# A Semantic Subgraphs Based Link Prediction Method for Heterogeneous Social Networks with Graph Attention Networks

1st Kai Zhu
*Department of Computer Science and Technology*
*Nanjing University*
Nanjing, China
zhukai@smail.nju.edu.cn

2nd Meng Cao
*Department of Computer Science and Technology*
*Nanjing University*
Nanjing, China
caomeng@smail.nju.edu.cn

***Abstract*—Link prediction is a very important research issue in social networks analysis, and it has a very wide range of applications. Real world social networks are usually heterogeneous networks which contain rich semantic information. Meta-paths are often used to characterize this semantic information in the analysis of heterogeneous social networks. Existing methods either use only topology information or use only a single meta path to extract semantic information in the network. In this paper, we propose a link prediction method based on SEmantic Subgraphs and Graph ATtention network (SESGAT). SESGAT not only makes full use of the different semantic information contained in different semantic subgraphs, but also uses the attention mechanism to learn the different importance of different semantic subgraphs for link prediction. Experiment results on real social networks show that our approach exhibits better predictive performance than other state-of-the-art methods.**

***Index Terms*—Social Link Prediction, Heterogeneous Social Networks, Meta Path, Graph Attention Networks**

## I. Introduction

Link prediction is an important issue that has been extensively studied in the analysis of data with network structure. Its goal is to predict the possibility of existence of edge between two nodes by analyzing nodes and edges in the network. Link prediction has a wide range of applications, such as social relationship mining, drug interaction prediction, financial abnormal behavior analysis, product recommendation, and so on. This paper focuses on the issue of social link prediction in social networks.

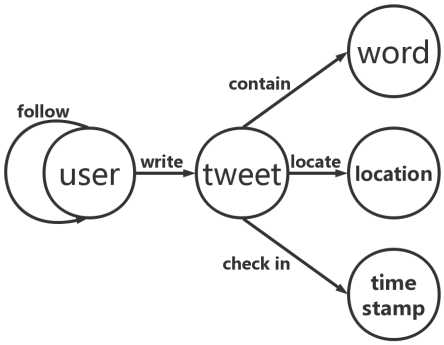
Some existing works suggest some heuristics for link prediction in homogeneous networks. For example, Preferential Attachment(PA) Index [1], Jaccard's Coefficient(JC) Index [2] and Common Neighbor(CN) Index [2] utilize the first-order neighbors of a node to predict the social links. Adamic/Adar(AA) Index [3] and Resource Allocation(RA) Index [4] use the second-order neighbors. SimRank (SR) [5] and Katz [6] employ high-order neighbors of a node, or even global topology information. Although these heuristics are effective in practice, they are all based on strong priori assumptions, which means they are not applicable in certain scenarios [7]. In recent years, some methods based on network embedding are used to learn the abstract representation of nodes and predict the links. For example, Deepwalk [8] and node2vec [9] obtain the abstract representation of nodes through random walks and use the abstract representation to predict links. However, these methods are all performed on homogeneous networks. Social networks usually contain different types of nodes and edges, which means that social networks are heterogeneous networks and contain many kinds of semantic information. So they fail to utilize the rich semantic information in the social networks.

In order to exploit these semantic information in heterogeneous social networks, meta-paths are used to learn the abstract representations of nodes. For example, metapath2vec [10] uses meta-path based random walk method to obtain the abstract representations of nodes. However, it uses only one type of meta-path and fails to take advantage of the various semantic information in heterogeneous social networks.

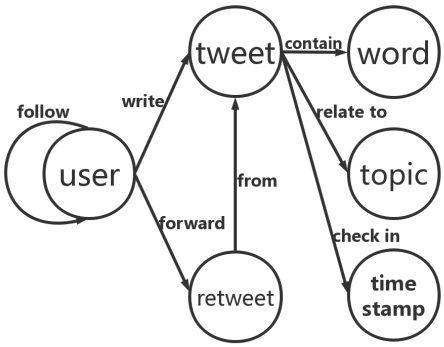
In [11] [12] [13], convolutional neural networks are applied to process data with network structure, which are called graph convolutional neural networks. For example, GraphSAGE [14] uses a graph convolutional neural network to aggregate the feature of a node's neighbors to generate an abstract representation of this node. However, this method only samples a fixed number of neighboring nodes, which means that useful information in some neighbors will be discarded. [15] proposes GAT, which uses the attention mechanism to let the model learn the importance of all neighbor nodes autonomously, thus avoiding the shortcomings of the above method. However, these methods are designed for homogeneous networks. HAN, a method used in heterogeneous networks, is proposed in [16], which uses meta-path based neighbors to learn abstract representation of nodes, and uses the attention mechanism to learn the importance of different meta-paths. However, these methods learn the representation of nodes by using the class of nodes as the objective function, which makes them unsuitable for the task of social links prediction, because the nodes at both ends of the social link are nodes of the same type (user nodes).

The prediction of social links in heterogeneous social net-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE




(a) Schema of Foursquare and Twitter




(b) Schema of Weibo

Fig. 1. The schema of heterogeneous social networks used in this paper.

works has the following challenges: 1) how to make full use of the rich semantic information in heterogeneous networks; 2) how to effectively integrate different semantic information; 3) how to make semantic information better serve the task of link prediction. In this paper, we proposed a link prediction model based on Semantic Subgraphs and Graph ATtention network (SESGAT), which solves the above challenges well. The contributions of our work can be summarized as follows:

- We define edge-centric semantic subgraphs to extract semantic information in heterogeneous social networks for social link prediction.
- We use the attention mechanism to learn the importance coefficients of different nodes and semantic subgraphs for integrating different information in the networks.
- We use the existence of edges as the objective function of our model, so that the model can learn the parameters that better serve link prediction task.

The rest of this paper is organized as follows. The problem formulation is discussed in Section II. In Section III, we describe the detail of the method we propose. The experiment results are shown in Section IV. And we conclude this paper in Section V.

## II. Problem Formulation

In this section, we first introduce the definitions that are needed in this paper, and then describe the form of the problem we need to solve.

### A. Terminology Definition

*a) Heterogeneous Social Network:* A heterogeneous social network is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{R}, \phi, \psi)$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $\mathbf{A}$ is the set of node types, and $\mathbf{R}$ is the set of edge types and satisfy $|\mathbf{A}| > 1$ and $|\mathbf{R}| > 1$. $\phi$ is a mapping function from $\mathcal{V}$ to $\mathbf{A}$, $\phi : \mathcal{V} \to \mathbf{A}$, and $\psi$ is a mapping function from $\mathcal{E}$ to $\mathbf{R}$, $\psi : \mathcal{E} \to \mathbf{R}$.

Since the datasets used in this article are from Foursquare, Twitter and Weibo, the node types contained in $\mathbf{A}$ are user, tweet, word, location, timestamp, retweet and topic, denoted by $\mathbf{U}$, $\mathbf{T}$, $\mathbf{W}$, $\mathbf{L}$, $\mathbf{S}$, $\mathbf{E}$ and $\mathbf{P}$ respectively. The edge types contained in $\mathbf{R}$ are social links between users, write links between users and tweets, forward links between users and retweets, locate links between tweets and locations, check-in links between tweets and timestamps, contain links between tweets and words, from links between tweets and retweets and relate links between tweets and topic, represented by $\mathbf{R_{U,U}}$, $\mathbf{R_{U,T}}$, $\mathbf{R_{U,E}}$, $\mathbf{R_{T,L}}$, $\mathbf{R_{T,S}}$, $\mathbf{R_{T,W}}$, $\mathbf{R_{T,E}}$ and $\mathbf{R_{T,P}}$ respectively. The schema is shown in Figure 1.

*b) Meta Path:* In the $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{R}, \phi, \psi)$, A meta path is a sequence of nodes and edges that connect two nodes in a heterogeneous social network. It can be formally defined as $v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \cdots \xrightarrow{e_l} v_{l+1}$, $v_i \in \mathcal{V}$, $e_i \in \mathcal{E}$, $\phi(v_i) \in \mathbf{A}$, $\psi(e_i) \in \mathbf{R}$, where $i = 1, 2, \cdots, |\mathcal{V}|$.

According to the schema shown in Figure 1, we have selected the following eight meta paths for the experiment in section 4.

- $\mathcal{M}_0$: $\mathbf{U} \xrightarrow[followee]{follower} \mathbf{U}$
- $\mathcal{M}_1$: $\mathbf{U} \xrightarrow{write} \mathbf{T} \xrightarrow{checkinat} \mathbf{L} \xleftarrow{checkinat} \mathbf{T} \xleftarrow{write} \mathbf{U}$.
- $\mathcal{M}_2$: $\mathbf{U} \xrightarrow{write} \mathbf{T} \xrightarrow{writeat} \mathbf{S} \xleftarrow{writeat} \mathbf{T} \xleftarrow{write} \mathbf{U}$.
- $\mathcal{M}_3$: $\mathbf{U} \xrightarrow{write} \mathbf{T} \xrightarrow{contain} \mathbf{W} \xleftarrow{contain} \mathbf{T} \xleftarrow{write} \mathbf{U}$.
- $\mathcal{M}_4$: $\mathbf{U} \xrightarrow{write} \mathbf{T} \xrightarrow{relateto} \mathbf{P} \xleftarrow{relateto} \mathbf{T} \xleftarrow{write} \mathbf{U}$.
- $\mathcal{M}_5$: $\mathbf{U} \xrightarrow{write} \mathbf{E} \xrightarrow{from} \mathbf{T} \xleftarrow{from} \mathbf{E} \xleftarrow{write} \mathbf{U}$.
- $\mathcal{M}_6$: $\mathbf{U} \xrightarrow{write} \mathbf{T} \xleftarrow{from} \mathbf{E} \xrightarrow{write} \mathbf{U}$.
- $\mathcal{M}_7$: $\mathbf{U} \xrightarrow{write} \mathbf{E} \xrightarrow{from} \mathbf{T} \xleftarrow{write} \mathbf{U}$.

*c) Edge Groove:* For any two nodes $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ in the heterogeneous social network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{R}, \phi, \psi)$, $\mathbb{E}_{i,j}$ is the edge groove between $v_i$ and $v_j$, $\mathbb{E}_{i,j}$ exists in two states: empty and full. If the state of $\mathbb{E}_{i,j}$ is full, then $\mathbb{E}_{i,j} \in \mathcal{E}$, which means there is a real edge between $v_i$ and $v_j$. In this paper, our goal is to predict social links between user nodes, so we make $\phi(v_i) = \mathbf{U}$, $\phi(v_j) = \mathbf{U}$ and $\psi(\mathbb{E}_{i,j}) = \mathbf{R_{U,U}}$.

*d) Edge-centric Semantic Subgraph:* Edge-centric semantic subgraphs can be defined as $\mathcal{S}_{i,j} = (\mathbb{E}_{i,j}, v_i, v_j, \mathcal{M}_k, \mathcal{N}_{v_i}^{\mathcal{M}_k}, \mathcal{N}_{v_j}^{\mathcal{M}_k})$, where $i, j = 1, 2, \cdots, |\mathbf{V_U}|, i \neq j$ and $k = 0, 1, \cdots, 7$. $v_i$ and $v_j$ are the user nodes, and $\mathcal{M}_k$ is the meta path mentioned above. $\mathcal{N}_{v_i}^{\mathcal{M}_k}$ is the set of $v_i$'s semantic neighbors based on meta path $\mathcal{M}_k$. We denote this subgraph as $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ and denote the nodes in this semantic subgraph as $\mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}$, where $\mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}} = \{v_i, v_j\} \cup \{\mathcal{N}_{v_i}^{\mathcal{M}_k}, \mathcal{N}_{v_j}^{\mathcal{M}_k}\}$. The example of Edge-centric semantic subgraphs is shown in Figure 2.

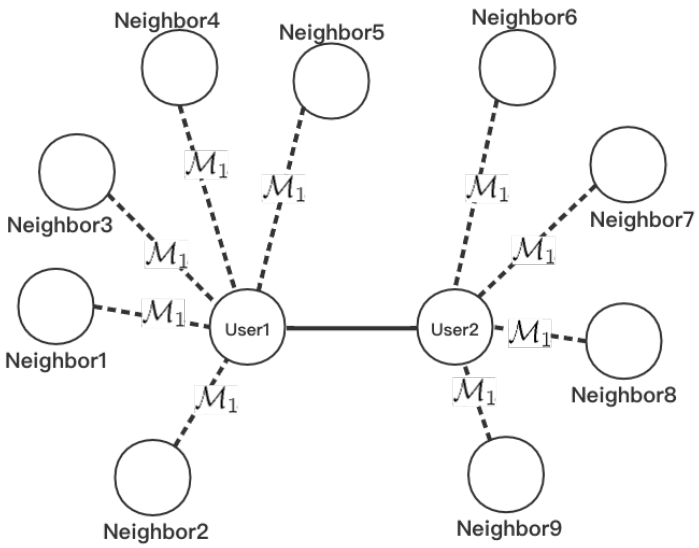



Fig. 2. An Example of Edge-centric Semantic Subgraph: $\mathcal{S}_{1,2}^{\mathcal{M}_1}$.

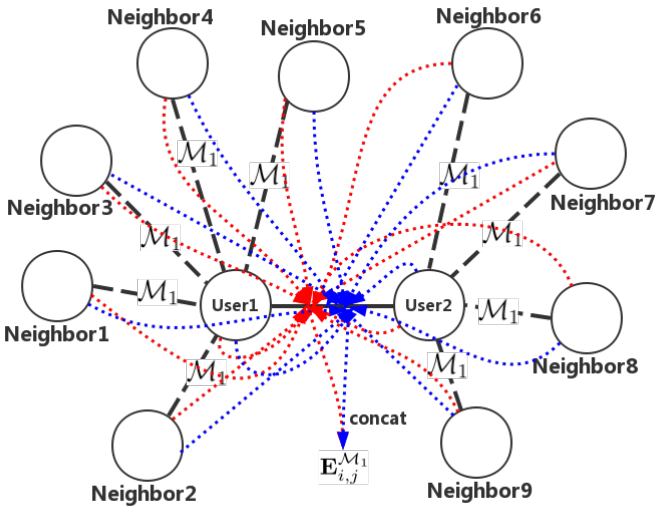



Fig. 3. An Example of Multi-head Attention Mechanism: $\mathcal{S}_{1,2}^{\mathcal{M}_1}$.

### B. Social Link Prediction in Heterogeneous Social Network

According to the definition in Section 2.1, in the heterogeneous social network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{R}, \phi, \psi)$, the set of users can be represented as $\mathcal{V}_{\mathbf{U}}$, and the set of existing social links can be represented as $\mathcal{E}_{\mathbf{U},\mathbf{U}}$, which is the set of $\mathbb{E}_{i,j}$ whose state is full . At the same time, we make the set of user node pairs with unobservable social links as $\mathcal{N}$, which is the set of $\mathbb{E}_{i,j}$ whose state is empty. Then, the set of $\mathbb{E}_{i,j}$ of the indeterminate state is $\mathcal{U}$, where $|\mathcal{U}| = |\mathcal{V}_{\mathbf{U}}| \times |\mathcal{V}_{\mathbf{U}}| - |\mathcal{E}_{\mathbf{U},\mathbf{U}}| - \mathcal{N}$. Our goal is to use $\mathcal{M}_k$, $k = 1, \cdots, 7$, and $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ to extract the rich topological and semantic information of $\mathcal{G}$, and learn the objective function $\mathcal{F}$. $\mathcal{F}$ can output the true state of $\mathbb{E}_{i,j} \in \mathcal{N}$, that is, the prediction of social links, which is essentially a Bi-classification problem of $\mathbb{E}_{i,j}$.

## III. Proposed Methods

### A. Intra-semantic-subgraph Attention Mechanism

In order to learn $\mathbb{E}_{i,j}$'s abstract representation based on specific semantic, we construct $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ based on specific meta path $\mathcal{M}_k$ for each $\mathbb{E}_{i,j}$. For each node in $\mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}$, its importance to $\mathbb{E}_{i,j}$ is different from other nodes. Therefore, the contribution of different nodes to abstract representation of $\mathbb{E}_{i,j}$ is different. So we use the attention mechanism to enable our model to automatically learn the importance weight of each node in $\mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}$. Because of the heterogeneity and complexity of heterogeneous social networks, different nodes may have different feature spaces [16] [17]. Therefore, we need to map the feature vectors of nodes to the same feature space before learning. Here, we also reduce the dimension of the feature vector while mapping, so that the abstract representation of $\mathbb{E}_{i,j}$ has a lower dimension. We use a mapping matrix to map and reduce the dimension of the node's feature vectors. The process is as follows:

$$\mathcal{X}_i' = \mathbf{M}_{\mathcal{M}_k} \times \mathcal{X}_i, k = 1, \cdots, 7 \tag{1}$$

where $\mathbf{M}_{\mathcal{M}_k}$ is the mapping matrix based on specific semantic. $\mathcal{X}_i$ is the feature vector of $v_i \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}$, and $\mathcal{X}_i'$ is the mapped feature vector, where $|\mathcal{X}_i| = d$ , $|\mathcal{X}_i'| = d'$. $d$ and $d'$ are the dimension of the feature vector, where $d' \ll d$.

After obtaining $\mathcal{X}_i'$, we use the attention mechanism to learn the weight of $v_i \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}$ for $\mathbb{E}_{i,j}$. Its form is as follows:

$$\alpha_l^{\mathcal{M}_k} = \mathcal{A}_{intra}(\mathcal{X}_l'; \mathcal{S}_{i,j}^{\mathcal{M}_k}), l \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}} \tag{2}$$

where the $\mathcal{A}_{intra}$ is the intra semantic subgraph attention mechanism implemented by the neural network. Its concrete form is as follows:

$$\alpha_l^{\mathcal{M}_k} = \frac{exp(\omega_l \times \mathcal{X}_l')}{\sum_{l \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}} exp(\omega_l \times \mathcal{X}_l')} \tag{3}$$

where the $\omega_l$ is the weight we need to learn.

After obtaining $\alpha_l^{\mathcal{M}_k}$, we can obtain the abstract representation of $\mathbb{E}_{i,j}$ under specific semantics by the following formula.

$$\mathbf{E}_{i,j}^{\mathcal{M}_k} = \sum_{l \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}} \alpha_l^{\mathcal{M}_k} \times \mathcal{X}_l' \tag{4}$$

where the $\mathbf{E}_{i,j}^{\mathcal{M}_k}$ is the abstract representation of $\mathbb{E}_{i,j}$ based on $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ that is generated from $\mathcal{M}_k$. In order to make the training process more stable, we use the multi-head attention mechanism inspired by [15] [16]. Specifically, we use T independent attention mechanisms to perform the above learning process. The obtained T abstract representations are then concatenated

to form the final abstract representation of $\mathbb{E}_{i,j}$. The form is as follows:

$$\mathbf{E}_{i,j}^{\mathcal{M}_k} = \|_{t=1}^{T} \Big( \sum_{l \in \mathcal{V}_{\mathcal{S}_{i,j}^{\mathcal{M}_k}}} \alpha_{l,t}^{\mathcal{M}_k} \times \mathcal{X}_l' \Big) \tag{5}$$

where the $\|$ denote the operator of concatenation and the $\alpha_{l,t}^{\mathcal{M}_k}$ denote the weight of user node $l$ that learned from $t$-th attention mechanism. The complete process of the intra-semantic-subgraph attention mechanism is shown in Figure 3.

### B. Inter-semantic-subgraphs Attention Mechanism

Using the method described in Section 3.1, we can obtain different abstract representations of $\mathbb{E}_{i,j}$ based on different $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ and denote as $\mathbf{E}_{i,j}^{\mathcal{M}_k}$. Then we integrate these $\mathbf{E}_{i,j}^{\mathcal{M}_k}$ of $\mathbb{E}_{i,j}$ learned from different $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ to generate the final abstract representation of $\mathbb{E}_{i,j}$. To achieve this goal, we propose the inter-semantic-subgraphs attention mechanism.

First we map $\mathbf{E}_{i,j}^{\mathcal{M}_k}$ into the same feature space. Inspired by [18], we use an FCL(Fully Connected Layer) to perform mapping operations to obtain their implicit representations. Here, we are only mapping the features without reducing the dimensions. The specific form is as follows:

$$\mathcal{H}_{i,j}^{\mathcal{M}_k} = tanh(\mathbf{M}'_{\mathcal{M}_k} \times \mathbf{E}_{i,j}^{\mathcal{M}_k} + bias) \tag{6}$$

where the $tanh$ is the activate function of FCL, $\mathcal{M}_k{}', bias$ are the parameters of FCL and the $\mathcal{H}_{i,j}^{\mathcal{M}_k}$ is the implicit representation of $\mathbb{E}_{i,j}$ after mapping.

After that, we used the inter-semantic-subgraphs attention mechanism to learn the importance of each $\mathbf{E}_{i,j}^{\mathcal{M}_k}$:

$$\beta_{\mathcal{M}_k} = \mathcal{A}_{inter}(\mathcal{H}_{i,j}^{\mathcal{M}_k}; \mathcal{S}_{i,j}^{\mathcal{M}_k}), k = 0, \cdots, 7 \tag{7}$$

where $\mathcal{A}_{inter}$ is the inter-semantic-subgraphs attention mechanism. Its implementation is as follows:

$$\beta_{\mathcal{M}_k} = \frac{exp(\omega_k' \times \mathcal{H}_{i,j}^{\mathcal{M}_k})}{\sum_{k=0}^{7} exp(\omega_k' \times \mathcal{H}_{i,j}^{\mathcal{M}_k})} \tag{8}$$

after getting the $\beta_{\mathcal{M}_k}$, we use them to get the final abstract representation of $\mathbb{E}_{i,j}$, as shown below:

$$\mathcal{X}_{i,j}^{final} = \sum_{k=0}^{7} \beta_{\mathcal{M}_k} \times \mathbf{E}_{i,j}^{\mathcal{M}_k} \tag{9}$$

### C. Framework of Our Method

As explained in Section 2.2, the prediction of social links is actually a Bi-classification problem of $\mathbb{E}_{i,j}$. We consider the two states of $\mathbb{E}_{i,j}$, empty and full, as the two categories, 0 and 1, of $\mathbb{E}_{i,j}$. Thus, the $\mathcal{X}_{i,j}^{final}$ is taken as the input to the classifier, and then the following objective function is used as the loss function:

$$loss = - \sum_{\mathbb{E}_{i,j} \in \{\mathcal{E}_{\mathbf{U},\mathbf{U}} \cup \mathcal{U}\}} \mathbf{Y}_{\mathbb{E}_{i,j}} \ln(\omega_c \times \mathcal{X}_{i,j}^{final} + b_c) \tag{10}$$

where $\omega_c$ and $b_c$ are the parameters of the classifier, $\mathbf{Y}_{\mathbb{E}_{i,j}}$ is the true label of the $\mathbb{E}_{i,j}$. We use the MLP(Multilayer Perceptron) to implement the classifier. Finally, we combine the classifier and the attention mechanisms in Section 3.1, Section 3.2 into a unified model named SESGAT and the overall flow of our method is shown in Figure 4. For the sake of convenience, we denote the trained model as $\mathcal{F}$ and the process of training and social link prediction for our model is shown in Algorithm 1.

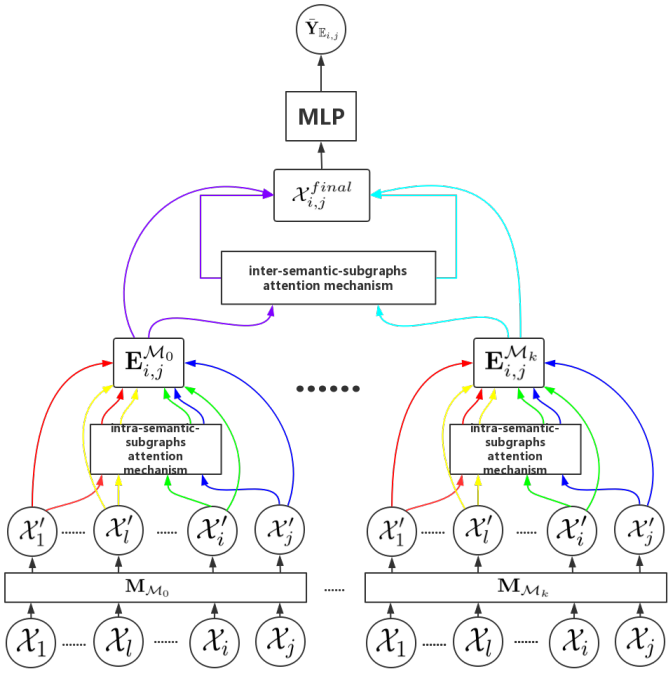



Fig. 4. The Overall Fow of Our Method.

## IV. Experiment

### A. Data Sets

In this paper, we used three real world social network datasets for experiments and the detailed descriptions are as follows.

Foursquare[1]: We extracted a subset of heterogeneous social network from Foursquare and its statistical information is shown in Table 1. We map the timestamp to a twenty-four hour interval to form 24 timestamp nodes. User feature vectors are statistics of the 64 themes they involve. The meta-paths we used are $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$.

Twitter[2]: We extracted a subset of heterogeneous social network from Twitter and its statistical information is also shown in Table 1. Other processing is similar to the Foursquare. The meta-paths we used are $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$.

Weibo[3]: We extracted a subset of heterogeneous social network from the Weibo dataset offered by [19] and its statistical information is also shown in Table 1. The user

[1]https://foursquare.com
[2]https://twitter.com
[3]https://weibo.com

**Algorithm 1** SESGAT

**Input:** $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{R}, \phi, \psi)$, $\{\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_k, \cdots\}$, T
**Output:** state set of $\mathbb{E}_{i,j} \in \mathcal{U}$, $i, j \in \mathcal{V}_{\mathbf{U}}$

Generate $\mathcal{E}_{\mathbf{U}}$, $\mathcal{N}$ and $\mathcal{U}$
**for** $\mathbb{E}_{i,j} \in \{\mathcal{E}_{\mathbf{U}} \cup \mathcal{N}\}$ **do**
  **for** $\mathcal{M}_k \in \{\mathcal{M}_0, \mathcal{M}_1, \cdots\}$ **do**
    Generate $\mathcal{S}_{i,j}^{\mathcal{M}_k}$ for $\mathbb{E}_{i,j}$ based on $\mathcal{M}_k$
    **for** t=0, t<T, t++ **do**
      Calculate the $\alpha_{l,t}^{\mathcal{M}_k}$ by Equation (3)
      Calculate the $\mathbf{E}_{i,j,t}^{\mathcal{M}_k}$ by Equation (4)
    **end for**
    Calculate the $\mathbf{E}_{i,j}^{\mathcal{M}_k}$ by Equation (5)
  **end for**
  Calculate the $\beta_{\mathcal{M}_k}$ by Equation (8)
  Calculate the $\mathcal{X}_{i,j}^{final}$ by Equation (9)
  Calculate the loss by Equation (10)
  Update all the parameters in SESGAT with the loss by back propagation
**end for**
Get the trained model $\mathcal{F}$
**for** $\mathbb{E}_{i,j} \in \mathcal{U}$ **do**
  Obtain the $\mathcal{F}(\mathbb{E}_{i,j})$ for all $\mathbb{E}_{i,j} \in \mathcal{U}$ and add them into the states set
**end for**
**return** states set

TABLE I
STATISTICAL INFORMATION OF HETEROGENEOUS SOCIAL NETWORKS

| Type | | Foursquare | Twitter | Weibo |
|---|---|---|---|---|
| Node | User | 5,734 | 6,164 | 5,923 |
| | Tweet | 159,401 | 2,082,189 | 8,545 |
| | Word | 97,311 | 802,741 | 50,979 |
| | TimeStamp | 24 | 24 | 24 |
| | Location | 9,139 | 7,205 | 0 |
| | Retweet | 0 | 0 | 14,555 |
| | Topic | 0 | 0 | 100 |
| Edge | Follow | 54,364 | 38,685 | 246,804 |
| | Write | 159,401 | 2,082,189 | 8,545 |
| | Contain | 97,311 | 802,741 | 50,979 |
| | Check in | 159,401 | 2,082,189 | 8,545 |
| | Locate | 9,139 | 7,205 | 0 |
| | Forward | 0 | 0 | 14,555 |
| | Relate to | 0 | 0 | 8,545 |
| | From | 0 | 0 | 14,555 |
| Meta Path | $\mathcal{M}_0$ | 54,364 | 38,685 | 246,804 |
| | $\mathcal{M}_1$ | 611,086 | 355,330 | 0 |
| | $\mathcal{M}_2$ | 4,831,840 | 31,654,470 | 4,521,550 |
| | $\mathcal{M}_3$ | 5,066,932 | 30,711,334 | 9,906,756 |
| | $\mathcal{M}_4$ | 0 | 0 | 151,934 |
| | $\mathcal{M}_5$ | 0 | 0 | 174,550 |
| | $\mathcal{M}_6$ | 0 | 0 | 10,010 |
| | $\mathcal{M}_7$ | 0 | 0 | 10,010 |

feature vector has a dimension of 9 which includes different personal information of the user. The meta-paths we used are $\mathcal{M}_0$, $\mathcal{M}_2$, $\mathcal{M}_3$, $\mathcal{M}_4$, $\mathcal{M}_5$, $\mathcal{M}_6$ and $\mathcal{M}_7$.

### B. Comparison Methods

We compared SESGAT with three heuristic methods, two based on network embedding, one based on graph neural network and its variants. They are as follows:

- SESGAT: SESGAT is the method proposed in this paper.
- SESGAT-$\mathcal{M}_0$: It is a variant of SESGAT that uses only semantic subgraph based on topological neighbors and intra-semantic-subgraph attention mechanism.
- SESGAT-$\mathcal{M}_k$: It is another variant of SESGAT that only uses semantic subgraph based on $\mathcal{M}_k$ and intra-semantic-subgraph attention mechanism. The value of k is given in the subsection $Analysis\ of\ SESGAT$ of this section.
- SEAL [7]: SEAL is a graph neural network based method that utilizes the topological neighbors in homogeneous networks to learn the abstract representation of edge.
- SEAL-$\mathcal{M}_k$: It is a variant of SEAL. We use meta-path-based neighbors instead of topological neighbors to make it suitable for heterogeneous networks.
- metapath2vec++ [10]: metapath2vec++ is a node embedding method based on random walk. It performs random walks on heterogeneous networks based on a specific meta-path to generate the embedding of nodes.
- Deepwalk [8]: Deepwalk is also a node embedding method based on random walks. It generates a node's embedding by performing random walks based on topological links on a homogeneous network.
- Katz [6]: Katz is a heuristic method for utilizing global topology information in homogeneous networks.
- RA [4]: Resource Allocation index(RA) is a heuristic method that utilizes second-order neighbors in homogeneous networks.
- JC [2]: Jaccard's Coefficient index(JC) is a heuristic method that utilizes first-order neighbors in homogeneous networks

### C. Evaluation Metric

Real social links and real non-existent links in the dataset are used as ground truth to evaluate the result of social link prediction. As stated in Section 3, this is a Bi-classification problem for $\mathbb{E}_{i,j}$. We use ACU, Accuracy and F1 score as the evaluation metrics in this paper.

### D. Experiment Setups

In the experiment, we randomly selected 2500 $\mathbb{E}_{i,j}$ in full states and 2500 $\mathbb{E}_{i,j}$ in empty states from each dataset. Then we select a certain proportion $\rho \in (0, 1)$ of $\mathbb{E}_{i,j}$ in these two states as the training set, and the remaining $(1 - \rho)$ as the test set. We treat the training set as $\{\mathcal{E}_{\mathbf{U}} \cup \mathcal{N}\}$ and treat the test set as $\mathcal{U}$.

For the heuristics, we take the calculated score as the feature of $\mathbb{E}_{i,j}$ and then use SVM for training and testing with $\{\mathcal{E}_{\mathbf{U}} \cup \mathcal{N}\}$ and $\mathcal{U}$ respectively. For Deepwalk, metapath2vec++, we hide the links in $\mathcal{U}$, which means, we set the sate of $\mathbb{E}_{i,j} \in \mathcal{U}$, whose state is full, to empty. Then we used these methods to get the abstract representation of each user node. We concatenate the abstract representations of the two endpoints of $\mathbb{E}_{i,j}$ as the feature vector of $\mathbb{E}_{i,j}$, and use $\{\mathcal{E}_{\mathbf{U}} \cup \mathcal{N}\}$ to train the SVM and then use $\mathcal{U}$ to test the performance of the trained SVM. For SEAL and SEAL-$\mathcal{M}_k$,

TABLE II
COMPARISON OF SOCIAL LINK PREDICTION PERFORMANCE BETWEEN DIFFERENT METHODS UNDER DIFFERENT $\rho$ IN ALL THE DATASET

| measure | method | ratio of training set $\rho$ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Foursquare | | | | Twitter | | | | Weibo | | | |
| | | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 | 0.6 | 0.8 |
| AUC | SESGAT | **0.832±0.002** | **0.865±0.003** | **0.887±0.005** | **0.908±0.002** | **0.843±0.002** | **0.857±0.004** | **0.875±0.001** | **0.896±0.002** | **0.861±0.003** | **0.876±0.002** | **0.889±0.004** | **0.913±0.002** |
| | SESGAT-$\mathcal{M}_0$ | 0.812±0.004 | 0.829±0.002 | 0.854±0.004 | 0.865±0.003 | 0.827±0.003 | 0.841±0.003 | 0.851±0.003 | 0.865±0.002 | 0.839±0.003 | 0.853±0.004 | 0.866±0.003 | 0.878±0.004 |
| | SESGAT-$\mathcal{M}_k$ | 0.806±0.002 | 0.825±0.001 | 0.849±0.003 | 0.861±0.003 | 0.821±0.004 | 0.836±0.002 | 0.846±0.002 | 0.859±0.003 | 0.831±0.001 | 0.849±0.002 | 0.859±0.002 | 0.868±0.003 |
| | SEAL | 0.808±0.002 | 0.820±0.002 | 0.838±0.007 | 0.858±0.002 | 0.807±0.001 | 0.835±0.002 | 0.843±0.002 | 0.849±0.002 | 0.831±0.007 | 0.838±0.007 | 0.851±0.001 | 0.872±0.006 |
| | SEAL-$\mathcal{M}_k$ | 0.792±0.003 | 0.818±0.002 | 0.833±0.003 | 0.853±0.004 | 0.814±0.001 | 0.833±0.002 | 0.841±0.006 | 0.852±0.003 | 0.823±0.003 | 0.849±0.002 | 0.851±0.004 | 0.860±0.003 |
| | metapath2vec++ | 0.785±0.005 | 0.821±0.004 | 0.848±0.001 | 0.859±0.004 | 0.777±0.006 | 0.827±0.005 | 0.849±0.005 | 0.845±0.001 | 0.828±0.002 | 0.846±0.001 | 0.863±0.004 | 0.876±0.002 |
| | Deepwalk | 0.786±0.001 | 0.819±0.003 | 0.834±0.002 | 0.859±0.002 | 0.742±0.005 | 0.753±0.002 | 0.758±0.004 | 0.768±0.004 | 0.773±0.005 | 0.837±0.004 | 0.847±0.003 | 0.857±0.003 |
| | Katz | 0.622±0.003 | 0.626±0.002 | 0.632±0.002 | 0.652±0.003 | 0.589±0.003 | 0.579±0.004 | 0.609±0.002 | 0.613±0.001 | 0.563±0.003 | 0.579±0.003 | 0.605±0.003 | 0.628±0.002 |
| | RA | 0.764±0.005 | 0.787±0.007 | 0.795±0.006 | 0.804±0.004 | 0.672±0.007 | 0.695±0.005 | 0.694±0.003 | 0.704±0.003 | 0.746±0.007 | 0.745±0.006 | 0.751±0.006 | 0.768±0.004 |
| | JC | 0.753±0.003 | 0.769±0.001 | 0.776±0.006 | 0.770±0.001 | 0.568±0.006 | 0.615±0.003 | 0.606±0.001 | 0.594±0.005 | 0.583±0.003 | 0.620±0.003 | 0.614±0.003 | 0.644±0.001 |
| Acc | SESGAT | **0.816±0.002** | **0.829±0.002** | **0.845±0.003** | **0.865±0.002** | **0.817±0.002** | **0.830±0.001** | **0.848±0.004** | **0.867±0.002** | **0.805±0.002** | **0.819±0.004** | **0.844±0.003** | **0.867±0.003** |
| | SESGAT-$\mathcal{M}_0$ | 0.796±0.003 | 0.811±0.001 | 0.824±0.003 | 0.841±0.003 | 0.789±0.003 | 0.807±0.004 | 0.827±0.003 | 0.839±0.004 | 0.781±0.006 | 0.802±0.002 | 0.811±0.003 | 0.823±0.004 |
| | SESGAT-$\mathcal{M}_k$ | 0.786±0.004 | 0.808±0.004 | 0.815±0.001 | 0.832±0.004 | 0.764±0.003 | 0.787±0.002 | 0.798±0.003 | 0.810±0.002 | 0.769±0.003 | 0.787±0.002 | 0.795±0.002 | 0.809±0.001 |
| | SEAL | 0.774±0.002 | 0.803±0.003 | 0.813±0.001 | 0.826±0.001 | 0.747±0.002 | 0.758±0.003 | 0.766±0.002 | 0.777±0.005 | 0.772±0.007 | 0.776±0.005 | 0.776±0.008 | 0.806±0.004 |
| | SEAL-$\mathcal{M}_k$ | 0.777±0.003 | 0.791±0.002 | 0.806±0.001 | 0.819±0.001 | 0.749±0.005 | 0.765±0.007 | 0.785±0.005 | 0.789±0.001 | 0.758±0.001 | 774±0.003 | 0.777±0.006 | 0.801±0.002 |
| | metapath2vec++ | 0.762±0.001 | 0.779±0.005 | 0.796±0.001 | 0.811±0.003 | 0.743±0.005 | 0.775±0.007 | 0.776±0.003 | 0.784±0.004 | 0.778±0.006 | 0.787±0.003 | 0.806±0.002 | 0.815±0.004 |
| | Deepwalk | 0.741±0.003 | 0.761±0.001 | 0.786±0.003 | 0.795±0.001 | 0.743±0.002 | 0.752±0.002 | 0.762±0.001 | 0.765±0.002 | 0.771±0.006 | 0.791±0.005 | 0.805±0.003 | 0.814±0.005 |
| | Katz | 0.628±0.004 | 0.661±0.006 | 0.683±0.003 | 0.643±0.002 | 0.591±0.001 | 0.583±0.002 | 0.611±0.005 | 0.603±0.003 | 0.564±0.002 | 0.607±0.005 | 0.588±0.002 | 0.632±0.004 |
| | RA | 0.682±0.001 | 0.719±0.002 | 0.759±0.001 | 0.758±0.001 | 0.673±0.004 | 0.698±0.001 | 0.699±0.002 | 0.692±0.001 | 0.746±0.003 | 0.748±0.007 | 0.755±0.003 | 0.771±0.004 |
| | JC | 0.649±0.006 | 0.667±0.001 | 0.699±0.004 | 0.737±0.001 | 0.570±0.002 | 0.619±0.003 | 0.632±0.002 | 0.658±0.004 | 0.587±0.003 | 0.623±0.004 | 0.617±0.004 | 0.646±0.001 |
| F1 | SESGAT | **0.801±0.003** | **0.813±0.004** | **0.826±0.001** | **0.841±0.003** | **0.785±0.003** | **0.802±0.003** | **0.819±0.002** | **0.843±0.004** | **0.802±0.001** | **0.817±0.004** | **0.841±0.005** | **0.853±0.002** |
| | SESGAT-$\mathcal{M}_0$ | 0.786±0.002 | 0.797±0.003 | 0.806±0.002 | 0.821±0.002 | 0.751±0.001 | 0.768±0.004 | 0.784±0.003 | 0.805±0.002 | 0.778±0.002 | 0.797±0.002 | 0.811±0.002 | 0.826±0.004 |
| | SESGAT-$\mathcal{M}_k$ | 0.764±0.003 | 0.778±0.004 | 0.796±0.003 | 0.812±0.002 | 0.744±0.003 | 0.767±0.001 | 0.773±0.002 | 0.795±0.003 | 0.779±0.003 | 0.792±0.003 | 0.807±0.002 | 0.815±0.002 |
| | SEAL | 0.717±0.003 | 0.734±0.003 | 0.755±0.005 | 0.779±0.004 | 0.739±0.004 | 0.763±0.004 | 0.778±0.003 | 0.786±0.003 | 0.764±0.001 | 0.785±0.003 | 0.796±0.006 | 0.801±0.006 |
| | SEAL-$\mathcal{M}_k$ | 0.741±0.002 | 0.748±0.008 | 0.766±0.005 | 0.802±0.005 | 0.729±0.003 | 0.766±0.002 | 0.758±0.005 | 0.784±0.005 | 0.774±0.001 | 0.789±0.001 | 0.801±0.001 | 0.804±0.004 |
| | metapath2vec++ | 0.727±0.005 | 0.757±0.003 | 0.776±0.004 | 0.791±0.006 | 0.708±0.004 | 0.735±0.006 | 0.741±0.004 | 0.789±0.003 | 0.773±0.005 | 0.778±0.005 | 0.797±0.004 | 0.807±0.003 |
| | Deepwalk | 0.706±0.001 | 0.707±0.003 | 0.756±0.003 | 0.792±0.006 | 0.729±0.006 | 0.753±0.001 | 0.761±0.006 | 0.789±0.004 | 0.759±0.005 | 0.777±0.003 | 0.773±0.002 | 0.792±0.006 |
| | Katz | 0.512±0.001 | 0.581±0.008 | 0.597±0.003 | 0.619±0.003 | 0.667±0.004 | 0.669±0.003 | 0.663±0.003 | 0.691±0.002 | 0.471±0.004 | 0.543±0.004 | 0.605±0.004 | 0.612±0.004 |
| | RA | 0.649±0.002 | 0.683±0.001 | 0.741±0.004 | 0.756±0.008 | 0.615±0.001 | 0.664±0.007 | 0.662±0.008 | 0.681±0.003 | 0.689±0.004 | 0.701±0.003 | 0.716±0.005 | 0.732±0.003 |
| | JC | 0.676±0.005 | 0.679±0.006 | 0.704±0.005 | 0.756±0.007 | 0.573±0.004 | 0.601±0.005 | 0.699±0.004 | 0.686±0.006 | 0.455±0.004 | 0.547±0.007 | 0.564±0.004 | 0.587±0.002 |

we hide the links in $\mathcal{E}_{\mathbf{U}}$ in the same way. The processed social network was used to train the model. For a fair comparison, we set the window in Deepwalk and metapath2vec++ to 10, the length of the walk to 80, and the length of the embedding to 32. The specific meta path used in metapath2vec++ and SEAL-$\mathcal{M}_k$ are $\mathcal{M}_1$ (on Foursquare and Twitter datasets) and $\mathcal{M}_4$ (on Weibo dataset). The reason for our choice is in Section 4.6. For our method SESGAT, we use $\{\mathcal{E}_{\mathbf{U}} \cup \mathcal{N}\}$ directly to train the model and then use $\mathcal{U}$ to test its performance. We set the value of the attention head T to 4, the learning rate of the model to 0.005, and the regularization coefficient to 0.001. The dropout rate of our model is set to 0.6.

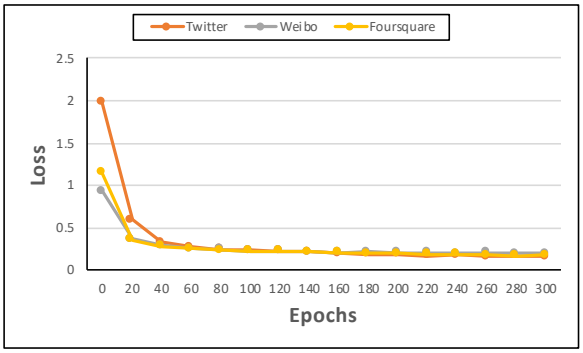



Fig. 5. The convergence of SESGAT with epochs.

## E. Experiment Results

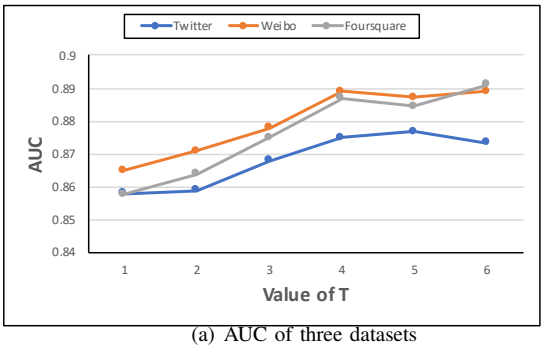
In the experiment, we took $\rho$ as 0.2, 0.4, 0.6, and 0.8, respectively, and recorded the AUC, Accuracy, and F1 score for each method. As can be seen from Table 2, our method performs better than other methods. From the comparison between SESGAT, heuristic methods, Deepwalk and SEAL, we can see that the semantic information can significantly improve the predictive performance of social links. Through the comparison of SESGAT-$\mathcal{M}_0$ and SEAL, SESGAT-$\mathcal{M}_k$ and SEAL-$\mathcal{M}_k$, it can be seen that the intra-semantic-subgraph attention mechanism can improve the performance. From the comparison of SESGAT, SESGAT-$\mathcal{M}_k$, metapath2vec++ and SEAL-$\mathcal{M}_1$, we can see that effectively integrating and utilizing different semantic information can improve the performance of social link prediction which means that the inter-semantic-subgraph attention mechanism is functional.

## F. Analysis of SESGAT

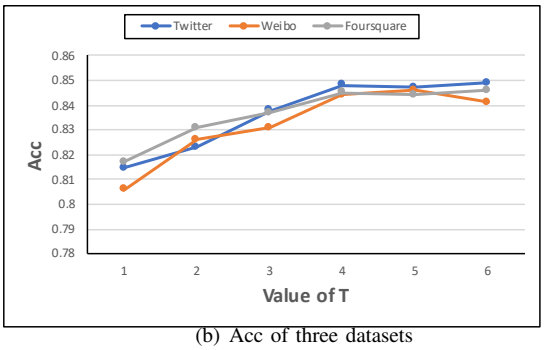
In this section, we did some detailed analysis of SESGAT. As can be seen from Figure 5, SESGAT's training loss on all datasets converges after about 200 epochs. In Figure 6, we analyze the effect of the different attention head T on the performance of SESGAT. From the figure we can see that the performance change of SESGAT is small after $T > 4$, so we make $T = 4$ in the comparative experiment.

In Figure 7, we extract the weight values of different meta-path-based semantic subgraphs in Twitter and Weibo. As can be seen from the figure, different semantics contribute differently to the final representation of $\mathbb{E}_{i,j}$. Based on the results in Figure 7, we chose $\mathcal{M}_1$ for metapath2vec++, SEAL-$\mathcal{M}_k$ and SESGAT-$\mathcal{M}_k$ on Foursquare and Twitter. And we selected $\mathcal{M}_4$ for metapath2vec++, SEAL-$\mathcal{M}_k$ and SESGAT-$\mathcal{M}_k$ on Weibo.

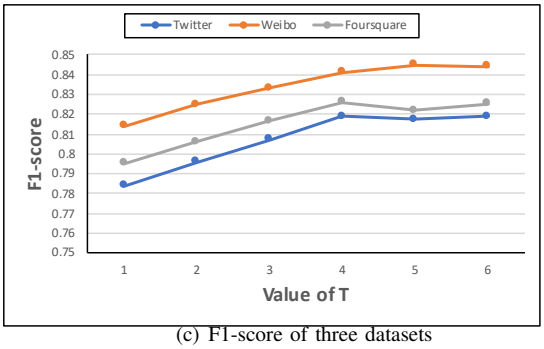
In addition, we also analyzed the performance of SESGAT under different number of meta paths and different combina-




(a) AUC of three datasets

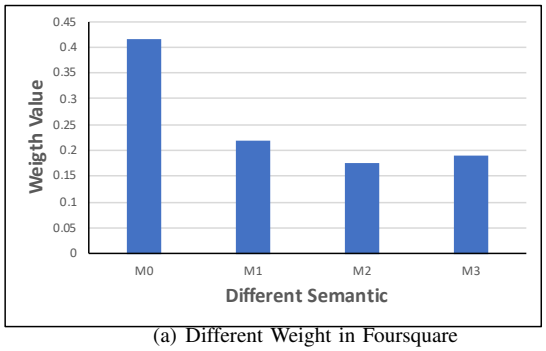



(b) Acc of three datasets

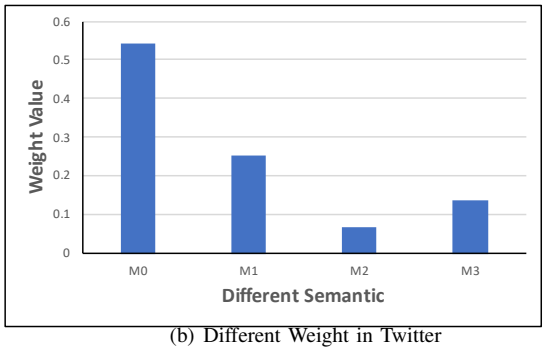



(c) F1-score of three datasets

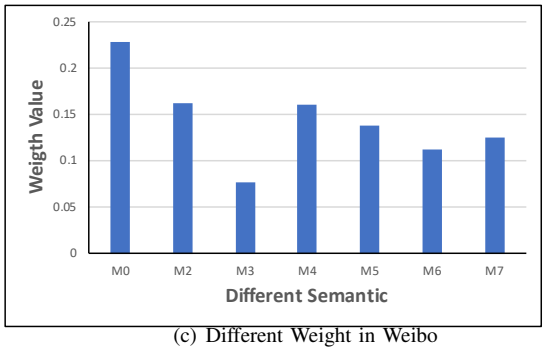
Fig. 6. The effect of different attention head T on social link prediction performance.




(a) Different Weight in Foursquare




(b) Different Weight in Twitter




(c) Different Weight in Weibo

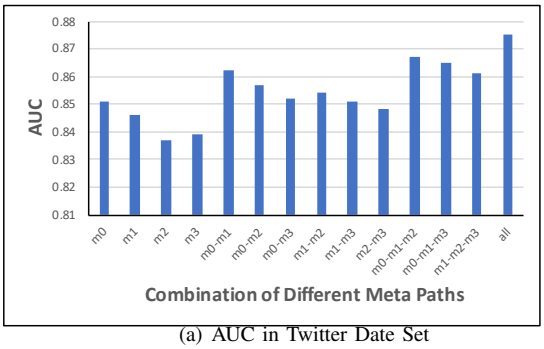
Fig. 7. Example of the weight of different sematic.

tions of meta paths. Taking the Twitter dataset as an example, the results are shown in Figure 8, and we can draw the following conclusions. On the one hand, as the number of metapaths used increases, that is, the semantic information used by SESGAT increases, the prediction performance of social links is gradually improved. On the other hand, the prediction performance of different metapath combinations is also different, which means that different semantic information has different contributions to the prediction of social links.

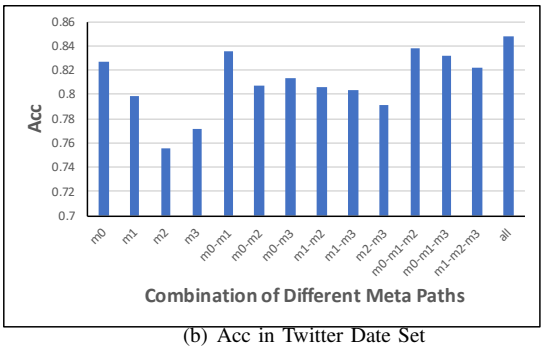
## V. Related work

Heuristic methods calculate the possibility of link formation by using different order neighbors of user nodes on the homogeneous social networks. CN Index [2] usese the number of common neighbors of two nodes as a measure of the likelihood of link formation. PA Index [1] usese the product of the degrees of two nodes in the network as the possibility of link formation between them. JC Index [2]use the ratio of the common neighbors of two nodes to the total number of their neighbors as a measure. RA Index [4] and AA Index [3] use different methods to give a weight to each common neighbor of the two nodes, and then count the sum of the weights of all common neighbors as a measure of the possibility of link formation. Considering that two nodes are similar if they are similar to two similar nodes, SimRank [5] iteratively calculates the probability of link formation between each pair of nodes. Katz [6] measures the possibility of link formation by counting the number of paths of different lengths between two nodes.

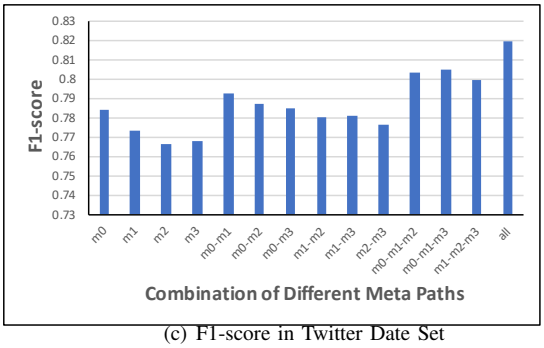
Deepwalk [8], node2vec [9] and LINE [20] use the topology information of the homogeneous networks to learn the low-dimensional node embeddings. [10], [21] and [22] use semantic information in heterogeneous networks to learn low-dimensional node embeddings. [15] proposes a node embedding method using graph neural network and attention mechanism in homogeneous networks. [16] uses metapath and graph attention mechanisms to learn low-dimensional node embeddings from heterogeneous networks. The low-dimensional node embeddings are then used to link the prediction tasks.




(a) AUC in Twitter Date Set




(b) Acc in Twitter Date Set




(c) F1-score in Twitter Date Set

Fig. 8. Performance of SESGAT under different metapath combinations.

WLNM [23] extracts locally closed subgraphs of links from the homogeneous networks, and then uses fully-linked neural networks to use these closed subgraphs for model training and link prediction. SEAL [7] usese graph convolutional neural networks to learn abstract representations of edges from such locally closed subgraphs and perform link prediction.

## VI. Conclusion

In this paper, we proposed a new method, called SESGAT, for social link prediction in heterogeneous social networks. It makes full use of the topological and semantic information in the networks by meta-paths based edge-centric semantic subgraphs, intra-semantic-subgraphs attention mechanism and inter-semantic-subgraphs attention mechanism. The experimental results on three real world datasets show the superior performance of SESGAT over other methods in real heterogeneous social networks.

## VII. Acknowledgment

This paper is supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization at Nanjing University.

## References

[1] A.-L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, *Evolution of the social network of scientific collaboration.*, In Physica A, 2002.

[2] M. Hasan and M. Zaki, *A Survey of Link Prediction in Social Networks.* In Social Network Data Analytics, 2011.

[3] Adamic, Lada A. and Adar, Eytan, *Friends and neighbors on the web*, Social Networks, 25(3):211-230, 2003.

[4] T. Zhou, L. Lu , and Y. Zhang, *Predicting missing links via local information*, The European Physical Journal B, 2009.

[5] Glen Jeh and Jennifer Widom, *Simrank: a measure of structural-context similarity*, In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 538543. ACM, 2002.

[6] L. Katz, *A new status index derived from sociometric analysis*, Psychometrika, 1953.

[7] M. Zhang and Y. Chen, *Link Prediction Based on Graph Neural Networks*, In NeurIPS 2018: 5171-5181.

[8] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, *Deepwalk: Online learning of social representations*, In SIGKDD. 701710, 2014 .

[9] Aditya Grover and Jure Leskovec, *node2vec: Scalable feature learning for networks*, In SIGKDD. 855864, 2016.

[10] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami, *metapath2vec: Scalable representation learning for heterogeneous networks*, In SIGKDD. 135144, 2017.

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

[12] Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst, *Convolutional neural networks on graphs with fast localized spectral filtering*, In NIPS. 38443852, 2016.

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

[14] William L. Hamilton, Rex Ying, and Jure Leskovec, *Inductive Representation Learning on Large Graphs*, In NIPS. 10241034, 2017.

[15] Petar Velickovic , Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, *Graph Attention Networks*, In ICLR, 2018.

[16] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, Philip S. Yu, *Heterogeneous Graph Attention Network*, In WWW 2019: 2022-2032.

[17] Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec, *Embedding logical queries on knowledge graphs*, In Advances in Neural Information Processing Systems. 20302041, 2018.

[18] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, Eduard H. Hovy, *Hierarchical Attention Networks for Document Classification*, In HLT-NAACL 2016: 1480-1489.

[19] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, Juanzi Li: Social Influence Locality for Modeling Retweeting Behaviors. IJCAI 2013: 2761-2767

[20] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei: LINE: Large-scale Information Network Embedding. WWW 2015: 1067-1077

[21] Yujie Fan, Shifu Hou, Yiming Zhang, Yanfang Ye, Melih Abdulhayoglu: Gotcha - Sly Malware!: Scorpion A Metagraph2vec Based Malware Detection System. KDD 2018: 253-262

[22] Lichao Sun, Lifang He, Zhipeng Huang, Bokai Cao, Congying Xia, Xiaokai Wei, Philip S. Yu: Joint Embedding of Meta-Path and Meta-Graph for Heterogeneous Information Networks. ICBK 2018: 131-138

[23] Muhan Zhang, Yixin Chen: Weisfeiler-Lehman Neural Machine for Link Prediction. KDD 2017: 575-583