Graph Few-shot Learning with Attribute Matching

Ning Wang
wangning059@stu.xjtu.edu.cn
School of Computer Science and Technology, Xi'an Jiaotong University

Minnan Luo*
minnluo@xjtu.edu.cn
School of Computer Science and Technology, Xi'an Jiaotong University

Kaize Ding
kaize.ding@asu.edu
Computer Science and Engineering, Arizona State University

Lingling Zhang
zhanglling@xjtu.edu.cn
School of Computer Science and Technology, Xi'an Jiaotong University

Jundong Li
jundong@virginia.edu
Dept. of Electrical and Computer Engineering, Dept. of Computer Science, School of Data Science, University of Virginia

Qinghua Zheng
qhzheng@xjtu.edu.cn
Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University

ABSTRACT

Due to the expensive cost of data annotation, few-shot learning has attracted increasing research interests in recent years. Various meta-learning approaches have been proposed to tackle this problem and have become the de facto practice. However, most of the existing approaches along this line mainly focus on image and text data in the Euclidean domain. However, in many real-world scenarios, a vast amount of data can be represented as attributed networks defined in the non-Euclidean domain, and the few-shot learning studies in such structured data have largely remained nascent. Although some recent studies have tried to combine meta-learning with graph neural networks to enable few-shot learning on attributed networks, they fail to account for the unique properties of attributed networks when creating diverse tasks in the meta-training phase—the feature distributions of different tasks could be quite different as instances (i.e., nodes) do not follow the data i.i.d. assumption on attributed networks. Hence, it may inevitably result in suboptimal performance in the meta-testing phase. To tackle the aforementioned problem, we propose a novel graph meta-learning framework–Attribute Matching Meta-learning Graph Neural Networks (AMM-GNN). Specifically, the proposed AMM-GNN leverages an attribute-level attention mechanism to capture the distinct information of each task and thus learns more effective transferable knowledge for meta-learning. We conduct extensive experiments on real-world datasets under a wide range of settings and the experimental results demonstrate the effectiveness of the proposed AMM-GNN framework.

CCS CONCEPTS

• **Computing methodologies → Transfer learning**; **Neural networks**.

KEYWORDS

Few-shot learning; Node classification; Graph neural networks

ACM Reference Format:
Ning Wang, Minnan Luo, Kaize Ding, Lingling Zhang, Jundong Li, and Qinghua Zheng. 2020. Graph Few-shot Learning with Attribute Matching. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411923

1 INTRODUCTION

With the rapid development of modern information systems, we are experiencing a rocketing growth of networked applications. A vast majority of these systems can be abstracted and represented as the so-called *attributed networks*, examples include citation networks, social networks, biological networks, and critical infrastructure networks, to name a few [16, 24, 36, 57]. For this type of data, node classification is considered as an essential task in previous research as it is used in a wide spectrum of applications. Recently, various types of graph neural networks (GNNs) [52] have been proposed to tackle the problem by learning high-level feature representations of nodes and addressing the classification task in an end-to-end manner. Despite their empirical success, GNNs [6, 21, 27, 47] normally require a large amount of training data (i.e., labeled nodes) to achieve a satisfactory classification performance. When the labeled training data is limited, these models will inevitably face a dilemma of over-fitting [22, 44] such that the developed GNNs cannot be well generalized to unlabeled nodes. In fact, in many real-world scenarios, labeled data is often costly to obtain, requiring a lot of human efforts. For example, in protein-protein interaction (PPI) networks, providing functions for proteins in the interactome is a time- and labor-consuming task [15] even for experienced experts. As such, we may only have a limited amount of labeled genes for a certain function in this context.

Considering the high cost of data annotation [48, 53, 55, 62], few-shot learning (FSL) [12, 13] has aroused wide interests in the research community. The primary goal of FSL is to learn an effective model to classify data with extremely few labels. It is contrary to the standard practice of supervised learning that uses a large amount of labeled data to feed a learning model [50]. To tackle the FSL problem, many meta-learning frameworks [8, 14, 46, 56] have been proposed in recent years and have become the de facto practice. In FSL, a good

*Corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411923

meta-learner is often trained on a variety of tasks with few labeled data, and then it transfers the accumulated knowledge to a new task that has never been encountered before—these two stages are often called *meta-training* and *meta-testing* [14]. However, a vast majority of existing meta-learning approaches for FSL are predominately focused on image data [18, 41, 42] and text data [25, 26, 38] while the FSL studies for the graph-structured data are rather limited.

In this paper, we focus on investigating the FSL problem on attributed networks. The problem, however, is much more challenging than performing FSL on image and text data. The major reason is that unlike image or text data that is often defined in the Euclidean domain, different data modalities (i.e., graph structure in the non-Euclidean domain and node attribute information in the Euclidean domain) are often coupled together on attributed networks with complex interactions. Existing research efforts are not equipped to characterize their inherent correlations for FSL. The recently developed Meta-GNN [59] attempts to fill this gap and address the few-shot node classification problem on attributed networks. Nonetheless, there is one major limitation of Meta-GNN—it simply combines the prevalent meta-learning method MAML with GNNs while fails to account for the unique properties of attributed networks when creating diverse meta-training tasks. Unlike image or text data, different data instances (i.e., nodes) on attributed networks often do not follow the data i.i.d. assumption, thus the feature distributions of different sampled tasks could be quite different. As Meta-GNN treats different sampled tasks equally in the meta-training phase, it may inevitably yield suboptimal classification performance in the meta-testing phase.

To tackle the aforementioned problem, in this paper, we propose a novel meta-learning framework—Attribute Matching Meta-learning Graph Neural Networks (AMM-GNN), which can effectively perform few-shot learning on attributed networks. Our proposed AMM-GNN framework is based on the Model-Agnostic Meta-Learning (MAML) framework [14], which learns a good model parameter initialization for different tasks and is compatible with any optimized learning model through gradient descent. Specifically, in AMM-GNN, we introduce a novel attribute-level attention mechanism to better capture the unique properties of each sampled meta-learning task. Thus, the inherent feature distribution differences between different sampled tasks can be well characterized and the meta-learning model can better match the current task in learning more effective transferable knowledge. As a summary, the overall contributions of this paper can be summarized as follows:

- We study the important problem of few-shot learning on attributed networks, which has significant implications when the number of labeled nodes is very limited.
- We show the major limitation of Meta-GNN [59] in the context of FSL on graphs and propose a novel meta-learning framework AMM-GNN. Specifically, AMM-GNN relies on the attribute-level attention mechanism to characterize the feature distribution differences between different tasks and learns more meaningful transferable knowledge across tasks. Therefore, it can greatly facilitate the classification performance in the meta-testing stage.
- We conduct extensive experiments on a wide range of experimental settings and demonstrate the effectiveness and superiority of our proposed framework.

The rest of this paper is organized as follows. In Section 2, we review related work on few-shot learning and graph neural networks. We introduce the problem definition and the proposed few-shot learning framework AMM-GNN for node classification in Section 3 and Section 4, respectively. Empirical evaluations are presented in Section 5, and the conclusion are shown in Section 6.

2 RELATED WORK

In this section, we summarize some typical few-shot learning frameworks as well as related works on graph neural networks.

2.1 Few-shot Learning

The primary goal of few-shot learning is to learn a classification model that performs well for the data having seen only a few training examples. But in most cases, using a small number of samples to train the model will cause the over-fitting problem. As such, meta-learning has become the prevailing practice for the FSL problem. It uses a unique learning paradigm to divide the entire learning process into meta-training and meta-testing. More specifically, it guides the learning of new tasks in meta-testing using prior knowledge accumulated from meta-training. Given that, many sophisticated meta-learning methods have been proposed in recent years, and these efforts can be mainly categorized as: metric-based approaches, optimization-based approaches, and memory-based approaches. Metric-based approaches [28, 41, 42, 46] model the metric space of data instances and perform downstream tasks with the learned distance metric. For example, Matching Networks [46] build different encoders for both support sets and query sets and then use the similarity between instances in the query sets and support sets as the weight to get the classification result. Prototypical Networks [41] learn a mapping function in meta-training by taking the centers of support sets as prototypes, and then perform the K-nearest neighbors algorithm using the Euclidean distance. Optimization-based approaches [14, 37] use fine-tuning to realize few-shot learning and avoid over-fitting. For example, in [37], Meta-Learner interprets stochastic gradient descent update rules as a gated recursive model with trainable parameters, so as to learn the update rules of model parameters. MAML [14] seeks a proper parameter initialization by second-order gradient descent method so that the model can achieve better generalization performance after a few steps of gradient descent. Memory-based approaches use the memory mechanism to extract valuable knowledge acquired in the meta-training phase to assist meta-testing. For instance, CMN [61] uses the key-value memory network paradigm to obtain an optimal video representation in a larger space. MM-net [5] utilizes memory mechanism to employ the memory slots sequentially to predict the parameters of CNNs.

2.2 Graph Neural Networks

As graphs provide a general language to describe various complex systems, the analysis and mining of graph-structured data have raised great attention in different disciplines. However, the discrete nature of graph data also brings many computational challenges for existing learning algorithms. Among various existing learning algorithms for graph data, graph neural networks (GNNs) have achieved superior learning performance in diverse settings [4, 9, 10, 20, 27, 49], which extend the traditional convolution operations [29, 30] for grid data to the non-Euclidean domain [19, 39]. The seminal work of Graph Convolutional Networks (GCNs) [27] use the localized first-order approximation of spectral graph convolutions to learn node embedding representations for the semi-supervised node classification task. Many variants of GCNs have been proposed over the past few years [7, 32]. GraphSAGE [21] proposes different types of aggregation functions (e.g., mean, pooling, and LSTM aggregators) to obtain node embedding representation of each node from its neighbors, and the proposed inductive learning paradigm naturally generalizes to unseen nodes in the graph. Graph Attention Networks (GAT) [45] uses the self-attention mechanism [1] to assign different weights to neighbors of each node and the developed embedding approach is applicable for both transductive and inductive learning settings. Learnable Graph Convolutional Networks (LGCN) [17] transform graph data to grid-like structures and enable the use of regular convolutional operations by selecting a fixed number of neighbors for each node. Simple Graph Convolution (SGC) [51] is a simplified version of GCNs that reduces the computational complexity of GCNs by repeatedly eliminating non-linear activation operations between GCN layers and folding the resulting function into a linear transformation. A more comprehensive review of GNNs can be referred to [52, 58, 60].

However, the performance of most existing GNNs may suffer from a catastrophic decline when the number of labeled nodes is small. For this reason, several existing methods [3, 54, 59] utilize meta-learning to solve the FSL problem on graphs. Graph Few-shot Learning (GFL) [54] proposes to transfer knowledge accumulated from auxiliary graphs to improve the classification performance on the target graph. However, it is not always feasible to collect such auxiliary graphs for extracting graph-level knowledge in practice. Meta-Graph [3] is a few-shot link prediction model that learns the signature functions of graphs with node embeddings to assist the meta-learning model. Meta-GNN [59] is most similar to our method, which also studies the few-shot node classification problem. However, Meta-GNN does not consider the distinct feature distributions of different tasks, which may yield suboptimal FSL performance.

3 PROBLEM DEFINITION

In this section, we formally introduce the studied problem of few-shot node classification on attributed networks using the meta-learning paradigm. For convenience, some important notations and explanations are summarized in Table 1.

We assume an input attributed network can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the node set ($n = |\mathcal{V}|$); $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set between nodes in $\mathcal{V}$; $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ denotes the adjacency matrix of the underlying graph structure and each entry a_{ij} represents the state of connection between node v_i and v_j; $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n \times d}$ is the feature matrix of all nodes, where $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes a d-dimensional feature vector for node v_i.

With the definition of attributed networks, we now formally define the investigated problem of few-shot node classification on attributed networks. Generally speaking, given an attributed network $\mathcal{G}$ in which a subset of nodes are labeled, our goal is to learn an effective classification model that works well for unlabeled nodes whose ground truth class labels are seen only a few times in the data. The problem is also referred as few-shot learning.

Table 1: Important notations and explanations.

Notations	Explanations
$\mathcal{G}$	Attributed network
$\mathcal{V}, \mathcal{E}$	The node set and edge set
$\boldsymbol{A}, \boldsymbol{X}$	Graph adjacency matrix and node feature matrix
$\mathcal{T}_t$	The t-th task in meta-learning
$\mathcal{S}_t, \mathcal{Q}_t$	Support node set and query node set of $\mathcal{T}_t$
N	The number of classes (ways)
K	The number of labeled nodes in each class (shots)
$f(\cdot)$	Graph neural network classifier
$g(\cdot)$	The generator of matching vectors
$\boldsymbol{\theta}, \boldsymbol{\phi}$	Parameters of $f(\cdot)$ and $g(\cdot)$
$\boldsymbol{\alpha}, \boldsymbol{b}$	Matching vectors
β_0, β_1	Inner-loop learning rate, outer-loop learning rate

Currently, the meta-learning paradigm is often utilized to solve such few-shot learning problem [46]. The whole process of meta-learning can be divided into two parts: meta-training and meta-testing, referring to the process of learning the transferable knowledge and testing the model capability, respectively. To construct a meta-learning framework for few-shot node classification, the nodes in graph $\mathcal{G}$ are divided into two disjoint sets $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, which correspond to the node sets used in meta-training and meta-testing, respectively. In $\mathcal{D}_{train}$, we assume abundant labeled nodes for each class are available, but in $\mathcal{D}_{test}$ there are extremely few labeled nodes for each class. When the number of classes in $\mathcal{D}_{test}$ is N and the number of labeled nodes in each of these classes is K (K is often specified as a small number), the problem is often called as an N-way K-shot learning problem. And the final task in meta-testing is called $\mathcal{T}_{test}$.

In order to make the model capable of solving the few-shot learning problem, we need to sample multiple tasks from distribution $p(\mathcal{T})$ composed by the classes of nodes in $\mathcal{D}_{train}$. Any task $\mathcal{T}_t$ drawn from $p(\mathcal{T})$ has N different classes and follows the same paradigm as the task $\mathcal{T}_{test}$. For a particular sampled task $\mathcal{T}_t$, we randomly sample K nodes from each class to create the support set $\mathcal{S}_t$ and M nodes from each class to create the query set $\mathcal{Q}_t$. The nodes in $\mathcal{S}_t$ will perform as labeled nodes for training in task $\mathcal{T}_t$, to simulate the few-shot situation in meta-testing task $\mathcal{T}_{test}$. The nodes in $\mathcal{Q}_t$ will act as test data being predicted in task $\mathcal{T}_t$, to verify and update the meta-learning model.

4 ATTRIBUTE MATCHING META-LEARNING GRAPH NEURAL NETWORKS

In traditional meta-learning paradigm for few-shot learning, the sampled tasks are often assumed to follow the same distribution as the data instances are independent and identically distributed (i.i.d.). However, instances (i.e., nodes) on attributed networks are inherently connected, which naturally makes the data i.i.d. assumption

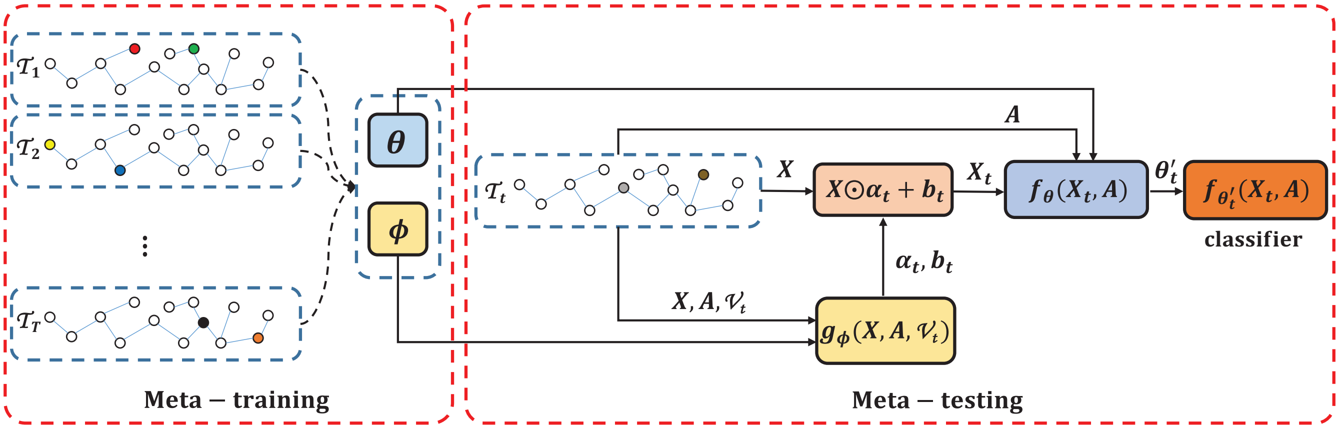


Figure 1: The overview of the proposed AMM-GNN framework. *Left*: In the meta-training phase, multiple tasks are sampled to train the meta-learning model, and we obtain two parameter sets θ and ϕ. *Right*: In the meta-testing phase, we use parameter sets ϕ and θ for attribute matching and gradient descent respectively, and obtain the classifier $f_{\theta'_t}(\cdot)$ for a new sampled task $\mathcal{T}_t$.

invalid. In this regard, if we directly apply existing meta-learning frameworks on attributed networks for few-shot learning, the feature distributions of different sampled tasks could vary significantly. Here, we use a detailed example to further illustrate it.

Suppose we need to perform a classification task on a citation network, where nodes represent papers, edges represent citation relationships between papers, and the attributes of each node are represented as a bag-of-words vector. In meta-training, assume we need to distinguish the papers in the disciplines of "mathematics" and "literature" in one sampled task, the word "formula" apparently occupy important status among all node attributes, because it can well classify the papers in these two disciplines. For a bag-of-words vector, each word corresponds to a certain position. In such case, the trained model will pay more attention to the position of the word "formula" in the feature vector. However, in meta-testing, we may need to classify the papers of "physics" and "chemistry". At this time, the word "formula" no longer separates them well, and another word "reaction" may play an important role. In other words, important features for different sampled tasks could vary remarkably, and the reason can be attributed to the feature distribution differences between different sampled tasks from non-i.i.d. data. This phenomenon considerably reduces the credibility of the model in learning useful transferable knowledge from diverse meta-training tasks for meta-testing.

To tackle the aforementioned problem, we exploit an attribute-level attention mechanism to characterize the feature distribution differences between different tasks in learning more effective transferable knowledge. Based on the attribute-level attention mechanism, a novel meta-learning framework—Attribute Matching Meta-learning Graph Neural Networks (AMM-GNN) is proposed, and the overview of AMM-GNN framework is provided in Figure 1.

4.1 Capturing Task Differences

To make up for the cross-task feature distribution differences for few-shot learning, we attempt to learn a task-wise feature matrix $X_t \in \mathbb{R}^{n \times d}$ for each sampled task $\mathcal{T}_t$, so as to enable the meta-learning model to better match the current task and achieve better performance in the stage of meta-testing. Specifically, we introduce two matching vectors $\alpha_t \in \mathbb{R}^d$ and $b_t \in \mathbb{R}^d$, and obtain a task-wise feature matrix regarding the t-th task $\mathcal{T}_t$ by

$$X_t = X \odot \alpha_t + b_t. \tag{1}$$

To simplify the formulation, the broadcasting rule is used to represent the element-wise product of a row vector α_t and each row of the matrix X by the operation of "$\odot$". The same is true for the operation of "+". These two matching vectors α_t and b_t encode the importance information (i.e., importance of different node attributes) pertaining to the current task $\mathcal{T}_t$.

The matching vectors for task $\mathcal{T}_t$ can be calculated by

$$\begin{cases} \alpha_t = g_{\phi_\alpha}(X, A, \mathcal{V}_t) \\ b_t = g_{\phi_b}(X, A, \mathcal{V}_t), \end{cases} \tag{2}$$

where $g_{\phi_\alpha}(\cdot)$ and $g_{\phi_b}(\cdot)$ are two generators which will be described in more detail in the following subsection. $\mathcal{V}_t$ refers to the node set in the t-th task $\mathcal{T}_t$ in meta-training. With the new task-wise feature matrix X_t, graph neural network (GNN) [27] is leveraged to learn a task-wise representation $Z_t \in \mathbb{R}^{n \times N}$ regarding the t-th task $\mathcal{T}_t$ in meta-learning by

$$Z_t = f_\theta(X_t, A) = \mathrm{softmax}(\hat{A}\,\mathrm{ReLU}(\hat{A}X_tW^{(0)})W^{(1)}) \tag{3}$$

where N denotes the number of ways (i.e., classes) in the task $\mathcal{T}_t$. f_θ represents a deep architecture of GNN, parameterized by $\theta = \left\{W^{(0)}, W^{(1)}\right\}$; $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ with $\tilde{A} = A + I_n$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

4.2 Obtaining Matching Vectors

In this subsection, we discuss how to obtain the matching vectors for task $\mathcal{T}_t$ as described in Eq. (2). It is noteworthy that in the context of few-shot learning on attributed networks, the attribute information alone is not sufficient to capturing the unique properties of the sampled task. Motivated by the strategy used in [51], we utilize a parameterless simplified graph convolution before generating the matching vectors, i.e.,

$$\hat{X} = \hat{A}\hat{A}X. \tag{4}$$

Correspondingly, we denote the aggregated feature matrix regarding to the nodes used in the t-th task $\mathcal{T}_t$ by $\hat{\boldsymbol{X}}_t \in \mathbb{R}^{|\mathcal{V}_t| \times d}$.

To enable the matching vectors to conduct the attribute-level attention on the t-th task $\mathcal{T}_t$ in meta-learning, the following two properties should be satisfied:

1. **Attribute Independence:** To perform attribute-level attention through the matching vectors $\boldsymbol{\alpha}_t$ and $\boldsymbol{b}_t$, we have to guarantee that each entry in these two vectors correspond to one specific attribute in the original attribute space. In other words, we assume that different attributes are independent of each other when designing certain mechanisms to obtain the matching vectors (i.e., avoiding any possible cross-attribute operations).
2. **Sample Consistency:** The matching vectors capture the distinct information for nodes in the task $\mathcal{T}_t$. As different nodes could be sampled in the task $\mathcal{T}_t$, we need to ensure the obtained matching vectors are similar for different selected nodes (i.e., eliminating randomness of node selection).

For the first property (1), we can just perform left multiplication[1] on matrix $\hat{\boldsymbol{X}}_t$ in the generators of $g_{\boldsymbol{\phi}_\alpha}$ and $g_{\boldsymbol{\phi}_b}$, to avoid the intersection between different attributes in the obtained matching vectors. For property (2), we conduct two random samplings of nodes[2] in the t-th task $\mathcal{T}_t$ to reduce the randomness. Specifically, let $\mathcal{V}_t^i$ be the node set for the i-th sampling ($i = 1, 2$), and $\hat{\boldsymbol{X}}_t^i$ be the aggregated feature matrix correspondingly. We design the matching vectors $\boldsymbol{\alpha}_t^i$ and $\boldsymbol{b}_t^i$ for the i-th sampling of the t-th task $\mathcal{T}_t$ in meta-training by the following formulation

$$\begin{cases} \boldsymbol{\alpha}_t^i = \text{MLP}_{\boldsymbol{\phi}_\alpha}(\hat{\boldsymbol{X}}_t^i) \\ \boldsymbol{b}_t^i = \text{MLP}_{\boldsymbol{\phi}_b}(\hat{\boldsymbol{X}}_t^i) \end{cases} \tag{5}$$

for $i = 1, 2$, where $\text{MLP}_{\boldsymbol{\phi}_\alpha}(\cdot)$ and $\text{MLP}_{\boldsymbol{\phi}_b}(\cdot)$ are multi-layer perceptron (MLP) with only left multiplication, parameterized by $\boldsymbol{\phi}_\alpha$ and $\boldsymbol{\phi}_b$ respectively. Then, the matching vectors $\boldsymbol{\alpha}_t$ and $\boldsymbol{b}_t$ with respect to the t-th task $\mathcal{T}_t$ in meta-training is calculated as the the average pooling of the random sampling, i.e.,

$$\begin{cases} \boldsymbol{\alpha}_t = \text{meanpool}\left(\boldsymbol{\alpha}_t^1, \boldsymbol{\alpha}_t^2\right) \\ \boldsymbol{b}_t = \text{meanpool}\left(\boldsymbol{b}_t^1\ , \ \boldsymbol{b}_t^2\right) \end{cases} \tag{6}$$

where meanpool $(\cdot, \cdot)$ is a mean value function.

4.3 Objective Function of AMM-GNN

Following the MAML framework in few-shot learning [14], we use a second-order gradient descent method in the meta-training phase to obtain a good initialization parameter $\boldsymbol{\theta}$ of GNN model $f_{\boldsymbol{\theta}}$. Typically, the inner-loop loss function for the t-th task $\mathcal{T}_t$ is calculated on the basis of the cross-entropy loss of support set $\mathcal{S}_t$,

$$\mathcal{L}_{\mathcal{T}_t}(f_{\boldsymbol{\theta}}) = -\sum_{(\tilde{x}_i, y_i) \in \mathcal{S}_t} \Big(y_i \log f_{\boldsymbol{\theta}}(\tilde{x}_i) + (1 - y_i) \log(1 - f_{\boldsymbol{\theta}}(\tilde{x}_i)) \Big), \tag{7}$$

where $\tilde{\boldsymbol{x}}_i$ denotes the attribute information of the task-wise feature matrix (from Eq. (1)) of i-th node in the support set $\mathcal{S}_t$, and y_i denotes the corresponding class label of the node.

To adapt the inner-loop model to the t-th task $\mathcal{T}_t$, we need to perform gradient descent update based on the above loss function one or a few times, and then obtain a new parameter $\boldsymbol{\theta}_t^{'}$. For simplicity, if we only perform the gradient descent update once, we have the following

$$\boldsymbol{\theta}_t' = \boldsymbol{\theta} - \beta_0 \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{T}_t}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{X}_t, \boldsymbol{A}\right)\right), \tag{8}$$

where β_0 is the inner-loop learning rate.

Algorithm 1 Meta-training in AMM-GNN

```
Require: $(\boldsymbol{A}, \boldsymbol{X}) \in \mathcal{G}$; node set $\mathcal{V}$
Require: task distribution $p(\mathcal{T})$
Require: Inner-loop learning rate $\beta_0$, Outer-loop learning rate $\beta_1$
  Randomly initialize $\boldsymbol{\theta}, \boldsymbol{\phi}$
  while not done do
    Randomly sample batch of tasks $\mathcal{T}_t \sim p(\mathcal{T})$
    for all $\mathcal{T}_t$ do
      Compute matching vectors $\boldsymbol{\alpha}_t, \boldsymbol{b}_t$ by function $g(\cdot)$ with $\boldsymbol{\phi}, \boldsymbol{X}, \boldsymbol{A}, \mathcal{V}_t$
      Calculate $\boldsymbol{X}_t$ with $\boldsymbol{X}, \boldsymbol{\alpha}_t, \boldsymbol{b}_t$
      Evaluate $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{T}_t}(f_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{A}))$ with support set $\mathcal{S}_t$
      Compute adapted parameters with gradient descent: $\boldsymbol{\theta}_t^{'} = \boldsymbol{\theta} - \beta_0 \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{T}_t}(f_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{A}))$
    end for
    Perform stochastic gradient descent with the batch of tasks:
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta_1 \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(f_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}\right)$;  $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \beta_1 \nabla_{\boldsymbol{\phi}} \mathcal{L}\left(f_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}\right)$
  end while
```

In the outer-loop of meta-learning framework, we validate the current model on each query set $\boldsymbol{Q}_t$ of the t-th task $\mathcal{T}_t$ and minimize the sum of query sets loss as

$$\mathcal{L}\left(f_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}\right) = \sum_{\mathcal{T}_t \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_t}\left(f_{\boldsymbol{\theta}_t'}\left(\boldsymbol{X}_t, \boldsymbol{A}\right)\right) - \lambda_1 \mathcal{L}_{sim} \tag{9}$$

where $\boldsymbol{\phi} = \{\boldsymbol{\phi}_\alpha, \boldsymbol{\phi}_b\}$ collects the parameters of generators for matching vectors. The regularization term $\lambda_1 \mathcal{L}_{sim}$ is defined as follows to make the framework more robust

$$\mathcal{L}_{sim} = sim(\boldsymbol{\alpha}_t^1, \boldsymbol{\alpha}_t^2) + sim(\boldsymbol{b}_t^1, \boldsymbol{b}_t^2) \tag{10}$$

where λ_1 is a trade-off parameter, $sim(\cdot, \cdot)$ refers to the cosine similarity between two vectors. This regularization term is incorporated to ensure that the matching vectors are as similar as possible between different sampling.

By optimizing the above objective function, we can obtain a novel meta-learning framework with attribute-level attention, which makes the knowledge accumulation and transfer across different tasks more effective. The detailed description of the proposed meta-training framework is given in Algorithm 1.

4.4 Complexity Analysis

Since the inner-loop model in the proposed framework is a GNN, the complexity of AMM-GNN is proportional to the complexity of the inner-loop model. Note that although we introduce matching vectors $\boldsymbol{\alpha}_t$ and $\boldsymbol{b}_t$ for the t-th task in meta-training, they do not participate in the update of inner-loop. The added computation complexity of matching vectors is $O(n_r dh)$ (if we use a 2-layer MLP), where n_r denotes to the number of sampled nodes, d denotes the number of input attributes, and h denotes the number of hidden layers in the first layer. For Eq. (4), we only need to perform the

[1] We only consider parameter matrix in the left side of $\boldsymbol{X}_t$.

[2] In practice, we can include more random samplings. For the sake of simplicity, we choose the number of random samplings as 2, which achieves good performance.

Table 2: Detailed statistics of datasets.

Datasets	# Nodes	# Edges	# Features	# Labels	Domain
Amazon-Clothing	24,919	91,680	9,034	77	E-Commerce Network
DBLP	40,672	288,270	7,202	137	Citation Network
Reddit	232,965	11,606,919	602	41	Social Network
Cora	2,708	5,429	1,433	7	Citation Network
Citeseer	3,327	4,732	3,703	6	Citation Network

calculation once in the whole process, which can save the computational cost. In summary, the complexity brought by attribute matching is negligible, so the computation complexity of AMM-GNN can be regarded as the same as that of Meta-GNN. In fact, using GCN in the MAML-based framework will get a fairly high computational complexity. Therefore, according to the suggestion of Meta-GNN, we use Simple Graph Convolution (SGC) [51], a simplified version of GCN, as the inner-loop GNN of the proposed AMM-GNN in this paper.

5 EXPERIMENTAL EVALUATIONS

In this section, we validate the effectiveness of our proposed AMM-GNN framework on real-world datasets and with a wide range of experimental settings. We first introduce the used datasets and compared baseline methods and then present the detailed experimental results and findings.

5.1 Datasets

In the experiments, we use a variety of large and small attributed networks that are collected from different domains, including e-commerce networks, social networks, and citation networks.

It should be noted that few-shot learning problems often require a large number of classes for meta-training while some widely used attributed networks [23, 31, 40] only have a few number of classes. Therefore, it is not appropriate to use them directly in the FSL problem. To validate the effectiveness of the proposed framework for AMM-GNN, we use three large attributed networks Amazon-Clothing, DBLP, and Reddit that contain more classes. Meanwhile, we also adopt two small attributed networks Cora and Citeseer that are used in Meta-GNN [59] for a fair comparison. The detailed descriptions of these datasets are as follows:

- **Amazon-Clothing** is a network among Amazon[3] products in the category of "Clothing, Shoes and Jewelry". The dataset is originally collected by [33] and has been preprocessed by [11] for the few-shot learning study. In this dataset, each node corresponds to a product and different products are connected if they have ever viewed by the same user. The product descriptions are used to generate node attributes. Additionally, the node class labels are defined by fine-grained product categories under "Clothing, Shoes and Jewelry". We follow the settings of [11] to use 40/17/20 classes as training/validation/test sets.
- **DBLP**[4] is a citation network extracted from "DBLP dataset (version v11)" [43], where each node represents a paper and edges represent citations. We use the Bag-of-Words model to generate node features from the abstract of each paper. Meanwhile, we take the venues of the paper as its class label. For this dataset, we follow the settings of [11] to use 80/27/30 classes as training/validation/test sets.
- **Reddit**[5] is a dataset extracted from the large online forum Reddit [21]. Each node represents a post and two posts are connected if they have been commented by the same user. The features of nodes are built upon the off-the-shelf 300-dimensional GloVe word vectors [34]. The node labels are the communities (subreddit) where the posts locate. Following the settings of [11], we use 21/10/10 classes as training/validation/test sets.
- **Cora** and **Citeseer**[6] are two classic citation networks that have been extensively used in previous research [2, 27, 40]. In these two datasets, each node represents a paper and the edges represent citations between papers. The node features are also obtained based on the Bag-of-Words models and the papers in these two datasets are categorized into 7 and 6 different classes. As the number of node classes is small, we set 5/2 and 4/2 node categories as training/test sets for these two datasets respectively, which is consistent with the experimental settings in [59].

All these datasets are publicly available and the detailed statistics of these datasets are summarized in Table 2.

5.2 Baseline Methods

To demonstrate the superiority of the proposed AMM-GNN framework, we compare it against the following baseline methods. These methods can be divided into the following three categories: node representation learning methods, traditional meta-learning methods, and graph meta-learning methods. To the best of our knowledge, Meta-GNN [59] is the only meta-learning model for few-shot node classification when the paper is submitted for review, thus we only have one baseline method in the category of graph few shot learning. The detailed information of the compared baseline methods are as follows:

- **Deepwalk** [35] first obtains truncated random walks and learns node embedding representations with the skip-gram model. Then it uses a one-vs-rest logistic regression for the multi-label classification.
- **Node2vec** [20] further extends DeepWalk by using biased random walks. After obtaining node embeddings, it also uses the one-vs-rest logistic regression classifier.

[3] https://www.amazon.com/

[4] https://dblp.uni-trier.de/

[5] https://www.reddit.com/

[6] https://linqs.soe.ucsc.edu/data

Table 3: The few-shot classification accuracy (%) on three large attributed networks.

Datasets	Methods	5-way 1-shot	5-way 3-shot	5-way 5-shot	10-way 1-shot	10-way 3-shot	10-way 5-shot
Amazon-Clothing	Deepwalk	28.6	36.7	46.5	17.1	21.3	35.3
	Node2vec	28.2	36.2	41.9	16.7	17.5	32.6
	GCN	46.7	54.3	59.3	27.7	41.3	44.8
	SGC	47.9	56.8	62.2	30.3	43.1	46.3
	Protonet	43.7	53.7	63.5	24.8	41.5	44.8
	MAML	46.8	55.2	66.1	27.9	45.6	46.8
	Meta-GNN	70.3	74.1	77.3	56.3	61.4	64.2
	AMM-GNN	**72.2**	**79.7**	**81.7**	**58.6**	**67.8**	**69.6**
DBLP	Deepwalk	35.7	44.7	62.4	18.9	33.8	45.1
	Node2vec	29.9	40.7	58.6	17.2	31.5	41.2
	GCN	42.4	59.6	68.3	29.7	43.9	51.2
	SGC	38.1	57.3	65.0	24.1	40.2	50.3
	Protonet	31.0	37.2	43.4	21.2	26.2	32.6
	MAML	33.1	39.7	45.5	22.3	30.8	34.7
	Meta-GNN	64.9	70.9	78.2	55.2	60.7	68.1
	AMM-GNN	**66.5**	**76.5**	**80.4**	**58.9**	**65.8**	**69.9**
Reddit	Deepwalk	24.6	26.7	30.1	13.7	17.6	18.8
	Node2vec	25.5	27.1	31.2	14.7	19.8	23.4
	GCN	25.8	38.8	45.5	20.6	29.0	35.7
	SGC	33.0	44.4	46.8	22.0	29.7	31.6
	Protonet	29.3	34.6	37.6	16.1	19.8	23.3
	MAML	26.9	29.1	31.1	14.7	15.2	17.9
	Meta-GNN	56.1	60.8	62.7	33.8	44.9	51.5
	AMM-GNN	**59.8**	**65.0**	**66.9**	**35.9**	**49.0**	**54.7**

- **GCN** [27] uses the first-order approximation of spectral graph convolution to obtain node embedding representations. Through the layer-wise propagation rule, it aggregates the features of the adjacent nodes layer by layer.
- **SGC** [51] is a variant of GCN. It reduces the extra complexity of GCN by repeatedly eliminating the nonlinearity between the GCN layers and folding the embedding functions into a linear transformation.
- **Prototypical Network** [41] (a.k.a. Protonet) is a traditional metric-based meta-learning method that performs well on the few-shot image classification problem. We instantiate it using a two-layer neural network (with ReLU as the activation function). To apply this method, we only utilize node attribute information.
- **MAML** [14] is an optimization-based meta-learning method for few-shot learning. We use a two-layer neural network (with ReLU as the activation function) as the encoder. Similarly to the Prototypical Network, we only leverage the node attribute information for the few-shot learning.
- **Meta-GNN** [59] is a graph few-shot learning method that combines graph neural networks with MAML. According to the suggestions of the original paper, we use SGC [51] as the inner-loop semi-supervised node classification model.

For AMM-GNN, in all datasets except Reddit, we follow the hyper-parameter settings used in Meta-GNN [59], i.e., β_0 = 0.5, β_1 = 0.003. And λ_1 is set to 0.1. In particular, for Reddit, we set β_1 = 0.001 in both Meta-GNN and AMM-GNN, to stabilize the iterative process in meta-training.

5.3 Few-Shot Learning Results

Here, we present the experimental results of different methods. We show the superiority of the proposed AMM-GNN framework in few-shot learning by comparing its classification performance with other baseline methods.

5.3.1 Experiments on Large Attributed Networks. We first conduct experiments on three large datasets Amazon-Clothing, DBLP, and Reddit that contain more classes. For each dataset, we perform experiments under six different settings (N-way K-shot), where N and K are varied among $\{5, 10\}$ and $\{1, 3, 5\}$, respectively. In order to ensure the experimental fairness and stability, for each method, we randomly select 50 tasks in the meta-testing phase and report their average classification results. In Table 3, we report the classification accuracy of all methods on these three datasets. The detailed analysis is as follows:

Firstly, we can find that the few-shot classification results of AMM-GNN are apparently higher than other baselines. The reason is that AMM-GNN takes into account the feature distribution differences between tasks and learns more meaningful transferable knowledge across tasks for few-shot learning. In general, the classification accuracy of AMM-GNN is around 2% $\sim$ 6% higher than the results of the best competitor. We also observe that the improvement of AMM-GNN at 1-shot is relatively lower than the improvement at 3-shot and 5-shot.

Secondly, we find that the results of random walk based methods Deepwalk and node2vec are significantly lower than those of other baseline methods as they perform embedding learning and classification at two different stages. Meanwhile, we find that GCN and SGC achieve better performance as they use the end-to-end training paradigm. However, in general, the performance of GCN and SGC shows greater fluctuations when the number of training samples in each task varies. It can be attributed to the fact that these two models are not designed for the few-shot learning problem and will face serious over-fitting problem when the number of labeled instances in each task is small.

Thirdly, the results of the traditional meta-learning methods Protonet and MAML are in line with our expectations. These methods do not consider the topological information of graphs during training, thus their experimental results are generally worse than those of Meta-GNN. Besides, GCN and SGC still outperform Protonet and MAML in certain cases as they well characterize the inherent correlations among node attributes and graph topology information through nonlinear graph convolution operations.

Lastly, Meta-GNN is the only baseline method that is designed for few-shot learning on attributed networks. We find that the results of Meta-GNN are better than other baselines in most cases. It uses both the meta-learning approach and the topology information of the graph, that is why the accuracy of Meta-GNN is relatively high. Furthermore, AMM-GNN considers the distinctions between tasks, thus improves the results of Meta-GNN.

5.3.2 Experiments on Small Attributed Networks. In addition to the aforementioned three datasets, we also conduct experiments on two small attributed networks Cora and Citeseer which has fewer number of classes. We follow the same experimental settings as [59] and report the few-shot classification results under the settings of 2-way 1-shot and 2-way 3-shot. We compare the experimental results of AMM-GNN with baselines GCN [27], SGC [51], Protonet [41], MAML [14], and Meta-GNN [59]. The detailed experimental results are shown in Table 4.

We can see that on these smaller datasets, due to the lack of utilization of the adjacency matrix, the traditional meta-learning methods Protonet and MAML still do not perform as well as GNNs. And the MAML-based method, Meta-GNN, applies the meta-learning approach to the graph neural networks and achieves better results. Note that there are few categories in meta-training of small datasets, so the generalization ability of the meta-learning model is relatively poor. In this case, AMM-GNN can still effectively capture the feature distribution differences between tasks and achieves better performance than the baselines.

5.4 Further Studies

In this subsection, we perform further studies to have a more comprehensive understanding of the effectivness of the proposed AMM-GNN framework.

Table 4: The few-short classification accuracy (%) on Cora and Citeseer.

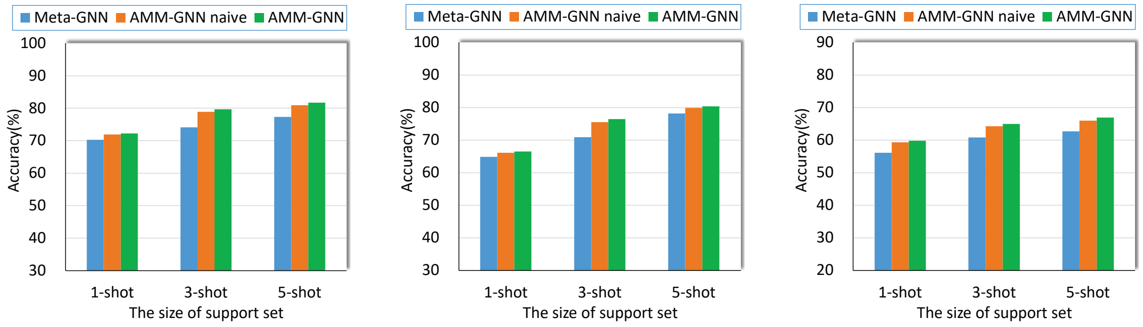
Methods	**Cora**		**Citeseer**	
	1-shot	3-shot	1-shot	3-shot
SGC	61.64	75.67	56.91	65.67
GCN	60.33	75.15	58.44	67.99
Protonet	56.21	63.47	54.29	58.35
MAML	58.29	68.15	56.88	62.78
Meta-GNN	65.27	77.19	61.91	69.43
AMM-GNN	**67.19**	**80.87**	**64.44**	**73.04**

5.4.1 Effect of Graph Topology Information on Matching Vectors. In the proposed AMM-GNN framework, we perform parameterless graph convolutions to obtain the new feature matrix $\hat{\boldsymbol{X}}$ based on the original node attribute matrix $\boldsymbol{X}$ and graph adjacency matrix $\boldsymbol{A}$, and then use $\hat{\boldsymbol{X}}$ to obtain the matching vectors $\boldsymbol{\alpha}$ and $\boldsymbol{b}$. To better show the advantages of using the graph topology information, we conduct comparative experiments with a variant of AMM-GNN—named AMM-GNN naive. Specifically, in AMM-GNN naive, we directly use $\boldsymbol{X}$ instead of $\hat{\boldsymbol{X}}$ to obtain the matching vectors (that is, for Eq. (4), we have $\hat{\boldsymbol{X}} = \boldsymbol{X}$). Here, we set the number of ways N as 5 in the experiments (i.e., 5-way K-shot). As shown in Figure 2, the classification results of AMM-GNN are always higher than the variant AMM-GNN naive which does not use the graph topological information when obtaining the matching vectors. It demonstrates that in AMM-GNN, the utilization graph topology information can help us learn better matching vectors. Meanwhile, the variant AMM-GNN naive still outperforms Meta-GNN, which shows the effectiveness of capturing the unique properties of different tasks with attention mechanism.

5.4.2 Effect on the size of classes N. In Table 3, we have found that the few-shot learning performance generally increases when the size of support set K increases. Here, we conduct further experiments to assess how the variation of class size N will impact the classification performance. In the experiments, we fix K as 3 while vary the value of N from 5 to 10. The experimental results of AMM-GNN and Meta-GNN are shown in Figure 3. We find that their classification performance decreases when the number of classes in the data increases. The reason is that when the number of classes in the data becomes larger, it becomes more and more difficult for the meta-learning models to make accurate predictions.

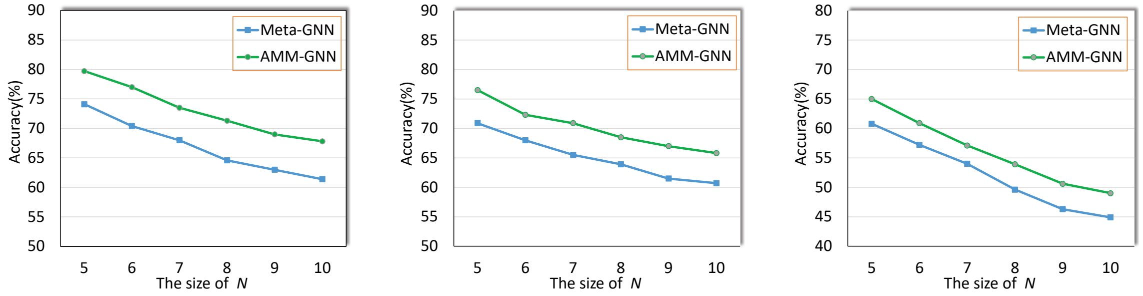
6 CONCLUSION

Few-shot learning for graph-structured data is an essential but underexplored research problem. The major obstacle is that graph data often lies in the non-Euclidean domain while existing few-shot learning algorithms are mainly designed for data in the Euclidean domain. Meanwhile, the complex coupling effect of graph topology and node attributes make the few-shot learning problem much more difficult. Although some recent effort attempts to address the problem by combing the optimization-based meta-learning framework MAML and graph neural networks, it does not consider the



(a) **Amazon-Clothing** (b) **DBLP** (c) **Reddit**

Figure 2: The effect of graph topology information for the matching vectors.



(a) **Amazon-Clothing** (b) **DBLP** (c) **Reddit**

Figure 3: The few-shot classification performance w.r.t. different class size N.

unique feature distributions of different tasks (as a result of the non-i.i.d. nature of graph data), which inevitably yields suboptimal few-shot classification results in the meta-testing. To tackle the aforementioned problem, in this paper, we develop a novel meta-learning framework AMM-GNN to facilitate few-shot learning on attributed networks. In particular, we propose a novel attention mechanism called attribute matching to characterize the differences between feature distributions of different tasks, enabling more effective cross-task knowledge transfer for unseen tasks in meta-testing. We perform extensive experiments on real-world attributed networks from different domains to validate the effectiveness of the proposed framework.

Future work includes generalizing the proposed framework for more complex networks (e.g., heterogeneous networks, signed networks, and multi-dimensional networks) and investigating the feasibility of incorporating feature interactions for few-shot learning on attributed networks.

ACKNOWLEDGMENTS

This work was supported by National Nature Science Foundation of China (No. 61872287 and No. 61772411), Innovative Research Group of the National Natural Science Foundation of China (No. 61721002), Innovation Research Team of Ministry of Education (IRT_17R86), and Project of China Knowledge Center for Engineering Science and Technology.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 5–es.

[3] Avishek Joey Bose, Ankit Jain, Piero Molino, and William L Hamilton. 2019. Meta-graph: Few shot link prediction via meta learning. *arXiv preprint arXiv:1912.09867* (2019).

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR).*

[5] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. 2018. Memory matching networks for one-shot image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[6] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning graph representations with global structural information. In *ACM International on Conference on Information and Knowledge Management (CIKM).*

[7] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247* (2018).

[8] Noy Cohen-Shapira, Lior Rokach, Bracha Shapira, Gilad Katz, and Roman Vainshtein. 2019. AutoGRD: Model recommendation through graphical dataset representation. In *ACM International Conference on Information and Knowledge Management (CIKM).*

[9] Kaize Ding, Jundong Li, Nitin Agarwal, and Huan Liu. 2020. Inductive Anomaly Detection on Attributed Networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*. 1288–1294.

[10] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*. 594–602.

[11] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. 2020. Graph Prototypical Networks for Few-shot Learning on Attributed Networks. *arXiv preprint arXiv:2006.12739* (2020).

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 28, 4 (2006), 594–611.

[13] Michael Fink. 2005. Object classification from a single example utilizing class relevance metrics. In *Conference on Neural Information Processing Systems (NIPS)*.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*.

[15] Madhavi K Ganapathiraju, Mohamed Thahir, Adam Handen, Saumendra N Sarkar, Robert A Sweet, Vishwajit L Nimgaonkar, Christine E Loscher, Eileen M Bauer, and Srilakshmi Chaparala. 2016. Schizophrenia interactome with 504 novel protein–protein interactions. *NPJ schizophrenia* 2, 1 (2016), 1–10.

[16] Hongchang Gao and Heng Huang. 2018. Deep Attributed Network Embedding.. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 3364–3370.

[17] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 1416–1424.

[18] Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043* (2017).

[19] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *International Joint Conference on Neural Networks (IJCNN)*, Vol. 2. IEEE, 729–734.

[20] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Conference on Neural Information Processing Systems (NIPS)*.

[22] Douglas M Hawkins. 2004. The problem of overfitting. *Journal of Chemical Information and Computer Sciences* 44, 1 (2004), 1–12.

[23] Xiao Huang, Jundong Li, and Xia Hu. 2017. Accelerated attributed network embedding. In *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*. 633–641.

[24] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *ACM International Conference on Web Search and Data Mining*. 731–739 (WSDM).

[25] Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. *arXiv preprint arXiv:1805.06556* (2018).

[26] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. 2017. Learning to remember rare events. *arXiv preprint arXiv:1703.03129* (2017).

[27] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[28] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems (NIPS)*.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[31] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. 2017. Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*. 387–396.

[32] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive graph convolutional neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[33] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[34] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[35] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[36] Joseph J Pfeiffer III, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. 2014. Attributed graph models: Modeling network structure with correlated attributes. In *International Conference on World Wide Web (WWW)*. 831–842.

[37] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

[38] Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3132–3142.

[39] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[40] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine* 29, 3 (2008), 93–93.

[41] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Conference on Neural Information Processing Systems (NIPS)*.

[42] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[43] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

[44] Igor V Tetko, David J Livingstone, and Alexander I Luik. 1995. Neural network studies. 1. Comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences (JCIM)* 35, 5 (1995), 826–833.

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[46] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Conference on Neural Information Processing Systems (NIPS)*.

[47] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. 2017. MGAE: Marginalized graph autoencoder for graph clustering. In *ACM International Conference on Information and Knowledge Management (CIKM)*.

[48] Fei Wang and Changshui Zhang. 2007. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering* 20, 1 (2007), 55–67.

[49] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1101–1110.

[50] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–34.

[51] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. 2019. Simplifying graph convolutional networks. In *International Conference on Machine Learning (ICML)*.

[52] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2020).

[53] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning (ICML)*. 40–48.

[54] Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh V Chawla, and Zhenhui Li. 2020. Graph few-shot learning via knowledge transfer. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[55] Lingling Zhang, Xiaojun Chang, Jun Liu, Minnan Luo, and Alexander Hauptmann. 2020. ZSTAD: Zero-shot temporal activity detection. (2020).

[56] Lingling Zhang, Jun Liu, Minnan Luo, Xiaojun Chang, and Qinghua Zheng. 2018. Deep semisupervised zero-shot learning with maximum mean discrepancy. *Neural Computation* 30, 5 (2018), 1426–1447.

[57] Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 1345–1354.

[58] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[59] Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. 2019. Meta-GNN: On few-shot node classification in graph meta-learning. In *ACM International Conference on Information and Knowledge Management (CIKM)*.

[60] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* (2018).

[61] Linchao Zhu and Yi Yang. 2018. Compound memory networks for few-shot video classification. In *European Conference on Computer Vision (ECCV)*.

[62] Xiaojin Zhu. 2005. Semi-supervised learning with graphs. (2005).