GSim: A Graph Neural Network based Relevance Measure for Heterogeneous Graphs

Linhao Luo, Yixiang Fang, Moli Lu, Xin Cao, Xiaofeng Zhang, Wenjie Zhang

Abstract—Heterogeneous graphs, which contain nodes and edges of multiple types, are prevalent in various domains, including bibliographic networks, social media, and knowledge graphs. As a fundamental task in analyzing heterogeneous graphs, relevance measure aims to calculate the relevance between two objects of different types, which has been used in many applications such as web search, recommendation, and community detection. Most of existing relevance measures focus on homogeneous networks where objects are of the same type, and a few measures are developed for heterogeneous graphs, but they often need the pre-defined meta-path. Defining meaningful meta-paths requires much domain knowledge, which largely limits their applications, especially on schema-rich heterogeneous graphs like knowledge graphs. Recently, the Graph Neural Network (GNN) has been widely applied in many graph mining tasks, but it has not been applied for measuring relevance yet. To address the aforementioned problems, we propose a novel GNN-based relevance measure, namely GSim. Specifically, we first theoretically analyze and show that GNN is effective for measuring the relevance of nodes in the graph. We then propose a context path-based graph neural network (CP-GNN) to automatically leverage the semantics in heterogeneous graphs. Moreover, we exploit CP-GNN to support relevance measures between two objects of any type. Extensive experiments demonstrate that GSim outperforms existing measures.

Index Terms—Relevance measure, graph neural network, heterogeneous graphs, context path

1 Introduction

Nowadays, many real-world data are often modeled as heterogeneous graphs, which contain multiple typed nodes and edges. As a fundamental topic in network science, similarity measure has been studied for decades and found in various real-world applications, such as web search [1], recommendation [2], and community detection [3]. Conventional studies on similarity measures (e.g., SimRank [4] and P-rank [5]) mainly focus on homogeneous graphs, where objects are of the same type. However, due to the heterogeneous types of nodes and edges in heterogeneous graphs, it is meaningless to measure the "similarity" between two nodes of different types by directly applying these existing similarity measures. As a result, it is necessary to develop novel measures for quantifying the relevance between two nodes in heterogeneous graphs. For example, Fig. 1 (a) depicts a bibliographic network with four types of nodes (i.e., *paper*, *author*, *venue*, and *subject*) and five relations (edge types) among them. As shown in Fig. 1 (b), we may want to find an author that is the most relevant to a certain venue. Unlike the similarity measures in homogeneous graphs which focus on objects of the same type, the relevance measure in heterogeneous graphs should measure the relevance between two objects of different types [6].

Compared to measuring the similarity in homogeneous graphs, measuring the relevance in heterogeneous graphs is more challenging, because the multiple types of nodes and edges carry abundant semantic information. This kind of semantic information plays an important role in measuring the relevance, but it is ignored by many similarity measures in homogeneous networks [1], [4]. For instance, in Fig. 1 (a), there is no direct connection between authors and venues in the graph, making it hard to measure their relevance directly. However, if considering the path *Author-Paper-Venue*, we can claim that an author is relevant to a venue, as long as she/he has written some papers published in this venue. The path that reveals the semantic information above in the heterogeneous graph is called the *meta-path* [7].

In recent years, several works have employed the meta-path for measuring the relevance of nodes in heterogeneous graphs, such as PathSim [7], HeteSim [6], RelSim [8],

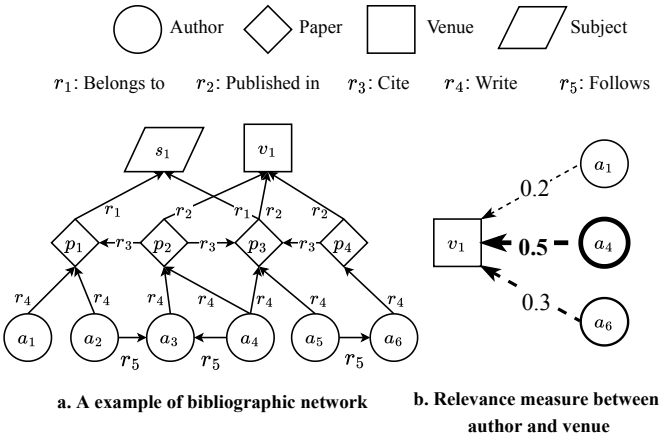


Fig. 1: An example bibliographic information network and relevance measure.

- *Linhao Luo, Moli Lu and Xiaofeng Zhang are with the Department of Computer Science, Harbin Institute of Technology, Shenzhen (E-mail: {luolinhao,21S051052}@stu.hit.edu.cn; zhangxiaofeng@hit.edu.cn.).*
- *Yixiang Fang is with the School of Data Science, The Chinese University of Hong Kong, Shenzhen (E-mail: fangyixiang@cuhk.edu.cn).*
- *Xin Cao and Wenjie Zhang are with the University of New South Wales, Australia (E-mail: {xin.cao,wenjie.zhang}@unsw.edu.au).*

and AvgSim [9]). Nevertheless, a major limitation of these measures is that their performance highly depends on the quality of the pre-defined meta-paths, which needs to be selected manually by domain experts. Moreover, different meta-paths reveal different semantics, and the number of possible meta-paths increases exponentially with the path length, meaning that it is almost infeasible to find all meta-paths to comprehensively capture the semantics. Furthermore, they fail to incorporate several meta-paths at the same time to measure the relevance in a collective manner. In addition, different meta-paths contribute differently to the relevance measure, which imposes great challenges for distinguishing their importance. Therefore, it is desirable to develop new relevance measures without using pre-defined meta-paths.

With the development of graph representation learning [10], the node embeddings that encode the structure and semantics information have been shown effectiveness for measuring the relevance between nodes. Esim [11] is a such kind of measure, but it still requires pre-defined meta-paths. Recently, Graph Neural Network (GNN) has shown great potential in graph representation learning area [12], [13], [14], [15]. GNN can not only embed the graph structure information [16], [17], [18] into the node embeddings, but also consider the semantic information [19] in embedding process. These embeddings can be used for various downstream tasks [20], [21], [22], [23]. An earlier version of this work [24] adopts the GNN to capture the semantic information and uses the learned embeddings for community detection in heterogeneous graphs. Despite the success in many applications, GNN has not been applied to address the relevance measure problem in heterogeneous graphs.

Motivated by the above, in this paper we propose a novel GNN-based relevance measure for heterogeneous graphs, which is called GSim. GSim can automatically leverage the semantics in heterogeneous graphs without using pre-defined meta-paths and learn node embeddings used for the relevance measure. Specifically, we first theoretically prove that GNN can simulate the meeting probability of *pair-wise random walk*, which has been followed by many previous measure methods. Based on the theoretical analysis, we propose a context path-based graph neural network (CP-GNN), which adopts the *context path* to capture the semantics between nodes automatically. Context path [25] links two nodes with the same type via a sequence of nodes of auxiliary type. It can not only well capture the semantics in heterogeneous graphs, but also avoid selecting meta-paths manually by domain experts.

Besides, we introduce the *relation attention* module in GSim, which calculates the attention score of each relation, so the contributions of different relations are well differentiated. In other words, GSim not only avoids using meta-paths, but also well captures the semantics between nodes preserved by context paths with different importance. Moreover, to support relevance measures between two nodes of any type, we present the *relation message passing* mechanism that extends the context path to measure the relevance under asymmetric context paths, whose effectiveness has been shown by both theoretical analyses and empirical study. In addition, we design a *type-length attention* module to capture the fine-grained importance of context paths of different lengths. Last, we adopt the framework of supervised contrast learning to optimize GSim, so that the relative relevance between nodes can be well-preserved.

In summary, our principal contributions are as follow:

- We propose a novel GNN-based relevance measure method, called GSim, for heterogeneous graphs. To our best knowledge, this is the first relevance measure for heterogeneous graphs that exploits GNN.
- We theoretically show that GNN can be used to measure the relevance effectively, based on which a context path-based GNN (CP-GNN) is designed for capturing the semantics of relevance measure.
- We conduct extensive experiments on four real-world heterogeneous graphs, and the results on tasks like relevance search and community detection demonstrate the superior performance of GSim.

An earlier version of this work was published in the conference CIKM'2021 [24], where we proposed the CP-GNN for community detection in heterogeneous graphs. In this paper, we extend the CP-GNN to measure the relevance in heterogeneous graphs and provide theoretical analysis as well as extensive experiments to guarantee effectiveness. The paper is organized as follows. We review the related works in Section 2. We describe the related notations and problem definition in Section 3. Section 4 discusses our proposed GSim. We report the experimental results in Section 5 and conclude in Section 6.

2 Related work

2.1 Similarity and Relevance Measures

Similarity measure, aiming to calculate the similarity between objects of the same type, has been studied for decades [5], [26], [27]. Previous methods often utilize the structure to measure the similarity. For example, SimRank [4] calculates the similarity of two nodes by recursively averaging the similarity of their neighbors. Personalized PageRank (PPR) [1] measures the similarity by using the probability of random walks starting from the source node to the target node. However, since these approaches only consider objects of the same type, they cannot be directly used in heterogeneous graphs.

To extend the application to heterogeneous graphs, a few relevance measures have been developed. For example, PathSim [7] introduces the concept of meta-path-based similarity by calculating possible meta-path instances, but it can only use symmetric meta-paths to measure the similarity between nodes of the same type. To measure the relevance of different-type objects, HeteSim [6] computes the meeting probability of two nodes following a given meta-path. RelSim [8] further introduces the latent semantic relation (LSR) by combining weighted meta-paths to catch different semantics for relevance measure. SCHAIN [28] and CMOC-AHIN [29] calculate the relevance as a weighted sum of all attributes and meta-paths. However, all the aforementioned methods need user-defined meta-paths. Besides, HeteSim and PathSim can use only one meta-path at a time, which does not fully excavate the semantics of different meta-paths. In addition, when multiple meta-paths are used, how to properly assign their weights is also a challenging task.

Recently, some meta-path-free measures have been studied. nSimGram [30] makes use of q-gram and applies the Bray-Curits similarity index to measure the relevance, but it can only be applied on the labeled heterogeneous graphs. HowSim [31] is another meta-path-free method that aggregates similarities over relations to catch different semantics, but it needs the two query nodes to be of the same type, thus cannot handle our relevance measure problem.

2.2 Graph Neural Network

Conventional graph representation learning (e.g., DeepWalk [32], LINE [33], Node2Vec [10], and Metapath2vec [34]) aims at mapping node into a low-dimensional vector, which preserves the original graph information and can be used for various tasks. For instance, ESim [11] learns node embeddings under the guidance of meta-paths and uses the embeddings for relevance measure in heterogeneous graphs.

Recently, Graph Neural Network (GNN) has been shown powerful for learning node representation [35]. The key idea of GNN is to aggregate information from node's neighbors via neural networks [36]. For example, GAT [37] applies the attention mechanism on each node and its neighbors, which can help the center node aggregate information from the important neighbors. In this way, GNN can embed both the structure and feature information into the node embedding as discussed in previous works [16], [17]. GNN has also been studied on heterogeneous graphs. For example, MEIRec [38] is a meta-path guided heterogeneous GNN that learns the embeddings of objects for intent recommendation. HAN [14] adopts the meta-path to leverage the semantic information and uses the attention mechanism to differentiate them. MAGNN [39] further aggregates information along the meta-path to incorporate fine-grained semantics. ie-HGCN [15] designs a hierarchical aggregation architecture to automatically extract useful meta-paths, which presents good interpretability and model efficacy. However, all these methods still require the meta-path, which largely limits their applications.

Although some recent GNN-based models (e.g., HGT [19]) have considered the semantics in heterogeneous graphs, none of them is designed for relevance measurement. On the other hand, the GNN-based embedding models have been shown more powerful than traditional similarity metric-based embedding models, which can be designed by applying cosine similarity [40], Jaccard coefficient [41], and the p-norm distance [42] on them.

Motivated by the above, in this paper we propose a GNN-based relevance measure, namely GSim, which is not only meta-path-free but also universal that can be used on different-typed objects, allowing us to measure the node relevance in heterogeneous graphs in an end-to-end manner.

3 Preliminary

In this section, we first briefly introduce the data model of the heterogeneous graph, and then present some key concepts and the problem that we study in this paper.

Definition 3.1 (**Heterogeneous graph**)**.** The heterogeneous graph is defined as a graph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ with a node mapping function $\phi(v) : \mathcal{V} \rightarrow \mathcal{A}$ and an edge mapping function $\psi(e) : \mathcal{E} \rightarrow \mathcal{R}$, where $|\mathcal{A}| + |\mathcal{R}| > 2$, each node $v \in \mathcal{V}$ belongs to a node type $\phi(v) \in \mathcal{A}$, and each edge $e \in \mathcal{E}$ belongs to an edge type (also called relation) $\psi(e) \in \mathcal{R}$.

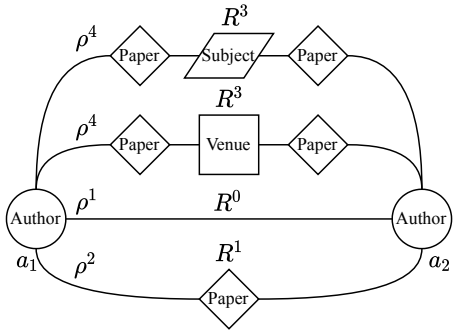


Fig. 2: An example of four context paths between a_1 and a_2.

Definition 3.2 (**Meta-path [7]**)**.** Given a heterogeneous graph $\mathcal{H}$, the meta-path is a path with the form $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \cdots \xrightarrow{R_k} A_k$, where $A_i \in \mathcal{A}$, $R_i \in \mathcal{R}$ $(1 \leq i \leq k)$, and $R_1 \circ R_2 \circ \cdots \circ R_k$ defines the composite relation between node type A_0 to node type A_k. The length of the meta-path is the number of composite relations $|R_1 \circ R_2 \circ \cdots \circ R_k|$.

Definition 3.3 (**Context path [25]**)**.** Given a heterogeneous graph $\mathcal{H}$, a *context path* is a path connecting two nodes v_i and v_j with the same type A, formulated as $\rho^k = \{v_i, R^{k-1}, v_j\}$. The R^{k-1} is any path connecting v_i and v_j that contains $k-1$ $(k \geq 1)$ nodes of *auxiliary types* $\mathcal{A}' = \mathcal{A} \backslash \{A\}$, where nodes in R^{k-1} is also denoted as the *auxiliary nodes*. The length of a context path is k (when $k = 1$, $R^k = \emptyset$).

Example 1. Fig. 2 depicts four possible context paths ρ^* of different lengths that connect authors a_1 and a_2, where R^* denotes the auxiliary nodes that constitute the path.

Difference between context path and meta-path. Intuitively, different semantic relationships revealed from the context path come from different auxiliary nodes. The purpose of the meta-path is to manually define the combination of auxiliary nodes in the context path. However, the number of combinations explodes exponentially as the node types and path length increase. Thus, the context path relaxes the restriction of the auxiliary nodes. Given a length k, the context path contains all the possible k-order relationships. For example, in Fig. 2, two meta-paths *Author-Paper-Subject-Paper-Author* and *Author-Paper-Venue-Paper-Author* can both be represented by a 4-length context path.

Besides, specifying an integer value of k is much easier than defining the meta-path. This is because the number of meta-paths of different node/edge types grows exponentially as the meta-path length increases, while the number of possible values of k is rather limited since the average length of the shortest paths between any two nodes in real-world networks is often between 4 to 6, according to [43].

Noticeably, the original definition of context path only considers the path between nodes of the same type, which cannot measure nodes of heterogeneous types. Therefore, we propose the *generalized context path* to extend the CP-GNN model in Section 4.2.3, which is defined as follows:

Definition 3.4 (Generalized context path)**.** Generalized context path relaxes the same type constraint for nodes v_i, v_j in the contex path $\rho^k = \{v_i, R^k, v_j\}$. Thus, it can capture the semantics between nodes of different types.

Problem 1. Given a heterogeneous graph $\mathcal{H}$ and two nodes v_i and v_j with any type in $\mathcal{H}$, we want to design a measure $S(v_i, v_j)$ such that it can accurately capture the relevance between v_i and v_j in $\mathcal{H}$.

4 Our Approach

In this section, we first discuss how GNN can measure the relevance on graph. Then, we propose a context path-based graph neural network (CP-GNN) to automatically leverage the semantics in heterogeneous graphs. Finally, we propose the CP-GNN+ to support relevance measures between any type of object.

4.1 GNN for relevance measure

Previous measure methods (e.g., SimRank [4] and HeteSim [6]) are based on the theory of *pair-wise random walk*.

Definition 4.1 (**Pair-wise random walk (PRW) [4]**)**.** Pair-wise random walk PRW(v_i, v_j) measures the probability that two random walks starting from v_i and v_j meet at the same node on the graph. This can be formulated as

$$PRW(v_i, v_j|\pi^k) = \sum_{\pi^k} p(v_i, v_j|\pi^k), \tag{1}$$

where $\pi^k : \{v_i, \cdots, v_m, \cdots v_j\}$ denotes arbitrary k-length walks (path contains k edges) that v_i, v_j meet at intermediate node v_m. The higher the probability, the more likely the two nodes are similar.

The π^k can be seen as the composition of two walks $\pi_{i:m}, \pi_{j:m}$ respectively starting from v_i and v_j and ending at v_m. Thus, the $p(v_i, v_j|\pi^k)$ can be written as

$$p(v_i, v_j|\pi^k) = p(v_m|v_i, \pi_{i:m})p(v_m|v_j, \pi_{j:m}) \tag{2}$$

where $p(v_m|v_*, \pi_{*:m})$ denotes the probability of v_* visits at node v_m under path $\pi_{*:m}$. Following the idea of random walk, each node will randomly visit one of its neighbors, we formulate $p(v_m|v_*, \pi_{*:m})$ as

$$p(v_m|v_*, \pi_{*:m}) = \prod_{w_l \in \pi_{*:m}} \frac{1}{O(w_l)}, \tag{3}$$

where $O(w_l)$ denotes the out degree of l-th node in $\pi_{*:m}$.

HeteSim, to capture the semantics in heterogeneous graphs, uses the meta-path $\mathcal{P}$ as constraint, which measures how likely v_i and v_j will meet at the same node when they travel along the meta-path. Given a meta-path $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \cdots \xrightarrow{R_k} A_k$, it can be formulated as

$$\text{HeteSim}(v_i, v_j|\mathcal{P}) = \sum_{\varphi^k \in \mathcal{P}} p(v_i, v_j|\varphi^k), \tag{4}$$

where φ^k denotes the k-length meta-path instance constrained by $\mathcal{P}$. Equation 4 shows that HeteSim needs to iterate over all possible meta-path instances and sum up the relevance.

Extending the theory of pair-wise random walk, we propose the theorem that GNN can simulate pair-wise random walk and measure the relevance between two nodes v_i and v_j.

Theorem 4.1. The inner-product of two node representations h_i^k, h_j^k generated by a k-layer GNN is equal to the probability of pair-wise random walk under $2k$-length paths. This can be formulated as

$$H^k = \text{k-layer GNN}(Z), \tag{5}$$

$$PRW(v_i, v_j|\pi^{2k}) = \left\langle h_i^k, h_j^k \right\rangle, \tag{6}$$

where Z denotes the initial node embedding, $h_i^k, h_j^k \in H^k$, and $\langle \cdot, \cdot \rangle$ denotes the inner-product operation.

Proof. Following the Equation 2, a $2k$-length path π^{2k} can be divided into two k-length walks, formulated as

$$\pi^{2k} = \{\pi_{i:m}^k, \pi_{m:j}^k\}. \tag{7}$$

Thus, we can prove that each entry of the node representation $h_i^k[l]$ denotes the probability that node v_i visits $v_l \in \mathcal{V}$ under k-length walks, written as $p(v_l|v_i, \pi^k)$.

A k-layer GNN can be written as

$$H^k = \sigma\Big(L \cdots \sigma\big(L\sigma(LZW^1)W^2\big) \cdots W^k\Big), \tag{8}$$

where Z denotes the initialized node embedding, L denotes the Laplacian matrix, $\sigma(\cdot)$ denotes the activation function, and W^* denotes the weight matrix.

For simplicity, we can assign the Z with the unique *one-hot embedding*, where $Z \in \mathbb{R}^{|V|\times|V|} = \mathbf{I}$, and use $D^{-1}A$ as the Laplacian matrix, where D denotes the degree matrix, and A denotes the adjacency matrix. The $D^{-1}A$ is also known as the random walk transition matrix, which means each node will walk to its neighbors with equal chance. The $\sigma(\cdot)$ and W^* are also defined as the identity matrix $\mathbf{I}$.

When $k = 1$, the H^1 can be formulated as

$$H^1 = \mathbf{I}D^{-1}AZ\mathbf{I} = D^{-1}AZ = D^{-1}A, \tag{9}$$

where $H^1 \in \mathbb{R}^{|V|\times|V|}$. Given a node v_i, each entry of its node representation can be written as

$$h_i^1[l] = \begin{cases} \dfrac{1}{O(v_i)}, & A[i][l] = 1, \\ 0, & else. \end{cases} \tag{10}$$

Therefore, $h_i^1[l]$ denotes the probability that node v_i directly transits to v_l, written as $p(v_l|v_i, \pi^1)$.

Assuming at layer $k-1$, $h_i^{k-1}[l]$ denotes the probability that node v_i visits v_l under $(k-1)$-length path, written as $p(v_l|v_i, \pi^{k-1})$. At k layer, the $h_i^k[l]$ is formulated as

$$\begin{aligned} h_i^k[l] &= \sum_{v_l' \in N(v_l)} \frac{1}{O(v_l')} h_i^{k-1}[l'] \\ &= \sum_{v_l' \in N(v_l)} p(v_l|v_l', \pi^1)p(v_l'|v_i, \pi^{k-1}) \\ &= p(v_l|v_i, \pi^k), \end{aligned} \tag{11}$$

where $N(v_l)$ denotes the direct neighbors of v_l. From Equation 11, we can see that GNN summarizes the transition probability from neighbors to synthesize visit probability.

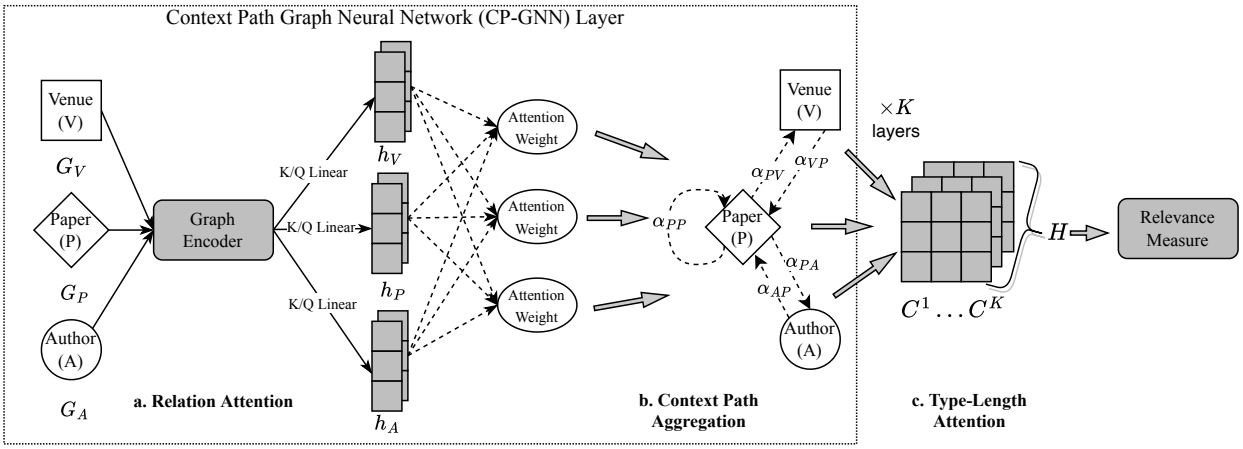


Fig. 3: The overall framework of the CP-GNN. a. Relation attention assesses the importance of different relations by calculating the attention score. b. Context path aggregation aggregates the information to generate the context information vector. c. Type-length attention summarizes the vectors under different lengths and types to obtain the final node embeddings.

Therefore, the inner production of h_i^k, h_j^k can be formulated as

$$\begin{aligned}\left\langle h_i^k, h_j^k \right\rangle &= \sum_{l=1}^{|\mathcal{V}|} h_i^k[l] h_j^k[l] \\ &= \sum_{l=1}^{|\mathcal{V}|} p(v_l|v_i, \pi^k) p(v_l|v_j, \pi^k) \\ &= \sum_{\pi^{2k}} p(v_i, v_j|\pi^{2k}) \\ &= PRW(v_i, v_j|\pi^{2k}).\end{aligned} \tag{12}$$

From Equation 12, we can see that $\left\langle h_i^k, h_j^k \right\rangle$ summarizes the meeting possibility of every node in the graph, which is the same as the Definition 4.1. □

From Theorem 4.1, we can see that by stacking k layers, GNN can simulate the k-length random walk and inject the reaching possibility into the node embedding. Thus, the inner-product of any two embeddings summarizes the meeting possibility under $2k$-length paths, which is equal to the definition of the pair-wise random walk. Based on Theorem 4.1, we can define a GNN-based relevance measure method, which is formulated as

$$H^k = \text{k-layer GNN}(Z), \tag{13}$$

$$S(v_i, v_j) = \left\langle h_i^k, h_j^k \right\rangle, h_i^k, h_j^k \in H^k. \tag{14}$$

4.2 Context path-based graph neural network

To extend the Theorem 4.1 for heterogeneous graphs, we propose a novel Context Path-based Graph Neural Network (CP-GNN), which aims to learn node representations that are able to well capture the semantic information for relevance measure. The overall framework of CP-GNN is depicted in Fig. 3.

As we discussed in section 3, the context path can well represent the semantic information in heterogeneous graphs. Therefore, we propose the CP-GNN that recursively embeds the semantics of each node into a context information vector c^k, which can be formulated as

$$C^k = \text{k-layer CP-GNN}(Z), \tag{15}$$

where $Z \in \mathbb{R}^{|V| \times d}$ denotes the initialized node embeddings, and $c^k \in C^k$ denotes each nodes's k-length context information vector.

Instead of using the random walk transition matrix, CP-GNN consists of two major components (i.e., *relation attention* and *context path aggregation*) to control the transition probability. In this way, we can automatically differentiate the importance of different semantics in the context paths. Then, we propose a *type-length attention* mechanism to synthesize information of different length context paths for relevance measure.

4.2.1 Relation Attention

Relation attention aims to calculate the attention score of each relation, so that the contributions of different relations are well differentiated. We first use a graph encoder to encode the graph of each node type into a summary vector. After that, the attention score of each relation is calculated based on the graph summary vectors.

Given a node type A, the G_A is denoted by G_A=$(\mathcal{V}_A, \mathcal{E}_A)$ where each node $v \in \mathcal{V}_A$ is of the node type A and each edge $e \in \mathcal{E}_A \subseteq \{\mathcal{V}_A \times \mathcal{V}_A\}$. To enhance the model robustness, the graph encoder contains a node dropout mechanism that randomly drops nodes from the original graph. Then an averaging operation is adopted to calculate the graph summary vector h_A. Although there exist several techniques to generate the graph summary vector h_A, the simple averaging operation demonstrates superior performance [44], and thus h_A is calculated as

$$G'_A = NodeDropout(G_A), \tag{16}$$

$$h_A = Mean(C'_A), \tag{17}$$

where C'_A denotes context information vectors of all the nodes in the remaining graph G'_A.

After calculating the h_A, for the l-th CP-GNN layer, we calculate the h-head attention score $\alpha_{S,T}^{h,l}$ for each relation $r_{S,T} \in \mathcal{R}$ by

$$h_S^l = GraphEncoder(C_S^{l-1}), \tag{18}$$

$$h_T^l = GraphEncoder(C_T^{l-1}), \tag{19}$$

$$\alpha_{S,T}^{h,l} = \underset{S \in \mathcal{A}}{Softmax} \frac{Q^h(h_T^l)^\top K^h(h_S^l)}{\sqrt{d}}, \tag{20}$$

$$Q^h(h_T^l) = QLinear_T^h(h_T^l), \tag{21}$$

$$K^h(h_S^l) = KLinear_S^h(h_S^l), \tag{22}$$

where S and T respectively denote the source and target node types in the relation $r_{S,T}$, h_S^l and h_T^l denote the graph summary vector of G_S and G_T at layer l respectively, C_S^{l-1} and C_T^{l-1} are the context information vectors at the $(l-1)$-th layer, $QLinear$ and $KLinear$ are the linear projection functions that project the graph summary vectors to a Query vector and a Key vector. We want to learn more diverse importance of the relations, thus we adopt total H different heads of relation attention with their own parameters to be learned during the training. $\alpha_{S,T}^{h,l}$ is the attention weight in head h at layer l for the relation $r_{S,T}$.

4.2.2 Context Path Aggregation

Context path aggregation aims to aggregate the information along relations to generate the context information vectors for all nodes. After calculating the scores of different relationships, we aggregate the information for a node v_i of type T from its one-hop neighbors by adopting the widely used GNN message passing method.

Assuming at layer l, we are going to obtain the context information vector c_i^l for v_i by aggregating the information from its neighbors along different relations. We utilize its neighbors' context information vectors obtained at layer $l-1$, which can be formulated as

$$c_i^l = W_2^l \Big(||_{h\in[1,H]} \sigma(W_1^l \sum_{r_{S,T}\in\mathcal{R}} \alpha_{S,T}^{h,l} \sum_{v_j \in N_S(i)} c_j^{l-1} + B_1^l) + B_2^l \Big), \tag{23}$$

where $N_S(i)$ denotes the adjacent neighbors of v_i in graph G_S for each relation $r_{S,T} \in \mathcal{R}$ relevant to node type T, W_1^l, W_2^l, B_1^l, and B_2^l are the trainable parameters in the l-th layer, and H is the number of attention heads. Note that finally we only use the embedding of nodes at the k-th layer to obtain k-length context information vectors C^k.

In order to get the information of k-length context paths, we stack k CP-GNN layers to obtain the C^k at layer k. The GRU mechanism [45] is also utilized to alleviate the over smoothing problem that unusually occurred in GNN model [46]. This can be formulated as

$$\hat{C}^l = \text{CP-GNN Layer}(C^{l-1}), l \in [1,k] \tag{24}$$

$$C^l = \text{GRU}\Big(C^{l-1}, \hat{C}^l\Big). \tag{25}$$

Therefore, the final context information vector c_i^k of each node in $\mathcal{H}$ can be taken from C^k.

4.2.3 Relation Message Passing

From the Definition 3.3 of context path, we can see that CP-GNN can only measure the relevance between the nodes of the same type. To measure nodes of heterogeneous types, we can use the *generalized context path*[1] in Definition 3.4 to extend CP-GNN. However, the context paths are often asymmetric between nodes of different types. For example, one possible context path between author and paper is *Author-Paper-Subject-Paper*. There are not intermediate nodes between *Paper* and *Subject* for walks meeting. Thus, we cannot adopt CP-GNN to capture such semantics. To address this challenge, we propose the *relation message passing* mechanism and integrate it with CP-GNN to come up with the CP-GNN+.

Inspired by HeteSim [6], we can simply add an intermediate node E to each edge in the original graph. For example, path *Author-Paper-Subject-Paper* can be transformed into *Author-E_{AP}-Paper-E_{PS}-Subject-E_{SF}-Paper*. Thus, arbitrary context paths can be transformed into even-length paths, where walks can meet. We can prove that adding intermediate nodes would not change the relevance of nodes in the original graph.

Theorem 4.2. Adding intermediate nodes would not affect the relevance calculated by GNN.

Proof. We can denote the set of added intermediate nodes by $\mathcal{V}_{new}$ and the graph after adding nodes by $\tilde{\mathcal{H}}$. The nodes in $\tilde{\mathcal{H}}$ are denoted by $\tilde{\mathcal{V}} = \mathcal{V} \cup \mathcal{V}_{new}$. We can extend the dimension of initialized node embedding to $|\tilde{\mathcal{V}}|$, and assign one-hot embeddings for v_{new}.

Before adding intermediate nodes, the visit probability under k-length paths is written as

$$p(v_l|v_i,\pi^k) = \prod_{w_l\in\pi^k} \frac{1}{O(w_l)}. \tag{26}$$

After adding intermediate nodes, the k-length path is extended to $2k$-length. Since the out-degree of an intermediate node is 1, the probability under extended $2k$-path is written as

$$p(v_l|v_i,\pi^{2k}) = \prod_{w_l\in\pi^{2k}} \frac{1}{O(w_l)} \tag{27}$$

$$= \prod_{w_l\in\pi^{2k}\wedge w_l\in\mathcal{V}} \frac{1}{O(w_l)} \prod_{w_l\in\pi^{2k}\wedge w_l'\in\mathcal{V}_{new}} \frac{1}{O(w_l')} \tag{28}$$

$$= \prod_{w_l\in\pi^{2k}\wedge w_l\in\mathcal{V}} \frac{1}{O(w_l)} \tag{29}$$

$$= p(v_l|v_i,\pi^k). \tag{30}$$

According to Theorem 4.1, we can simply use a $2k$-layer GNN on $\tilde{\mathcal{H}}$ to obtain the relevance in original graph. □

Though adding intermediate nodes would not affect the relevance, the storage complexity of the graph will increase from $O(|\mathcal{V}|+|\mathcal{E}|)$ to $O(|\mathcal{V}|+2|\mathcal{E}|)$. This is unacceptable for large-scale graphs.

To address the aforementioned challenge, we propose the *relation message passing* mechanism, which would not increase the storage complexity. We introduce the Theorem 4.3 to facilitate the analysis.

1. We still use the term of *context path* to denote *generalized context path* in the following sections for simplicity.

Theorem 4.3. There exists an injective function f that can represent each intermediate node $v_{e_{ij}}$ with its neighbor nodes, which can be formulated as

$$m_{e_{ij}} = f(z_i, z_j), v_{e_{ij} \in V_{new}}, v_i, v_j \in N(v_{e_{ij}}), \quad (31)$$

where $v_{e_{ij}}$ denotes the added intermediate node for original edge $e_{ij} = (v_i, v_j)$, and z_* denotes the node embedding.

Proof. According to Theorem 4.2, added nodes with unique representation would not affect the relevance results. In Theorem 4.1, we assign the unique one-hot embedding for $v \in \mathcal{V}$. Additionally, each $v_{e_{ij}}$ only connects to nodes v_i and v_j. Thus, by defining the f as a simple add operation, we can generate at most $\binom{|\mathcal{V}|}{2}$ unique embedding vectors for each intermediate node. This can be formulated as

$$m_{e_{ij}} = z_i + z_j. \quad (32)$$

□

Based on Theorem 4.2 and 4.3, we introduce the *relation message passing* mechanism, which is shown in Fig. 4. In relation message passing, we first synthesize the representation for intermediate node. Practically, the dimension of node embedding is $d << |\mathcal{V}|$. The node representation cannot be a one-hot embedding. Thus, simply adding them cannot generate unique embeddings for intermediate nodes. Besides, due to different types of nodes and edges, a simple adding operation in Equation 32 would ignore such semantic information. Based on the universal approximation theorem [47], we adopt the MLP to learn the mapping function f, which can be formulated as

$$m_{e_{ij}}^{l-1} = W_r(c_i^{l-1} || c_j^{l-1}), v_i, v_j \in N(v_{e_{ij}}), \quad (33)$$

where we use the relation-specific parameter W_r to learn mapping functions for different types of relations.

Then, we integrate the representation with context path aggregation to generate the final node embedding. This can be formulated as

$$c_i^l = W_2^l\Big(||_{h\in[1,H]}\sigma(W_1^l \sum_{r_{S,T}\in\mathcal{R}} \alpha_{S,T}^{h,l} \sum_{v_j\in N_S(i)} m_{e_{ij}}^{l-1} + B_1^l) + B_2^l\Big). \quad (34)$$

The relation message passing extends the context path and CP-GNN to measure the relevance between nodes of different types. Besides, it would not increase the storage complexity.

By using the relation message passing, we propose the CP-GNN+ for relevance measure in heterogeneous graphs. From Theorem 4.1, 4.2, and 4.3, we can easily have the following corollary.

Corollary 4.4. A k-layer CP-GNN+ can simulate the probability that pair-wise random walk meets at $(k+1)$-length context path.

The Corollary 4.4 theoretically ensures the effectiveness of CP-GNN+.

4.3 GSim for relevance measure

In this section, we introduce the process of GSim for relevance measure. Given a heterogeneous graph $\mathcal{H}$, we want to capture the semantics under different lengths. By defining a maximum length K, we obtain context information vectors of different lengths, which are formulated as

$$C^k = \text{k-layer CP-GNN+}(Z), k \in [1, K]. \quad (35)$$

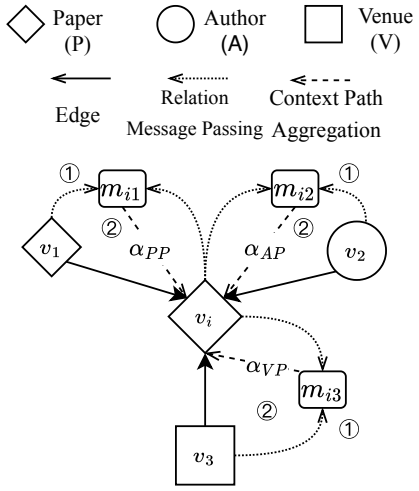


Fig. 4: The illustration of relation message passing. It first synthesizes the representation for intermediate nodes, and then integrates the representation with context path aggregation to generate the final node embedding.

Then, we propose a *type-length attention* $\alpha_A^k, A \in \mathcal{A}$ to differentiate the importance of various lengths for each node type, which is formulated as

$$H_A = \sum_{k=1}^{K} \alpha_A^k C_A^k, A \in \mathcal{A}, \quad (36)$$

$$H = \{H_A | A \in \mathcal{A}\}, \quad (37)$$

where the α_A^k is a learnable parameter that will be optimized in the training. The final relevance measure function S is formulated as

$$S(v_i, v_j) = \sigma(\langle h_i, h_j \rangle), h_i, h_j \in H, \quad (38)$$

where $H = \{H_A | A \in \mathcal{A}\}$, and σ denotes the sigmoid function.

To optimize the parameters in GSim, we adopt the framework of supervised contrast learning [48]. We assume that nodes with the same labeled class should be relevant to each other, and vice versa. By contrasting nodes within the same class and different classes, we can force relevant nodes closer to each other, while pushing away the irrelevant nodes. This is formulated as

$$\mathcal{L}_S = -\sum_{v_i\in\mathcal{V}} log \frac{\mathbb{E}\sum_{v_j\in I(v_i)} S(v_i, v_j)}{\mathbb{E}\sum_{v_j'\notin I(v_i)} 1 - S(v_i, v_j')}, \quad (39)$$

where $I(v_i)$ denotes a set of nodes sharing the same class of v_i. In addition, each node should be relevant to itself [6]. Thus, we propose the self-maximizing loss, which is formulated as

$$\mathcal{L}_U = trace\big(\sigma(H^\top H)\big) - \mathbf{I}. \quad (40)$$

The overall loss is formulated as

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_U. \quad (41)$$

4.4 Overall Algorithm

The overall training process of GSim is shown in Algorithm 1. Given a heterogeneous graph, we first randomly initialize the node embeddings Z (line 1), and assign all the type-length attention scores α_A^k to 1 (line 2). Then, we use a while-

Algorithm 1: The training process of GSim

Input: heterogeneous graph $\mathcal{H} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}\}$, Maximum length K. Max epoch ME
Output: Measurement function S

1 Randomly initialize the Z;
2 Initialize $\alpha_A^k \leftarrow 1, k \in [1, K], A \in \mathcal{A}$;
3 Epoch = 0;
4 **while** *Epoch* $< ME$ **do**
5 **for** $k = 1, \ldots, K$ **do**
6 $C^k =$ k-layer CP-GNN+(Z);
7 **end**
8 $H_A = \sum_{k=1}^{K} \alpha_A^k C_A^k, A \in \mathcal{A}$;
9 $H = \{H_A | A \in \mathcal{A}\}$;
10 $S(v_i, v_j) = \sigma(\langle h_i, h_j \rangle), h_i, h_j \in H$;
11 Optimize parameters using Equation 41;
12 Epoch ++;
13 **end**
14 **return** S

loop to finish the training process (lines 4-11). Specifically, we first adopt the K-layer CP-GNN+ to generate the context information vectors of different lengths C^k (lines 5-7). After that, we use the type-length attention to summarize them into the final node embedding H (lines 8-9). Finally, we use the relevance measure function in Equation 38 to measure the relevance between nodes (line 10), and the parameters in GSim will be optimized by Equation 41 (line 11).

In GSim, the computation complexity of each layer GNN is $\mathcal{O}(4|\mathcal{E}|)$. Thus, the overall computation complexity is $\mathcal{O}(4K|\mathcal{E}|)$, where K is the maximum length of context path. According to the statistic of average shortest paths in real-world networks [43], the K can be chosen between 4 to 6 empirically. Thanks to the relation message passing, the storage complexity of the graph is still $O(|\mathcal{V}| + |\mathcal{E}|)$.

5 Experiment

To evaluate the performance of our method, we conduct experiments on four real-world heterogeneous graphs with two downstream tasks: relevance search and community detection.

5.1 Dataset

We choose three widely used real-world heterogeneous graph datasets, i.e., ACM [14], DBLP [49], IMDB [50], together with a schema-rich knowledge graph dataset AIFB [51] to evaluate the performance of GSim and other baseline models. We report their statistics in Table 1, and discuss their details as follows.

- ACM dataset [14] is a bibliographic information network with four types of nodes (i.e., Paper, Author, Subject, and Facility). The paper nodes are categorized into 3 classes, i.e., *database*, *wireless communication* and *data mining*.
- DBLP dataset [49] is a monthly updated citation network consisting of four node types. Nodes are labeled with four classes, i.e., *database* (DB), *data mining* (DM), *information retrieval* (IR) and *machine learning* (ML).
- IMDB dataset [50] consists of four types of nodes. The movie nodes are labeled with three classes, i.e., *Action*, *Comedy*, and *Drama*.
- AIFB dataset [51] is a knowledge graph dataset consisting of 7 types of nodes and 104 types of edges. We choose the "Personen" node that is labeled with four classes. Due to the complexity of the graph, we do not provide detail illustration in Table 1, and predefine the meta-paths by ourselves.

TABLE 1: Statistics of datasets. The labeled node type are highlighted by $*$.

Dataset	Node type	# Nodes	Edge type	# Edges	Meta-path
ACM	Paper* (**P**)	12,499	Paper - Paper	30,789	
	Author (**A**)	17,431	Paper - Author	37,055	PAP
	Subject (**S**)	73	Paper - Subject	12,499	PSP
	Facility (**F**)	1,804	Author - Facility	30,424	
DBLP	Author* (**A**)	14,475	Author - Paper	41,794	APA
	Paper* (**P**)	14,736	Paper - Conference	14,736	APCPA
	Conference* (**C**)	20	Paper - Term	114,624	APTPA
	Term (**T**)	8,920			
IMDB	Movie* (**M**)	4,275	Movie - Actor	12,831	MAM
	Actor (**A**)	5,432	Movie - Director	4,181	MDM
	Director (**D**)	2,083	Movie - Keyword	20,428	MKM
	Keyword (**K**)	7,313			
AIFB	7 different types	Total 7,262	104 different types	Total 48,810	-

Because only the DBLP dataset provides labels for different types of nodes, we form two datasets (i.e., DBLP-A and DBLP-Multi) where we measure the relevance on author-only and multi-type nodes. For other datasets, we only use one type of labeled nodes for relevance measure.

We adopt the meta-paths in ACM, DBLP, and IMDP defined by previous works to evaluate the meta-path-based methods. Due to the rich-schema property of AIFB, we do not define the meta-path by ourselves. Therefore, some meta-path-based methods cannot be evaluated on the AIFB dataset. Besides, since the unsupervised methods do not need the training data, to make a fair comparison between the unsupervised and supervised methods, the splits of the labeled nodes with 25%/25%/50% for the training, validation, and testing.

5.2 Baselines

To evaluate the effectiveness of our measure, we compare it with a list of the state-of-the-art node embedding methods (i.e., Node2vec [10], Metapath2vec [34], and HIN2vec [52]) and GNN-based methods (i.e., GCN [12], GAT [37], LGNN [53], HAN [14], HGT [19], and CP-GNN [24]). Especially, HGT is a semi-supervised neural network model which adopts the transformer mechanism to differentiate the importance of relations. CP-GNN is the earlier version of our work that utilizes the context path to excavate semantics in heterogeneous graphs. We adopt Equation 38 on the learned embeddings to calculate the relevance.

Except for graph embedding-based and GNN-based baselines, we also choose two traditional relevance measure methods (i.e., SimRank [4] and HeteSim [6]) for comparison.

5.3 Parameters Settings

We now briefly discuss the settings of model parameters. For graph embedding-based approaches such as Node2vec

and Metapath2vec, we respectively set the length of random walk to 100, the sampling window size to 5, the number of walks per node to 120, and the number of negative samplings to 5. For GNN-based methods such as GCN, GAT, LGNN, HAN, and HGT, the number of graph convolution layers is set to 2, and their node features are first randomly initialized, and then updated during the model learning process. The dimension of node feature embedding for all compared methods is set to 128. We set these parameters following previous research [14], [19], [34].

For our GSim, the number of attention heads is set to 2, the dimension of the output vectors of K/Q-Linear components is set to 128, and the node dropout rate is set to 0.3. The maximum K is set to 4. The Adam [54] is adopted to optimize all models, and the learning rate is set to 0.05. We analyze the impact of these parameters in section 5.6.

5.4 Relevance Search Results

Another application of relevance measure is the relevance search. In relevance search, given a query node, we want to find its top-N relevant nodes. We adopt the average recall as the metric, which is computed as

$$recall@N = \frac{|r_i \in R \wedge f_I(r_i) = f_I(q)|}{N}, \tag{42}$$

where q denotes the query node, $f_I(\cdot)$ denotes the class label of node, and R denotes the set of top-N relevant results. In experiments, we randomly select 50 query nodes for each dataset, then we return top-10 nodes based on the measurement results. The experiment results are shown in Table 2.

From the results in Table 2, we can find that GSim outperforms other baselines on all datasets. However, comparing the GNN-based methods with the node embedding methods and conventional methods, we can see that GNN-based methods achieve inferior performance in the relevance search task. The possible reason could be that the GNN-based methods suffer from the over-smoothing problem [46]. This means that the final node representations tend to converge to the same value, making them hard to be distinguished. Therefore, the relative relevance between nodes is erased, resulting in weak performance in relevance search.

As for GSim, it adopts the supervised contrast learning to contrast nodes within the same class and different classes during training. This will force relevant nodes closer to each other while pushing away the irrelevant nodes. In this way, GSim enables us to capture the relative relevance between nodes and achieve the best performance in relevance search.

5.4.1 Case Study 1: Find same type relevant nodes

In this study, we illustrate the top-10 nodes of the same type as the query node. We first query the top-10 relevant movies for the movie "Twilight" in IMDB dataset, which is one of the most famous drama movies. The results are shown in Table 3. We respectively list the names, relevance scores, and labels of each movie found.

From the results, we can see that the results returned by GSim are better than the other methods, of which the "Drama" movies are the most (i.e., 8/10). On the contrary, other baselines contain many "Adventure" and "Action" movies in their results. Specifically, SimRank returns the most diverse results containing movies of different genera. HeteSim returns "The Twilight Sega: New Moon" and "The Twilight Saga: Eclipse", which are the sequels to "Twilight". The possible reason is that these movies share similar actors, and such relations are captured by the MAM meta-path. But these actors might also act in movies of other genera, which deteriorates the search results. HGT tries to automatically discover the semantics, but the found semantics might not be suitable for the query node. Thus, it returns lots of "Adventure" movies and the results are not very distinctive. GSim adopts the context path to excavate the semantics that is crucial to the query node. For example, the 4th and 6th results ("Blood and Chocolate" and "Blood Ties") both contain horror and action scenes which are also the major themes of "Twilight". These underlying relations cannot be well revealed by the meta-path.

TABLE 2: Performance of different measures on relevance search task.

Method	ACM	DBLP-A	DBLP-Multi	IMDB	AIFB
	Recall				
SimRank	0.450	0.826	0.850	0.430	0.412
Hetesim	0.646	0.358	0.367	0.410	-
Node2Vec	0.450	0.556	0.497	0.520	0.330
Metapath2vec	0.468	0.292	0.751	0.484	-
GCN	0.382	0.312	0.354	0.382	0.455
GAT	0.436	0.470	0.307	0.332	0.396
LGNN	0.528	0.272	0.293	0.278	0.594
HAN	0.534	0.286	-	0.512	-
HGT	0.504	0.472	0.710	0.388	0.626
CP-GNN	0.646	0.644	-	0.454	0.356
GSim	**0.782**	**0.900**	**0.888**	**0.524**	**0.664**

In Table 4, we try to find relevant authors for author "Weidong Chen" who is a researcher in the database area (DB). From the results, we can see that all the authors found by GSim are in DB area, whereas other baselines return some authors in the information retrieval (IR), data mining (DM), and artificial intelligence (AI) areas. In addition, we can see that the first result returned by the HGT is not "Weidong Chen". This shows that HGT does not satisfy the self-maximizing property which is important in previous methods (SimRank and HeteSim). Thanks to the self-maximizing loss shown in Equation 40, GSim enables to make sure that each node is most relevant to itself.

5.4.2 Case Study 2: Find different types relevant nodes

In relevance measure, we need to evaluate the relevance of different types of nodes. Therefore, in DBLP-Multi dataset, given an author "Weidong Chen", we want to find the relevant conferences, authors, and papers for him. The top-5 results of different types of nodes are shown in Table 5. Due to the limitation of space, we replace the actual paper names with P1-P5.

From the results in Table 5, we can see that SimRank still achieves the worst results. The reason is that SimRank disregards the node type in heterogeneous graphs. Thus, it cannot find relations between different types of nodes. HeteSim behaves better than SimRank. But it requires defining different meta-paths for different types of nodes. It is

TABLE 3: Top-10 query results for movie:"Twilight" (label: Drama) on IMDB dataset.

Rank	SimRank			HeteSim (MAM)			HGT			GSim		
	Movie	Score	Label	Movie	Score	Label	Author	Score	Label	Movie	Score	Label
1	**Twilight**	0.148	Drama	**Twilight**	1.000	Drama	Trash	1.000	Drama	**Twilight**	1.000	Drama
2	We Need to Talk About Kevin	0.055	Drama	The Twilight Saga: New Moon	0.667	Drama	Modern Problems	1.000	Adventure	Shallow Hal	1.000	Adventure
3	Gangster Squad	0.026	Action	The Twilight Saga: Eclipse	0.667	Drama	Unleashed	1.000	Action	The Bridges of Madison County	1.000	Drama
4	A Thin Line Between Love and Hate	0.010	Adventure	Zathura: A Space Adventure	0.333	Action	The Other Woman	1.000	Adventure	Blood and Chocolate	1.000	Drama
5	Get Carter	0.007	Action	What to Expect When You're Expecting	0.333	Adventure	Pain & Gain	1.000	Adventure	A Walk to Remember	1.000	Drama
6	Girls Gone Dead	0.005	Adventure	Drinking Buddies	0.333	Adventure	Rio	1.000	Adventure	Blood Ties	1.000	Drama
7	Jab Tak Hai Jaan	0.005	Drama	The Ridiculous 6	0.333	Adventure	Guess Who	1.000	Adventure	The Brothers Bloom	1.000	Adventure
8	Red Dog	0.005	Adventure	Welcome to the Rileys	0.333	Drama	Faithful	1.000	Adventure	Titanic	1.000	Drama
9	Captain America: The First Avenger	0.004	Action	The Last Five Years	0.333	Adventure	Inside Out	1.000	Adventure	Womb	1.000	Drama
10	Ironclad	0.004	Action	On the Road	0.333	Drama	Death Sentence	1.000	Action	Code 46	1.000	Drama

TABLE 4: Top-10 query results for author: "Weidong Chen" (label: DB) on DBLP-A dataset.

Rank	SimRank			HeteSim (APA)			HGT			GSim		
	Author	Score	Label	Author	Score	Label	Author	Score	Label	Author	Score	Label
1	**Weidong Chen**	0.129	DB	**Weidong Chen**	1.000	DB	Matthew Denny	1.000	DB	**Weidong Chen**	0.982	DB
2	Serge Rielau	0.007	DB	Serge Rielau	0.224	DB	Danette Chimenti	1.000	DB	Kurt Ingenthron	0.926	DB
3	Ben Hutchinson	0.002	AI	Eliezer Levy	0.000	DB	Christian Zimmer	1.000	IR	Adam Silberstein	0.924	DB
4	Eckhard D. Falkenberg	0.002	DB	Richard Sidle	0.000	DB	Volker Linnemann	1.000	DB	David E. Bakkom	0.924	DB
5	Hal R. Varian	0.002	IR	Roger Nasr	0.000	IR	Mihalis Yannakakis	1.000	DB	Xin Luna Dong	0.923	DB
6	Balder ten Cate	0.001	DB	Keizo Oyama	0.000	IR	Stephen C. North	1.000	DM	Pavel Avgustinov	0.923	DB
7	Bennet Vance	0.001	DB	Hai Leong Chieu	0.000	IR	Cyril Goutte	1.000	IR	Jen-Yao Chung	0.921	DB
8	Stephen J. Hegner	0.001	DB	Mark Coyle	0.000	DB	Armin B. Cremers	1.000	IR	H. V. Jagadish	0.921	DB
9	Amit Ramesh	0.001	AI	Fabrizio Angiulli	0.000	DM	Bruno Defude	1.000	IR	Mary Tork Roth	0.920	DB
10	Marc Spielmann	0.001	DB	Denver Dash	0.000	AI	Shanshan Wang	1.000	DM	Nancy D. Griffeth	0.920	DB

TABLE 5: Top-5 multi-type query results for author: "Weidong Chen" (label: DB) on DBLP-Multi dataset. (Due to the limitation of space, we replace the actual paper names with P1-P5.)

Method	SimRank						HeteSim					
Rank	Conference	Label	Author	Label	Paper	Label	Conference (APC)	Label	Author (APA)	Label	Paper (APAP)	Label
1	SDM	DM	Weidong Chen	DB	P1	DB	VLDB	DB	Weidong Chen	DB	P1	DB
2	EDBT	DB	Serge Rielau	DB	P2	DB	SDM	DM	Serge Rielau	DB	P2	DB
3	ECML	AI	Ben Hutchinson	AI	P3	IR	EDBT	DB	Eliezer Levy	DB	P3	IR
4	VLDB	DB	Eckhard D. Falkenberg	DB	P4	DB	ECML	AI	Richard Sidle	DB	P4	DB
5	ICML	AI	Hal R. Varian	IR	P5	DM	ICML	AI	Roger Nasr	IR	P5	DM
Method	HGT						GSim					
Rank	Conference	Label	Author	Label	Paper	Label	Conference	Label	Author	Label	Paper	Label
1	VLDB	DB	Matthew Denny	DB	P1	DB	WSDM	IR	Weidong Chen	DB	P1	DB
2	SDM	DM	Danette Chimenti	DB	P2	DB	ICML	AI	Kurt Ingenthron	DB	P2	DB
3	EDBT	DB	Christian Zimmer	IR	P3	IR	EDBT	DB	Adam Silberstein	DB	P3	DB
4	ECML	AI	Volker Linnemann	DB	P4	DB	SDM	DM	David E. Bakkom	DB	P4	DB
5	ICML	AI	Mihalis Yannakakis	DB	P5	DM	SIGIR	IR	Xin Luna Dong	DB	P5	DB

infeasible when the number of node types increases. On the contrary, HGT can find the relationship between different types of nodes when calculating their relevance. But it fails to capture the asymmetric semantics, thus performing poorly for the relevance measure between *Author-Paper*. GSim adopts the relation message passing, which could comprehensively capture both the asymmetric and symmetric semantics and reach great results in relevance measures between *Author-Paper* and *Author-Author*. However, GSim fails on the *Author-Conference*. The possible reason is that there are only 5 labeled conference nodes in DBLP-Multi dataset for training. Due to the limited number of training nodes, GSim cannot fully learn the relevance of conference nodes.

5.5 Community Detection Results

Relevance measure plays an essential role in community detection. Following previous research [6], [31], we apply the Spectral Clustering [55] on the relevance scores generated by different methods to perform the community detection. We choose the widely used F-score, NMI, ARI, and Purity as the metrics. The results are reported in Table 6, where the best results are highlighted in bold and the second-best results are labeled with underlines.

From the results in Table 6, we can see that GSim outperforms other baselines on most datasets concerning different metrics. This demonstrates that GSim can generate high-quality measure results via automatically excavating semantics in heterogeneous graphs.

As for node embedding methods (i.e., Node2vec and Metapath2vec), they perform better than the conventional measure methods (i.e., SimRank and HeteSim). This indicates that the learned node embedding can successfully consider the structure information in the graph which is essential for the relevance measure. Besides, the Metapath2vec only beats Node2vec in ACM and DBLP-Multi datasets. This shows that even when semantics is considered by adopting meta-paths, the choice of meta-path will also severely influence the performance of Metapath2vec. Additionally, the Metapath2vec uses only one meta-path, which cannot capture the semantics comprehensively.

For GNN-based methods, we can see that the GCN, GAT,

TABLE 6: Performance of different measures on community detection task.

Method	ACM				DBLP-A				DBLP-Multi				IMDB				AIFB			
	F-score	NMI	ARI	Purity	F-score	NMI	ARI	Purity	F-score	NMI	ARI	Purity	F-score	NMI	ARI	Purity	F-score	NMI	ARI	Purity
SimRank	0.5395	0.0033	0.0003	0.4908	0.5055	0.3798	0.2125	0.4993	0.404	0.0085	0	0.3078	**0.5101**	0.0034	-0.0003	0.3803	0.3457	0.0648	-0.004	0.4773
HeteSim	0.5353	0.0038	0.003	0.4933	0.3978	0.0058	-0.002	0.3075	0.4092	0.0083	0.0011	0.3030	0.5070	0.0015	-0.0003	0.3804		-		
Node2Vec	0.3506	0.1324	0.1056	0.5413	0.3196	0.0972	0.08	0.4352	0.3164	0.0795	0.0742	0.4313	0.3938	0.0777	0.0817	0.4920	0.3834	0.2011	0.1863	0.6013
Metapath2vec	0.4397	0.1586	0.1276	0.5718	0.3093	0.0739	0.0484	0.3982	0.3536	0.2834	0.2672	0.4012	0.3761	0.0219	0.0186	0.4387		-		
GCN	0.4090	0.0375	0.0385	0.5161	0.2616	0.0071	0.0048	0.3144	0.2754	0.0101	0.0123	0.3236	0.3539	0.0013	0.0006	0.3934	0.4157	0.1444	0.1325	0.5795
GAT	0.5595	0.1676	0.0926	0.5693	0.6439	0.5174	0.5087	0.7536	0.3945	0.1561	0.0925	0.4294	0.495	0.0002	-0.0004	0.3803	0.4506	0.2286	0.1935	0.6023
LGNN	0.6243	0.4013	0.4026	0.7139	0.4071	0.0088	-0.0009	0.3016	0.3959	0.0073	-0.0007	0.3030	0.4854	0.0156	-0.008	0.3803	0.4572	0.2437	0.1166	0.6136
HAN	0.6379	0.4062	0.4369	0.7695	0.2774	0.0028	0.0011	0.3135			-		0.3832	0.065	0.0678	0.5005		-		
HGT	0.5589	0.2847	0.2632	0.6239	0.6721	0.5853	0.5548	0.7496	0.7797	0.6754	0.7025	0.8679	0.4941	0.0013	0	0.3798	0.4471	0.2688	0.1902	0.6250
CP-GNN	0.6170	0.4199	0.3718	0.7069	0.4507	0.2738	0.2597	0.6175			-		0.405	0.0899	0.096	0.5047	0.3836	0.1146	0.1034	0.5341
GSim	**0.6968**	**0.5065**	**0.5136**	**0.8058**	**0.8820**	**0.7857**	**0.8411**	**0.9354**	**0.8620**	**0.7539**	**0.8140**	**0.9239**	0.4304	**0.1181**	**0.1234**	**0.5697**	**0.6688**	**0.5001**	**0.4855**	**0.7386**

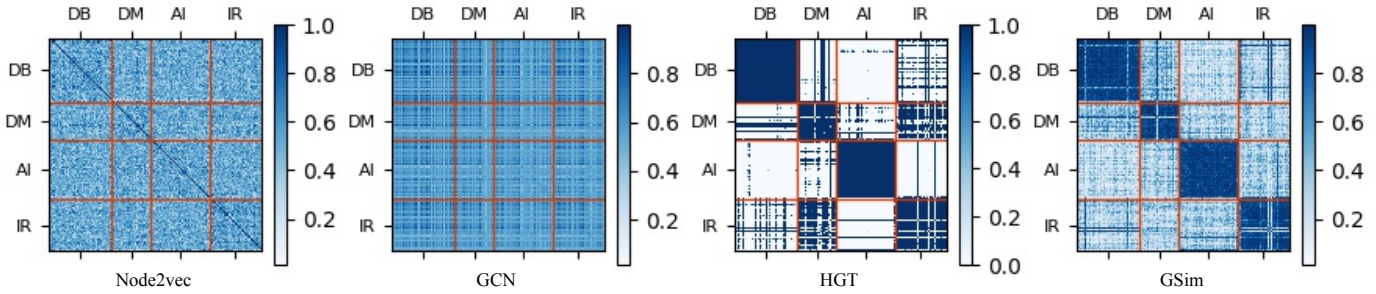


Fig. 5: The relevance matrices generated by different methods on DBLP-Multi dataset. (DB: "Database", DM: "Data Mining", AI: "Artificial Intelligence", IR: "Information retrieval").

and LGNN achieve relatively worse results. The possible reason is that they are originally proposed for homogeneous graphs, thus they do not consider the complex semantics in heterogeneous graphs. GAT performs better than GCN, which reflects the importance of the attention mechanism. The attention mechanism used in GAT can be regarded as a simple way to differentiate the node type and edge type in heterogeneous graphs. Thanks to the meta-path, HAN can explicitly excavate complex semantic information to achieve a better result. But due to the symmetry constraint of meta-path, HAN cannot handle the relevance measure between multiple node types in DBLP-Multi. On the other hand, HGT and CP-GNN do not require the meta-path. They adopt the designed attention mechanism to automatically capture the semantics in graphs, which helps them outperform other GNN-based methods in most datasets. But due to the constraint of context path, CP-GNN cannot be used in DBLP-Multi.

For GSim, it adopts the context path to spontaneously leverage the semantics in the graph and utilizes the relation attention to differentiate their importance. Besides, thanks to the relation message passing, GSim can capture both the symmetric and asymmetric semantics, thus it can measure the relevance between any node type.

5.5.1 Case Study 3: Relevance matrix visualization

To intuitively understand the relevance measure results of community detection, we first visualize the relevance matrices generated by different methods on the DBLP-Multi dataset in Fig. 5. Then, we selected several authors with different labels from DBLP-Multi and illustrate the relevance scores between them in Fig. 6.

In Fig. 5, we demonstrate the relevance matrices generated by Node2vec, GCN, HGT, and GSim. Each entry of the relevance matrix depicts the relevance score between two nodes. The darker the color, the more relevant the two nodes are. The nodes in DBLP-Multi can be classified into four classes (i.e., DB: "Database", DM: "Data Mining", AI: "Artificial Intelligence", IR: "Information retrieval"). Therefore, we group the nodes of the same class and divide them using orange lines.

From the results in Fig. 5, we can see that Node2vec and GCN generate inferior results. The relevance matrix of Node2vec is quite random except for the diagonal entries. This shows that Node2vec fails to consider the complex semantics in heterogeneous graphs and only captures simple structure information. Thus, it only captures the relevance between nodes and themselves. GCN generates a quite smooth relevance matrix, where each node shares a similar relevance score. This is because GCN adopts the graph convolutional layer that would smoothen the node embeddings. Thus, node embedding of each node will become similar and the relevance is measured inaccurately following the embeddings. The results of HGT are better, where nodes of the same class have higher relevant scores. However, it still has some error results. For example, HGT generates high relevance scores for some nodes between IR and DM. GSim achieves relative better results, where nodes within the same classes are highly related whereas nodes between different classes have lower relevance scores.

To better demonstrate the relevance measure results, we select several authors with different labels and use the GSim to measure their relevance scores. In Fig. 6, we can see that each author is most relevant to himself, which is consistent with the self-maximizing property. Then, we can see that authors with the same labels are highly related. For example, in the area of DM or IR, authors are highly relevant to each other. In addition, GSim can also capture some underlying

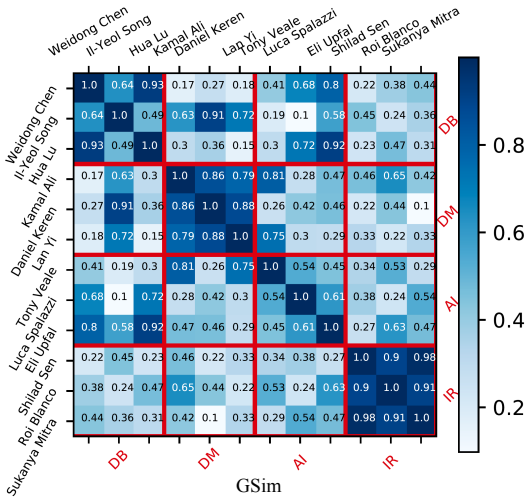


Fig. 6: Relevance matrix between selected authors in DBLP-Multi dataset.

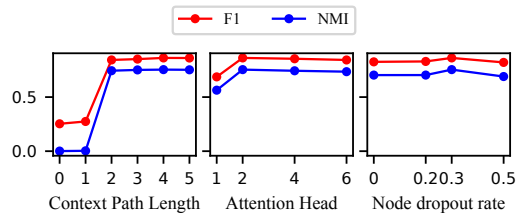


Fig. 7: Parameter sensitivity w.r.t. different parameters.

relevance between nodes. For instance, the author "Il-Yeol Song" is labeled with DB. However, he has also published lots of papers in the data mining area. Thus, he has relatively high relevance scores with authors in the DM area. From the aforementioned results, we can see that GSim can not only accurately discover the relevance between nodes but is also able to capture the hidden relevance.

5.6 Parameters Analysis and Ablation Study

In this section, we discuss the parameters analysis and ablation study. As shown in Fig. 7, with the increase of parameter values, GSim's performances raise first and then drop slightly; the best performances are reached when the Context Path Length, Attention head, and Node dropout rate reach 4, 8, and 0.3, respectively. The reason is that an over large context path length may contain redundant semantics which deteriorates the performance. Although more attention heads can capture more diverse relation importance and increase the representation ability, they also introduce more parameters to the model, making it hard to train. Besides, too many nodes are dropped, which causes graph summary vectors cannot be well generated from the remaining nodes.

In Table 7, we study the impact of balance between $\mathcal{L}_S$ and $\mathcal{L}_U$ in Equation 41. We can see that the performance of GSim is the best when the balance is 1:1. This is because the two losses are complementary to each other and need to be equally considered during the training. When the balance is 0:1, the model only focuses on self-relevnace, which largely impairs performance. When the balance is 1:0, the model ignores the self-relevance, which also deteriorates the performance. Besides, reducing the weight ratio of both $\mathcal{L}_S$ and $\mathcal{L}_U$ results in a decrease in performance, emphasizing their equal importance in achieving optimal results.

TABLE 7: Balance between $\mathcal{L}_S$ and $\mathcal{L}_U$.

$\mathcal{L}_S : \mathcal{L}_U$	F1	NMI	ARI	Purity
1:1	**0.8856**	**0.7914**	**0.8539**	**0.9404**
0:1	0.2731	0.0285	0.0252	0.3736
1:0	0.8801	0.7808	0.8386	0.9345
0.8:1	0.8827	0.7857	0.8421	0.9354
0.6:1	0.8853	0.7911	0.8476	0.9379
1:0.8	0.8807	0.7821	0.8393	0.9349
1:0.6	0.8803	0.7810	0.8387	0.9340

TABLE 8: Effectiveness of different components.

Method	NMI	ARI
GSim	**0.8620**	**0.7539**
w/o type-length attention	0.8574	0.7497
w/o relation attention	0.8405	0.7318
w/o relation message passing	0.8284	0.7104

In Table 8, to evaluate the effectiveness of different components, we gradually remove the *type-length attention*, *relation attention*, and *relation message passing*. From the results, we can see that the performance of the model decreases with components removed. Specifically, without the type-length attention and relation attention, GSim fails to differentiate the importance of different relations. Moreover, without the relation message passing, GSim cannot capture the asymmetric semantics as analyzed in section 4.2.3.

5.7 Impact of Relation Message Passing

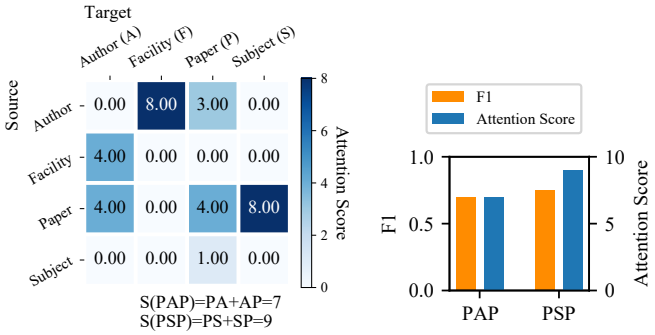
In this section, we further study the impact of relation message passing. Relation message passing is proposed to measure nodes of heterogeneous types without increasing complexity. We show the improvement of GPU memory usage and training time brought by relation message passing in Table 9. From Table 9, we can see that, by using relation message passing, memory usage and training time are reduced by 68.33% and 63.63%, respectively. This is because the model w/o relation message passing needs $2K$ layers to achieve the same results after adding intermediate nodes, which introduces additional memory consumption and training time.

5.8 Relation Attention Visualization

To further analyze whether GSim can differentiate the context paths, we first present the corresponding relations at-

TABLE 9: Improvement of memory usage and training time brought by relation message passing.

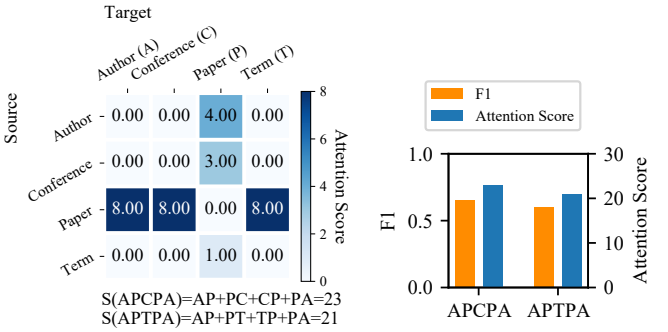
Method	Memory (Mb)	Time/Epoch (s)
w/o relation message passing	7090	0.33
GSim	2245	0.12
Improvement	↓ 68.33%	↓ 63.63%



a Attention matrix of 1-length context path on ACM.

b Metapath2vec F1 values on ACM.

Fig. 8: Visualization of the context relation attention matrix on ACM.



a Attention matrix of 3-length context path on DBLP.

b Metapath2vec F1 values on DBLP.

Fig. 9: Visualization of the context relation attention matrix on DBLP.

tention matrix acquired from GSim in Fig. 8a and 9a, where each entry is the attention score of the relation with a source node type and a target node type. The attention score of each context path can be computed by summarizing the scores of its relations. Then, to justify whether the paths with higher attention scores are more meaningful for community detection, we adopt the Metapath2vec to evaluate the effect of each path. The results are shown in Fig. 8b and 9b

For example, the paths PAP and PSP in ACM are both 2-length context path. Therefore, there attention scores can be computed from the relation attention matrix of 2-length context path shown in Fig. 8a where $S(PAP) = PA + AP = 7$ and $S(PSP) = PS + SP = 9$. Clearly, we can find that the attention score of PSP is higher than PAP, which means the path PSP is slightly more important than PAP for community detection. This can be justified by the result shown in Fig. 8b where the F1 score of PSP is higher than PAP. Similarly, from Fig. 9a, the attention scores of paths APCPA and APTPA are $S(APCPA) = AP + PC + CP + PA = 23$ and $S(APTPA) = AP + PT + TP + PA = 21$. This indicates that the relationship reflected by APCPA is a little bit more important than that of APTPA. This finding can be proved by the result shown in Fig. 9b where the F1 of APCPA is higher than APTAP.

In summary, the above analysis demonstrates that relation attention can discover many context paths of different importance and capture the context path of higher importance that are more meaningful and useful for relevance measure.

6 Conclusion

In this paper, we propose a context path-based graph neural network model for relevance measure in heterogeneous graphs, called GSim. GSim can automatically capture the semantics among nodes, and differentiate their importance. Furthermore, we introduce the relation message passing mechanism that enables GSim to measure relevance between nodes with different types. The effectiveness of GSim is guaranteed by both theoretical analysis and empirical study. Extensive experiments on four real-world datasets show that GSim outperforms other baselines and reaches the best performance in both the community detection and relevance search task.

In the future, we will consider the relevance measure in the unsupervised setting. Besides, we will try to explore the potential of GSim for other graph mining tasks, such as node importance measure and node ranking. These will require us to consider the fine-grinned relative relevance in the heterogeneous graphs.

References

[1] G. Jeh and J. Widom, "Scaling personalized web search," in *WWW*, 2003, pp. 271–279.
[2] M. S. Pera and Y.-K. Ng, "A group recommender for movies based on content similarity and popularity," *Information Processing & Management*, vol. 49, no. 3, pp. 673–687, 2013.
[3] F. D. Zarandi and M. K. Rafsanjani, "Community detection in complex networks using structural similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 882–891, 2018.
[4] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *KDD*, 2002, pp. 538–543.
[5] P. Zhao, J. Han, and Y. Sun, "P-rank: a comprehensive structural similarity measure over information networks," in *CIKM*, 2009, pp. 553–562.
[6] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *TKDE*, vol. 26, no. 10, pp. 2479–2492, 2014.
[7] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *PVLDP*, vol. 4, no. 11, pp. 992–1003, 2011.
[8] C. Wang, Y. Sun, Y. Song, J. Han, Y. Song, L. Wang, and M. Zhang, "Relsim: relation similarity search in schema-rich heterogeneous information networks," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 621–629.
[9] D. Xiao, X. Meng, Y. Li, C. Shi, and B. Wu, "Avgsim: Relevance measurement on massive data in heterogeneous networks." *Journal of Theoretical & Applied Information Technology*, vol. 84, no. 1, 2016.
[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *SIGKDD*, 2016, pp. 855–864.
[11] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, "Meta-path guided embedding for similarity search in large-scale heterogeneous information networks," *arXiv preprint arXiv:1610.09769*, 2016.
[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[13] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.
[14] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

[15] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, "Interpretable and efficient heterogeneous graph convolutional network," *TKDE*, 2021.
[16] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2018.
[17] J. You *et al.*, "Identity-aware graph neural networks," in *AAAI*, 2021.
[18] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, and A. Mian, "Bi-clkt: Bi-graph contrastive learning based knowledge tracing," *Knowledge-Based Systems*, vol. 241, p. 108274, 2022.
[19] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *WWW*, 2020, pp. 2704–2710.
[20] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *TNNLS*, vol. 33, no. 6, pp. 2378–2392, 2021.
[21] D. Jiang, Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, and T. Hou, "Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models," *Journal of cheminformatics*, vol. 13, no. 1, pp. 1–23, 2021.
[22] L. Luo, K. Liu, D. Peng, Y. Ying, and X. Zhang, "A motif-based graph neural network to reciprocal recommendation for online dating," in *ICONIP*. Springer, 2020, pp. 102–114.
[23] L. Yang, L. Luo, X. Zhang, F. Li, X. Zhang, Z. Jiang, and S. Tang, "Why do semantically unrelated categories appear in the same session? a demand-aware method," in *SIGIR*, 2022, pp. 2065–2069.
[24] L. Luo, Y. Fang, X. Cao, X. Zhang, and W. Zhang, "Detecting communities from heterogeneous graphs: A context path-based graph neural network model," in *CIKM*, 2021, pp. 1170–1180.
[25] D. Barman, S. Bhattacharya, R. Sarkar, and N. Chowdhury, "k-context technique: A method for identifying dense subgraphs in a heterogeneous information network," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1190–1205, 2019.
[26] S. Rothe and H. Schütze, "Cosimrank: A flexible & efficient graph-theoretic similarity measure," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1392–1402.
[27] D. Li, Z. Gu, Y. Wang, C. Ren, and F. C. Lau, "One model packs thousands of items with recurrent conditional query learning," *Knowledge-Based Systems*, vol. 235, p. 107683, 2022.
[28] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, and Y. Zheng, "Semi-supervised clustering in attributed heterogeneous information networks," in *WWW*, 2017, pp. 1621–1629.
[29] S. Zhou, J. Bu, Z. Zhang, C. Wang, L. Ma, and J. Zhang, "Cross multi-type objects clustering in attributed heterogeneous information network," *Knowledge-Based Systems*, vol. 194, p. 105458, 2020.
[30] A. Conte, G. Ferraro, R. Grossi, A. Marino, K. Sadakane, and T. Uno, "Node similarity with q-grams for real-world labeled networks," in *KDD*, 2018, pp. 1282–1291.
[31] Y. Wang, Z. Wang, Z. Zhao, Z. Li, X. Jian, H. Xin, L. Chen, J. Song, Z. Chen, and M. Zhao, "Effective similarity search on heterogeneous networks: A meta-path free approach," *TKDE*, 2020.
[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.
[33] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.
[34] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017, pp. 135–144.
[35] C. Xu, W. Zhao, J. Zhao, Z. Guan, X. Song, and J. Li, "Uncertainty-aware multi-view deep learning for internet of things applications," *IEEE Transactions on Industrial Informatics*, 2022.
[36] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *Trans. Neur. Netw.*, vol. 20, no. 1, pp. 61–80, 2009.
[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
[38] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, "Metapath-guided heterogeneous graph neural network for intent recommendation," in *KDD*, 2019, pp. 2478–2486.
[39] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *WWW*, 2020, pp. 2331–2341.
[40] A. Singhal *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.
[41] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, pp. 34–35, 1971.
[42] P. L. Duren, *Theory of H p Spaces*. Academic press, 1970.
[43] Q. Ye *et al.*, "Distance distribution and average shortest path length estimation in real-world networks," in *ADMA*, 2010, pp. 322–333.
[44] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax," *arXiv preprint arXiv:1911.08538*, 2019.
[45] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
[46] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *AAAI*, vol. 34, no. 04, 2020, pp. 3438–3445.
[47] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
[48] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *NIPS*, vol. 33, pp. 18 661–18 673, 2020.
[49] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, "Graph-based consensus maximization among multiple supervised and unsupervised models," in *NIPS*, 2009, pp. 585–593.
[50] C.-Y. Wu, A. Beutel, A. Ahmed, and A. J. Smola, "Explaining reviews and ratings with paco: Poisson additive co-clustering," in *WWW*, 2016, pp. 127–128.
[51] P. Ristoski, G. K. D. De Vries, and H. Paulheim, "A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web," in *International Semantic Web Conference*. Springer, 2016, pp. 186–194.
[52] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *CIKM*, 2017, pp. 1797–1806.
[53] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *ICLR*, 2019.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[55] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

Linhao Luo received the bachelor degree from the Harbin Institute of Technology, Shenzhen in 2021. He is now a PhD student in the faculty of information and technology at Monash University. His research interests include machine learning, data mining, and graph neural networks.

Yixiang Fang received the Ph.D. degree from the University of Hong Kong (HKU) in 2017. Currently, he is an associate professor in the School of Data Science at the Chinese University of Hong Kong, Shenzhen. His research interests mainly focus on the areas of data management, data mining, and artificial intelligence over big data.

Moli Lu is a master student in the Harbin Institute of Technology, Shenzhen. Her research interests focus on graph mining.

Xin Cao is a senior lecturer in the School of Computer Science and Engineering at the University of New South Wales. His research interests include database management, data mining, big data analytics, and artificial intelligence.

Xiaofeng Zhang is currently an associate professor with the Department of Computer Science, Harbin Institute of Technology, Shenzhen. His research interests include data mining, machine learning, and graph mining.

Wenjie Zhang is a Professor in the School of Computer Science and Engineering at the University of New South Wales. Her research interests lie in data management and analytics for large-scale data, especially graph, spatial-temporal, and image data. She serves as an Associate Editor for TKDE, Associate Editor for PVLDB 2022, and (senior) PC member for leading conferences in database and data mining.