widetilde{\mathbf{H}}_v)], \tag{4}$$

where $H(\cdot)$ is the entropy term. In this way, the change of the correlation between the selected embedding $\widetilde{\mathbf{H}}_v$ and the embedding $\mathbf{H}_v$ can be measured. When the GNN is trained and fixed in an epoch, $\mathbf{H}_v$ is constant. Therefore, we just need to minimize the upper bound of the second term in Eq. (4) by applying Jensen's inequality with the convexity assumption,

$$\min_{\widetilde{\mathbf{H}}_v \subset \mathbf{H}} H\left(\mathbf{H}_v | E[\widetilde{\mathbf{H}}_v]\right). \tag{5}$$

To tractably estimate $E[\widetilde{\mathbf{H}}_v]$, the probability of forming the subgraph $\widetilde{\mathbf{H}}_v$ with different embedding sampled from $\mathbf{H}$ can be decomposed into a multivariate Bernoulli distribution, $P(\widetilde{\mathbf{H}}_v) = \prod_{\mathbf{H}_u \in \mathbf{H}} P(\mathbf{H}_u)$, where $P(\mathbf{H}_u)$ means the probability of whether the embedding $\mathbf{H}_u$ exists in $\widetilde{\mathbf{H}}_v$. We can estimate $P(\mathbf{H}_u)$ by masking $\mathbf{H}$ with a mask vector $\mathbf{M}_v \in \mathbb{R}^{|\mathbf{V}|\times 1}$ in which each entry $\mathbf{M}_v[u]$ represents the probability of existence of $\mathbf{H}_u$ existing in $\widetilde{\mathbf{H}}$. By masking, the conditional entropy in Eq. (5) can be replaced with

$$\min_{\mathbf{M}_v} H\left(\mathbf{H}_v | \widetilde{\mathbf{H}}_v = \sigma(\mathbf{M}_v) \odot \mathbf{H}\right), \tag{6}$$

where $\odot$ denotes element-wise multiplication, and $\sigma(\cdot)$ denotes the sigmoid function that maps the mask entry to $[0, 1]$. If the entry u, i.e., $\sigma(\mathbf{M}_v)[u]$ is lower than a threshold t_m ($t_m = 0.2$), we can remove $\mathbf{H}_u$ from $\widetilde{\mathbf{H}}_v$.

To optimize Eq. (6), we can approximate it with the Kullback–Leibler divergence between the selected embedding $\widetilde{\mathbf{H}}_v$ and the embedding $\mathbf{H}_v$, so that $\mathbf{M}_v$ can be optimized by a few steps of gradient descent,

$$\min_{\mathbf{M}_v} KL(\mathbf{H}_v || \widetilde{\mathbf{H}}_v) = \sum_k \mathbf{H}_{v,k} \log \frac{\mathbf{H}_{v,k}}{\sum_{\mathbf{H}_u \in \widetilde{\mathbf{H}}_v} \mathbf{H}_{u,k}}, \tag{7}$$

where $\mathbf{H}_{u,k}$ is the k-th element of $\mathbf{H}_u$. Each entry in $\mathbf{M}_v$ indicates the degree of correlation of the corresponding node to node v. Therefore, we regard each entry in $\sigma(\mathbf{M}_v)$ as the weight of the node to form the smooth subgraph embedding $\mathbf{S}_v^{sm}$,

$$\mathbf{S}_v^{(sm)} = Readout(\widetilde{\mathbf{H}}_v) = \frac{\sum_{\mathbf{H}_u \in \widetilde{\mathbf{H}}_v} \mathbf{H}_u}{\sum_u \sigma\left(\mathbf{M}_v\right)[u]}. \tag{8}$$

Suppressing dissimilar nodes will lose the inherent pattern of a class and get a straightforward manifold, while the manifold of the noisy subgraph embedding is rugged. To address the issue, we generate the proxy subgraph to compromise the biases of the noisy subgraph embedding $\mathbf{S}^{(no)}$ and the smooth subgraph embedding $\mathbf{S}^{(sm)}$.

Proxy Subgraph Embedding

To compromise the biases of $\mathbf{S}_v^{(no)}$ and $\mathbf{S}_v^{(sm)}$, we propose a subgraph generator to generate proxy subgraph $\mathbf{S}_v^{(pr)}$. The subgraph generator f_{θ_p} takes the subgraph mask $\sigma(\mathbf{M}_v)$ and raw nodes' features of $\Omega(v) \cup v$ as inputs, and the generated proxy subgraph is denoted as,

$$\mathbf{S}_v^{(pr)} = f_{\theta_p}\left(\frac{\sum_{u \in \Omega(v) \cup v} \sigma(\mathbf{M}_v)[u] \odot \mathbf{X}_u}{\sum_{u \in \Omega(v) \cup v} \sigma\left(\mathbf{M}_v\right)[u]}\right). \tag{9}$$

Because optimal transport distance provides a weaker topology which is important for data residing on a low dimensional manifold (Canas and Rosasco 2012), we minimize the optimal transport distance to compromise the discrepancy between the noisy subgraph distribution $\mu_{\mathbf{S}_v^{(no)}}$ and the proxy subgraph distribution $\mu_{\mathbf{S}_v^{(pr)}}$, and the proxy subgraph distribution $\mu_{\mathbf{S}_v^{(pr)}}$ and the smooth subgraph distribution $\mu_{\mathbf{S}_v^{(sm)}}$,

$$\begin{aligned}\min \mathcal{D}(\mu_{\mathbf{S}_v^{(pr)}}, \mu_{\mathbf{S}_v^{(no)}}) + \mathcal{D}(\mu_{\mathbf{S}_v^{(pr)}}, \mu_{\mathbf{S}_v^{(sm)}})\\ = \inf_{\gamma \in \Pi(\mu_{\mathbf{S}_v^{(pr)}}, \mu_{\mathbf{S}_v^{(no)}})} \mathbb{E}_{(\mathbf{S}_v^{(pr)}, \mathbf{S}_v^{(no)}) \sim \gamma} ||\mathbf{S}_v^{(pr)} - \mathbf{S}_v^{(no)}||\\ + \inf_{\gamma \in \Pi(\mu_{\mathbf{S}_v^{(pr)}}, \mu_{\mathbf{S}_v^{(sm)}})} \mathbb{E}_{(\mathbf{S}_v^{(pr)}, \mathbf{S}_v^{(sm)}) \sim \gamma} ||\mathbf{S}_v^{(pr)} - \mathbf{S}_v^{(sm)}||.\end{aligned} \tag{10}$$

The infimum operation in Eq. (10) is highly intractable. According to the Kantorovich-Rubinstein duality (Avraham, Zuo, and Drummond 2019), we can learn a classifier $f_w \in \mathcal{F}_w$ parameterized by θ_w, where $\mathcal{F}_w$ is function family. With the classifier f_w, we can minimize the following proxy denoising loss,

$$\begin{aligned}\min_{\theta_w, \theta_p} \mathcal{L}_d = \mathbb{E}_{\mathbf{S}_v^{(pr)} \sim \mu_{\mathbf{S}_v^{(pr)}}} [f_w(\mathbf{S}_v^{(pr)})] - \mathbb{E}_{\mathbf{S}_v^{(no)} \sim \mu_{\mathbf{S}_v^{(no)}}} [f_w(\mathbf{S}_v^{(no)})]\\ + \mathbb{E}_{\mathbf{S}_v^{(pr)} \sim \mu_{\mathbf{S}_v^{(pr)}}} [f_w(\mathbf{S}_v^{(pr)})] - \mathbb{E}_{\mathbf{S}_v^{(sm)} \sim \mu_{\mathbf{S}_v^{(sm)}}} [f_w(\mathbf{S}_v^{(sm)})],\end{aligned} \tag{11}$$

where a softmax activation function is performed on f_w. Let $\mathbf{S}^{(pr)} = f_w(\mathbf{S}^{(pr)})$ be the final output. For predicting the probability of each class for each node, the cross-entropy loss of $\mathbf{S}^{(pr)}$ over the support nodes S_t is minimized,

$$\min_{\theta_w, \theta_p} \mathcal{L}_{cl} = -\sum_{v \in S_t} \sum_{c} \mathbf{Y}_{vc} \ln(\mathbf{S}_{vc}^{(pr)}), \tag{12}$$

where $\mathbf{Y}$ is the corresponding label indicator matrix, and $\mathbf{S}_{vc}^{(pr)}$ is the probability of node v belonging to class c.

Meta Objective Function

The episodic meta-learning framework is incorporated with our proposed Pro-MC. The meta objective function is a bilevel optimization: Firstly, when θ_g is fixed, $\sigma(\mathbf{M}_v)$ and $\mathbf{S}_v^{(sm)}$ are obtained by Eq. (7) and Eq. (8), respectively. Based on the $\sigma(\mathbf{M}_v)$ and $\mathbf{S}_v^{(sm)}$, $\mathbf{S}_v^{(pr)}$ is generated by minimizing the summation of $\mathcal{L}_{cl}$ and $\mathcal{L}_d$,

$$\min_{\theta_p, \theta_w} \mathcal{L} = \mathcal{L}_{cl} + \lambda_d \mathcal{L}_d, \tag{13}$$

Then, θ_p, θ_w are fixed to conduct the optimization of θ_g. $\mathbf{S}_v^{(no)}$ is generated by minimizing the following loss function,

$$\min_{\theta_g} \mathcal{L}_{no} = -\sum_{v \in S_t} \sum_{c} \mathbf{Y}_{vc} \ln(\mathbf{S}_{vc}^{(no)}), \tag{14}$$

where a softmax activation function is performed on $\mathbf{S}^{(no)}$. Besides, the squared Euclidean distance $\mathcal{L}_p$ between $\mathbf{S}^{(no)}$ and $\mathbf{S}^{(sm)}$, and $\mathbf{S}^{(no)}$ and $\mathbf{S}^{(pr)}$ is minimized. Therefore, the θ_g can be optimized by $\mathcal{L}_g = \mathcal{L}_{no} + \mathcal{L}_p$.

The details of the meta-learning processes can be seen in Algorithm 1 of the supplementary material.

Theoretical Analysis

We provide the generalization bound of the proposed Pro-MC to illustrate the effectiveness of using proxy subgraph embedding to calibrate the smoothness of the manifold. The proof is given in the supplementary material.

Theorem 1 *(Generalization bound of the proposed Pro-MC). For the m-way k-shot node classification task, assume that $\mathcal{F}$ is a function class consisting of functions with range $[a, b]$. $\mathbf{S}^{(pr)}$, $\mathbf{S}^{(sm)}$, and $\mathbf{S}^{(no)}$ are drawn from domain $\mathcal{Z}^{(pr)}$, $\mathcal{Z}^{(sm)}$, and $\mathcal{Z}^{(no)}$, respectively. Assume that the proposed Pro-MC algorithm $\Theta = \{\theta_g, \theta_p, \theta_w\}$ has n meta-train tasks $\{\mathcal{T}_t\}_{t=1}^{n}$ that are drawn from any task distribution τ, if $\Theta \in \mathcal{F}$ has uniform stability β w.r.t a loss function $\mathcal{L}$ bounded M. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the expected generalization bound $\mathcal{R}$ is given by*

$$\begin{aligned}\mathcal{R}(\Theta, \tau) \leq \hat{\mathcal{R}}(\Theta, \mathbf{S}^{(pr)})\\ +2\beta + (4n\beta + M)\sqrt{\frac{\ln(1/\delta)}{2n}}\\ +D_{\mathcal{F}}(\mathcal{Z}^{(sm)}, \mathcal{Z}^{(no)})\\ +\sqrt{\frac{(b-a)^2 \ln(4/\delta)}{2kmn}},\end{aligned} \tag{15}$$

where $\hat{\mathcal{R}}$ is the empirical error and $D_{\mathcal{F}}(\mathcal{Z}^{(sm)}, \mathcal{Z}^{(no)})$ measures the gap between the domain $\mathcal{Z}^{(sm)}$ and the domain $\mathcal{Z}^{(no)}$. In this work, we minimize the empirical error $\hat{\mathcal{R}}$ by Eq. (12), and implicitly minimize $D_{\mathcal{F}}$ by Eq. (11), so that the expected generalization bound can be minimized.

EXPERIMENTS

Benchmark Datasets

We evaluate our proposed method on three real-world graph datasets: Cora (Sen et al. 2008), Citeseer (Sen et al. 2008), and Amazon Photo (McAuley et al. 2015). Cora dataset and Citeseer dataset are citation networks for node classification. Amazon Photo is the Amazon co-purchase graph. The details of the three graph datasets including the number of nodes, edges, features, and classes, are listed in Table 1 of the supplementary material.

Experimental Settings

To endow existing robust GNNs with the ability to handle long-tail distribution, we incorporate the model-agnostic meta-learning framework MAML into robust GNNs for comparison. We select the following algorithms as the baselines,

1. Meta-GCN (Zhou et al. 2019b): Meta-GCN incorporates the MAML paradigm into GNNs, enabling tackling the few-shot node classification tasks.
2. GCN-SVD (Wu et al. 2019): GCN-SVD is proposed to vaccinate GCNs with the low-rank approximation adjacency matrix of the perturbed graph.
3. GCN-Jaccard (Wu et al. 2019): GCN-Jaccard calculates the Jaccard similarity of pairs of nodes and eliminates edges between nodes with low similarity.

Table 1: Few-shot node classification accuracy (Mean±Std) under various structural noises (%)

	Dataset	Perturbation	Meta-GCN	Meta-GCN-SVD	Meta-GCN-Jaccard	Meta-RGCN	Meta-Pro-GNN	Meta-RS-GNN	Pro-MC
2-way-1-shot	Cora	Random Attack	61.19±1.02	64.49±2.30	61.60±1.65	66.81±1.27	66.42±1.50	66.83±2.20	**67.75±2.16**
		Non-Targeted Attack	60.08±0.98	60.58±1.44	61.34±1.34	64.54±1.41	60.27±2.73	64.50±1.64	**66.85±1.73**
		Targeted Attack	52.08±1.57	53.19±1.87	54.31±1.18	54.78±1.39	54.65±1.07	53.51±1.86	**55.20±2.02**
	Citeseer	Random Attack	55.40±1.90	57.02±1.73	56.10±1.85	57.35±3.04	56.15±1.42	56.42±2.32	**58.21±2.55**
		Non-Targeted Attack	52.93±1.52	54.74±2.83	55.25±2.66	**57.44±2.96**	54.97±1.22	55.53±1.50	54.59±1.67
		Targeted Attack	50.25±2.33	53.15±1.40	51.57±1.19	52.81±1.97	51.49±1.70	54.11±2.30	**54.25±2.10**
	Amazon Photo	Random Attack	57.22±2.34	60.19±1.52	59.97±1.64	60.12±1.66	58.54±1.98	59.03±1.62	**60.71±1.34**
		Non-Targeted Attack	51.26±1.79	56.42±1.47	55.37±1.52	56.11±1.70	53.97±1.81	52.08±1.18	**57.42±1.08**
		Targeted Attack	50.72±2.05	55.19±1.76	55.01±1.84	55.22±1.20	51.81±2.30	53.29±1.21	**56.89±0.66**
2-way-5-shot	Cora	Random Attack	66.81±2.09	67.72±2.70	68.79±2.65	68.09±1.29	69.48±1.63	67.95±2.55	**71.53±1.59**
		Non-Targeted Attack	64.35±1.80	63.79±3.16	68.44±1.34	67.10±3.67	66.20±1.86	69.10±1.58	**69.83±1.41**
		Targeted Attack	58.86±1.64	60.60±2.31	60.19±3.13	61.98±1.88	60.37±2.24	59.98±2.75	**63.37±1.40**
	Citeseer	Random Attack	62.54±2.12	64.30±1.90	65.01±1.65	66.54±1.94	63.61±2.29	68.22±2.59	**69.78±1.68**
		Non-Targeted Attack	60.28±2.02	61.48±2.62	66.79±2.46	66.09±1.07	60.45±1.43	67.17±1.48	**67.68±1.54**
		Targeted Attack	59.22±1.96	60.33±1.82	65.78±2.07	63.78±1.33	59.84±1.74	66.79±2.15	**67.33±2.16**
	Amazon Photo	Random Attack	63.38±1.36	**66.57±1.38**	65.37±1.95	65.58±1.70	64.98±2.24	65.20±1.48	66.45±2.35
		Non-Targeted Attack	59.05±2.20	63.49±1.49	63.83±1.01	63.74±1.14	62.91±1.99	62.08±1.16	**64.44±1.88**
		Targeted Attack	57.90±1.97	58.43±2.18	59.37±2.44	**60.63±1.11**	58.80±1.53	59.13±1.28	59.46±1.40

4. RGCN (Zhu et al. 2019): RGCN aims to defend against noisy edges by introducing Gaussian distributions to absorb the negative effects of noisy edges.
5. Pro-GNN (Jin et al. 2020): Pro-GNN learns a robust GNN model by imposing the low-rank constraint and smoothing the embedding.
6. RS-GNN (Dai et al. 2022): RS-GNN eliminates noisy edges by imposing the low-rank constraint and smoothing the embedding to handle both noisy graphs and label sparsity issues.

The compared robust GNNs are renamed as ”Meta-” with their original name in the following part.

To show the effectiveness of the compared algorithms in resisting various structural noises, three types of structural noises are imposed on the datasets:

1. Random attack: Random attack randomly removes edges connecting nodes with the same label and then adds noisy edges connecting a node with a different label. The perturbation rate (i.e., for each node, the ratio of edges be flipped) is set to 20%.
2. Non-targeted attack: Metattack (Zügner and Günnemann 2018) is adopted to poison the graph globally by changing 10% edges.
3. Targeted attack: Targeted attack focuses on misclassifying specific target nodes. We adopt nettack (Zügner, Akbarnejad, and Günnemann 2018) as the targeted attack method to perturb 20% nodes.

The learning rate, dropout rate, and λ_d are set to 0.01, 0.5, and 0.5, respectively. We employ GraphSAGE (Hamilton, Ying, and Leskovec 2017) as the graph encoder θ_g. The subgraph generator θ_p includes an encoder with two FC layers and a decoder with two FC layers. The experiments in the paper are run 5 times independently, and we report the average classification accuracy over these repetitions.

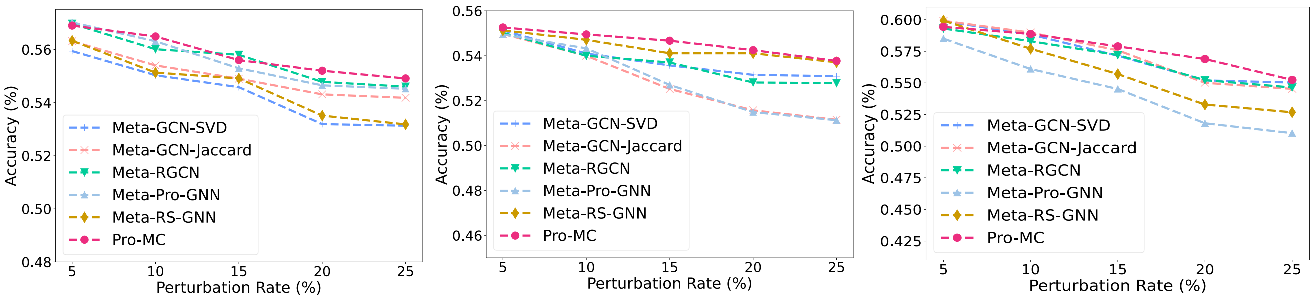
Comparisons with Baselines

Table 1 shows the performance comparison of few-shot node classification with three types of structural noises on three datasets. The best performance is highlighted in bold. The findings are listed as follows.

Firstly, experimental results show that Pro-MC outperforms other methods under different types of structural noises in 15 out of 18 cases. Specifically, compared with vanilla Meta-GCN, our proposed Pro-MC improves accuracy by 2%∼7%, demonstrating the ability to handle both long-tail distribution and structural noise.

Secondly, our proposed Pro-MC can achieve better results than low-rank constraint-based methods such as Meta-GCN-SVD, Meta-Pro-GNN, and Meta-RS-GNN. Low-rank constraint-based methods focus on refining the graph topology, while often ignoring node features that can provide complementary information for purifying the graphs. Meta-Pro-GNN and Meta-RS-GNN smooth features by minimizing Eq. (1), and this way can bring limited improvement as GNNs have the intrinsic ability to minimize Eq. (1) (Zhu et al. 2021).

Lastly, Pro-MC steadily performs better than other baselines under various types of structural noises. The targeted attack adds noisy edges on targeted nodes, which makes the local manifold of the attacked nodes damaged. Non-targeted attack generates poisoning attacks based on meta-learning to globally perturb the graph, which skews the global manifold. For both two attacks, our proposed method can still generate a proper manifold. We can conclude that Pro-MC is able to resist various types of structural noises, which is the desired property in practice.



(a) Cora

(b) Citeseer

(c) Amazon Photo

Figure 4: Node classification accuracy of different algorithms under the targeted attack with varying ratios of noises.

Impacts of Perturbation Rates

We also investigate the performance of compared algorithms under the targeted attack with varying ratios from 5% to 25%. The accuracy results on the Cora dataset, Citeseer dataset, and Amazon Photo dataset are plotted in Fig. 4. The experimental results indicate that Pro-MC can consistently achieve promising results and effectively resists structural noises. Together with the above experimental findings, we can conclude that Pro-MC achieves comparable performance under different degrees of structural noises, showing its advantage.

Ablation Study

Pro-MC has three types of subgraph-level embedding, noisy subgraph embedding, smooth subgraph embedding, and proxy subgraph embedding. To examine the behaviors of different types of embedding, we evaluate the accuracy by using noisy subgraph embedding or smooth subgraph embedding for classification. The corresponding ablation algorithms are renamed as Pro-MC ($\mathbf{S}^{(no)}$) and Pro-MC ($\mathbf{S}^{(sm)}$), respectively.

The experimental results are shown in Table 2 of the supplementary material. It can be obviously found Pro-MC can achieve a better performance than Pro-MC ($\mathbf{S}^{(no)}$) and Pro-MC ($\mathbf{S}^{(sm)}$), which verifies the effectiveness of the proxy subgraph embedding. Therefore, compromising the biases of $\mathbf{S}^{(no)}$ and $\mathbf{S}^{(sm)}$ is helpful in generating a proper manifold to improve the accuracy. We also perform ablation learning on $\mathcal{L}_p$, and Pro-MC ($-\mathcal{L}_p$) demonstrates the effectiveness of $\mathcal{L}_p$. The smoothness $S(\mathcal{M})$ of the generated manifold of Pro-MC ($\mathbf{S}^{(no)}$), Pro-MC ($\mathbf{S}^{(sm)}$), and Pro-MC is plotted in Fig. 2 of the supplementary material. We can see that Pro-MC can generate the manifold with moderate smoothness.

To verify the effectiveness of $\mathcal{L}_d$, we perform ablation learning on this part to evaluate the contribution of $\mathcal{L}_d$ in the Pro-MC. The corresponding ablation algorithm is renamed as Pro-MC ($-\mathcal{L}_d$). From Table 2 of the supplementary material, we can see that $\mathcal{L}_d$ has a positive impact on the accuracy of few-shot classification for the proposed Pro-MC.

We also analyze the effectiveness of using subgraph-level embedding. As can be seen from Table 2 of the supplementary material, we can see that compared with Pro-MC (NE) (where $\mathbf{S}_v^{(no)}$ and $\mathbf{S}_v^{(sm)}$ are replaced with the corresponding embedding of the node v, respectively), the improvement of subgraph-level embedding is significant.

Parameter Sensitivity

The sensitivity results regarding the threshold t_m in $\mathbf{M}_v$ and λ_d on Cora, Citeseer, and Amazon Photo datasets are shown in Fig. 1 (see supplementary material). We can conclude that different datasets have different optimal values of the threshold values, which need to be set carefully. When λ_d increases from the corresponding minimum value to a maximum value, the classification accuracy firstly improves and then drops down. The choice of parameter λ_d also depends on the specific dataset.

CONCLUSION

In this work, we advanced graph meta-learning that is robust against structural noise. We found that the structural noise induces the manifold to become rugged, thus impairing the transferable priori of the meta-learner. Taking this cue, we propose Pro-MC to learn a proper manifold that can resist structural noise. Specifically, we generate proxy subgraph embedding to compromise the biases of smooth subgraph embedding and noisy subgraph embedding to prevent the manifold from being rugged and straightforward. In this way, the smoothness of the manifold can be calibrated. With the manifold with proper smoothness, the meta-learner can learn a generalizable priori. We also provide a theoretical generalized error of the proposed Pro-MC. The experiments on canonical few-shot node classification tasks with various structural noises exhibit substantial improvements.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant U21A20512, Grant 62276222, and Grant 61876162, and in part by the Research Grants Council of the Hong Kong SAR under Grant PolyU11211521, and in part by PolyU Internal Research Fund under Grant P0042687 and PolyU Start-up Fund under Grant P0035763.

References

Avraham, G.; Zuo, Y.; and Drummond, T. 2019. Parallel Optimal Transport GAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Canas, G.; and Rosasco, L. 2012. Learning Probability Measures with respect to Optimal Transport Metrics. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Chapelle, O.; Scholkopf, B.; and Zien, A., Eds. 2009. Semi-Supervised Learning. *IEEE Transactions on Neural Networks*, 20(3): 542–542.

Chauhan, J.; Nathani, D.; and Kaul, M. 2020. Few-shot Learning on Graphs via Super-classes based on Graph Spectral Measures. In *International Conference on Learning Representations*.

Dai, E.; Jin, W.; Liu, H.; and Wang, S. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 181–191.

Ding, K.; Wang, J.; Caverlee, J.; and Liu, H. 2022. Meta Propagation Networks for Graph Few-shot Semi-supervised Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 169–177.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Huang, K.; and Zitnik, M. 2020. Graph Meta Learning via Local Subgraphs. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 5862–5874. Curran Associates, Inc.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2020*, 66–74. Association for Computing Machinery.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Lan, L.; Wang, P.; Du, X.; Song, K.; Tao, J.; and Guan, X. 2020. Node Classification on Graphs with Few-Shot Novel Labels via Meta Transformed Network Embedding. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 16520–16531. Curran Associates, Inc.

Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.; and Yang, Y. 2019. Learning to propagate labels: Transductive propagation network for few-shot learning. In *7th International Conference on Learning Representations, ICLR 2019.*

Liu, Z.; Fang, Y.; Liu, C.; and Hoi, S. C. 2021. Relative and absolute location embedding for few-shot node classification on graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4267–4275.

Liu, Z.; Zhang, W.; Fang, Y.; Zhang, X.; and Hoi, S. C. 2020. Towards locality-aware meta-learning of tail node embeddings on networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 975–984.

Luo, D.; Cheng, W.; Yu, W.; Zong, B.; Ni, J.; Chen, H.; and Zhang, X. 2021. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM international conference on web search and data mining*, 779–787.

Ma, N.; Bu, J.; Yang, J.; Zhang, Z.; Yao, C.; Yu, Z.; Zhou, S.; and Yan, X. 2020. Adaptive-Step Graph Meta-Learner for Few-Shot Graph Classification. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, CIKM '20, 1055–1064. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368599.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Tang, X.; Li, Y.; Sun, Y.; Yao, H.; Mitra, P.; and Wang, S. 2020. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th international conference on web search and data mining*, 600–608.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019. Adversarial examples for graph data: deep insights into attack and defense. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4816–4823.

Xie, T.; and Grossman, J. C. 2018. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.*, 120: 145301.

Yao, H.; Zhang, C.; Wei, Y.; Jiang, M.; Wang, S.; Huang, J.; Chawla, N.; and Li, Z. 2020. Graph few-shot learning via knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6656–6663.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Zhang, C.; Yao, H.; Huang, C.; Jiang, M.; Li, Z.; and Chawla, N. V. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3041–3048.

Zhao, F.; Wang, D.; and Xiang, X. 2021. Multi-Initialization Graph Meta-Learning for Node Classification. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ICMR '21, 402–410. New York, NY, USA: Association for Computing Machinery. ISBN 9781450384636.

Zhao, J.; Yang, Y.; Lin, X.; Yang, J.; and He, L. 2021. Looking wider for better adaptive representation in few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10981–10989.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019a. Meta-GNN: On Few-Shot Node Classification in Graph Meta-Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, 2357–2360. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369763.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019b. Meta-gnn: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2357–2360.

Zhou, K.; Zha, H.; and Song, L. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, 641–649. PMLR.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.

Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021. Interpreting and unifying graph neural networks with an optimization framework. In *Proceedings of the Web Conference 2021*, 1215–1226.

Zhu, Y.; Wang, X.; Zhong, E.; Liu, N.; Li, H.; and Yang, Q. 2012. Discovering spammers in social networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 171–177.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *SIGKDD*, 2847–2856.

Zügner, D.; and Günnemann, S. 2018. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations*.