Heterogeneous Graph Attention Network

Xiao Wang, Houye Ji
Beijing University of Posts and Telecommunications
Beijing, China
{xiaowang,jhy1993}@bupt.edu.cn

Chuan Shi*, Bai Wang
Beijing University of Posts and Telecommunications
Beijing, China
{shichuan,wangbai}@bupt.edu.cn

Peng Cui, P. Yu
Tsinghua University
Beijing, China
{cuip,psyu}@tsinghua.edu.cn

Yanfang Ye
West Virginia University
WV, USA
yanfang.ye@mail.wvu.edu

ABSTRACT

Graph neural network, as a powerful graph representation technique based on deep learning, has shown superior performance and attracted considerable research interest. However, it has not been fully considered in graph neural network for heterogeneous graph which contains different types of nodes and links. The heterogeneity and rich semantic information bring great challenges for designing a graph neural network for heterogeneous graph. Recently, one of the most exciting advancements in deep learning is the attention mechanism, whose great potential has been well demonstrated in various areas. In this paper, we first propose a novel heterogeneous graph neural network based on the hierarchical attention, including node-level and semantic-level attentions. Specifically, the node-level attention aims to learn the importance between a node and its meta-path based neighbors, while the semantic-level attention is able to learn the importance of different meta-paths. With the learned importance from both node-level and semantic-level attention, the importance of node and meta-path can be fully considered. Then the proposed model can generate node embedding by aggregating features from meta-path based neighbors in a hierarchical manner. Extensive experimental results on three real-world heterogeneous graphs not only show the superior performance of our proposed model over the state-of-the-arts, but also demonstrate its potentially good interpretability for graph analysis.

KEYWORDS

Social Network, Neural Network, Graph Analysis

1 INTRODUCTION

The real-world data usually come together with the graph structure, such as social networks, citation networks, and the world wide web. Graph neural network (GNN), as a powerful deep representation learning method for such graph data, has shown superior performance on network analysis and aroused considerable research interest. For example, [10, 20, 24] leverage deep neural network to learn node representations based on node features and the graph structure. Some works [6, 14, 18] propose the graph convolutional networks by generalizing the convolutional operation to graph. A recent research trend in deep learning is the attention mechanism, which deals with variable sized data and encourages the model to focus on the most salient parts of data. It has demonstrated the effectiveness in deep neural network framework and is widely applied to various applications, such as text analysis [1], knowledge graph [25] and image processing [38]. Graph Attention Network (GAT) [35], a novel convolution-style graph neural network, leverages attention mechanism for the homogeneous graph which includes only one type of nodes or links.

Despite the success of attention mechanism in deep learning, it has not been considered in the graph neural network framework for heterogeneous graph. As a matter of fact, the real-world graph usually comes with multi-types of nodes and edges, also widely known as heterogeneous information network (HIN) [28]. For convenience, we uniformly call it heterogeneous graph in this paper. Because the heterogeneous graph contains more comprehensive information and rich semantics, it has been widely used in many data mining tasks. Meta-path [32], a composite relation connecting two objects, is a widely used structure to capture the semantics. Taking the movie data IMDB[1] shown in Figure 1(a) as an example, it contains three types of nodes include movie, actor and director. A relation between two movies can be revealed by meta-path Movie-Actor-Movie (*MAM*) which describes the co-actor relation, while Movie-Director-Movie (*MDM*) means that they are directed by the same director. As can be seen, depending on the meta-paths, the relation between nodes in the heterogeneous graph can have different semantics. Due to the complexity of heterogeneous graph, traditional graph neural network cannot be directly applied to heterogeneous graph.

Based on the above analysis, when designing graph neural network architecture with attention mechanism for heterogeneous graph, we need to address the following new requirements.

Heterogeneity of graph. The heterogeneity is an intrinsic property of heterogeneous graph, i.e., various types of nodes and edges. For example, different types of nodes have different traits and their features may fall in different feature space. Still taking IMDB as
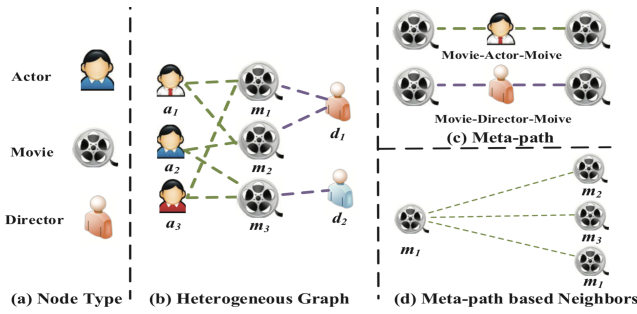
---

[1] https://www.imdb.com

Figure 1: An illustrative example of a heterogenous graph (IMDB). (a) Three types of nodes (i.e., actor, movie, director). (b) A heterogenous graph IMDB consists three types of nodes and two types of connections. (c) Two meta-paths involved in IMDB (i.e., Moive-Actor-Moive and Movie-Director-Movie). (d) Moive m_1 and its meta-path based neighbors (i.e., m_1, m_2 and m_3).

an example, the feature of an actor may involve in sex, age and nationality. On the other hand, the feature of movie may involve in plot and actors. How to handle such complex structural information and preserve the diverse feature information simultaneously is an urgent problem that needs to be solved.

Semantic-level attention. Different meaningful and complex semantic information are involved in heterogeneous graph, which are usually reflected by meta-paths [32]. Different meta-paths in heterogeneous graph may extract diverse semantic information. How to select the most meaningful meta-paths and fuse the semantic information for the specific task is an open problem [4, 19, 26]. Semantic-level attention aims to learn the importance of each meta-path and assign proper weights to them. Still taking IMDB as an example, *The Terminator* can either connect to *The Terminator 2* via Movie-Actor-Movie (both starred by *Schwarzenegger*) or connect to *Birdy* via Movie-Year-Movie (both shot in *1984*). However, when identifying the genre of the movie *The Terminator*, *MAM* usually plays more important role, rather than *MYM*. Therefore, treating different meta-paths equally is unpractical and will weaken the semantic information provided by some useful meta-paths.

Node-level attention. In a heterogeneous graph, nodes can be connected via various types of relation, e.g., meta-path. Given a meta-path, each node has lots of meta-path based neighbors. How to distinguish the subtle difference of there neighbors and select some informative neighors is required. For each node, node-level attention aims to learn the importance of meta-path based neighbors and assign different attention values to them. Still taking IMDB as an example, when using the meta-path Movie-Director-Movie (the movies are with the same director), *The Terminator* will connect to *Titanic* and *The Terminator 2* via director *James Cameron*. To better identify the genre of *The Terminator* as sci-fi movie, the model should pay more attention to *The Terminator 2*, rather than *Titanic*. Therefore, how to design a model which can discover the subtle differences of neighbors and learn their weights properly will be desired.

In this paper, we propose a novel **H**eterogeneous graph **A**ttention **N**etwork, named HAN, which considers both of node-level and

semantic-level attentions. In particular, given the node features as input, we use the type-specific transformation matrix to project different types of node features into the same space. Then the node-level attention is able to learn the attention values between the nodes and their meta-path based neighbors, while the semantic-level attention aims to learn the attention values of different meta-paths for the specific task in the heterogeneous graph. Based on the learned attention values in terms of the two levels, our model can get the optimal combination of neighbors and multiple meta-paths in a hierarchical manner, which enables the learned node embeddings to better capture the complex structure and rich semantic information in a heterogeneous graph. After that, the overall model can be optimized via backpropagation in an end-to-end manner.

The contributions of our work are summarized as follows:

• To our best knowledge, this is the first attempt to study the heterogeneous graph neural network based on attention mechanism. Our work enables the graph neural network to be directly applied to the heterogeneous graph, and further facilitates the heterogeneous graph based applications.

• We propose a novel heterogeneous graph attention network (HAN) which includes both of the node-level and semantic-level attentions. Benefitting from such hierarchical attentions, the proposed HAN can take the importance of nodes and meta-paths into consideration simultaneously. Moreover, our model is high efficiency, with the linear complexity with respect to the number of meta-path based node pairs, which can be applied to large-scale heterogeneous graph.

• We conduct extensive experiments to evaluate the performance of the proposed model. The results show the superiority of the proposed model by comparing with the state-of-the-art models. More importantly, by analysing the hierarchical attention mechanism, the proposed HAN demonstrates its potentially good interpretability for heterogeneous graph analysis.

2 RELATED WORK

2.1 Graph Neural Network

Graph neural networks (GNNs) which aim to extend the deep neural network to deal with arbitrary graph-structured data are introduced in [10, 24]. Yujia Li et al. [20] proposes a propagation model which can incorporate gated recurrent units to propagate information across all nodes. Recently, there is a surge of generalizing convolutional operation on the graph-structured data. The graph convolutional neural work generally falls into two categories, namely spectral domain and non-spectral domain. On one hand, spectral approaches work with a spectral representation of the graphs. Joan Bruna et al. [2] extends convolution to general graphs by finding the corresponding Fourier basis. Michaël et al. [6] utilizes K-order Chebyshev polynomials to approximate smooth filters in the spectral domain. Kipf et al. [18] proposes a spectral approach, named Graph Convolutional Network, which designs a graph convolutional network via a localized first-order approximation of spectral graph convolutions. On the other hand, we also have non-spectral approaches, which define convolutions directly on the graph, operating on groups of spatially close neighbors. Hamilton et al. [14] introduces GraphSAGE which performs a neural network based aggregator over a fixed size node neighbor. It can learn a function that generates embeddings by aggregating features from a node's local neighborhood.

Attention mechanisms, e.g., self-attention [34] and soft-attention [1], have become one of the most influential mechanisms in deep learning. Some previous works introduce the attention mechanism for graph based applications, e.g., the recommendation [15, 16]. Inspired by attention mechanism, Graph Attention Network [35] is proposed to learn the importance between nodes and its neighbors and fuse the neighbors to perform node classification. However, the above graph neural network cannot deal with various types of nodes and edges and can only be applied to the homogeneous graphs.

2.2 Network Embedding

Network embedding, i.e., network representation learning (NRL), is proposed to embed network into a low dimensional space while preserving the network structure and property so that the learned embeddings can be applied to the downstream network tasks. For example, the random walk based methods [12, 23], the deep neural network based methods [36], the matrix factorization based methods [22, 37], and others, e.g., LINE [33]. However, all these algorithms are proposed for the homogeneous graphs. Some elaborate reviews can be found in [5, 11].

Heterogeneous graph embedding mainly focuses on preserving the meta-path based structural information. ESim [26] accepts user-defined meta-paths as guidance to learn vertex vectors in a user-preferred embedding space for similarity search. Even through ESim can utilize multiple meta-paths, it cannot learn the importance of meta-paths. To achieve the best performance, ESim needs to conduct grid search to find the optimal weights of hmeta-paths. It is pretty hard to find the optimal combination for specific task. Meta-path2vec [7] designs a meta-path based random walk and utilizes skip-gram to perform heterogeneous graph embedding. However, metapath2vec can only utilize one meta-path and may ignore some useful information. Similar to metapath2vec, HERec [27] proposes a type constraint strategy to filter the node sequence and capture the complex semantics reflected in heterogeneous graph. HIN2Vec [9] carries out multiple prediction training tasks which learn the latent vectors of nodes and meta-paths simultaneously. Chen et al. [3] proposes a projected metric embedding model, named PME, which can preserve node proximities via Euclidian Distance. PME projects different types of node into the same relation space and conducts heterogeneous link prediction. To study the problem of comprehensive describe heterogeneous graph, Chen et al. [29] proposes HEER which can embed heterogeneous graph via edge representations. Fan et al. [8] proposes a embedding model metagraph2vec, where both the structures and semantics are maximally preserved for malware detection. Sun et al. [30] proposes meta-graph-based network embedding models, which simultaneously considers the hidden relations of all meta information of a meta-graph. In summary, all these aforementioned algorithms do not consider the attention mechanism in heterogeneous graph representation learning.

3 PRELIMINARY

A heterogeneous graph is a special kind of information network, which contains either multiple types of objects or multiple types of links.

Definition 3.1. **Heterogeneous Graph** [31]. A heterogeneous graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of an object set $\mathcal{V}$ and a link

set $\mathcal{E}$. A heterogeneous graph is also associated with a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and a link type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the sets of predefined object types and link types, where $|\mathcal{A}| + |\mathcal{R}| > 2$.

Example. As shown in Figure 1(a), we construct a heterogeneous graph to model the IMDB. It consists of multiple types of objects (Actor (A), Movie (M), Director (D)) and relations (shoot relation between movies and directors, role-play relation between actors and movies).

In heterogeneous graph, two objects can be connected via different semantic paths, which are called meta-paths.

Definition 3.2. **Meta-path** [32]. A meta-path Φ is defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ (abbreviated as $A_1 A_2 \cdots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between objects A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.

Example. As shown in Figure 1(a), two movies can be connected via multiple meta-paths, e.g., Movie-Actor-Movie (*MAM*) and Movie-Director-Movie (*MDM*). Different meta-paths always reveal different semantics. For example, the *MAM* means the co-actor relation, while Movie-Director-Movie (*MDM*) means they are directed by the same director.

Given a meta-path Φ, there exists a set of meta-path based neighbors of each node which can reveal diverse structure information and rich semantics in a heterogeneous graph.

Definition 3.3. **Meta-path based Neighbors.** Givien a node i and a meta-path Φ in a heterogeneous graph, the meta-path based neighbors $\mathcal{N}_i^{\Phi}$ of node i are defined as the set of nodes which connect with node i via meta-path Φ. Note that the node's neighbors includes itself.

Example. Taking Figure 1(d) as an example, given the meta-path Movie-Actor-Movie, the meta-path based neighbors of m_1 includes $m1$ (itself), m_2 and m_3. Similarly, the neighbors of m_1 based on meta-path Movie-Director-Movie includes m_1 and m_2. Obviously, meta-path based neighbors can exploit different aspects of structure information in heterogeneous graph. We can get meta-path based neighbors by the multiplication of a sequences of adjacency matrices.

Graph neural network has been proposed to deal with arbitrary graph-structured data. However, all of them are designed for homogeneous network [18, 35]. Since meta-path and meta-path based neighbors are two fundamental structures in a heterogeneous graph, next, we will present a novel graph neural network for heterogeneous graph data, which is able to exploit the subtle difference of nodes and meta-paths. The notations we will use throughout the article are summarized in Table 1.

4 THE PROPOSED MODEL

In this section, we propose a novel semi-supervised graph neural network for heterogeneous graph. Our model follows a hierarchical attention structure: node-level attention → semantic-level attention. Figure 2 presents the whole framework of HAN. First, we propose a node-level attention to learn the weight of meta-path based neighbors and aggregate them to get the semantic-specific node embedding.

Table 1: Notations and Explanations.

| Notation	Explanation
Φ	Meta-path
$\mathbf{h}$	Initial node feature
$\mathbf{M}_\phi$	Type-specific transformation matrix
$\mathbf{h}'$	Projected node feature
e_{ij}^Φ	Importance of meta-path based node pair (i,j)
$\mathbf{a}_\Phi$	Node-level attention vector for meta-path Φ
α_{ij}^Φ	Weight of meta-path based node pair (i,j)
$\mathcal{N}^\Phi$	Meta-path based neighbors
$\mathbf{Z}_\Phi$	Semantic-specific node embedding
$\mathbf{q}$	Semantic-level attention vector
w_Φ	Importance of meta-path Φ
β_Φ	Weight of meta-path Φ
$\mathbf{Z}$	The final embedding
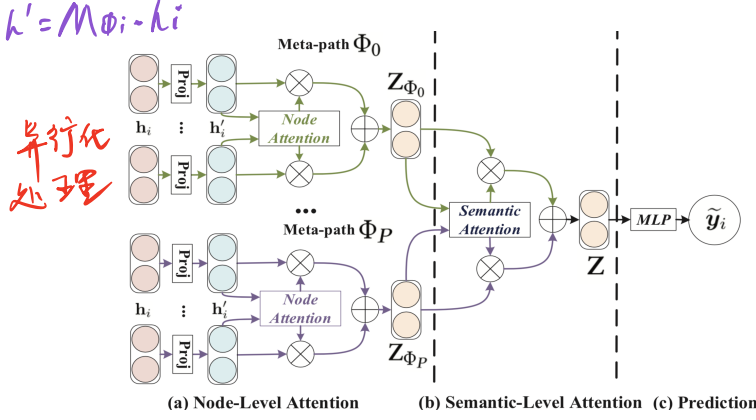
Figure 2: The overall framework of the proposed HAN. (a) All types of nodes are projected into a unified feature space and the weight of meta-path based node pair can be learned via node-level attention. (b) Joint learning the weight of each meta-path and fuse the semantic-specific node embedding via semantic-level attention. (c) Calculate the loss and end-to-end optimization for the proposed HAN.

After that, HAN can tell the difference of meta-paths via semantic-level attention and get the optimal weighted combination of the semantic-specific node embedding for the specific task.

4.1 Node-level Attention

Before aggregating the information from meta-path neighbors for each node, we should notice that the meta-path based neighbors of each node play a different role and show different importance in learning node embedding for the specific task. Here we introduce node-level attention can learn the importance of meta-path based neighbors for each node in a heterogeneous graph and aggregate the representation of these meaningful neighbors to form a node embedding.

Due to the heterogeneity of nodes, different types of nodes have different feature spaces. Therefore, for each type of nodes (e.g.,node

with type ϕ_i, we design the type-specific transformation matrix $\mathbf{M}_{\phi_i}$ to project the features of different types of nodes into the same feature space. Unlike [13], the type-specific transformation matrix is based on node-type rather than edge-type. The projection process can be shown as follows:

$$\mathbf{h}'_i = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i, \tag{1}$$

where $\mathbf{h}_i$ and $\mathbf{h}'_i$ are the original and projected feature of node i, respectively. By type-specific projection operation, the node-level attention can handle arbitrary types of nodes.

After that, we leverage self-attention [34] to learn the weight among various kinds of nodes. Given a node pair (i, j) which are connected via meta-path Φ, the node-level attention e_{ij}^Φ can learn the importance e_{ij}^Φ which means how important node j will be for node i. The importance of meta-path based node pair (i, j) can be formulated as follows:

$$e_{ij}^\Phi = att_{node}(\mathbf{h}'_i, \mathbf{h}'_j; \Phi). \tag{2}$$

Here att_{node} denotes the deep neural network which performs the node-level attention. Given meta-path Φ, att_{node} is shared for all meta-path based node pairs. It is because there are some similar connection patterns under one meta-path. The above Eq. (2) shows that given meta-path Φ, the weight of meta-path based node pair (i, j) depends on their features. Please note that, e_{ij}^Φ is asymmetric, i.e., the importance of node i to node j and the importance of node j to node i can be quite difference. It shows node-level attention can preserve the asymmetry which is a critical property of heterogenous graph.

Then we inject the structural information into the model via masked attention which means we only calculate the e_{ij}^Φ for nodes $j \in \mathcal{N}_i^\Phi$, where $\mathcal{N}_i^\Phi$ denotes the meta-path based neighbors of node i (include itself). After obtaining the importance between meta-path based node pairs, we normalize them to get the weight coefficient α_{ij}^Φ via softmax function:

$$\alpha_{ij}^\Phi = softmax_j(e_{ij}^\Phi) = \frac{\exp(\sigma(\mathbf{a}_\Phi^T \cdot [\mathbf{h}'_i \| \mathbf{h}'_j]))}{\sum_{k \in \mathcal{N}_i^\Phi} \exp(\sigma(\mathbf{a}_\Phi^T \cdot [\mathbf{h}'_i \| \mathbf{h}'_k]))}, \tag{3}$$

where σ denotes the activation function, $\|$ denotes the concatenate operation and $\mathbf{a}_\Phi$ is the node-level attention vector for meta-path Φ. As we can see from Eq. (3), the weight coefficient of (i, j) depends on their features. Also please note that the weight coefficient α_{ij}^Φ is asymmetric which means they make different contribution to each other. Not only because the concatenate order in the numerator, but also because they have different neighbors so the normalize term (denominator) will be quite difference.

Then, the meta-path based embedding of node i can be aggregated by the neighbor's projected features with the corresponding coefficients as follows:

$$\mathbf{z}_i^\Phi = \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j\right). \tag{4}$$

where $\mathbf{z}_i^\Phi$ is the learned embedding of node i for the meta-path Φ. To better understand the aggregating process of node-level, we also give a brief explanation in Figure 3 (a). Every node embedding is

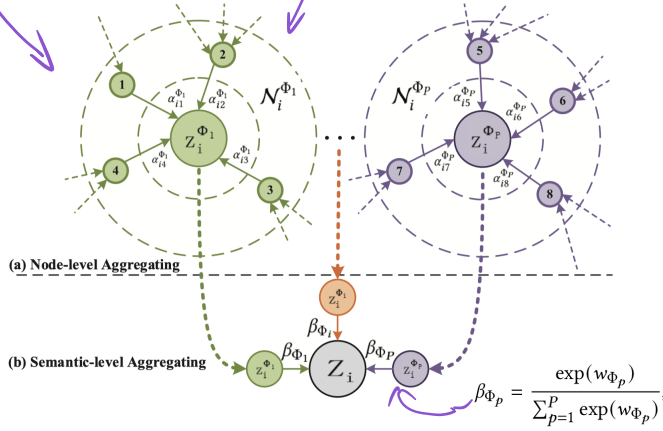$$z_i^\Phi = \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot h_j'\right).$$

Figure 3: Explanation of aggregating process in both node-level and semantic-level.

aggregated by its neighors. Since the attention weight α_{ij}^Φ is generated for single meta-path, it is semantic-specific and able to caputre one kind of semantic information.

Since heterogeneous graph present the property of scale free, the variance of graph data is quite high. To tackle the above challenge, we extend node-level attention to multihead attention so that the training process is more stable. Specifically, we repeat the node-level attention for K times and concatenate the learned embeddings as the semantic-specific embedding:

$$z_i^\Phi = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot h_j'\right). \tag{5}$$

Given the meta-path set $\{\Phi_1, \ldots, \Phi_P\}$, after feeding node features into node-level attention, we can obtain P groups of semantic-specific node embeddings, denoted as $\{Z_{\Phi_1}, \ldots, Z_{\Phi_P}\}$.

4.2 Semantic-level Attention

Generally, every node in a heterogeneous graph contains multiple types of semantic information and semantic-specific node embedding can only reflect node from one aspect. To learn a more comprehensive node embedding, we need to fuse multiple semantics which can be revealed by meta-paths. To address the challenge of meta-path selection and semantic fusion in a heterogeneous graph, we propose a novel semantic-level attention to automatically learn the importance of different meta-paths and fuse them for the specific task. Taking P groups of semantic-specific node embeddings learned from node-level attention as input, the learned weights of each meta-path $(\beta_{\Phi_1}, \ldots, \beta_{\Phi_P})$ can be shown as follows:

$$(\beta_{\Phi_1}, \ldots, \beta_{\Phi_P}) = att_{sem}(Z_{\Phi_1}, \ldots, Z_{\Phi_P}). \tag{6}$$

Here att_{sem} denotes the deep neural network which performs the semantic-level attention. It shows that the semantic-level attention can capture various types of semantic information behind a heterogeneous graph.

To learn the importance of each meta-path, we first transform semantic-specific embedding through a nonlinear transformation (e.g., one-layer MLP). Then we measure the importance of the

semantic-specific embedding as the similarity of transformed embedding with a semantic-level attention vector q. Furthermore, we average the importance of all the semantic-specific node embedding which can be explained as the importance of each meta-path. The importance of each meta-path, denoted as w_{Φ_i}, is shown as follows:

$$w_{\Phi_p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q^T \cdot \tanh(W \cdot z_i^{\Phi_p} + b), \tag{7}$$

where W is the weight matrix, b is the bias vector, q is the semantic-level attention vector. Note that for the meaningful comparation, all above parameters are shared for all meta-paths and semantic-specific embedding. After obtaining the importance of each meta-path, we normalize them via softmax function. The weight of meta-path Φ_i, denoted as β_{Φ_i}, can be obtained by normalizing the above importance of all meta-paths using softmax function,

$$\beta_{\Phi_p} = \frac{\exp(w_{\Phi_p})}{\sum_{p=1}^{P} \exp(w_{\Phi_p})}, \tag{8}$$

which can be interpreted as the contribution of the meta-path Φ_p for specific task. Obviously, the higher β_{Φ_p}, the more important meta-path Φ_p is. Note that for different tasks, meta-path Φ_p may has different weights. With the learned weights as coefficients, we can fuse these semantic-specific embeddings to obtain the final embedding Z as follows:

$$Z = \sum_{p=1}^{P} \beta_{\Phi_p} \cdot Z_{\Phi_p}. \tag{9}$$

To better understand the aggregating process of semantic-level, we also give a brief explanation in Figure 3 (b). The final embedding is aggregated by all semantic-specific embedding. Then we can apply the final embedding to specific tasks and design different loss fuction. For semi-supervised node classification, we can minimize the Cross-Entropy over all labeled node between the ground-truth and the prediction:

$$L = -\sum_{l \in \mathcal{Y}_L} Y^l \ln(C \cdot Z^l), \tag{10}$$

where C is the parameter of the classifier, $\mathcal{Y}_L$ is the set of node indices that have labels, Y^l and Z^l are the labels and embeddings of labeled nodes. With the guide of labeled data, we can optimize the proposed model via back propagation and learn the embeddings of nodes. The overall process of HAN in shown in Algorithm 1.

4.3 Analysis of the Proposed Model

Here we give the analysis of the proposed HAN as follows:
- The proposed model can deal with various types of nodes and relations and fuse rich semantics in a heterogeneous graph. The information can transfer from one kind of nodes to another kind of nodes via diverse relation. Benefitted from such a heterogeneous graph attention network, different types of node embedding can enhance the mutual integration, mutual promotion and mutual upgrade.
- The proposed HAN is highly efficient and can be easily parallelized. The computation of attention can compute individually across all nodes and meta-paths. Given a meta-path Φ, the time complexity of node-level attention is $O(V_\Phi F_1 F_2 K + E_\Phi F_1 K)$, where K is the number of attention head, V_Φ is the number of nodes, E_Φ is the

Table 2: Statistics of the datasets.

Dataset	Relations(A-B)	Number of A	Number of B	Number of A-B	Feature	Training	Validation	Test	Meta-paths
DBLP	Paper-Author	14328	4057	19645	334	800	400	2857	*APA*
	Paper-Conf	14328	20	14328					*APCPA*
	Paper-Term	14327	8789	88420					*APTPA*
IMDB	Movie-Actor	4780	5841	14340	1232	300	300	2687	*MAM*
	Movie-Director	4780	2269	4780					*MDM*
ACM	Paper-Author	3025	5835	9744	1830	600	300	2125	*PAP*
	Paper-Subject	3025	56	3025					*PSP*

Algorithm 1: The overall process of HAN.

Input : The heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
The node feature $\{\mathbf{h}_i, \forall i \in \mathcal{V}\}$,
The meta-path set $\{\Phi_0, \Phi_1, \ldots, \Phi_P\}$.
The number of attention head K,

Output : The final embedding $\mathbf{Z}$,
The node-level attention weight α ,
The semantic-level attention weight β .

1 **for** $\Phi_i \in \{\Phi_0, \Phi_1, \ldots, \Phi_P\}$ **do**
2   **for** $k = 1...K$ **do**
3   Type-specific transformation $\mathbf{h}'_i \leftarrow \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i$;
4   **for** $i \in \mathcal{V}$ **do**
5   Find the meta-path based neighbors $\mathcal{N}_i^\Phi$;
6   **for** $j \in \mathcal{N}_i^\Phi$ **do**
7   Calculate the weight coefficient α_{ij}^Φ ;
8   **end**
9   Calculate the semantic-specific node embedding
       $\mathbf{z}_i^\Phi \leftarrow \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j\right)$;
10   **end**
11   Concatenate the learned embeddings from all
     attention head $\mathbf{z}_i^\Phi \leftarrow \overset{K}{\underset{k=1}{\|}} \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j\right)$;
12   **end**
13   Calculate the weight of meta-path β_{Φ_i} ;
14   Fuse the semantic-specific embedding
   $\mathbf{Z} \leftarrow \sum_{i=1}^{P} \beta_{\Phi_i} \cdot \mathbf{Z}_{\Phi_i}$;
15 **end**
16 Calculate Cross-Entropy $L = -\sum_{l \in \mathcal{Y}_L} \mathbf{Y}_l \ln(\mathbf{C} \cdot \mathbf{Z}_l)$;
17 Back propagation and update parameters in HAN;
18 **return** Z, α, β.

---

number of meta-path based node pairs, F_1 and F_2 are the numbers of rows and columns of the transformation matrix, respectively. The overall complexity is linear to the number of nodes and meta-path based node pairs. The proposed model can be easily parallelized, because the node-level and semantic-level attention can be parallelized across node paris and meta-paths, respectively. The overall complexity is linear to the number of nodes and meta-path based node pairs.

• The hierarchical attention is shared for the whole heterogeneous graph which means the number of parameters is not dependent on the scale of a heterogeneous graph and can be used for the inductive problems [14]. Here inductive means the model can generate node embeddings for previous unseen nodes or even unseen graph.
• The proposed model has potentially good interpretability for the learned node embedding which is a big advantage for heterogeneous graph analysis. With the learned importance of nodes and meta-paths, the proposed model can pay more attention to some meaningful nodes or meta-paths for the specific task and give a more comprensive description of a heterogeneous graph. Based on the attention values, we can check which nodes or meta-paths make the higher (or lower) contributions for our task, which is beneficial to analyze and explain our results.

5 EXPERIMENTS

5.1 Datasets

The detailed descriptions of the heterogeneous graph used here are shown in Table 2.
• **DBLP**[2]. We extract a subset of DBLP which contains 14328 papers (P), 4057 authors (A), 20 conferences (C), 8789 terms (T). The authors are divided into four areas: *database, data mining, machine learning, information retrieval*. Also, we label each author's research area according to the conferences they submitted. Author features are the elements of a bag-of-words represented of keywords. Here we employ the meta-path set {*APA, APCPA, APTPA*} to perform experiments.
• **ACM**[3]. We extract papers published in KDD, SIGMOD, SIG-COMM, MobiCOMM, and VLDB and divide the papers into three classes (*Database, Wireless Communication, Data Mining*). Then we construct a heterogeneous graph that comprises 3025 papers (P), 5835 authors (A) and 56 subjects (S). Paper features correspond to elements of a bag-of-words represented of keywords. We employ the meta-path set {*PAP, PSP*} to perform experiments. Here we label the papers according to the conference they published.
• **IMDB.** Here we extract a subset of IMDB which contains 4780 movies (M), 5841 actors (A) and 2269 directors (D). The movies are divided into three classes (*Action, Comedy, Drama*) according to their genre. Movie features correspond to elements of a bag-of-words represented of plots. We employ the meta-path set {*MAM, MDM*} to perform experiments.

---

[2]https://dblp.uni-trier.de
[3]http://dl.acm.org/

Table 3: Qantitative results (%) on the node classification task.

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN$_{nd}$	HAN$_{sem}$	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	**89.40**
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	**89.79**
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	**90.00**	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	**90.63**
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	**89.22**
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	**89.64**
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	**89.85**	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	**90.54**
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	**92.24**
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	**92.40**
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	**92.80**
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	**93.08**
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	**93.11**
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	**93.30**
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	**93.70**
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	**93.99**
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	**50.87**	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	**52.71**
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	**54.24**
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	**54.38**
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	**55.73**
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	**57.97**
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	**58.32**
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	**58.51**

5.2 Baselines

We compare with some state-of-art baselines, include the (heterogeneous) network embedding methods and graph neural network based methods, to verfify the effectiveness of the proposed HAN. To verify the effectiveness of our node-level attention and semantic-level attention, respectively, we also test two variants of HAN.

- DeepWalk [23]: A random walk based network embedding method which designs for the homogeneous graphs. Here we ignore the heterogeneity of nodes and perform DeepWalk on the whole heterogeneous graph.
- ESim [26]: A heterogeneous graph embedding method which can capture semantic information from multiple meta-paths. Because it is difficult to search the weights of a set of meta-paths, we assign the weights learned from HAN to ESim.
- metapath2vec [7]: A heterogeneous graph embedding method which performs meta-path based random walk and utilizes skip-gram to embed the heterogeneous graphs. Here we test all the meta-paths for metapath2vec and report the best performance.
- HERec [27]: A heterogeneous graph embedding method which designs a type constraint strategy to filter the node sequence and utilizes skip-gram to embed the heterogeneous graphs. Here we test all the meta-paths for HERec and report the best performance.
- GCN [18]: It is a semi-supervised graph convolutional network that designed for the homogeneous graphs. Here we test all the meta-paths for GCN and report the best performance.
- GAT [35]: It is a semi-supervised neural network which considers the attention mechanism on the homogeneous graphs. Here we test all the meta-paths for GAT and report the best performance.

- HAN$_{nd}$: It is a variant of HAN, which removes node-level attention and assigns the same importance to each neighbor.
- HAN$_{sem}$: It is a variant of HAN, which removes the semantic-level attention and assigns the same importance to each meta-path.
- HAN: The proposed semi-supervised graph neural network which employs node-level attention and semantic-level attention simultaneously.

5.3 Implementation Details

For the proposed HAN, we randomly initialize parameters and optimize the model with Adam [17]. For the proposed HAN, we set the learning rate to 0.005, the regularization parameter to 0.001, the dimension of the semantic-level attention vector q to 128, the number of attention head K to 8, the dropout of attention to 0.6. And we use early stopping with a patience of 100, i.e. we stop training if the validation loss does not decrease for 100 consecutive epochs. To make our experiments repeatable, we make our dataset and codes publicly available at website[4]. For GCN and GAT, we optimize their parameters using the validation set. For semi-supervised graph neural network, including GCN, GAT and HAN, we split exactly the same training set, validation set and test set to ensure fairness. For random walk based methods include DeepWalk, ESim, metapath2vec, and HERec, we set window size to 5, walk length to 100, walks per node to 40, the number of negative samples to 5. For a fair comparison, we set the embedding dimension to 64 for all the above algorithms.

---

[4]https://github.com/Jhy1993/HAN

5.4 Classification

Here we employ KNN classifier with $k = 5$ to perform node classification. Since the variance of graph-structured data can be quite high, we repeat the process for 10 times and report the averaged *Macro-F1* and *Micro-F1* in Table 3.

Based on Table 3, we can see that HAN achieves the best performance. For traditional heterogeneous graph embedding method, ESim which can leverage multiple meta-paths performs better than metapath2vec. Generally, graph neural network based methods which combine the structure and feature information, e.g., GCN and GAT, usually perform better. To go deep into these methods, compared to simply average over node neighbors, e.g., GCN and HAN_{nd}, GAT and HAN can weigh the information properly and improve the performance of the learned embedding. Compared to GAT, the proposed HAN, which designs for heterogeneous graph, captures the rich semantics successfully and shows its superiority. Also, without node-level attention (HAN_{nd}) or semantic-level attention (HAN_{sem}), the performance becomes worse than HAN, which indicates the importance of modeling the attention mechanism on both of the nodes and semantics. Note that in ACM and IMDB, HAN improves classification results more significantly than in DBLP. Mainly because *APCPA* is the much more important than the rest meta-paths. We will explain this phenomenon in Section 5.7 by analyzing the semantic-level attention.

Through the above analysis, we can find that the proposed HAN achieves the best performance on all datasets. The results demonstrate that it is quite important to capture the importance of nodes and meta-paths in heterogeneous graph analysis.

5.5 Clustering

We also conduct the clustering task to evaluate the embeddings learned from the above algorithms. Once the proposed HAN trained, we can get all the node embedding via feed forward. Here we utilize the KMeans to perform node clustering and the number of clusters K is set to the number of classes. We use the same ground-truth as in node classification. And we adopt *NMI* and *ARI* to assess the quality of the clustering results. Since the performance of KMeans is affected by initial centroids, we repeat the process for 10 times and report the average results in Table 4.

As can be seen in Table 4, we can find that HAN performs consistently much better than all baselines. Also, graph neural network based algorithms usually achieve better performance. Besides, without distinguishing the importance of nodes or meta-paths, metapath2vec and GCN cannot perform well. With the guide of multiple meta-paths, HAN performs significantly better than GCN and GAT. On the other hand, without node-level attention (HAN_{nd}) or semantic-level attention (HAN_{sem}), the performance of HAN has shown various degrees of degeneration. It demonstrates that via assigning the different importance to nodes and meta-paths, the proposed HAN can learn a more meaningful node embedding.

Based on the above analysis, we can find that the propsed HAN can give a comprehensive description of heterogeneous graph and achieve a significant improvements.
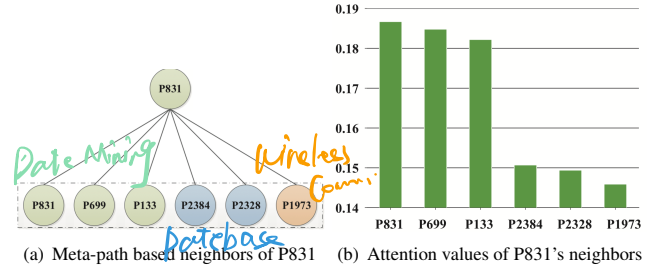

(a) Meta-path based neighbors of P831   (b) Attention values of P831's neighbors

Figure 4: Meta-path based neighbors of node P831 and corresponding attention values (Different colors mean different classes, e.g., *green* means Data Mining, *blue* means Database, *orange* means Wireless Communication).
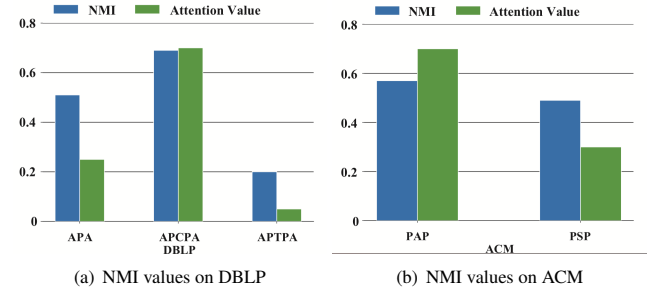

(a) NMI values on DBLP   (b) NMI values on ACM

Figure 5: Performance of single meta-path and corresponding attention value.

5.6 Analysis of Hierarchical Attention Mechanism

A salient property of HAN is the incorporation of the hierarchical mechanism, which takes the importance of node neighbors and meta-paths into consideration in learning representative embedding. Recall that we have learned the node-level attention weight α_{ij}^{Φ} and the semantic-level attention weight β_{Φ_i}. To better understand the importance of the neighbors and meta-paths, we provide a detailed analysis on the hierarchical attention mechanism.

Analysis of node-level attention. As mentioned before, given a specific task, our model can learn the attention values between nodes and its neighbors in a meta-path. Some important neighbors which are useful for the specific task tend to have larger attention values. Here we take the paper P831 [5] in ACM dataset as an illustrative example. Given a meta-path Paper-Author-Paper which describes the co-author of different papers, we enumerate the meta-path based neighbors of paper P831 and their attention values are shown in Figure 4. From Figure 4(a), we can see that P831 connects to P699 [6] and P133 [7], which all belong to *Data Mining*; conects to P2384 [8] and

[5]Xintao Wu, Daniel Barbara, Yong Ye. Screening and Interpreting Multi-item Associations Based on Log-linear Modeling, KDD'03
[6]Xintao Wu, Jianpin Fan, Kalpathi Subramanian. B-EM: a classifier incorporating bootstrap with EM approach for data mining, KDD'02
[7]Daniel Barbara, Carlotta Domeniconi, James P. Rogers. Detecting outliers using transduction and statistical testing, KDD'06
[8]Walid G. Aref, Daniel Barbara, Padmavathi Vallabhaneni. The Handwritten Trie: Indexing Electronic Ink, SIGMOD'95

Table 4: Qantitative results (%) on the node clustering task.

Datasets	Metrics	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN$_{nd}$	HAN$_{sem}$	HAN
ACM	NMI	41.61	39.14	21.22	40.70	51.40	57.29	60.99	61.05	**61.56**
	ARI	35.10	34.32	21.00	37.13	53.01	60.43	61.48	59.45	**64.39**
DBLP	NMI	76.53	66.32	74.30	76.73	75.01	71.50	75.30	77.31	**79.12**
	ARI	81.35	68.31	78.50	80.98	80.49	77.26	81.46	83.46	**84.76**
IMDB	NMI	1.45	0.55	1.20	1.20	5.45	8.45	9.16	10.31	**10.87**
	ARI	2.15	0.10	1.70	1.65	4.40	7.46	7.98	9.51	**10.01**

P2328 [9] while P2384 and P2328 both belong to *Database*; connects to P1973 [10] while P1973 belongs to *Wireless Communication*. From Figure 4(b), we can see that paper P831 gets the highest attention value from node-level attention which means the node itself plays the most important role in learning its representation. It is reasonable because all information supported by neighbors are usually viewed as a kind of supplementary information. Beyond itself, P699 and P133 get the second and third largest attention values. This is because P699 and P133 also belong to *Data Mining* and they can make significant contribution to identify the class of P831. The rest neighbors get minor attention values that because they do not belong to *Data Mining* and cannot make important contribution to identify the P831's class. Based on the above analysis, we can see that the node-level attention can tell the difference among neighbors and assign higher weights to some meaningful neighbors.

Analysis of semantic-level attention. As mentioned before, the proposed HAN can learn the importance of meta-paths for the specific task. To verify the ability of semantic-level attention, taking DBLP and ACM as examples, we report the clustering results (*NMI*) of single meta-path and corresponding attention values in Figure 5. Obviously, there is a positive correlation between the performance of a single meta-path and its attention value. For DBLP, HAN gives *APCPA* the largest weight, which means that HAN considers the *APCPA* as the most critical meta-path in identifying the author's research area. It makes sense because the author's research area and the conferences they submitted are highly correlated. For example, some natural language processing researchers mainly submit their papers to ACL or EMNLP, whereas some data mining researchers may submit their papers to KDD or WWW. Meanwhile, it is difficult for *APA* to identify the author's research area well. If we treat these meta-paths equally, e.g., HAN$_{sem}$, the performance will drop significantly. Based on the attention values of each meta-path, we can find that the meta-path *APCPA* is much more useful than *APA* and *APTPA*. So even the proposed HAN can fuse them, *APCPA* still plays a leading role in identifying the author's research area while *APA* and *APTPA* do not. It also explains why the performance of HAN in DBLP may not be as significant as in ACM and IMDB. We get the similar conclusions on ACM. For ACM, the results show that HAN gives the most considerable weight to *PAP*. Since the performance of *PAP* is slightly better than *PSP*, so HAN$_{sem}$ can achieve good performance by simple average operation. We can see that semantic-level attention can reveal the difference between these meta-paths and weights them adequately.

[9]Daniel Barbara, Tomasz Imielinski. Sleepers and Workaholics: Caching Strategies in Mobile Environments, VLDB'95
[10]Hector Garcia-Holina, Daniel Barbara. The cost of data replication, SIGCOMM'81

5.7 Visualization

For a more intuitively comparation, we conduct the task of visualization, which aims to layout a heterogeneous graph on a low dimensional space. Specifically, we learn the node embedding based on the proposed model and project the learned embedding into a 2-dimensional space. Here we utilize t-SNE [21] to visualize the author embedding in DBLP and coloured the nodes based on their research areas.

From Figure 6, we can find that GCN and GAT which design for the homogeneous graphs do not perform well. The authors belong to different research areas are mixed with each other. Metapath2vec performs much better than the above homogeneous graph neural networks. It demonstrates that the proper meta-path(e.g., *APCPA*) can make a significant contribution to heterogeneous graph analysis. However, since metapath2vec can only take only one meta-path into consideration, the boundary is still blurry. From Figure 6, we can see that the visualization of HAN peform best. With the guide of multiple meta-paths, the embedding learned by HAN has high intra-class similarity and separates the authors in different research area with distinct boundaries.

5.8 Parameters Experiments

In this section, we investigate the sensitivity of parameters and report the results of clustering (*NMI*) on ACM dataset with various parameters in Figure 7.

• **Dimension of the final embedding Z.** We first test the effect of the dimension of the final embedding Z. The result is shown in Figure 7(a). We can see that with the growth of the embedding dimension, the performance raises first and then starts to drop slowly. The reason is that HAN needs a suitable dimension to encode the semantics information and larger dimension may introduce additional redundancies.

• **Dimension of semantic-level attention vector q.** Since the ability of semantic-level attention is affected by the dimension of the semantic-level attention vector q, we explore the experimental results with various dimension. The result is shown in Figure 7(b). We can find that the performance of HAN grows with the dimension of semantic-level attention vector and achieves the best performance when the dimension of q is set to 128. After that, the performance of HAN starts to degenerate which may because of overfitting.

• **Number of attention head K.** In order to check the impact of multihead attention, we explore the performance of HAN with various number of attention head. The result is shown in Figure 7(c). Note that the multihead attention is removed when the number of attention head is set to 1. Based on the results, we can find
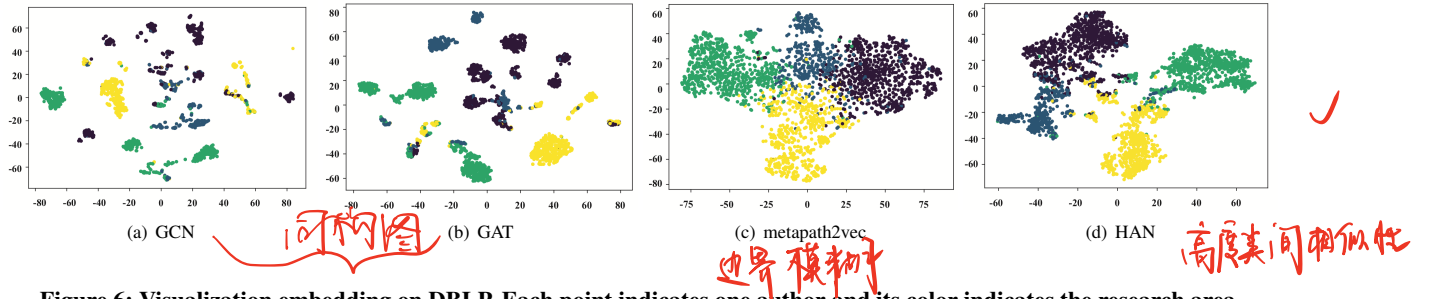
Figure 6: Visualization embedding on DBLP. Each point indicates one author and its color indicates the research area.
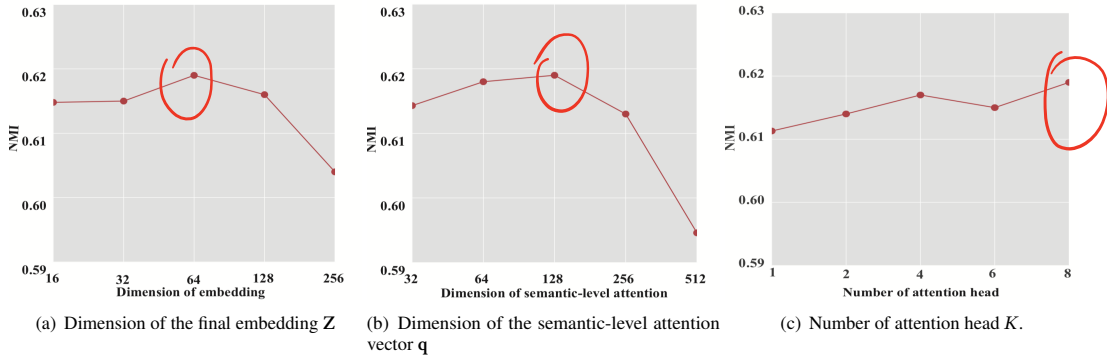

Figure 7: Parameter sensitivity of HAN w.r.t. Dimension of the final embedding Z, Dimension of the semantic-level attention vector q and Number of attention head K.

that the more number of attention head will generally improve the performance of HAN. However, with the change of attention head, the performance of HAN improve only slightly. Meanwhile, we also find that multihead attention can make the training process more stable.

6 CONCLUSION

In this paper, we tackle several fundamental problems in heterogeneous graph analysis and propose a semi-supervised heterogeneous graph neural network based solely on attention mechanism. The proposed HAN can capture complex structures and rich semantics behind heterogeneous graph. The proposed model leverages node-level attention and semantic-level attention to learn the importance of nodes and meta-paths, respectively. Meanwhile, the proposed model utilizes the structural information and the feature information in a uniform way. Experimental results include classification and clustering demonstrate the effectiveness of HAN. By analyzing the learned attention weights include both node-level and semantic-level, the proposed HAN has proven its potentially good interpretability.

7 ACKNOWLEDGMENTS

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR* (2015).

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[3] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. In *SIGKDD*. 1177–1186.

[4] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*. 295–304.

[5] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*. 3844–3852.

[7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*. 135–144.

[8] Yujie Fan, Shifu Hou, Yiming Zhang, Yanfang Ye, and Melih Abdulhayoglu. 2018. Gotcha-sly malware!: Scorpion a metagraph2vec based malware detection system. In *SIGKDD*. 253–262.

[9] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. In *CIKM*. 1797–1806.

[10] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *IJCNN*, Vol. 2. 729–734.

[11] Palash Goyal and Emilio Ferrara. 2017. Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801* (2017).

[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. 855–864.

[13] Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding logical queries on knowledge graphs. In *Advances in Neural Information Processing Systems*. 2030–2041.

[14] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[15] Xiaotian Han, Chuan Shi, Senzhang Wang, S Yu Philip, and Li Song. 2018. Aspect-Level Deep Collaborative Filtering via Heterogeneous Information Networks.. In *IJCAI*. 3393–3399.

[16] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging Meta-path based Context for Top-N Recommendation with A Neural Co-Attention Model. In *SIGKDD*. 1531–1540.

[17] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR* (2015).

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[19] Xiang Li, Yao Wu, Martin Ester, Ben Kao, Xin Wang, and Yudian Zheng. 2017. Semi-supervised clustering in attributed heterogeneous information networks. In *WWW*. 1621–1629.

[20] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2016. Gated graph sequence neural networks. *ICLR* (2016).

[21] Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 2605 (2008), 2579–2605.

[22] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *SIGKDD*. 1105–1114.

[23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

[25] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[26] Jingbo Shang, Meng Qu, Jialu Liu, Lance M. Kaplan, Jiawei Han, and Jian Peng. 2016. Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. *CoRR* abs/1610.09769 (2016).

[27] Chuan Shi, Binbin Hu, Xin Zhao, and Philip Yu. 2018. Heterogeneous Information Network Embedding for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[28] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A Survey of Heterogeneous Information Network Analysis. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017), 17–37.

[29] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *SIGKDD*. ACM, 2190–2199.

[30] Lichao Sun, Lifang He, Zhipeng Huang, Bokai Cao, Congying Xia, Xiaokai Wei, and S Yu Philip. 2018. Joint embedding of meta-path and meta-graph for heterogeneous information networks. In *2018 IEEE International Conference on Big Knowledge (ICBK)*. 131–138.

[31] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter* 14, 2 (2013), 20–28.

[32] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB* 4, 11 (2011), 992–1003.

[33] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *ICLR* (2018).

[36] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *SIGKDD*. 1225–1234.

[37] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding.. In *AAAI*. 203–209.

[38] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2048–2057.