MANE: Organizational Network Embedding with Multiplex Attentive Neural Networks

Yuyang Ye, Zheng Dong, Hengshu Zhu, *Senior Member, IEEE*, Tong Xu, Xin Song, Runlong Yu, Hui Xiong, *Fellow, IEEE*

Abstract—Every organization has organizational networks for exchange of ideas and information. It is believed that organizational network analysis (ONA) can help the business be more effective. While considerable research efforts have been made for visualizing and analyzing relationships in organizational networks, it lacks a holistic way to model the complex social structures and rich semantic information of these networks. Indeed, employee behaviors can occur across different communication platforms, such as email and instant messaging systems, which naturally lead to the multiplex structure of organizational social networks. Meanwhile, it is also a challenge to model the impact of semantic information, such as employee attributes and organization charts, and the collaboration relationships of employees. To this end, in this paper, we propose a Multiplex Attentive Network Embedding (MANE) approach for modeling organizational social networks in a holistic way. Specifically, we first develop a multiple attributed random walk approach to jointly model multiple networks, with the integration of external work information. Then, we preserve the network structure by maximizing the probability of predicting the central node based on the surrounding context nodes. In particular, we introduce an attention mechanism to assign a weight to each context node in the training process, according to its attributed relation and structural relation with the central node by utilizing the k-core algorithm and the shortest path algorithm. In this way, the embedding results can be kept consistent with their structural relationships. Furthermore, to solve some department-level tasks, we introduce an attentive relational transition method to learn the representation of departments in the organizational networks. Finally, we evaluate the performance of MANE with extensive experiments on real-world data for three important talent management tasks, namely employee performance prediction, employee turnover prediction and department performance prediction. We also conduct a link prediction task to validate the effectiveness of employee embedding. Experimental results clearly show the effectiveness and interpretability of MANE for organizational network analysis.

Index Terms—organizational network analysis, network embedding, human resource management

1 INTRODUCTION

In modern organizations, people always build informal "go-to" teams for facilitating business collaborations beyond organizational structures. These spontaneously-formed communication and socio-technical connections are usually referred as organizational social networks. Indeed, organizational network analysis (ONA) can help understand the importance of critical connections and flows in organizations, and thus makes those leaders aware of the importance of vibrant communities and employees to be much more targeted and effective in business actions. Therefore, ONA has attracted increasing research attention in recent years, as an emerging and diverse tool for enhancing talent management [1] [2], such as HIPO identification [3], turnover prediction [4], and performance evaluation [5]. While considerable research efforts have been made in this direction, it still lacks an effective approach to model the complex social structure and rich semantic information of organizational social networks.

In the literature, with the development of embedding techniques, large efforts have been made for better describing the networks as low-dimension vectors, and further support related applications, such as node classification [6], [7] and link prediction [8], [9]. However, these techniques cannot be well-adapted to model organizational social networks. Indeed, different from ordinary social networks, the organizational social network is consisted of multiple individual networks with greatly different structure, since employee behaviors are always scattered across different work platforms. For instance, employees are usually mutually connected via multiple ways, such as the instant message network for group discussion, and the email network for official notification. In this case, the multiple networks should be carefully integrated with consideration of their own characteristics. Therefore, traditional network embedding techniques which focus on the single network representation could not fully describe the multiplex structure of organizational social network, or even lead to bias. Meanwhile, how to model the impact of external work-related semantic information (e.g., employee attributes and organization charts) is also challenging. Furthermore, due to the characteristic of our organizational network modeling problem, solely learning the embedding of nodes (i.e., employees) cannot satisfy some tasks in ONA such as depart-

- *Y. Ye and H. Xiong (corresponding author) are with the Management Science and Information Systems Department, Rutgers University, Newark, NJ07102, USA.*
  E-mail: {yuyang.ye, hxiong}@rutgers.edu.
- *Z. Dong, H. Zhu (corresponding author) and X. Song are with Baidu Talent Intelligence Center (TIC), Baidu, Beijing, 100085, China.*
  E-mail: {dongzheng01, songxin06}@baidu.com, zhuhengshu@gmail.com.
- *R. Yu and T. Xu are with the Anhui Province Key Laboratory of Big Data Analysis and Application (BDAA), School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China.*
  E-mail: yrunl@mail.ustc.edu.cn, tongxu@ustc.edu.cn.

ment modeling because obtaining subgraph embedding by some simple operations (e.g., average) would lead to a poor performance. Thus, to better modeling the organizational network, we need to explore subgraph modeling problem in this work.

To this end, in this paper, we propose a Multiplex Attentive Network Embedding (MANE) approach to comprehensively describe the collaborative behaviors within organizational social networks. Specifically, we first build a bipartite attributed network to connect employees and working information, and then adopt a joint random walk [10] on multiple networks to generate the node sequences. Then, we preserve the network structure by maximizing the probability of central node prediction based on surrounding context nodes, where the attention mechanism is adapted to assign weight for each context node according to its structural similarity relations. In particular, the k-core [11] and approximated shortest path algorithm are utilized to estimate the structural relation, so that the network structure will be better preserved. Furthermore, we also extend our MANE framework to learn the representation of department in organizational network for some graph learning tasks. To be specific, on the basis of MANE approach, we propose a relational transition based approach with an attention mechanism to obtain the department embedding by modeling the impact of each employee to the whole department. The contributions of this paper can be summarized as follows:

- We study the problem of modeling organizational social network from the perspective of multiplex network embedding, which fills the void in previous studies and can benefit many data-driven applications of talent management, such as employee turnover prediction and performance evaluation.
- We propose a delicately-designed multiplex network embedding approach MANE, where rich semantic information can be intuitively integrated by adopting structural and attributed attention mechanisms.
- We also extend our MANE framework to learn the department embedding, which integrate the information of employees in the same department by designing an attention mechanism to obtain the impact of employee to the department while preserving the attribute of department through objective function.
- We evaluate the performance of our approach MANE with extensive experiments on a real-world dataset, including several important talent management tasks, namely employee turnover, performance prediction and department performance prediction. Experimental results clearly validate the effectiveness and interpretability of our approach MANE for organizational network embedding.

Overview. The rest of this paper is organized as follows. First, we briefly summarize the related works of this paper in Section 2. Then, preliminaries and formal definition of organizational network embedding will be introduced in Section 3. Afterwards, technical details of our MANE framework for employee modeling and department modeling will be respectively explained in Section 4 and Section 5. Section 6 will be the validation part to illustrate the experiments on a real-world dataset. Finally, we conclude this paper in Section 7.

2 Related Work

The related studies of this paper can be grouped into three categories, i.e., network embedding, graph representation learning and organizational network analysis.

2.1 Network Embedding

In recent years, network embedding has become a popular research subject in the field of network representation learning. Under the inspiration of neural network, many efficient and novel network embedding methods were proposed while the traditional methods usually have high time complexity, such as analyzing network spectral property [12]. Some of them adopted random walk algorithm on network and assumed the neighbor nodes which have the similar network structure have similar network embedding. This thought is first created by [10], which generated several node sequences as training samples and utilized the skip-gram algorithm proposed in Word2vec [13] on these sequence. On this basis, [14] conducted a biased random walk by introducing two parameters p and q to control the process, for exploring the homogeneity as well as the structure information of nodes. Besides, some models aimed to explore some specific structural relation between nodes in the network, such as LINE proposed by [15] which learned the node representation for preserving the first-order proximity or the second-order proximity, and Struct2vec proposed by [16] which captures the structural identity to measure the similarities between nodes and obtain the network embeddings. Meanwhile, some other network embedding methods were inspired by the rapidly evolving deep learning techniques. For example, SDNE [17] first proposed a semi-supervised auto-encoder to preserve both local and global network structures. GraphGAN [18] introduced a generative adversarial network based model where the generator learns the connectivity distribution while the discriminator predicts the existence of edges between each two nodes. GCN [19] defines a convolution operator on graph and updates node embeddings by its neighbors. GraphSAGE [20] learns the functions by sampling node neighborhoods with the inductive learning ability for unseen graphs. GAT [21] assigned different weights to different neighbors by attention mechanism with a better interpretability.

The methods we introduced above are all suitable for single network analysis but cannot adopted for multiple network directly. Furthermore, there are also some existing methods proposed for multiplex network embedding problem. PMNE [22] proposed three different methods to model a multiplex network into a unified vector space for learning a overall embedding. MVE [23] proposed a framework combine single network embedding of each view with using an attention mechanism to learn the weight of each view. MNE [24] proposed an unified model to jointly learn a common embedding and several additional embeddings of each view for each nodes, which need labels information to learn the weight and may have different node embeddings since it is task dependent. HAN [25] applies a two-level attention mechanism to heterogeneous graphs through metapath based neighbors. Besides, GATNE [26] proposes a unified framework to solve the multiplex attributed network by considering both network structure and node attributes, however they only generate node embedding on each view without combining.

2.2 Graph Representation Learning

Although tremendous efforts have been paid on network embedding, most of the existing methods in this field only focused on node embedding learning instead of simultaneously considering the representation of whole graph, which would probably result in the loss of some information for solving some graph based tasks, such as the department representation learning in our work. Previous graph analytical approaches were mainly based on graph kernel, which decompose the graphs into some substructures such as graphlets while defining a similarity function over these substructures for evaluating the similarity between this set of graphs [27] [28] [29]. However, these graph kernel based approaches have some limitations. First, most of them are incapable of providing the vector embedding of graphs so that some ML algorithms like GBDT cannot handle graph classification problems by using the results of these approaches. Furthermore, the similarity functions of these methods used usually lead to a high dimension and sparse graph embedding which would result in a poor performance. Recently, with the increasing number of graph learning application, some graph embedding approach were proposed. [30] learned a graph representation vector which composed of features that represent the distance to several predefined prototype graphs in order to classify and cluster the graphs. [31] proposed an unsupervised approach Graph2vec which extend Doc2vec [32] to arbitrary graph by taking the graph as a document while taking the rooted subgraphs in this graph as words to get the graph embedding through skim-gram optimization. Besides, Sub2vec proposed by [33] focus on preserving two intuitive properties of subgraph (i.e., neighborhood and structural) while formulating the embedding of subgraphs.

2.3 Organizational Network Analysis

Organizational network analysis (ONA) is a comprehensive exploration of the organizational social networks built in the interior of enterprises by using social network analysis techniques, in order to better carry out the development strategies and the talent management of the company. By using ONA, the managers could have a understanding management structure, information flows and employee interactions in the company. Traditionally studies collect data mainly from survey then apply straightforward statistical technologies to analyze the relationships of factors like job satisfaction and outcomes like turnover or revenue, and get the most value from their talent [34] [35]. Many literatures analyze talent and team behaviors through organizational networks [36] [37] [38] since social capital exists in companies need to be invested to help team grow, as valuable as financial and physical capital [36]. The understanding of the progress of organizational networks formed could benefit the personal and talent development, also help employees or managers to improve the structure of their organizations across companies [37] [38]. Recently, some network embedding based methods have proposed to solve the human resource management problems such as company competitive analysis [39] and person job fitting [40]. However, it still lacks of a comprehensive data-driven approach to quantitatively model the employees' behaviors in organizational social networks.

3 Preliminaries

In this section, we first introduce the details of our real-world dataset, and then formally define the problem of organizational network embedding.

3.1 Data Description

The real-world talent data used in this work, which consist of two datasets, were provided by a high-tech company in China. The first is the organizational social network dataset, including three types of networks respectively based on informal communication, formal communication and group collaboration network since January 2018 to December 2018. The network dataset only contains interaction count without textual information. Specifically each network shares the same vertices set V where each node stands for an employee in the company. Besides, there are some departments consisting of some particular employees while employees and departments in our network have some external working attributes.

In the second dataset, we have some working related records for model evaluation, including performance appraisal records and employee turnover records. For the privacy-protection purpose, all the sensitive information in our datasets (e.g. names) has been anonymized.

3.2 Problem Formulation

Based on the talent datasets described above, we can build the organizational networks, which reveal the relationship between employees on forming project teams or building alliances across groups [41], as follows.

Definition 1 (Organizational Networks). Organizational networks consist of several networks $G_k = (V, E_k)(k \in Z_n^+)$ where $e_k^{ij} \in E_k$ means the frequency of communication between employees $v_i, v_j \in V$ in the corresponding network G_k. Besides, there is a set of departments D in the organizational networks where each employee v_i belongs to one department $d \in D$, and each employee has a specific profession $p_i \in P$ which indicates the type of their work (e.g., engineer, marketing). Meanwhile, each employee and department has some work-related attributes (e.g, length of service, job level).

In this paper, we aim to model the employees and departments in the organizational social networks for solving three tasks in human resource management (i.e. employee and department performance prediction, employee turnover prediction) through analyzing multiple communication networks by utilizing network embedding methods. To be specific, the problem of organizational network embedding can be formally defined as follows.

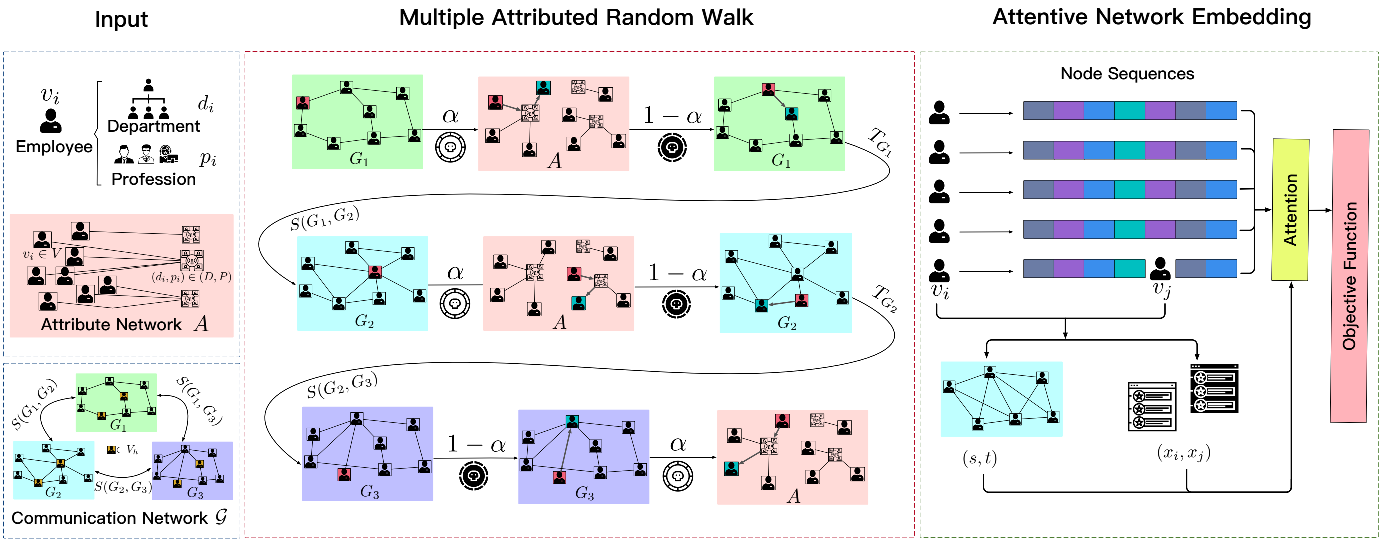


Fig. 1: The framework overview of our MANE approach.

Definition 2 (Organizational Network Embedding). Given a set of multiple organizational communication networks $\mathcal{G} = G_k = (V, E_k, D)(k \in Z_n^+)$ where V and $E_k \in V \times V$ respectively denote the set of nodes and edges in G_k, and D is the set of departments in the organization. We aim to learn a low-dimension vector representation $u_v \in \mathcal{R}_d(d << |V|)$ for each employee $v \in V$ and $r_d \in \mathcal{R}_d$ for each department $d \in D$, which not only preserves the network structure of each single network but keeps consistent across different types of networks as well.

4 Employee Modeling by Multiplex Attentive Network Embedding

In this section, we will introduce the technical details of our proposed model MANE for employee modeling problem. Specifically, the structure of our model is illustrated as Figure 1, consisting of two components. We first build an employee attribute network which connects the employee and their work-related information, and propose a joint random walk approach, adopted on multiple networks and attribute network by using two parameters to control the skip between each single network, to generate some node paths in the networks. Then, similar with the bag-of-words model (CBOW) model, we preserve the network structure by maximizing the probability of predicting the central node based on the surrounding context nodes, and fuse the attention mechanism according to the structural relations between employees, calculated by the k-core and the shortest path algorithms. Therefore, the representation of employees with the similar structure and attributes would be closer in the latent space.

4.1 Multiple Attributed Random Walk

Similar as some network embedding models proposed before, we firstly adopt random walk algorithm on the network. However, as employee collaborative behaviors are not showed in a single network but some different types of networks and we need to consider the influence brought by the additional information included in the attributes of employees, the previous method cannot be utilized directly in our problem. Thus we should propose a novel multiple attributed random walk approach. To track this issue, we first need to consider how to incorporate the attribute information in the random walk process. Our key idea is to get more walk paths between the employees in the same department and with the similar professions(i.e., employees who probably participate in the same project). Hence, inspired by [42], we build a bipartite network to extract the attributed interaction between employees, which is described in the following subsection.

4.1.1 Bipartite Network Construction

As mentioned in Section 3.2, each employee $v_i \in V$ in the multiple communication networks $\mathcal{G}$ belongs to a department $d_i \in D$ and has a specific profession $p_i \in P$. Then we combine (D, P) into W that denotes the work-related attribute information of employees and each element $w_i \in W$ equals to (d_i, p_i), thus we can define a bipartite network $A = (V, W, E_A)$ by taking $W \in \mathbb{R}^m$ as a set of nodes, where exists an edge between v_i and w_i. By building this network, we can get a more informative walking path through adopting random walk method and make the path more adaptive to our problem.

4.1.2 Random Walk Mechanism

To comprehensively incorporate all of useful information, we propose a novel random walk mechanism on multiple networks. We set two parameters $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ to control the next transition in random walk process. Assume that the current node is $v_i \in V$ in $G = (V, E) \in \mathcal{G}$, we first toss a coin which has probability α to show the obverse and $(1 - \alpha)$ otherwise. If the coin yields obverse, the path jumps to the attribute network A. We aim to find a path to $v_j \in V$ which also has an edge to w_i by two steps of walking. Let

$R_{v_i \to v_j}$ denotes the walking path and this process can be formulated in the following subsection.

$$R_{v_i \to v_j} = R_{v_i \to w_i} + R_{w_i \to v_j}, \tag{1}$$

thus we add the path (v_i, v_j) to the current walking path.

On the opposite, if the coin yields reverse, the random walk process would be continued on the communication network. We use a random number generator β to determine whether the next transition would occur on the current network or jump to one of other two networks. We set a threshold T_G based on the importance of each network which is measured by the degree centrality of the high-level employees in the whole organizations V_h in this network, as we assume the network where these employees tend to involve has a greater relation with work scene. Thus T_G is computed as,

$$T_G = \frac{\sum_{v \in V_h} deg(v \in G)}{\sum_{G \in \mathcal{G}} \sum_{v \in V_h} deg(v \in G)} \tag{2}$$

If the generated number $\beta < T_G$, the random walker would still continue on the current network. Adversely, the random walk process would jumps to another communication network $G^{'} \in (\mathcal{G} - G)$ with the probability as,

$$P_{G^{'}} = \frac{1 - S(G, G^{'})}{\sum_{G^{''} \in (\mathcal{G} - G)} (1 - S(G, G^{''}))}, \tag{3}$$

where S is the similarity function of two networks. This probability function indicates that the random walk process has the higher probability to transit to the network which is more dissimilar with the current network, because our random walk mechanism tend to integrate the additional information in different networks rather than cover the duplicated information. To be specific, inspired by [43], the similarity between two networks G_1 and G_2 which share the node set can be calculated as,

$$S(G_1, G_2) = \frac{\sum_i min(deg(v_i \in G_1), deg(v_i \in G_2))}{2 * max(E_{G_1}, E_{G_2})} \tag{4}$$

The next node in the current walking path v_j would be one of the neighbor of v_i in $G^{'}$.

Generally we can also calculate the probability of the walk from v_i to v_j on each single communication network as follows.

$$P(R_{v_i \to v_j}) = \frac{e_{ij}}{\sum_{l=1}^{n} e_{il}}, \tag{5}$$

where e_{ij} denotes the weight of the edge between node i and node j in the network G (0 if there does not exist an edge between i and j).

Through this random walking mechanism, the performance of random walk would be boosted due to that the walking path in each single network would supply the additional information and there would not be ignorance of any employee communication behavior. Hence by adopting this multiple random walk approach, we can incorporate the useful information in all of the organizational networks with different types.

4.2 Attentive Network Embedding

After above mentioned multiple attributed random walk, we generate the training node sequences. In this section, we will introduce how to learn the representation of employees. We first treat the generated random walk paths as the training corpus where the neighbour nodes have the similar network structure. Based on the assumption that the employees with similar network structure and attributes have the similar representation, in this paper we adopt the continuous CBOW model which uses the surrounding nodes to predict the central node while exploring some additional structural information and attributed information by designing an attention network. The objective function can be formulated in the following subsections.

4.2.1 Objective Function

The context node vector is obtained by averaging the embeddings of each node. For each node v_i in generated training sequences, the model utilize the surrounding context nodes $\{v_{i-k}, ..., v_{i+k}\} - \{v_i\}$ to predict the center node v_i, and maximize the log probability to preserve the structure of organizational networks, which is formulated as follows:

$$\begin{aligned} L &= -\log p(v_i | context(v_i)) \\ &= -\log \frac{\exp u_i^{'T} u_{context(i)}}{\sum_{|V|} \exp u_i^{'T} u_{context(i)}} \end{aligned} \tag{6}$$

where $u_i^{'}$ is node context embedding, $u_{context(i)}$ is the comprehensive representation of the surrounding context nodes. Different with the previous works that obtain this representation vector by averaging the embeddings of each surrounding node and treat these nodes equally, in this work we aim to assign the weight to each context node in the CBOW model to characterize the different importance of context nodes because different neighbour nodes (i.e., collegues) may contribute differently for a central node (i.e., a specific employee) due to their different working relations. Thus considering one node sequence sample $\{v_{i-b}, ..., v_i, ..., v_{i+b}\}$, we calculate the context node vector representation by the weighted sum of the surrounding node embeddings as follows:

$$u_{context(i)} = \sum_{j \in [i-b, i+b] - \{i\}} a_{ij} u_j \tag{7}$$

where a_{ij} is the weight assigned to the context node embedding u_j and b is the window size. [44] for word embedding has a similar idea, which explores the different influence of the words with different positions in a sentence. However, it has challenges in our problem since the position of nodes in the generated random walking sequences will not hold such many semantics. Hence, we need to propose a novel attention network to calculate the weight value of surrounding node, which we will introduce as follows.

4.2.2 Attention Network

Since the generated random walk paths do not have enough semantic information to judge how importantly a node could influence another node, we aim to explore some additional information by designing an attention network. Compared with other models like GAT [21], we calculate the

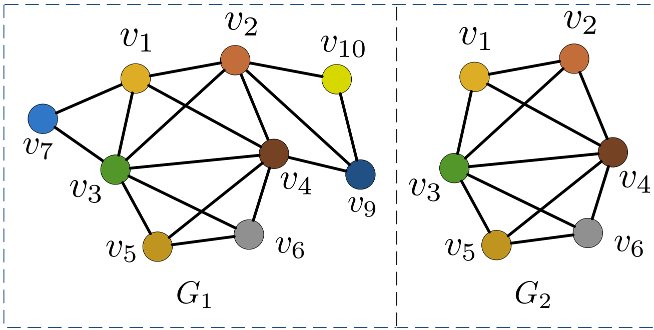


Fig. 2: Examples of employee collaborative graph.

attention by structural and attributes separately, then could better know where the importance come from, i.e., two employees play similar roles or come from the same team by structural attention; they work on similar projects or share similar skills by attributes attention. On the other hand, we model the structural attention in a novel way by grouping the nodes pair according to denseness and closeness, and thus we could analyse the importance of group working in employee modeling.

Structural Attention Network. For structural attention, considering an example of the organizational social network denoted as G_1 in Figure 2,

Example 1. In this example we focus on learning the embedding of node v_1. We can observe from the subgraph employee v_1 may collaborate more with v_2 than v_7 since v_1 may form a denser graph with v_2 than v_7. On the other hand, employee v_1 may collaborate more with v_2 than v_5 since the shortest path between v_1 and v_2 is shorter than v_1 and v_5.

Thus, for assigning the different weight of node pairs during model training, we model their relationship by the maximum k-core they can form, which can reflect the density of the networks, and the shortest path between them. We first introduce the definition of k-core as follows:

Definition 3 (k-core). Given a graph G and a positive integer k, the k-core of graph G denoted as H_k is the maximal subgraph of G, where $deg_{H_k}(v) \geq k$, for each $v \in V(H_k)$.

G_1 in Figure 2 is 2-core since each node has at least 2 neighbors and G_2 is a 3-core subgraph of G_1 since each node has at least 3 neighbors. So employee v_1 form a 3-core with v_2 in G_2 which is denser than the 2-core formed with v_7 in G_1. To model the densest k-core subgraph node pairs both exist in, we first need to compute the k-core decomposition to get each node's core value while there is a linear algorithm [11] for k-core in $O(|E|)$ time where $|E|$ is the number of edges.

K**-core index construction.** For the self-completeness of our techniques and reproducibility, we apply the existing core decomposition [11] on graph G to construct the k-core index for nodes in graph G. The algorithm is detailedly outlined in Algorithm 1. Particularly We implement the bulk deletion optimization algorithm [45] to speed up. Therefore, according to Theorem 1, we can obtain the coreness of subgraph where node pair (v_i, v_j) co-exist.

Theorem 1. For two nodes pairs $(v_i, v_j) \in G$, the maximum densest s-core subgraph they co-exist in is $min(\delta(v_i), \delta(v_j))$.

Proof 1. This follows the nested property of k-core, which means k_1-core of graph G must be a subgraph of k_2-core of graph G if $k_1 > k_2$.

After that, with BFS algorithm we could get the shortest path between the node pairs from random walk sequences. Thus for each node pair (v_i, v_j) in random walk paths, we could get a relative relation (s, t), which means this node pair form a maximum s-core, and the shortest path between v_i and v_j is t.

Then we could assign different weights to each context node based on the pre-computed coreness and shortest path length by proposing an attention network where each node v_j is assigned an attention level representing how much important it is in predicting v_i. Particularly, as coreness values s of nodes in our organizational networks are distributed nonuniformly as Figure 3, we separate the coreness value into several intervals and use the median of interval $m(s)$ to denote s so that the dimension and sparseness of the attention matrix would be reduced and the parameters of attention network would be better updated.

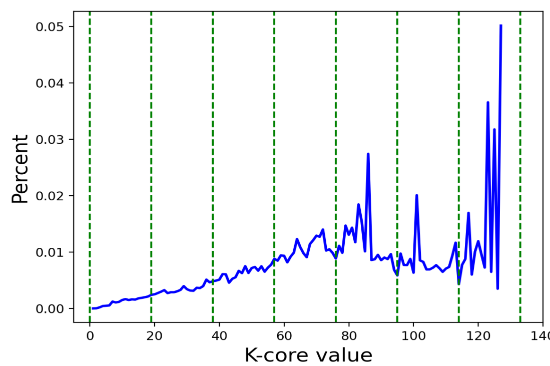


Fig. 3: Employee Collaborative Graph Example

Hence, the attention weight a^1_{ij} given to node $v_j \in V$ which has the relative relation (s, t) with the central node $v_i \in V$ is computed as:

$$a^1_{ij} = \frac{\exp K_{n,m(s),t} + c_{m(s),t}}{\sum_{j \in [-b,b]-\{0\}} \exp K_{n,m(s),t} + c_{m(s),t}} \tag{8}$$

where the dimension of K is $|V| \times ((|m|+1) * d)$, d is the diameter of graph and $|m|$ is the number of intervals. K is a set of parameters that determines the importance of each node in each relative network structure, c is a bias of dimension $(|m|+1) * d$, which is conditioned only on the relative relation.

Attributed Attention Network. As the assumption we mentioned above, we also aim to assign the weights to the employee pairs based on the similarity of their attributes. We indeed design the other attention network. Specifically, the major attributes used in this part are similar as [3]. For each numerical feature, we adopt a normalization operation, in order to prevent the prediction being dominated by the factor which has a large range of distribution. Meanwhile,

Algorithm 1 Core Index Construction

Require: A graph $G = (V, E)$.
Ensure: Coreness $\delta(v)$ for each $v \in V_G$.

```
Sort all nodes in G in ascending order of their degree;
while G ≠ ∅ do
    Let d be the minimum degree in G;
    while there exists deg_G(v) ≤ d do
        δ(v) ← d;
        Remove v and its incident edges from G;
        Re-order the remaining nodes in G in ascending
        order of their degree;
    end while
end while
Store δ(v) in index for each v ∈ V_G;
```

for the categorical features, we utilize the one-hot encoder as representation. In addition, for the missing value, we adopt the average pooling operation. Finally, all the features are concatenated into a representation vector and fed into a fully-connected embedding layer to obtain a 20-dimension feature vector x_i for each employee $v_i \in V$. Then we design an attention network to assign the attributed weights a_{ij}^2 to each context node u_j as

$$s_{ij} = \boldsymbol{h}^T ReLU(\boldsymbol{W}[x_i, x_j]), \tag{9}$$

$$a_{ij}^2 = \frac{exp(s_{ij})}{\sum_{(i,j)\in R_x} exp(s_{ij})}, \tag{10}$$

where $\boldsymbol{W}$ and $\boldsymbol{h}$ are parameters to be learnt and $[x_i, x_j]$ means the concatenation of two vectors.

Hence, the final weight assigned to context node v_j with the central node v_i is,

$$a_{ij} = \gamma_1 a_{ij}^1 + \gamma_2 a_{ij}^2 \tag{11}$$

where γ_1 and γ_2 are hyper-parameters to balance the weight of these two attention network, and $\gamma_1 + \gamma_2 = 1$.

4.3 Model Learning

By optimizing the objective function (5) we mentioned above, which maximize the log probability, we can learn the final representation of employee collaborative behavior. However, it is computationally expensive to directly optimize the function (5), because we need to traverse all nodes in the network when computing the conditional probability $p(v_i|context(i))$ and there are usually very large the number of nodes. For solving this problem, we adopt negative sampling mechanics proposed by [13], which selects negative samples according to the noisy distribution $P_{neg}^k(v)$ for each node context. Therefore, the objective function equation could be replaced by the following:

$$\begin{aligned} L_1 &= \log\sigma\left(u_i'^T u_{\text{context}}(v_i)\right) \\ &+ \sum_{i=1}^{K} E_{v_n \sim P_{neg}(v)}\left[\log\sigma\left(-u_n'^T u_{\text{context}}(v_i)\right)\right] \end{aligned} \tag{12}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function, N is the number of the negative samples and v_n is the sampled negative context node set for v_i according to the noisy distribution $P_{neg}(v) \propto d_v^{3/4}$ where d_v denotes the average degree of the nodes in all of the organizational social networks. This function aims to maximize the probability of positive samples while minimizing the probability of negative samples.

5 Department Modeling by Multiplex Attentive Network Embedding

As we discussed above, some tasks in ONA such as department modeling cannot be only exploring the representation of nodes, which would result in the loss of abundant information of some high-level substructures that would be useful in these tasks. Indeed, in this section, we extend our MANE framework for learning the representation of subgraphs (i.e., departments) in organizational networks. As Figure 4 shows, we set several transition matrices to embed all these representation vectors to a new relation space, where each transform matrix corresponds to a profession. In this relation space, we get the representation vector of each department by designing an attention network to get the weighted sum of the employee embeddings in the same department. In this transformation process, we aim to make the nodes in the same department close to each other while making the department with the similar attribute get the similar representation in the new vector space.

5.1 Relation Transformation

By MANE we obtain the representation vector u_i for each node v_i in organizational networks. Then in this task we aim to present the connection between the employees and the departments. Hence, we define a new relational space to reflect this connection by putting the node embedding to this space through projection. To be specific, we design several transform matrices. As mentioned before, in our organizational network, each employee belongs to a specific profession p where different profession corresponds to different type of work (e.g., marketing, software engineering) and make different contributions to the whole department as well. To fit this characteristic, we build a set of transition matrices $\{M_p\}, p \in P$ where the employees belongs to the same profession are processed by the same matrix. Therefore, for $v_i \in p$, its representation vector in the new relational vector space can be got as following,

$$t_i = u_i M_p \tag{13}$$

Then, since the employees with different levels would bring different impact on the department modeling, we design an attention mechanism to assign the different weight to employees according to their levels in department modeling process. To obtain the level of employees, we explore the report chain with tree structure in the organization where parent node is the superior of the child node and the employees in the same layer have the same level. Hence, assuming that v_i is in the j-th layer in the report chain, the weight α_i that denotes the importance of v_i in department modeling can be formulated as follows,

$$\begin{aligned} e_i &= tanh(W_j t_i + b_j), \\ \alpha_i &= \frac{\exp(e_i)\, v_\alpha}{\sum_{v_l \in D_k} \exp(e_l) v_\alpha}, \end{aligned} \tag{14}$$

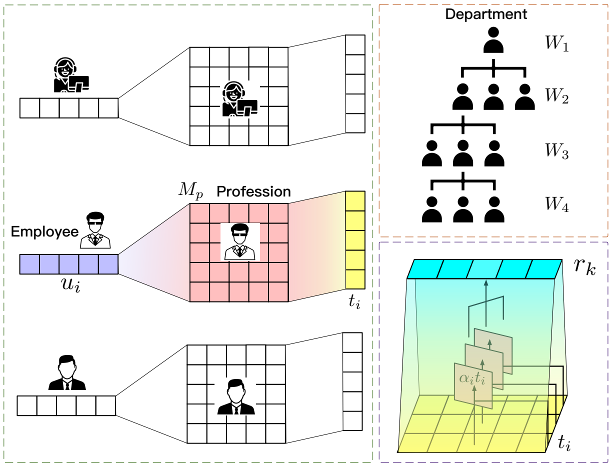


Fig. 4: An illustration of MANE on department modeling.

Fig. 5: An Example of Keyword Set.

where $\{W_j\}_j, \{b_j\}, v_\alpha$ are the randomly initialized parameters to learn. Therefore the representation vector of the department D_k can be computed as following,

$$r_k = \sum_{v_i \in D_k} \alpha_i t_i. \tag{15}$$

5.2 Model Learning

For effectively training MANE-D to accomplish the department modeling task, we propose a comprehensive objective function with a relational constraint. As we introduced above, the loss function of this task consist of two components because the model learning process is based on two criteria. First, for employees in the same department, our goal is making their representation vectors as close to each other as possible in the relational vector space. Besides, the similar department should have the similar representation.

To this end, we define a similarity measurement θ between departments based on some of their attributes. For modeling this similarity we first compute the number of employees of each department and corresponding proportion of each profession within these department. We put these attributes into a vector after normalization and get the cosine similarity θ_1 between them. Furthermore, we also obtain a textual description of each department from which we could get a keyword set by extracting some technical word in these description where we give an example in Figure 5. Then we adopt TF-IDF operation on these keyword sequence and we also obtain cosine similarity of textual attributes for two departments as θ_2. Finally, the similarity between two departments can be got as $\theta = (\theta_1 + \theta_2)/2$. For two departments $D_g, D_h \in D$, the higher θ_{gh} is, the more closer the distance between their representation vector is.

To achieve this similarity we described above, we minimize the difference between the corresponding representation vectors. Thus the objective function can be formulated as follows,

$$L_2 = \sum_k \sum_{v_i, v_j \in D_k} ||t_i - t_j||_2 + \sum_{D_g, D_h \in D} \theta_{gh} ||r_g - r_h||_2 \tag{16}$$

Furthermore, as we jointly learning the representations for both employees and departments, we design a relational constraints of them and integrate it with the aforementioned two objective functions as regularization norm. We minimize the Manhattan Distance between the employee embedding and their corresponding department embedding to make them distribute in one vector space as far as possible. The regularization norm can be formulate as:

$$L_3 = \sum_k \sum_{v_i \in D_k} ||r_k - u_i|| \tag{17}$$

Therefore, our final objective function is,

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \tag{18}$$

where λ_1, λ_2 and λ_3 are the hyper parameters to balance these three different constraints. Therefore, we could minimize this loss function for jointly training MANE and solving the two task synchronously. Gradients of the loss function with respect to all the parameters are computed with back-propagation, and we use Adam algorithm to optimize our model where parameters are updated after each training instance using a fixed learning rate.

6 Experiment

In this section, we evaluate our MANE model with extensive experiments on real-world talent data. We will give some discussion on the experimental results. Meanwhile, we also introduce some case studies and validate the interpretability of our model.

6.1 Experimental Settings

- **Data preprocessing.** As introduced in preliminaries, the dataset used in our experiments were provided by a high-tech company in China which we have briefly introduced in Section 3.1. In the experiments, to make networks fit the request of our multiplex network embedding problem, we only retained the employees who appeared in all networks. After this process, the statistics of these networks can be list as Table 1. In addition, there are 365 departments in our networks.
- **Evaluation Metrics.** Indeed, the performance prediction, the turnover prediction task and department performance task in this paper can be regarded as binary classification problems. Therefore, in these three experiments, we adopted the Accuracy, F1-score and AUC as the evaluation metrics to validate the performance of our model and other baselines. Besides, in link prediction task, we exploit the widely used top-k ranking metrics (i.e., precision@k and recall@k) as metrics where we set $k = 5$.

TABLE 1: Some statistics of the dataset.

Network	#Nodes	#Edges	#Max_core	#Importance
Formal	28,754	2,169,865	127	0.5657
Informal	28,754	1,685,635	86	0.2614
Group	28,754	402,358	49	0.1749

- **Parameter Settings.** As we know, parameter setting has great impact on the convergence and results of our model. Thus we would introduce some detailed parameters in the experiments. First, we set the dimension of embeddings for both employees and departments as 100. Meanwhile, we set the length of random walk as 30, the walk per nodes as 30 and window size in CBOW based optimization as 4. Moreover, we initialize the node embedding and all transform matrices as Gaussian distribution with a mean of 0 and standard deviation of 0.01. And we initialize the node context embedding as $\vec{0}$ vectors. Besides we set the hyper-parameters as $\alpha = 0.1, \gamma_1 = 0.6, \gamma_2 = 0.4, \lambda_1 = 0.05, \lambda_2 = 0.04, \lambda_3 = 0.02$. In the training process, we set the batch size as 64. Finally, we used the Adam algorithm to optimize our model where the learning rate α was set to 0.01.

6.2 Baseline Methods

In order to demonstrate the effectiveness of the proposed MANE model on two tasks introduced above respectively, we will compare our model with the following state-of-the-art baseline methods on some real-world applications.

For employee modeling task:

- **DeepWalk** [10] first adopts random walk on the network data, which makes the generated walking paths as sentences and then trains the node embeddings by using Skip-gram algorithms.
- **LINE** [15] learns the node embedding by preserving the first-order or the second-order proximity of the network structure.
- **TADW** [16] is a matrix completion based network embedding method that incorporates node attributes, which utilize both the network structure and node attributes information in network embedding learning.
- **PMNE** [22] proposes a cross layer network embedding method which utilizes hyper-parameters to obtain a random walk across each layer.
- **MNE** [24] uses a common embedding and several additional embedding for each type of relation, which is jointly learned in a united model, to learn the embedding of a node in multiple networks.
- **HAN** [25] leverages node-level and semantic-level attention to learn the node embedding in heterogeneous networks while capturing the complex structure and rich semantics, where in heterogeneous network employees with different professions have different node types while edges in different networks have different edge types.
- **HGT** [46] introduces type-specific transformation matrices to model heterogeneous networks while learning the importance of different nodes and relations based on the self-attention architecture. For this method, the setting of node types and edge types is the same as HAN.

For department modeling task:

- **Attribute** as a heuristic method to utilize some basic attributes of departments like textual descriptions to denote department embedding as input.
- **WL-kernel** [28] is a handcrafted feature based method which decomposes graphs into rooted subgraphs and computes their kernels.
- **Sub2vec** [33] can generate embedding of arbitrary subgraphs and preserve some intuitive properties by generating random walk within subgraph and maximizing objective functions as Skip-gram or CBOW.
- **Graph2Vec** [31] extends Doc2vec framework to learn the distributed representations of arbitrary sized graphs by treating the whole graph as a document.

Moreover, we utilized three variants of our model to validate the effectiveness of some components, including MANE-A (removing the attention mechanism adopted in MANE for learning the employee embedding), MANE-E (only using the average of employee embeddings learned by MANE as the department embedding) and MANE-P (only utilizing one transformation matrix for all node embeddings in drawing the department representation).

6.3 Employee Modeling

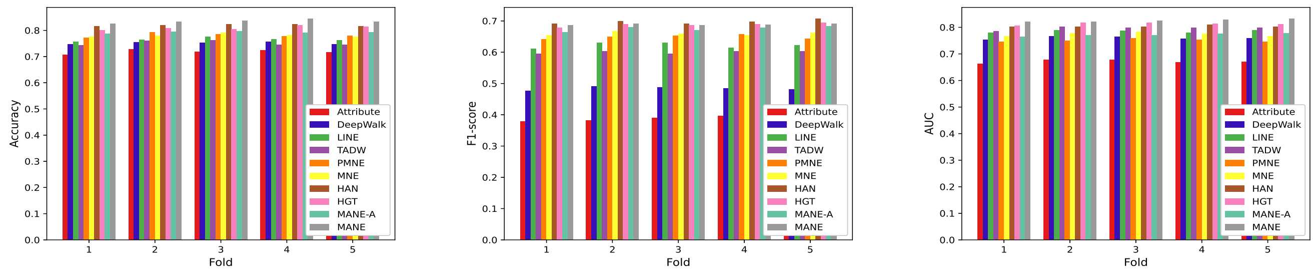
We evaluate the effectiveness of our model for employee modeling on two human resource tasks and report the results in this section.

6.3.1 Employee Performance Prediction

Employee performance prediction is always a major concern in the field of human resource management. Here we assume the performance of employees has a strong relation with their behaviors in organizational networks since communication between employees are mostly related to work events (e.g., project team). In the experiment, we totally obtained 14,745 employees as candidates and treated the performance prediction as a node classification problem. Besides, we had a list which recorded the employees who had been named as high performance by human resource management experts in 2018 and could be regarded as the ground-truth. Specifically, there are 4,088 positive samples (i.e. high performance employees) and 10,657 negative samples. Following the previous work [47] [48], we select the linear SVM [49] as the classifier, in order to reduce the impact of classifier on our classification task. And we select 80 percent of the labeled nodes as training data and use the rest as test. Besides, in this experiment, we utilize a five-fold cross-validation to evaluate each test approach. We list the average performance in Figure 6.

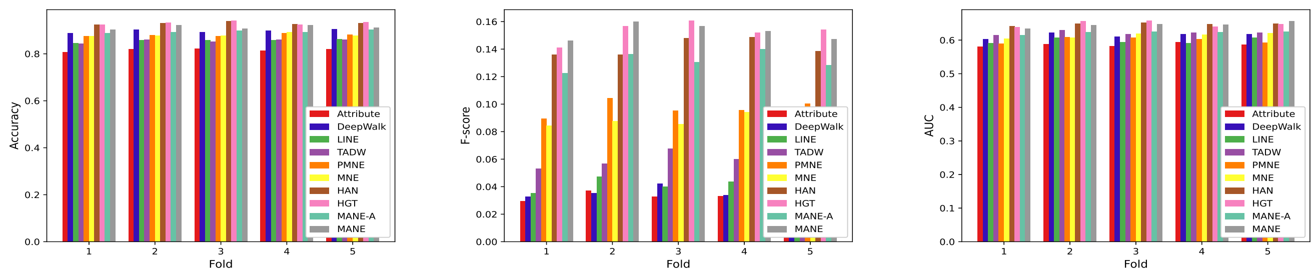
6.3.2 Employee Turnover Prediction

In the working environment, the turnover behaviors of employees have a great potential to be influenced by their colleagues through contagious effect, which has been already demonstrated by previous works [4]. Therefore, the analysis of collaborative behavior in organizational networks can play an important role in turnover prediction



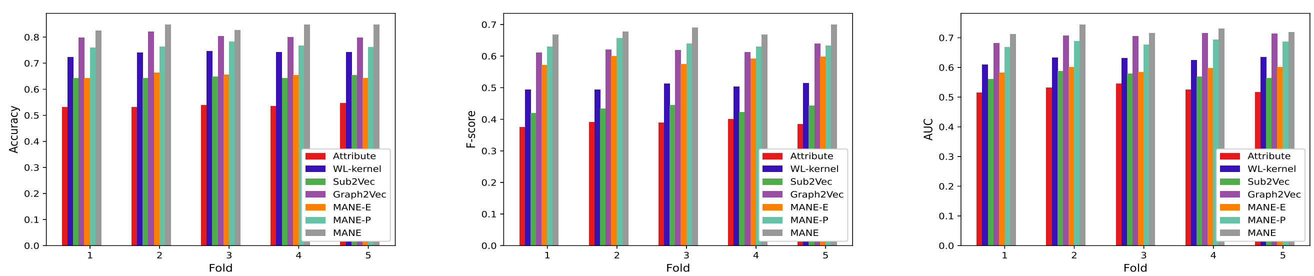
(a) Accuracy (b) F-score (c) AUC

Fig. 6: The overall performance of MANE on employee performance prediction under five-fold cross validation.



(a) Accuracy (b) F-score (c) AUC

Fig. 7: The overall performance of MANE on employee turnover prediction under five-fold cross validation.



(a) Accuracy (b) F-score (c) AUC

Fig. 8: The overall performance of MANE on department performance prediction under five-fold cross validation.

task. In the experiment, we totally obtained 8,961 employees as candidates, including 969 positive samples (i.e. employees who turnover from company during 2018) and 7,992 negative samples. As the turnover prediction problem is a binary classification problem, we can treat this task as a node classification problem. Besides, we had a list that recorded the employees who leave the company during the given period, which could be regarded as the ground-truth. Specifically, we first transform the basic features of each employee which are used in the turnover prediction task in real world into a vector. Then we concatenate them with the node embeddings learnt by the model into a new vector, as the input of classifier. The same as performance prediction task, we also select the linear SVM as the classifier and set the same training/test ratio as performance prediction task. We also show the performance of our model in five-fold cross validation in Figure 7.

6.3.3 *Result Analysis*

In this section we analyse the experimental results of employee modeling on the above two tasks.

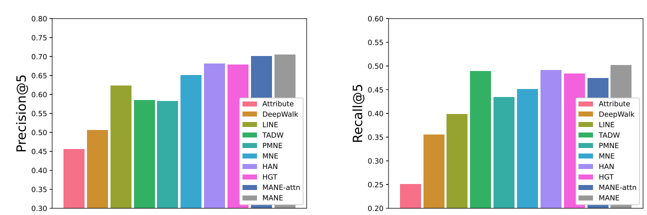


Fig. 9: Performance of link prediction.

First, we can observe that MANE constantly outperforms all baselines on each task, except that in turnover prediction task under accuracy where the probable reason is the imbalance between positive and negative samples results that most of employees would be easily classified as positive samples, which brings high accuracy and low AUC.

Second, compared with the method which only utilize basic attributes, MANE and other baselines which use additional information from organizational networks by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3140866, IEEE Transactions on Knowledge and Data Engineering

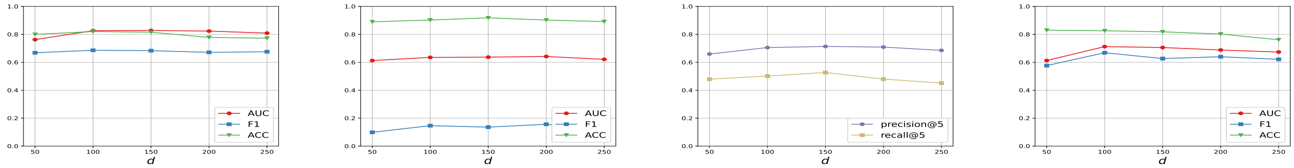


Fig. 10: Performance of MANE under different dimension of embedding d on different tasks.

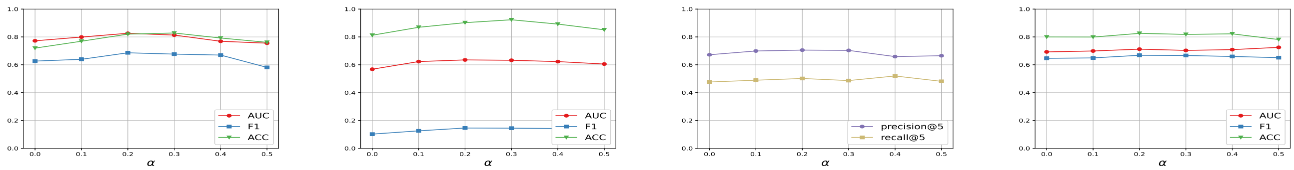


Fig. 11: Performance of MANE under different α on different tasks.

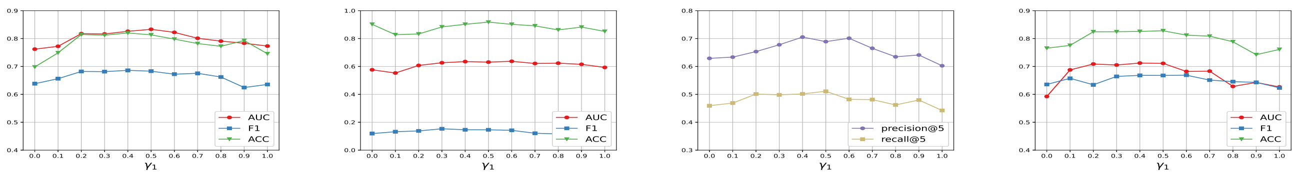


Fig. 12: Performance of MANE under different γ_1 on different tasks.

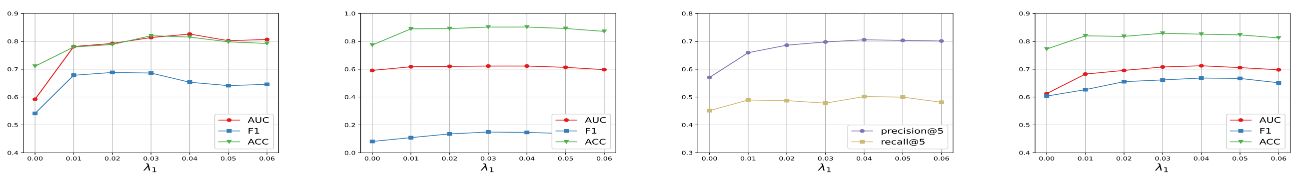


Fig. 13: Performance of MANE under different λ_1 on different tasks.

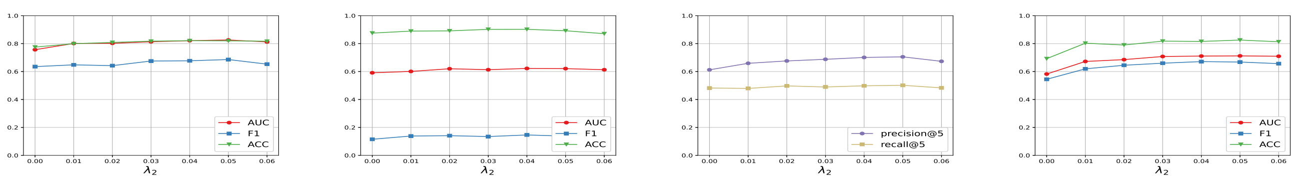


Fig. 14: Performance of MANE under different λ_2 on different tasks.

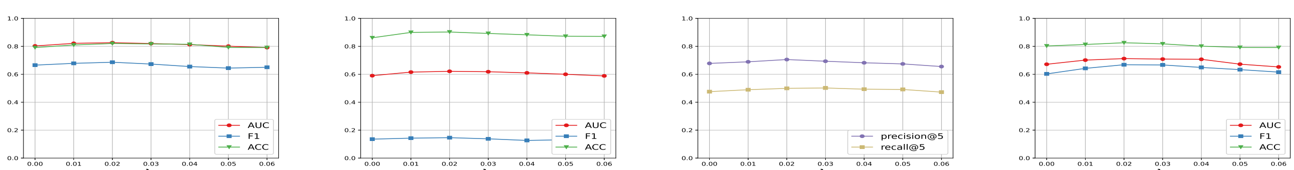


Fig. 15: Performance of MANE under different λ_3 on different tasks.

learning the representation of employees in the networks have significant improvement on performance, which indicates the effectiveness of analyzing employee's behaviors in organizational networks for real-world human resource management.

Third, the method based on multiple and heterogeneous network embedding stably keep better performance than those single network embedding based approaches, which validates our motivation that we need to integrate the information on different networks in a comprehensive way for modeling multiple organizational networks, rather than simply fusing the networks.

Besides, among single network embedding based baseline methods, TADW that leverages the node attributes in the representation learning process has a relatively better performance on each task, which demonstrates that analyzing attributes of employees plays an important role in modeling employees' behavior in organizational networks. Furthermore, the ablation study of removing attention mechanism in MANE also reveals the significance of exploring the characteristic of employees.

TABLE 2: Performance of employee modelling on business group.

	Performance Prediction			Turnover Prediction			Link Prediction	
	ACC	F1	AUC	ACC	F1	AUC	pre@5	rec@5
Attribute	0.7151	0.3657	0.6721	0.7577	0.0314	0.5536	0.3825	0.1819
DeepWalk	0.6753	0.4133	0.6811	0.7215	0.0541	0.5915	0.4611	0.3447
LINE	0.7105	0.5419	0.6997	0.7346	0.0483	0.6011	0.5734	0.3852
TADW	0.7483	0.6119	0.7155	0.7983	0.0689	0.5963	0.6122	0.4366
PMNE	0.7396	0.6183	0.7147	0.8089	0.0513	0.5787	0.6031	0.4218
MNE	0.7561	0.6328	0.7611	0.8151	0.1211	0.5919	0.6398	0.4358
HAN	0.8037	0.6631	0.7798	0.8667	0.1384	0.6338	0.6624	0.4657
HGT	0.8267	0.6544	0.7857	0.8364	0.1335	0.6411	0.6851	0.4816
MANE-A	0.8109	0.6871	0.7782	0.8397	0.1446	0.6225	0.7059	0.4997
MANE	**0.8366**	**0.7154**	**0.8155**	**0.8451**	**0.1568**	**0.6515**	**0.7115**	**0.5188**

6.4 Department Modeling

Finally, we aim to validate the effectiveness of department modeling by MANE. Similar with the employee performance prediction task, we assumed that the communication behavior within a department could reveal some implicit properties of the performance of this department. Hence we conducted an experiment on predicting the performance of each department based on the learnt department embedding in this section. Similar with employee performance prediction, in this experiment, we also used a list provided by human resource experts, containing the departments which were nominated as high performance departments in 2018, as the ground truth. To specific, among 365 departments we had 161 positive samples and 204 negative samples. The same as previous prediction tasks, we also used linear SVM as classifier since this task was also a binary classification problem. Furthermore, our experiment were conducted with five-fold cross validation where we randomly selected 80% samples as training data and the rest 20% as test data.

We report the result in Figure 8, and We have some observations based on the experimental results. First, MANE clearly outperforms all the baselines in this task. Compared with MANE-E which utilize the average of employee embedding to denote department embedding, our MANE and Graph2vec baseline achieve better performance which validate our motivation that node embedding based method cannot handle subgraph modeling based task (i.e., department modeling in this paper). Besides, the experiments conducted on the variants of our model MANE-P obtains comparatively lower value on each metrics which demonstrate the effectiveness of our relational transition method in department modeling that uses multiple transition matrices to process the different information of employees who belong to the different professions.

6.5 Link Prediction

To validate that our method can learn the representations of employees effectively, in this section, we conduct the link prediction experiments. Through this experiment, we can demonstrate that the representation of employees learnt by our model can capture the structural similarities between employees in the multiple organizational networks.

In this task, we first remove 30% edges in the multiple as the test set and use the rest of the edges to train the network embedding model. After training, we can learn the representation of each node and rank the potential nodes by calculating the cosine similarity. Specifically, as the number of nodes is very large, it's impractical to take all nodes as candidates in the experiments. Thus we adopt an approach which has been accepted by many previous works [50]. For each node, we randomly sample 100 negative linked nodes that are not connected to it and then mix all the positive linked nodes with the sampled negative nodes to select the top potential linked nodes of each node. We repeated the experiments for 10 times and report the average results in the Figure 9. From the results, we can firstly find our model outperforms all the baselines in our dataset, which proves the node embedding learnt by our MANE can measure the similarity between employees. Besides, performance of multiple network embedding methods is generally better than that of single network embedding methods since the property of employees should be learnt by co-analysis of multiple organizational networks. Furthermore, the methods that utilize the attributes in modeling employees (i.e., TADW, HAN and our MANE) can achieve better performance which validate our assumption that we need consider the external work-related semantic information to model the different impact of colleagues in employees representation learning.

6.6 Robustness Analysis

In order to analyse the robustness of our model, we evaluate our model on another dataset, which is a business group concentrating on one specific business direction (e.g., mobile application). In detail, this business group consists of 15035 employees and 111 departments. We evaluate the performance of MANE under experimental settings we reported above and show the results of employee and department modeling in Table 2 and Table 3 respectively, which reveals our MANE constantly outperforms other baselines on modeling organizational networks with different sizes.

TABLE 3: Department modelling on business group.

	ACC	F	AUC
Attribute	0.4713	0.3514	0.5345
WL-kernal	0.6128	0.4655	0.5787
sub2vec	0.7129	0.5114	0.6614
graph2vec	0.6772	0.6281	0.6114
MANE-E	0.7122	0.6115	0.6131
MANE-P	0.8205	0.6957	0.7182
MANE	**0.8556**	**0.7155**	**0.7522**

6.7 Study of Parameters

In this section, we evaluate the performance of our model under different settings of parameters. To specific, in each

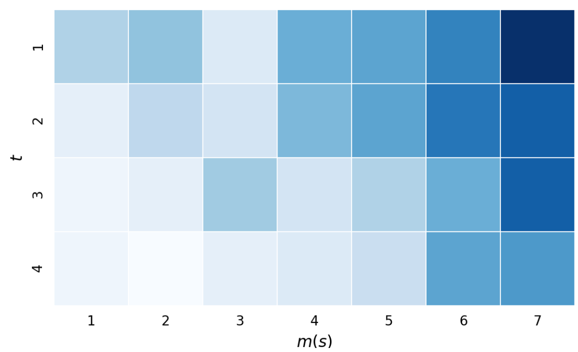


Fig. 16: The structural attention weights learnt by MANE.

experiment, we gradually change one parameter while keep other parameters invariant. In general we evaluate the impact of embedding dimension size d and hyper-parameters α, γ_1, γ_2, λ_1, λ_2, λ_3. It is worthy mentioning that in study of each parameter we change one parameter gradually while keeping others invariant and we would evaluate the impact of these parameters on all of tasks. First we show the results of all these experiments in Figures 10-15.

Impact of the dimension size d: From Figure 10, we can find that on almost every task, the performance of MANE rises with the increment of d. This is because the learnt embedding with higher dimension would contain more information of organizational networks. However it is clear that when the dimension size d increases continuously the performance will decrease, which probably results from the curse of dimensionality.

Impact of the hyper-parameter α: According to Figure 11, it is clear that MANE model has poor performance when $\alpha = 0$, which illustrates our designed attributed bipartite network A is useful to enhance the obtained relationship between similar employees. But the performance would drop rapidly when α gets larger. Based on our analysis, the probable reason is that the over utilization of attributed network would reduce randomness, thus the walk path would not achieve enough information of whole network, which is resulted from the unbalance of probability for each candidate node in random walk.

Impact of the hyper-parameter γ: Parameters γ_1 and γ_2 are utilized to control the weight of structural attention and attributed attention in attentive network embedding in MANE. In this part we only plot the performance of MANE on each task under different values of γ_1 because we could get γ_2 when γ_1 is given, as $\gamma_1 + \gamma_2 = 1$. From Figure 13, we can observe when $\gamma_1 = 0.4$ (i.e., $\gamma_2 = 0.6$) our MANE model achieves the best performance while MANE performs poorly when $\gamma_1 = 0$ and $\gamma_1 = 1$, which indicates that both these two parts of attention network in MANE are significantly useful for obtaining better representation of both employees and departments.

Impact of the hyper-parameter λ: Parameters λ_1, λ_2 and λ_3 reflect the importance of three constraint in model learning. From Figure we can observe that each component of loss function is proved to be effective, and in every evaluation the performance of our MANE increase first and then decrease where the best setting of λ is $\lambda_1 = 0.05$, $\lambda_2 = 0.04$, $\lambda = 0.02$. Besides, there is also an interesting finding that the performance of MANE on employee modeling task also get better with the increase of λ_2, which indicates the component of department embedding in MANE also has a positive impact for learning representations of employees.

The study of the structural attention weight: With the structural attention network, we can evaluate the contributes of introduced additional structural information to modeling employees in organizational networks. To this end, we visualized the average weights assigned to all the context nodes with the same relative relation $(m(s), t)$ to the central node, which were learnt by attention mechanism, in the heat map as Figure 16 where deeper color means higher value. We can clearly observe that the context nodes tend to be assigned larger weights when $m(s)$ get larger and t get smaller, which means it could form a denser subgraph and has shorter distance with the central node. This result validates our motivation that close colleagues in the same team with an employee would play more important role in modeling the behavior of this employee.

7 Conclusion

In this paper, we proposed a multiplex network embedding based quantitative approach (i.e., MANE) for modeling employee communication behavior in multiple types of working communication networks, while preserving some structural properties and attributes of the employees. Specifically, we first designed a multiple attributed random walk approach, which jointly adopts random walk algorithm on multiple networks and employee attribute network, to incorporate the information of employees on different networks. Then, an adaptive CBOW based optimization approach with an attention mechanism was proposed to preserve the network structure by maximizing the probability that utilized the surrounding context nodes to predict the central node, while preserving some structural and attribute relations of employees in the networks. Furthermore, we conducted a department modeling approach to get the embedding of departments in organizational networks by adopting a relational transformation. Finally, extensive experiments on real-world network data have clearly validated the effectiveness of the employee embeddings learnt by our model on two piratical business applications (i.e., appraisal prediction and turnover prediction) while verifying that representation of departments learnt by MANE is effective on department performance prediction as well.

Acknowledgments

This research was partially supported by the National Science Foundation (NSF) via the grant number III-2006387.

References

[1] B. Yamkovenko and S. Tavares, "To understand whether your company is inclusive, map how your employees interact," *Harvard Business Review.*, 2017.

[2] P. Leonardi and N. Contractor, "Better people analytics," *Harvard Business Review*, vol. 96, no. 6, pp. 70–81, 2018.

[3] Y. Ye, H. Zhu, T. Xu, F. Zhuang, R. Yu, and H. Xiong, "Identifying high potential talent: A neural network based dynamic social profiling approach," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 718–727.

[4] M. Teng, H. Zhu, C. Liu, C. Zhu, and H. Xiong, "Exploiting the contagious effect for employee turnover prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1166–1173.

[5] Y. Sun, F. Zhuang, H. Zhu, X. Song, Q. He, and H. Xiong, "The impact of person-organization fit on talent management: A structure-aware convolutional neural network approach," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1625–1633.

[6] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[7] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social network data analytics*. Springer, 2011, pp. 115–148.

[8] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[9] D. Du, H. Wang, T. Xu, Y. Lu, Q. Liu, and E. Chen, "Solving link-oriented tasks in signed network via an embedding approach," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 75–80.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[11] V. Batagelj and M. Zaversnik, "An o (m) algorithm for cores decomposition of networks," *arXiv preprint cs/0310049*, 2003.

[12] D. Spielman, "Spectral graph theory," *Lecture Notes, Yale University*, pp. 740–0776, 2009.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[15] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[16] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 385–394.

[17] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[18] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

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

[22] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, and L. Chen, "Principled multilayer network embedding," *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 134–141, 2017.

[23] M. Qu, J. Tang, J. Shang, X. Ren, M. Zhang, and J. Han, "An attention-based collaboration framework for multi-view network representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1767–1776.

[24] H. Zhang, L. Qiu, L. Yi, and Y. Song, "Scalable multiplex network embedding," in *IJCAI*, 2018.

[25] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[26] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," *arXiv preprint arXiv:1905.01669*, 2019.

[27] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Artificial Intelligence and Statistics*, 2009, pp. 488–495.

[28] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[29] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *The Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[30] K. Riesen and H. Bunke, "Graph classification and clustering based on vector space embedding," in *Series in Machine Perception and Artificial Intelligence*, 2010.

[31] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *ArXiv*, vol. abs/1707.05005, 2017.

[32] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *ArXiv*, vol. abs/1607.05368, 2016.

[33] B. Adhikari, Y. Zhang, N. Ramakrishnan, and B. Prakash, "Sub2vec: Feature learning for subgraphs," in *PAKDD*, 2018.

[34] T. H. Davenport, J. Harris, and J. Shapiro, "Competing on talent analytics," *Harvard business review*, vol. 88, no. 10, pp. 52–58, 2010.

[35] W. H. Mobley, S. O. Horner, and A. T. Hollingsworth, "An evaluation of precursors of hospital employee turnover." *Journal of Applied psychology*, vol. 63, no. 4, p. 408, 1978.

[36] L. Prusak and D. Cohen, "How to invest in social capital," *Harvard business review*, vol. 79, no. 6, pp. 86–97, 2001.

[37] H. Hunter, "How leaders create and use networks [internet]," *Harvard Business Review*, 2017.

[38] H. Ibarra and M. Hunter, "How leaders create and use networks," *Growth*, vol. 35, no. 1, pp. 101–103, 2007.

[39] L. Zhang, T. Xu, H. Zhu, C. Qin, Q. Meng, H. Xiong, and E. Chen, "Large-scale talent flow embedding for company competitive analysis," in *Proceedings of The Web Conference 2020*, 2020, pp. 2354–2364.

[40] S. Bian, X. Chen, W. X. Zhao, K. Zhou, Y. Hou, Y. Song, T. Zhang, and J.-R. Wen, "Learning to match jobs with resumes from sparse interaction data using multi-view co-teaching network," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 65–74.

[41] G. A. Janicik and R. P. Larrick, "Social network schemas and the learning of incomplete networks." *Journal of personality and social psychology*, vol. 88, no. 2, p. 348, 2005.

[42] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 732–740.

[43] T. E. Goldsmith and D. M. Davenport, "Assessing structural similarity of graphs." 1990.

[44] W. Ling, Y. Tsvetkov, S. Amir, R. Fermandez, C. Dyer, A. W. Black, I. Trancoso, and C.-C. Lin, "Not all contexts are created equal: Better word representations with variable attention," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1367–1372.

[45] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng, "Approximate closest community search in networks," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 276–287, 2015.

[46] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," *Proceedings of The Web Conference 2020*, 2020.

[47] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Network*, vol. 11, no. 9, p. 12, 2016.

[48] H. Wang, E. Chen, Q. Liu, T. Xu, D. Du, W. Su, and X. Zhang, "A united approach to learning sparse attributed network embedding," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 557–566.

[49] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *the Journal of machine Learning research*, vol. 9, pp. 1871–1874, 2008.

[50] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, and M. Wang, "Modeling the evolution of users' preferences and social links in social networking services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1240–1253, 2017.

Yuyang Ye received the matser's degree in computer science and technology from the University of Science and Technology of China (USTC), Hefei, China, in 2020. Before that, he received the BEng degree in computer science and technology from USTC, Hefei, China, in 2017. He is currently working toward the PhD degree in the Management Science and Information Systems Department, Rutgers University, Newark, New Jersey. His research interests include representation learning, social network analysis, and recommender systems. He was the recipient of KDD CUP 2019 PaddlePaddle Special Award.

Zheng Dong received the master degree in computer science from the University of British Columbia in 2017 and the bachelor of science degree in electronic science and technology from the University of Science and Technology of China in 2015. He is currently a software engineer at Baidu Inc. His research interests mainly focus on representation learning and graph data management.

Hengshu Zhu (SM'19) is currently a principal architect & scientist at Baidu Inc. He received the Ph.D. degree in 2014 and B.E. degree in 2009, both in Computer Science from University of Science and Technology of China (USTC), China. His general area of research is data mining and machine learning, with a focus on developing advanced data analysis techniques for innovative business applications. He has published prolifically in refereed journals and conference proceedings, including Proceedings of the IEEE (PIEEE), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Mobile Computing (TMC), ACM Transactions on Information Systems (ACM TOIS), ACM Transactions on Knowledge Discovery from Data (TKDD), ACM SIGKDD, ACM SIGIR, WWW, IJCAI, and AAAI. He is a reviewer of many leading academic journals, and has served regularly on the organization and program committees of numerous conferences. He was the recipient of the Distinguished Dissertation Award of CAS (2016), the Distinguished Dissertation Award of CAAI (2016), the Special Prize of President Scholarship for Postgraduate Students of CAS (2014), the Best Student Paper Award of KSEM-2011, WAIM-2013, CCDM-2014, and the Best Paper Nomination of ICDM-2014. He is the senior member of IEEE, ACM, and CCF.

Tong Xu (M'17) received the Ph.D. degree in University of Science and Technology of China (USTC), Hefei, China, in 2016. He is currently working as an Associate Professor of the Anhui Province Key Laboratory of Big Data Analysis and Application, USTC. He has authored 70+ journal and conference papers in the fields of social network and social media analysis, including IEEE TKDE, IEEE TMC, IEEE TMM, KDD, AAAI, ICDM, etc.

Xin Song received the master degree from Beihang University of China in 2017 and the B.E. degree from Shandong University in 2013, both in computer science. He is currently an engineer of Baidu Talent Intelligence Center (TIC). His research interests include data analysis techniques for Talent and Management Computing.

Runlong Yu received the BEng degree in computer science and technology from the University of Science and Technology of China (USTC), Hefei, China, in 2017. He is currently pursuing the PhD degree in the Anhui Province Key Laboratory of Big Data Analysis and Application (BDAA), School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. His research interests include recommender systems, data mining, and computational intelligence. He was the recipient of KDD CUP 2019 PaddlePaddle Special Award. He was also the recipient of China National Scholarship, in 2019.

Hui Xiong (F'20) is a Distinguished Professor at the Rutgers, the State University of New Jersey. Xiong's research interests include data mining, mobile computing, and their applications in business. Xiong received his PhD in Computer Science from University of Minnesota, USA. He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the 2015 IEEE International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He received the 2021 AAAI Best Paper Award and the 2011 IEEE ICDM Best Research Paper award. For his outstanding contributions to data mining and mobile computing, he was elected an AAAS Fellow and an IEEE Fellow in 2020.