2024 IEEE International Conference on Knowledge Graph (ICKG) | 979-8-3315-0882-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICKG63256.2024.00057

Heterogeneous Graph Neural Network based on Bandit Sampling for Attribute Completion

Ta-Yang Wang
University of Southern California
Los Angeles, USA
tayangwa@usc.edu

Rajgopal Kannan
DEVCOM Army Research Office
Los Angeles, USA
rajgopal.kannan.civ@army.mil

Viktor Prasanna
University of Southern California
Los Angeles, USA
prasanna@usc.edu

***Abstract*—Graph neural networks (GNNs) have gained significant attention across diverse areas due to their superior performance in learning graph representation. Many models have emerged to handle heterogeneous graphs using GNNs and have achieved significant success. However, the challenge of incomplete attribute values remains prevalent, particularly in heterogeneous graphs where some types of nodes lack attributes. Existing approaches often separate the attribute completion from the graph learning process. This separation falls short of fully exploiting the rich information inherent in heterogeneous graphs. In this paper, we propose a methodology by using a heterogeneous graph neural network based on bandit sampling for attribute completion. Our approach consists of three key modules: topological embedding, adaptive node sampling, and representation learning. It first learns topological information, and then by integrating a modified multi-armed bandit algorithm, our proposed method adaptively samples informative nodes. This enhances the attribute completion process. Finally, it learns the final embedding of the heterogeneous graph using the graph with completed attributes. We conduct extensive experiments on three real-world datasets. Compared with the state-of-the-art heterogeneous GNN models on the node classification task and the node clustering task, our approach improves the performance by up to 2% and 2.2%, respectively. This shows that the completed attributes from various node types will be aggregated into the target node types, hence improving the model's predictive performance.**

***Index Terms*—Graph representational learning, Graph neural networks, Heterogeneous graphs, Attribute Completion, Multi-armed Bandit**

I. Introduction

Heterogeneous graphs are extensively employed to model complex network systems, where objects are involved in distinct interactions, such as wireless networks, social networks, and recommendation systems [1]–[3]. Heterogeneous graphs consist of multiple node types and edge types, corresponding to various entities and their interactions in real-world applications. For example, the citation network dataset DBLP[1] consists of four types of nodes: papers, authors, terms, and venues, as well as three types of relationships: paper-author, paper-term, and paper-venue. Heterogeneous graphs can model real-world systems more accurately because they contain more comprehensive information and complex relationships compared to traditional homogeneous graphs.

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

Graph Neural Networks (GNNs) utilize deep neural networks to aggregate feature information from neighboring nodes, enhancing the power of the aggregated embeddings. Numerous efforts have been made to apply GNNs to heterogeneous networks [4]–[6]. These heterogeneous GNN-based models can learn node representation via nodes' attributes. However, some nodes have no attributes due to the high cost of obtaining them. Especially in heterogeneous graphs, it is challenging to acquire attributes for all types of nodes. Taking the DBLP dataset as an example, its heterogeneous graph is shown in Figure 1. For each paper node, the keywords are considered as its attribute. However, although we usually perform downstream tasks on author nodes, it is difficult to obtain their attributes. Recent research has shown that the attribute information of authors on node analysis tasks is important for learning the embeddings of heterogeneous graphs [7]. Therefore, addressing the issue of missing node attributes is crucial for improving the performance of heterogeneous GNNs.

Although some types of nodes lack attributes, existing work leverages their connectivity to attributed nodes to address this issue in heterogeneous graphs. However, since there is no relevant information in such a dataset, existing approaches provide less effective information, as the attributes obtained is just the mean of their neighbors' attributes. State-of-the-art methods based on heterogeneous GNNs primarily complete features by (1) filling the missing features with the average of directed connected nodes' features and (2) using their topological information as the node features. These methods are effective; however, there are some challenges for existing Heterogeneous graph neural networks. One challenge is that the heterogeneous information is not fully explored. For example, some heterogeneous GNNs rely on partial neighboring information on customized metapaths [7], [8], making it impossible to distinguish the importance of different node types. Moreover, most existing approaches for attribute completion rely on information from directly connected attributed nodes, neglecting other higher-order connected nodes that are also informative. For example, in the DBLP dataset, coauthors who collaborated on writing the same paper should share some similarities in their research interests, so the relation author-paper-author plays an important role in attribute completion. Therefore, there is a urgent need to develop methods that

979-8-3315-0882-1/24/$31.00 ©2024 IEEE
DOI 10.1109/ICKG63256.2024.00057

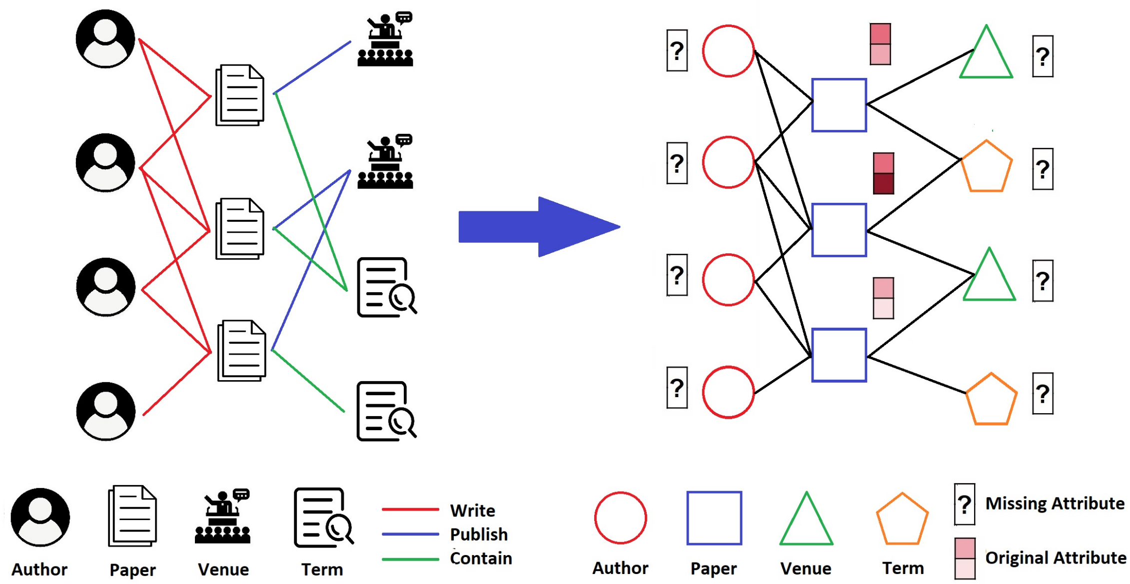

Fig. 1. A heterogenous graph of DBLP dataset.

can capture the heterogeneity in the graph to address these challenges.

In this paper, we propose a novel methodology by using a heterogeneous graph neural network based on bandit sampling for attribute completion. It consists of three key modules: topological embedding, adaptive nodes sampling, and representation learning. Specifically, we employ a heterogeneous graph neural network to incorporate the graph's topological information. The topological relationship help complete missing attributes by aggregating information from their neighboring attributed nodes. Then, instead of using the traditional neighbor attention mechanism as in existing work [7], [9]–[11], we integrate a modified multi-armed bandit algorithm to adaptively sample nodes, so that informative nodes can be captured. This adaptive sampling approach allows us to identify relevant nodes for attribute completion, even if they are not directly connected to the target node. Finally, the heterogeneous graph neural network model is used to learn the node embeddings on the completed heterogeneous graph.

The main contributions of this paper are:

- We propose a methodology of heterogeneous graph neural network based on bandit sampling for attribute completion, which can effectively complete missing attributes of certain types of nodes.
- By incorporating a modified multi-armed bandit algorithm, our proposed methodology adaptively samples informative nodes, resulting in more effective attribute completion and improved performance.
- We conduct extensive experiments on the DBLP, IMDB, and ACM datasets to evaluate the performance of the proposed methodology on several downstream tasks. Compared with the state-of-the-art heterogeneous GNN models on the node classification task and the node clustering task, our approach improves the performance by up to 2% and 2.2%, respectively.

The rest of this paper is organized as follows: Section II presents the background of a GNN model and defines attribute completion. Section III defines the heterogeneous graph representation learning problem and the attribute completion problem. Section IV provides an overview of heterogeneous graph neural networks and bandit sampling methods. Section V presents a detailed description of our proposed methodology. Section VI presents experimental results that validate the performance of the proposed algorithm. Finally, Section VII concludes the paper.

II. Background

In the following section, we introduce the formal notations that define our problem setting and provide an overview of graph neural networks, as well as the neighbor sampling problem. In particular, we focus on bandit sampling method that is applicable to graph neural networks. The notations are summarized in Table I.

A. Basic Notations

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $\mathcal{E}$ among these nodes. Let $(v_i, v_j) \in \mathcal{E}$ denote an edge going from node $v_i \in V$ to node $v_j \in \mathcal{V}$, denote $N(v_i) = \{v_i \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}\}$ as the neighborhood of node v_i. Assume that $\mathcal{G}$ is undirected, that is, $v_j \in N(v_i)$ if and only if $v_i \in N(v_j)$. Let $N(T) = \{v \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}, v_i \in T\}$ denote the neighborhoods of a set of nodes S. $[L]$ denotes $\{1, \ldots, L\}$ for a positive integer L.

TABLE I
NOTATIONS AND EXPLANATIONS

Notations	Explanations
$\mathcal{G}$	A heterogeneous graph
$\mathcal{V}$	The set of nodes
$\mathcal{E}$	The set of edges
$\mathcal{O}_\mathcal{V}$	The set of node types
$\mathcal{R}_\mathcal{E}$	The set of edge types
φ	The node type mapping function from $\mathcal{V}$ to $\mathcal{O}_\mathcal{V}$
ϕ	The edge type mapping function from $\mathcal{E}$ to $\mathcal{R}_\mathcal{E}$
$\mathcal{V}^+$	The set of nodes with attributes in $\mathcal{V}$
$\mathcal{V}^-$	The set of nodes without attributes in $\mathcal{V}$
N_v	The set of neighbors of node $v \in \mathcal{V}$
X	The node attribute matrix
A	Topological structure
H	Node embedding based on the graph topology
$\mathcal{Z}$	The final node embedding

B. Graph Neural Networks

Formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the forward propagation of a GNN is formulated as

$$\mathbf{h}_{v,t}^{(\ell+1)} = \sigma\left(\sum_{i \in N_v} a_{vi} \mathbf{h}_{i,t}^{(\ell)} W_t^{(\ell)}\right) \tag{1}$$

for the node $v \in \mathcal{V}$ at training iteration t. Here $\mathbf{h}_{i,t}^{(\ell)} \in \mathbb{R}^d$ is the hidden embedding of node i at the layer ℓ, $\mathbf{h}_{i,t}^{(0)} = \mathbf{x}_i$ is the node feature, and $\sigma(\cdot)$ is the activation function. $a_{vi} > 0$ is the edge weight between node v and i. $W_t^{(\ell)} \in \mathbb{R}^{d \times d}$ is the GNN weight matrix, learned by minimizing the stochastic loss $\hat{\mathcal{L}}$ with SGD. Finally, we denote $\mathbf{z}_{i,t}^{(\ell)} = a_{vi}\mathbf{h}_{it}^{(\ell)}$ as the weighted embedding, $[D_v] = \{i \mid 1 \le i \le D_v\}$. For a vector $x \in \mathbb{R}^{d_0}$, we refer to its 2-norm as $\|x\|$; for a matrix W, we refer to its spectral norm as $\|W\|$.

Sampling in the training of graph neural network can be formulated as follows:

$$SN^{(k)}(v) = Sampling^{(k)}(N_v) \tag{2}$$

$$a_v^{(k)} = Aggregate^{(k)}\left(\{h_u^{k-1} : u \in SN(v)\}\right) \tag{3}$$

$$h_v^{(k)} = Combine^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right), \tag{4}$$

where $SN(v)$ is the sampled neighbors from N_v, $a_v^{(k)}$ is the aggregation feature vector of node v in the k-th layer, $h_v^{(k)}$ is the representation feature of node v in the k-th layer.

C. Bandit Sampling

Design suitable sampling strategies that take into consideration the intrinsic properties of the graph is crucial for the efficiency and effectiveness of graph neural networks. Given that the contributions of nodes to the learning process can vary over time, it is intuitive to consider the adversarial setting in the multi-armed bandit problem. To adaptively choose the most informative nodes, we consider an importance weighted estimator called Exp4 [12], short for Exploration and Exploitation using Expert advice.

In a general contextual bandit setting, the learner selects an action a_t and observes $\ell_t(a_t)$ for some loss vector ℓ_t, then the main idea of the Exp4 algorithm is to construct the following importance weighting:

$$\hat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)} \mathbb{I}\{a = a_t\}, \tag{5}$$

where p_t is a distribution over actions at time t, and $\mathbb{I}\{a = a_t\}$ is the indicator function that is 1 if $a = a_t$ and 0 otherwise.

Algorithm 1: EXP4.P

```
Input: Learning rate η > 0, p_min ∈ [0, 1/K]
for t = 1, 2, ..., T do
    Compute P_t such that
      P_t(π) ∝ (1 − K p_min) exp(−η Σ_{s<t} ℓ̂_s(π(x_s)))
    Sample a_t from p_t where
      p_t(a) = Σ_{π∈Π:π(x)=a} P_t(π)
    Observe ℓ_t(a_t) and construct ℓ̂_t such that
      ℓ̂_t(a) = (ℓ_t(a) / p_t(a)) 𝕀{a = a_t}
```

In our work, we incorporate a modified version of Exp4. This algorithm with high probability gives a regret at most $O(\sqrt{KT \log N})$ (where there are T steps, the learner must choose one of K actions and have access to a class of N policies) in the adversarial contextual bandit setting. We will discuss this further in details in Section V-C.

III. PROBLEM DEFINITION

In this section, we define the problem of heterogeneous graph representation learning as well as attribute completion. We start with the definition of both heterogeneous graphs and missing attribute in heterogeneous graphs. The notations are summarized in Table I, where capital calligraphic letters denote sets, and capital bold letters represent matrices.

Heterogeneous graphs. A heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}_\mathcal{V}, \mathcal{R}_\mathcal{E})$ is composed of a set of nodes $\mathcal{V}$, a set of edges $\mathcal{E}$, a node type mapping function $\varphi : \mathcal{V} \to \mathcal{O}_\mathcal{V}$, where $\mathcal{O}_\mathcal{V}$ represents the set of node types, and an edge type mapping function $\phi : \mathcal{E} \to \mathcal{R}_\mathcal{E}$, where $\mathcal{R}_\mathcal{E}$ represents edge types that correspond to edges in $\mathcal{E}$. Each node $v \in \mathcal{V}$ is assigned a type via $\varphi(v)$, and each edge $e \in \mathcal{E}$ is assigned a type via $\phi(e)$. For example, in the DBLP dataset, the heterogeneous graph to model DBLP. It consists of four types of nodes (author, paper, venue, and term) and three types of edges (author-paper (write), paper-venue (publish), and paper-term (contain), as illustrated in Figure 1.

Attribute-missing. Given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}_\mathcal{V}, \mathcal{R}_\mathcal{E})$ with a node attribute $\mathcal{X}$, if there exists a subset of node types $\mathcal{S} \subseteq \mathcal{O}_\mathcal{V}$ and $\mathcal{S} \neq \varnothing$, such that for each node $v \in \mathcal{V}$ with $\varphi(v) \in \mathcal{S}$ has no attributes, then node types $\varphi(v) \in \mathcal{S}$ are called attribute-missing. As shown in Figure 1, in the DBLP dataset, only paper nodes have attributes, while authors, venues and terms are attribute-missing.

Heterogeneous graph representation learning. Given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}_{\mathcal{V}}, \mathcal{R}_{\mathcal{E}})$, the problem is to learn a d-dimensional node representation $\mathbf{h}_v \in \mathbb{R}^d$ for all $v \in \mathcal{V}$ with $d \ll |\mathcal{V}|$, which can capture rich structural information in $\mathcal{G}$.

Attribute completion. Given a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}_{\mathcal{V}}, \mathcal{R}_{\mathcal{E}})$ with some node types that are attribute-missing, we can partition the set of nodes into two categories: $\mathcal{V} = \mathcal{V}^+ \cup \mathcal{V}^-$, where $\mathcal{V}^-$ denotes the set of nodes without attributes (i.e., $\varphi(v) \in \mathcal{S}$) and $\mathcal{V}^+$ denotes the set of nodes with attributes. The attribute completion problem is to complete attributes of nodes $v \in \mathcal{V}^-$.

IV. Related work

In this section, we discuss some literature related to heterogeneous graph neural networks and sampling methods for graph neural networks.

Heterogeneous graph neural networks. In recent years, numerous studies have focused on investigating heterogeneous graphs for various applications, such as personalized recommendation [13], [14]. For example, Zhang et al. [4] proposed a heterogeneous graph neural network model to handle the issue of the structural information in heterogeneous graphs and attributes or contents correlated to each node. Hu et al. [5] designed a heterogeneous mini-batch graph sampling method to train web-scale heterogeneous graph efficiently. R-GCN [15] projects node embeddings into several relational spaces using multiple weight matrices, capturing the heterogeneity of the graph. Several approaches have been developed to enhance the representation learning of heterogeneous graphs. HAN [10] model learns the importance between meta-path based nodes and that of different meta-path on node-level and semantic-level aggregation, respectively. It aggregates attributes from meta-path based neighbors in a hierarchical manner. MAGNN [7] determines aggregation weights for different neighbors based on the attributes of all nodes along the meta-path and an attention mechanism, exploring the semantic importance of various meta-paths. GTN [16] generates new graph structure by identifying potentially useful edges in a heterogeneous graph to generate a new graph, learning effective node embeddings on the new graphs to obtain final embeddings. R-HGNN [17] uses relation encoding and designs a relation-aware representation learning framework based on a hierarchical attention mechanism.

However, these methods do not address the problem of missing attributes in heterogeneous graphs. They either complete attributes by an average imputation strategy or customized metapaths, where missing attributes are completed by averaging the attributes of neighboring nodes. In the process of node embedding, the phenomenon of node information loss could occur, which leads to suboptimal results.

Bandit Sampling Methods. Bandit sampling methods [18], [19] have explored the application of bandit algorithms to sample neighboring nodes during the aggregation, which takes the sum of the neighbor embeddings. Liu et al. [18] propose a novel formulation of neighbor sampling as multi-armed bandit problem (MAB) and apply EXP3 [12] and its variants to update sampler and reduce variance. They provide an asymptotic regret analysis on sampling variance, which show that the regret of their estimator, BanditSampler, approximates the optimal sampler within a factor of 3. Zhang et al. [19] propose a numerically-stable reward function that trades bias with variance, which enables the connection to sampling approximation error. ANS-GT [20] adaptively samples informative nodes and captures dependencies through graph coarsening algorithms. Specifically, they formulate the optimization strategy of node sampling as an adversary bandit problem, and then apply the modified Exp4.P algorithm to adaptively assign weights to several sampling heuristics. By combing these strategies together, informative nodes can be sampled.

V. Methodology

In this section, we describe our proposed methodology for attribute completion. Our methodology contains three key components: topological embedding, adaptive node sampling, and representation learning, designed to collaboratively address the problem of attribute completion and obtain the final node embeddings.

A. Overview

The overall methodology is shown in Figure 2. It consists of the following three key modules:

1) **Topological embedding**. The first step involves computing the embeddings of nodes based on the graph's topology. This is achieved using a heterogeneous graph neural network that captures the topological relationships among different node types. Formally, given the heterogeneous graph $\mathcal{G}$ and initial node features X, the model first learns the topological information of node topological embedding H.
2) **Adaptive node sampling**. Once the topological embeddings H are obtained, the next step is to evaluate the topological relationships to identify the most informative nodes. Adaptive node sampling is employed to sample relevant nodes $v \in \mathcal{V}^+$ to the target nodes with no attributes $v \in \mathcal{V}^-$. The model then aggregates the attributes of these informative nodes to complete the missing attributes in $\mathcal{V}^-$ and obtain the completed attribute H^+.
3) **Representation learning**. In the final step, the heterogeneous graph neural network model integrates the initial node attribute X with the completed attribute H^+ to learn the final node embeddings $\mathcal{Z}$. They are then applied to various downstream tasks such as node classification and node clustering.

B. Topological Embedding

The initial phase of our methodology is to obtain the topological embedding. In heterogeneous graphs, although some types of nodes are attribute missing, all nodes have topological information. The relationships among different node types, such as authors, papers, and venues in academic networks, are

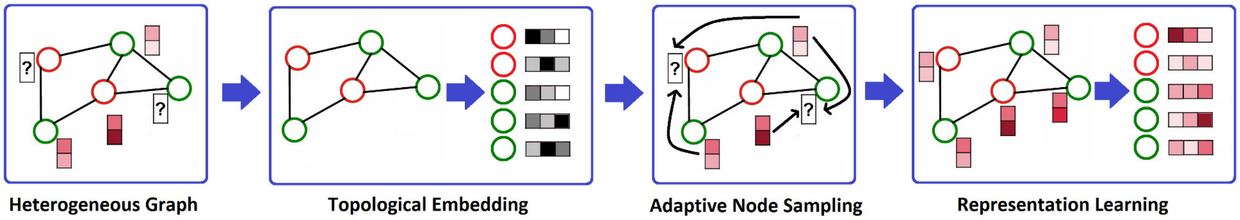

Fig. 2. Overview of the proposed methodology.

crucial for attribute completion, as these nodes share similar topological structures due to their role in the heterogeneous graph. For instance, nodes within the same field of study in the citation network will have similar attributes, hence, the relation among nodes' topological information can reflect the relations among their attribute information. In this paper, we adopt the proximity encoding method in Gophormer [21] to get node embeddings H based on network topology.

C. Adaptive Node Sampling

Adaptive Node Sampling plays a crucial role in our methodology by adaptively sampling and aggregating informative node attributes to help attribute completion. This process is essential for completing missing attributes in target nodes $v \in \mathcal{V}^-$ based on attributes from informative nodes with attributes $v \in \mathcal{V}^+$. Instead of simply averaging aggregate attributes of the directly connected neighbors, we consider graph transformer [20] which modifies a multi-armed bandit algorithm adaptively sample informative nodes, and then automatically learn the importance and aggregates attribute information for nodes in $\mathcal{V}^-$ from $\mathcal{V}^+$.

Traditional methods often use fixed strategies, which may overlook the importance of nodes in a heterogeneous graph. It is intuitive that the contribution of nodes to the learning performance can be time-sensitive. Therefore, to choose the most informative nodes with designed sampling strategies, we consider the ALBL method [22] as in the graph transformer.

Formally, let $w^t = (w_k^t)_{k=1}^K$ be the adaptive weight vector in iteration t, where the k-th non-negative element w_k^t is the weight corresponding to the k-th node sampling strategy. The weight vector w^t is then scaled to a probability vector $p^t = (p_k^t)_{k=1}^K$ where $p_k^t \in [p_{\min}, 1]$ with $p_{\min} > 0$. It adaptively sample nodes based on the probability vector and then obtain the reward of the action.

For each center node, the sampling probability matrix $Q^t \in \mathbb{R}^{K \times n}$, where K is the number of sampling heuristic and n is the number of nodes in the graph. $Q_{k,j}^t$ denotes the k-th sampling strategy's preference on selecting node j in iteration t and Q^t is normalized to satisfy $\sum_{j=1}^n Q_{k,j}^t = 1$. The representative sampling heuristics include m-hop neighbors: we adopt the normalized adjacency matrix $\tilde{A} = \hat{D}^{1/2}\hat{A}\hat{D}^{-1/2}$ for 1-hop neighbors and $\tilde{A}^m$ for m-hop neighbors. Given the probability vector p^t and the node sampling matrices Q^t, the final node sampling probability is

$$\psi_i^t = \sum_{k=1}^{K} p_k^t Q_{ki}^t. \tag{6}$$

Given an attention matrix $\text{Softmax}(QK^\top)/\sqrt{d}$, we use the first row of the attention matrix multiplying the magnitude of corresponding value V_i to represent the significance of each node to the center node s_i. We average the significance scores in multiple attention matrices. The reward to the k-th sampling strategy is

$$r_k = \sum_{i=1}^{N} \frac{s_i Q_{ki}^t}{\phi_i^t}, \tag{7}$$

where N is the number of sampled nodes for each center node. r_k can be viewed as the dot product between the significance score vector and the normalized sampling probability vector. Intuitively, the reward to a certain sampling heuristic is higher if the sampling probability distribution and the node significance score distribution is closer. Finally, we update w^t with the reward. The pseudo-code is listed in Algorithm 2. We then proceed to aggregate the attributes of the informative nodes for each $v \in \mathcal{V}^-$ to complete the missing attributes, eventually obtain the completed attribute H^+.

D. Representation Learning

The final stage integrates the completed node attributes H^+ with the initial node attributes X to obtain final node embedding using the heterogeneous graph neural network model. Define the new attributes of all nodes as: $\tilde{X} = \{X_i, H_j^+ \mid \forall i \in \mathcal{V}^+, \forall j \in \mathcal{V}^-\}$. We then send the initial topological structure A as long as the new attributes $\tilde{X}$ to the heterogeneous graph neural network model: $\tilde{Y} = \Phi(A, \tilde{X}), \mathcal{L}_{\text{prediction}} = f(\tilde{Y}, Y)$, where Φ denotes an arbitrary heterogeneous neural network, f is the loss function, $\tilde{Y}$ and Y are model's prediction and label, respectively. Specifically, in the semi-supervised node classification task, we compute the prediction loss of the heterogeneous graph neural network model for all labeled nodes using the cross-entropy: $\mathcal{L}_{\text{prediction}} = -\sum_{i \in \mathcal{V}_L} \sum_{k=1}^K Y_{ik} \log \hat{Y}_{ik}$, where $\mathcal{V}_L$ is the set of nodes that have labels. K is the set of classes for labeled nodes, Y_{ik} and $\hat{Y}_{ik}$ are the indicator of the label and of the prediction result of node i on class k, respectively.

Finally, by combining the loss of attribute completion $\mathcal{L}_{\text{completion}}$ and the loss of the heterogeneous graph neural

Algorithm 2: Adaptive Node Sampling

Input: Training epochs E; $p_{\min}$; update period T; number of sampled nodes N
Output: Trained model, optimized w^t
Set $w_k^1 = 1$ for $k = 1, \ldots, K$
Calculate the sampling probability matrix Q^t.
for $t = 1, 2, \ldots, E$ **do**
  Train model with the sampled node sequences
  if $t \,\%\, T = 0$ **then**
    Obtain the attention matrices
    Calculate the significance score $s_i = \mathcal{A}_{1,i} \times \|\mathcal{V}_i\|$
    Set $W^t = \sum_{k=1}^{K} w_k^t$ and p_k^t for $k = 1, \ldots, K$
    Calculate ψ_i^t in equation (6) and sample N nodes
    Set $r_k = \sum_{i=1}^{N} s_i Q_{ki}^t / \psi_i^t$
    Update the weight vector w_k^{t+1} using $w_k^{t+1} = w_k^t \exp(\frac{p_{\min}}{2}(r_k + 1/p_k^t)\sqrt{\log N/KT})$

network model $\mathcal{L}_{\text{prediction}}$, the total loss of our proposed methodology is: $\mathcal{L} = \gamma\mathcal{L}_{\text{completion}} + \mathcal{L}_{\text{prediction}}$, where γ is a hyperparameter that control the trade-off between the two losses.

The overall process of our proposed methodology is shown in Algorithm 3.

Algorithm 3: The overall process of our methodology

Input: A heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the initial node attribute X, a heterogeneous graph neural network model Φ
Output: The node embedding $\mathcal{Z}$
Compute the topological embeddings H
Identify informative nodes using Alg 2 and obtain importance score for each node pair
for $v_i \in \mathcal{V}^-$ **do**
  Complete the node feature H_i^+ based on the important score (v_i, v_j) for all $v_j \in \mathcal{V}^+$
Obtain the new attribute $\tilde{X} = \{X, H^+\}$
return $\mathcal{Z}$

VI. Experiments

In this section, we evaluate the proposed methodology in Section V.

A. Datasets

To evaluate the effectiveness of attribute completion by the proposed methodology, we use the following three common datasets: We summarize the statistics in Table II.

1) DBLP [10]. We use a subset of DBLP with 14,328 papers (P), 4057 authors (A), 8789 terms (T), and 20 venues (V). The attributes of papers are bag-of-words representation of their keywords. Only papers' attributes are directly derived from the dataset.
2) IMDB [2] [8]. We obtain a subset of IMDB with 4780 movies (M), 5841 actors (A), and 2269 directors (D). Attributes of movies are bag-of-words representations of their keywords. Only movies' attributes are directly derived from the dataset.
3) ACM [3] [10]. We construct a heterogeneous graph that consists of 4019 papers (P), 7167 authors (A), and 60 subjects (S). Attributes of papers are bag-of-words representations of their keywords. Only papers' attributes are directly derived from the dataset.

B. Baselines

We compare our methodology proposed in Section V with several state-of-the-art heterogeneous graph neural network models, including R-GCN [15], HetGNN [4], HAN [5], and HGNN-AC [8].

1) R-GCN [15]: It extends GCN to graphs with multiple edge types. R-GCN assigns distinct weights to different edge types, and then performs weighted summation to update the node information.
2) HetGNN [4]: It jointly considers node heterogeneous contents encoding, type-based neighbors aggregation, and heterogeneous types combination.
3) HAN [10]: It leverages meta-relations of the given heterogeneous graph to parameterize weight matrices for several critical steps: heterogeneous mutual attention, heterogeneous message passing, and target specific aggregation.
4) HGNN-AC [8]: It first learns topological embeddings of nodes through a pre-trained model, then completes attributes using topological embeddings. Finally, the heterogeneous graph with completed attributes is sent into metapath aggregated GNN to learn node representations.

C. Node Classification

We perform node classification task on the evaluation datasets to compare the performance of our proposed methodology with the baselines. We first complete the missing attributes for target nodes by performing the model, and then obtain embeddings of nodes. Finally, we feed the embeddings to a linear support vector machine (SVM) classifier with training ratio ranging from 10% to 80% to get the classification results. We repeat the process several times and report the average Macro-F1 and Micro-F1.

As shown in Table III, Our methodology generally outperforms the baseline models. It demonstrates that attribute completion is helpful in heterogeneous graph representation learning. Notably, when compared with HGNN-AC, our proposed methodology has 0.36% and 1.28% higher value in terms of Micro-F1 metrics for DBLP, IMDB and ACM, which

[2]https://www.imdb.com/
[3]http://dl.acm.org/

TABLE II
STATISTICS OF THE EXPERIMENTAL DATASETS

Datasets	Nodes	Edges	Attributes
DBLP	author (A): 4,057 paper (P): 14,328 term (T): 7,723 venue (V): 20	#A-P: 19,645 #P-T: 85,810 #P-V: 14,328	A: missing P: original T: missing V: missing
IMDB	movie (M): 4,278 director (D): 2,081 actor (A): 5,257	#M-D: 4278 M-A: 12,828	M: original D: missing A: missing
ACM	author (A): 7,167 paper (P): 4,019 subject (S): 60	#A-P: 13,407 #P-P: 9,615 #P-S: 4,019	A: missing P: original S: missing

TABLE III
CLASSIFICATION RESULTS (%) WITH DIFFERENT METHODS ON THE NODE CLASSIFICATION TASK.

Datasets	Metrics	Training	R-GCN	HetGNN	HAN	HGNN-AC	This paper
DBLP	Macro-F1	10 %	89.72	92.39	92.01	**93.56**	93.50
		20 %	91.15	92.91	92.67	93.80	**94.05**
		40 %	91.52	93.37	93.45	93.92	**94.51**
		60 %	92.03	93.61	93.77	94.01	**94.53**
		80 %	92.33	93.79	93.92	94.20	**94.71**
	Micro-F1	10 %	90.51	92.73	92.53	**94.18**	94.03
		20 %	92.06	93.54	93.48	94.32	**94.45**
		40 %	92.33	93.73	93.85	94.42	**94.78**
		60 %	92.89	94.05	94.21	94.67	**94.94**
		80 %	93.17	94.37	94.55	94.72	**95.01**
IMDB	Macro-F1	10 %	43.87	45.58	56.82	57.82	**59.12**
		20 %	45.98	48.82	57.52	58.64	**59.77**
		40 %	46.12	51.43	57.65	59.23	**60.54**
		60 %	47.74	52.94	57.63	60.12	**61.17**
		80 %	47.90	53.08	57.71	60.60	**61.45**
	Micro-F1	10 %	47.02	46.55	57.10	57.95	**59.15**
		20 %	47.34	49.70	57.66	58.72	**59.91**
		40 %	47.58	52.39	57.72	59.31	**60.72**
		60 %	48.27	53.82	57.89	60.58	**61.13**
		80 %	48.69	54.17	57.77	60.84	**61.47**
ACM	Macro-F1	10 %	84.53	88.20	89.21	**92.29**	92.08
		20 %	85.19	89.25	90.91	**92.95**	92.84
		40 %	86.55	90.23	91.39	93.31	**93.45**
		60 %	87.23	90.80	91.77	93.29	**93.61**
		80 %	87.81	91.10	91.84	93.88	**93.97**
	Micro-F1	10 %	84.47	88.18	89.35	**92.31**	92.11
		20 %	85.23	89.33	91.02	**92.93**	92.78
		40 %	86.75	90.41	91.54	**93.45**	93.38
		60 %	87.30	91.02	91.90	93.47	**93.65**
		80 %	87.99	91.22	92.08	**93.92**	93.89

shows the effectiveness of attribute completion via bandit sampling.

D. Node clustering

To demonstrate the effectiveness of our proposed method, we perform node clustering experiments on DBLP, IMDB, and ACM. We use the K-means algorithm to cluster the labeled node embeddings, where K is the number of node labels. Table IV reports the results of the node clustering task.

Compared with state-of-the-art baselines, our proposed methodology improves NMI and ARI scores by 0.91% and 1.61% on DBLP dataset, and 2.29% and 1.57% on ACM dataset. The results indicate that attribute completion helps improve the clustering performance. The jointly optimization of the topological embedding, adaptive node sampling, and

TABLE IV
CLUSTERING RESULTS WITH DIFFERENT METHODS.

Datasets	Metrics	HetGNN	HAN	HGNN-AC	This paper
DBLP	NMI	0.7867	0.7762	0.7912	**0.8003**
	ARI	0.8402	0.8374	0.8511	**0.8672**
IMDB	NMI	0.1308	0.1289	0.1387	**0.1391**
	ARI	0.1276	0.1339	**0.1472**	0.1421
ACM	NMI	0.7025	0.7145	0.7234	**0.7463**
	ARI	0.7236	0.7488	0.6922	**0.7645**

representation learning makes full use of the attributes of imformative nodes. Therefore, our proposed methodology can learn better embeddings for downstream tasks than other methods.

VII. CONCLUSION

In this paper, we proposed a novel methodology of heterogeneous graph neural networks based on bandit sampling for attribute completion, which fully exploits the information of neighboring structure to solve the attribute completion problem in heterogeneous graphs. Our methodology consists of three key modules: topological embedding, adaptive node sampling, and representation learning, collectively designed to address the challenge of missing attributes in heterogeneous graphs. By integrating a modified multi-armed bandit algorithm, our methodology adaptively samples informative nodes, resulting in more effective attribute completion and improved performance. We conducted extensive experiments on the DBLP, IMDB, and ACM datasets to evaluate the effectiveness of our methodology. Compared with the state-of-the-art heterogeneous GNN models on the node classification task and the node clustering task, our approach improves the performance by up to 2% and 2.2%, respectively.

ACKNOWLEDGMENT

This work has been supported by the U.S. National Science Foundation (NSF) under grant OAC-2209563 and CNS-2009057, as well as the DEVCOM Army Research Office (ARO) under grant W911NF2220159.

Distribution Statement A: Approved for public release. Distribution is unlimited.

REFERENCES

[1] T.-Y. Wang, H. Zhou, R. Kannan, A. Swami, and V. Prasanna, "Throughput optimization in heterogeneous mimo networks: a gnn-based approach," in *Proceedings of the 1st International Workshop on Graph Neural Networking*, 2022, pp. 42–47.

[2] Y. Zhang, L. Wu, Q. Shen, Y. Pang, Z. Wei, F. Xu, E. Chang, and B. Long, "Graph learning augmented heterogeneous graph neural network for social recommendation," *ACM Transactions on Recommender Systems*, vol. 1, no. 4, pp. 1–22, 2023.

[3] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 544–552.

[4] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 793–803.

[5] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of the web conference 2020*, 2020, pp. 2704–2710.

[6] X. Chen, G. Yu, J. Wang, C. Domeniconi, Z. Li, and X. Zhang, "Activehne: Active heterogeneous network embedding," *arXiv preprint arXiv:1905.05659*, 2019.

[7] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of the web conference 2020*, 2020, pp. 2331–2341.

[8] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion," in *Proceedings of the web conference 2021*, 2021, pp. 391–400.

[9] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[10] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032.

[11] C. Li, Y. Yan, J. Fu, Z. Zhao, and Q. Zeng, "Hetregat-fc: Heterogeneous residual graph attention network via feature completion," *Information Sciences*, vol. 632, pp. 424–438, 2023.

[12] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[13] X. Ren, J. Liu, X. Yu, U. Khandelwal, Q. Gu, L. Wang, and J. Han, "Cluscite: Effective citation recommendation by information network-based clustering," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 821–830.

[14] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1531–1540.

[15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 2018, pp. 593–607.

[16] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[17] L. Yu, L. Sun, B. Du, C. Liu, W. Lv, and H. Xiong, "Heterogeneous graph representation learning with relation awareness," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[18] Z. Liu, Z. Wu, Z. Zhang, J. Zhou, S. Yang, L. Song, and Y. Qi, "Bandit samplers for training graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6878–6888, 2020.

[19] Q. Zhang, D. Wipf, Q. Gan, and L. Song, "A biased graph neural network sampler with near-optimal regret," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8833–8844, 2021.

[20] Z. Zhang, Q. Liu, Q. Hu, and C.-K. Lee, "Hierarchical graph transformer with adaptive node sampling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 21 171–21 183, 2022.

[21] J. Zhao, C. Li, Q. Wen, Y. Wang, Y. Liu, H. Sun, X. Xie, and Y. Ye, "Gophormer: Ego-graph transformer for node classification," *arXiv preprint arXiv:2110.13094*, 2021.

[22] W.-N. Hsu and H.-T. Lin, "Active learning by learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.