

# Cross-Network Embedding for Multi-Network Alignment

Xiaokai Chu<sup>1,2</sup>, Xinxin Fan<sup>1</sup>, Di Yao<sup>1,2</sup>, Zhihua Zhu<sup>1,2</sup>, Jianhui Huang<sup>1</sup>, Jingping Bi<sup>1,\*</sup>

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China<sup>1</sup>

University of Chinese Academy of Sciences, China<sup>2</sup>

{chuxiaokai,fanxinxin,yaodi,zhuzhuhua,huangjianhui,bjp}@ict.ac.cn

## ABSTRACT

Recently, data mining through analyzing the complex structure and diverse relationships on multi-network has attracted much attention in both academia and industry. One crucial prerequisite for this kind of multi-network mining is to map the nodes across different networks, i.e., so-called network alignment. In this paper, we propose a cross-network embedding method CrossMNA for multi-network alignment problem through investigating structural information only. Unlike previous methods focusing on pair-wise learning and holding the topology consistent assumption, our proposed CrossMNA considers the multi-network scenarios which involve at least two types of networks with diverse network structures. CrossMNA leverages the cross-network information to refine two types of node embedding vectors, i.e., *inter-vector* for network alignment and *intra-vector* for other downstream network analysis tasks. Finally, we verify the effectiveness and efficiency of our proposed method using several real-world datasets. The extensive experiments show that our CrossMNA can significantly outperform the existing baseline methods on multi-network alignment task, and also achieve better performance for link prediction task with less memory usage.

## CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; • **Information systems** → *Data mining*.

## KEYWORDS

multi-network alignment, network embedding, node representation, network mining

### ACM Reference Format:

Xiaokai Chu<sup>1,2</sup>, Xinxin Fan<sup>1</sup>, Di Yao<sup>1,2</sup>, Zhihua Zhu<sup>1,2</sup>, Jianhui Huang<sup>1</sup>, Jingping Bi<sup>1,\*</sup>. 2019. Cross-Network Embedding for Multi-Network Alignment. In *Proceedings of the 2019 World Wide Web Conference (WWW '19)*, May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3308558.3313499>

## 1 INTRODUCTION

The unprecedented growth of the diverse information has produced a large volume of networks, such as social networks, citation networks, and biological networks, etc. Nowadays, people usually participate in multiple networks, on the one hand, each network can depict the topological structure of all participants corresponding to

some particular relationships; on the other hand, all the networks can be related by the same participants/nodes. In recent years, many approaches have been proposed to mine the potential information inherent in these related networks, such as cross-network recommendation [42], mutual community detection [39] and genetic diseases classification [33], etc. Although the related networks can share the same participants, they are mostly isolated in different networks without any known connections accordingly among them. Therefore, a crucial prerequisite for multi-network mining is to map the nodes/participants among these related networks, i.e., so-called network alignment. In the field of network alignment, the shared participants among the networks are defined as anchor nodes, as they act like anchors aligning the networks they participate in, and the relationships among anchor nodes across networks are called anchor links [12]. In many cases, a few anchor links can be known beforehand, for example, some users in Foursquare may leave their corresponding accounts of Twitter, but most of the correspondences are unknown. Therefore network alignment aims at inferring these unknown or potential anchor links among the networks.

In recent years, many literatures have been proposed to handle the network alignment problem [13, 16, 37, 40, 43, 44]. However, there still exist some issues that need further concern. First, most existing works only consider the two-network scenarios or perform **pair-wise learning** in multi-network applications. However, in real-world a network is usually related to multiple networks. Thus the pair-wise learning methods can ignore much valuable complementary information across the multiple networks. Second, many previous works hold the assumption of **topology consistency**, which indicates that a node tends to have a consistent connectivity structure across different networks. Although the same node may show some similar features among the networks, the differences between the network semantic meanings can lead to quite diverse local structure of this node in each network, and this assumption can be easily violated in many applications [16, 23]. Thereby these methods can make misleading alignment in the multi-network scenarios. Third, most previous works heavily **rely on attributes**, e.g., the username, gender or other profile information. However, the attribute information is usually incomplete and unreliable [16] or unavailable [1], thus these attribute-based methods are not applicable in many realistic scenarios.

Keeping these problems in our mind, we, in this paper, study the network alignment problem in a multi-network scenario, wherein the number of networks is at least two with no attribute information available, in addition that the topology structure of each network can be diverse. Taking into account the effectiveness of preserving network structures and the efficiency of calculation in

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-6674-8/19/05.

<https://doi.org/10.1145/3308558.3313499>

\*Jingping Bi (bjp@ict.ac.cn) is the corresponding author

continuous vector space, we propose a novel **cross-network** embedding method for **multi-network alignment**, namely CrossMNA. CrossMNA can integrate cross-network information to refine much more powerful embedding vectors for alignment tasks, and can also be effectively applied to other downstream multi-network analysis tasks, such as link prediction. Compared to previous embedding-based network alignment methods [10, 15, 45] or multi-network embedding methods [19, 35, 38], CrossMNA can also dramatically decline the space overhead especially in large-scale multi-network scenarios.

In CrossMNA, an additional vector named *network vector* is proposed to extract the semantic meaning of the network, which can reflect the difference of global structure among the networks. What’s more, two kinds of embedding vectors are refined for each node: (i) *inter-vector*, which reflects the common features of the anchor nodes in different networks and is shared among the known anchor nodes; and (ii) *intra-vector*, which preserves the specific structural feature for a node in its selected network and is generated through the combination of network vector and intra-vector. This combination strategy can save much more space overhead but not sacrifice the performance. We also propose a transformation matrix to align these vectors with different dimensions. We summarize our contributions as follows:

- (i) We propose a novel cross-network embedding approach CrossMNA to deal with network alignment in a more general scenario, where the number of networks is at least two with no attribute information available, in addition that the structure of each network can be quite diverse.
- (ii) CrossMNA declines the physical memory overhead for model storage to a large extent compared to other embedding based network alignment methods or multi-network embedding models, which enables our CrossMNA appropriate to tackle today’s large-scale multi-network applications.
- (iii) Extensive experiments show that our CrossMNA significantly outperforms existing network alignment methods, i.e., achieves 10% to 20% improvement with the Twitter dataset and 10% with the arXiv datasets. Furthermore, CrossMNA can achieve 5% improvement in link prediction mission with the Twitter dataset compared to existing multi-network embedding methods.

We organize the remains of our work as follows. Section 2 states the problems of multi-network alignment. We investigate several inherent vulnerabilities in previous works and propose the challenges in multi-network alignment. We formally describe our CrossMNA in Section 3, validate our approach by analyzing extensive experiments in Section 4, present related work in Section 5 and conclude our work in Section 6.

## 2 PROBLEM STATEMENT

As a crucial prerequisite for many cross-network applications, network alignment aims to establish the node correspondences across different networks. Therefore, in recent years, network alignment has become a hot spot in both academia and industry. However, there still exist some problems in currently existing works:

- (i) **Topology Consistency.** Many previous works [2, 3, 10, 13, 14, 29, 41, 45] hold the assumption of topology consistency, namely

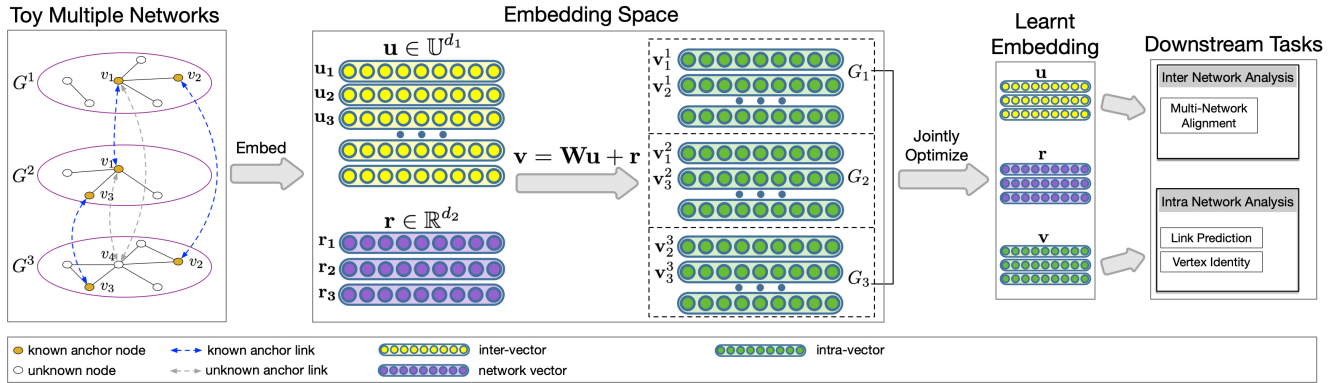
the same node has a consistent connectivity structure across different networks. For example, NetAlign [3] utilizes max-product belief propagation based on the network topology. REGAL [10] proposes an embedding-based method based on the assumption that nodes with similar structural connectivity or degrees have a high probability to be aligned. Although one node may share some similar features in related networks, its local structural connections can be entirely different in each network due to the distinctiveness in network semantic meanings, such as the types of interaction between proteins in bioinformatics. Some works have proved that topology consistency can be easily violated in many multi-network scenarios. For example, in bioinformatics, different types of genetic interactions can construct diverse network structures [23]. Another similar example can also be found in online social network sites [16], such as Twitter and Facebook, where users’s behavior may be divergent and platform dependent, making different social platforms show various connection relationships. Thereby, previous methods may lead to sub-optimal or even misleading alignments in these scenarios.

- (ii) **Pair-Wise Learning.** Most previous works only consider the two-network scenarios or perform pair-wise learning in the multi-network scenario. However, if we jointly consider all related networks, we can obtain much more useful information to benefit the node matching. Figure 1 illustrates a toy example with three related networks. If we only consider each pair of networks, it is hard to infer the node  $v_4$  in network  $G^3$  is the counterpart of the node  $v_1$  in network  $G^1$  and  $G^2$ . However, if we consider the three networks together, this alignment can be easily found.

- (iii) **Attributes Dependence.** Many previous works utilize the attribute information, like username or gender in social networks, to directly match nodes in different networks [26] or make use of the attributes to guide model learning structure information [44]. However, on the one hand, users may deliberately hide certain pieces of personal information or provide false data on their selected networks [16]. On the other hand, network data available for research is usually anonymized by the service providers for privacy concerns, where the attribute information is removed or replaced with meaningless unique identifiers [41]. Therefore, the methods based on attribute information are not general in many applications.

Considering the issues above, we intend to study the network alignment problem in a more general scenario wherein the number of networks is at least two, only topological structure available, even the networks may have different structures. In addition to the importance and novelty of this problem, we present the following challenges:

- **Semantics diversity.** The diversities in network semantics lead to the different interactional behaviors of the same node in each network, which can effect the accuracy of the node matching. Moreover, the more networks taken into consideration, the more complicated the problem is. Therefore, how to alleviate the impact of diverse network semantics on node matching is a big challenge.
- **Data imbalance.** The data imbalance has two aspects. First, the size of each network may vary considerably, including the numbers of nodes or edges in each network. Second, the



**Figure 1: An illustration of a multi-network scenario. CrossMNA learns two types of embedding vectors, i.e., network vector for each network and inter-vector for each node, which is used for node matching. The intra-vector, which can be adopted for other downstream tasks, is combined by these two types of embedding vectors**

number of anchor links between each pair of networks can be unequal. It is obvious that the previous pair-wise learning methods inevitably suffer from the data imbalance problem as they only consider each pair of networks. Thus how to make full use of the information across all the networks to deal with the data imbalance problem is another challenge.

- **Model Storage.** Network embedding is a practical approach to extract structural features of the node and has been applied in some network alignment methods [10, 15, 45]. However, in large-scale multi-network scenarios, it is essential to take into account the space overhead of the methods. Previous embedding-based methods need to generate the embedding vector for each node in each pair of networks in multi-network scenarios, and this takes too much storage space. Thereby, how to make overhead cheaper but not sacrifice the performance is what we should concern.

To address the challenges above, we propose a cross-network embedding based multi-network alignment method CrossMNA. It extracts an extra feature vector named *network vector* for each network, which reflects the difference of global structure among the networks. This means if the global structure of two networks is similar, their network vectors will be close in vector space. For each node in a network, we propose two types of embedding vectors: *inter-vector* and *intra-vector*. The former depicts the commonness of anchor nodes in different networks and is shared among the known anchor nodes. The latter reflects the specific structural feature of this node in its selected network but is generated through a combination of the network vector and inter-vector. By jointly training all the networks, CrossMNA can refine powerful inter-vector for network alignment tasks, and intra-vector for other downstream multi-network analysis tasks. What's more, the shared inter-vector and the combining intra-vector can significantly save space overhead without sacrificing the performance.

### 3 CROSSMNA: A MULTI-NETWORK ALIGNMENT APPROACH

#### 3.1 Problem Formulation

In this work, we suppose the networks are unweighted and all the edges are directed, as an undirected edge can be divided into two directed edges. For the sake of easy understanding, we follow the definitions of network alignment as reported in [41, 42].

**Definition 1 (Multiple Aligned Networks).** We define a set of networks  $\mathcal{G} = ((G^1, G^2, \dots, G^N), (\mathcal{A}^{(1,2)}, \mathcal{A}^{(1,3)}, \dots, \mathcal{A}^{(N-1,N)}))$  as multiple aligned networks, where  $G^i, i \in \{1, 2, \dots, N\}$  represents the  $i$ -th network in the set,  $N$  is the number of related networks, and  $\mathcal{A}^{(i,j)}, i, j \in \{1, 2, \dots, N\}$  denotes the set of anchor links between  $G^i$  and  $G^j$ . We define each network  $G^i$  as  $\{V^i, E^i\}$ , where  $V^i$  is the set of nodes in  $G^i$  and  $E^i$  is the set of links.

**Definition 2 (Anchor Link).** Given each two networks  $G^i$  and  $G^j$ , we define an anchor link between them as  $(v_k^i, v_k^j) \in \mathcal{A}^{(i,j)}$ , where  $v_k^i$  and  $v_k^j$  is the anchor node in networks  $G^i$  and  $G^j$  respectively. The anchor links in multiple networks following the *transitivity law* defined in [41], where if  $(v_k^i, v_k^j) \in \mathcal{A}^{(i,j)}$  and  $(v_k^j, v_k^h) \in \mathcal{A}^{(j,h)}$ , then  $(v_k^i, v_k^h) \in \mathcal{A}^{(i,h)}$ . For notational convenience, the nodes in different networks with the same subscript are the known anchor nodes.

**Definition 3 (Multi-Network Alignment Problem).** Given a set of networks and part of known anchor links among the networks, this problem is to discover the unknown or potential anchor links. It is noteworthy that the networks are partially aligned [42], which means not all the nodes have counterparts in other networks, and the anchor nodes follow the one-to-one matching constraint [41].

#### 3.2 Cross-Network Embedding

As stated before, the different network semantic meanings lead to diverse interactional behaviors of the anchor nodes, which influences the global structure of the networks. For instance, in Figure 1 the anchor node  $v_1$  shows different connection relationships in  $G^1$  and  $G^2$ . However, as the same entity, the anchor node should also display some common features across the networks. For example,

suppose that we have known that node  $v_4$  in  $G^3$  is the counterpart of node  $v_1$ , we can observe that the anchor node  $v_1$  tends to interact with nodes  $v_2$  or  $v_3$  in each network. Therefore, we propose CrossMNA in multi-network scenarios under the assumption that an anchor node in a selected network can both show some similar structural features with its counterparts and distinctive connection relationships due to its network semantic meaning. It is noteworthy that if the related networks have the same semantic meaning, the anchor nodes will display consistent connection relationships among these networks, which is the topology consistent assumption as previous works hold.

It is obvious that the common features among the anchor nodes are what we need for network alignment. To this end, we propose an *inter-vector*  $\mathbf{u}$  to preserve the common features among the anchor nodes. Through training, we hope the inter-vector of an unknown anchor node can be close to its counterparts in vector space. Nevertheless, this inter-vector is hard to be learnt directly, as there is no direct correlation between the unknown anchor nodes. For example, there is no direct relationship between the node  $v_1$  in  $G^1$  and the node  $v_4$  in  $G^3$ . Therefore, we have to learn the inter-vector indirectly.

On the other side, it is straightforward to extract structural features of nodes in a network via network embedding methods, which named *intra-vector*  $\mathbf{v}$  in CrossMNA. This type of vector contains both the commonness among counterparts and the specific local connections in its selected network due to the semantics, so it cannot be applied for node matching unless we can divestiture the impact of network semantic. Thus, we are motivated to present the following equation to build a correlation among intra-vector, inter-vector, and network semantics:

$$\mathbf{v}_i^k = \mathbf{u}_i + \mathbf{r}^k, \quad (1)$$

where  $\mathbf{v}_i^k$  is the intra-vector of node  $v_i$  in network  $G^k$  which can be easily learnt,  $\mathbf{u}_i$  is the inter-vector of node  $v_i$  and its known counterparts are shared this vector,  $\mathbf{r}^k$  is the *network vector* which can extract the unique characteristics of  $G^k$  and reflects the global difference among the networks. Thus, we can refine the inter-vector of the anchor nodes indirectly by training the combining-based intra-vectors.

We take the toy multi-network in Figure 1 as example for further explanation. Considering the anchor node  $v_1$ , it should share some common features among the networks and this commonness is what the inter-vector should represent. Meanwhile, as  $G^1$  and  $G^2$  are two different networks, the local connections of  $v_1$  can also be distinct because of the diverse network semantic meanings. Following Equation (1), we can give the intra-vector of node  $v_1$  in  $G^1$  and  $G^2$  as:

$$\begin{aligned} \mathbf{v}_1^1 &= \mathbf{u}_1 + \mathbf{r}^1, \\ \mathbf{v}_1^2 &= \mathbf{u}_1 + \mathbf{r}^2. \end{aligned} \quad (2)$$

On the one hand, through jointly training the intra-vectors, the shared  $\mathbf{u}_1$  can store the complementary information between the two networks: for example  $v_1$  can be connected to  $v_2$  and  $v_3$  at the same time. On the other hand,  $\mathbf{r}^1$  and  $\mathbf{r}^2$  can reflect the global differences between the two networks owing to the common information transmitting in  $\mathbf{u}_1$ .

Then we consider the node  $v_4$  in  $G^3$ . Its intra-vector can be written as:

$$\mathbf{v}_4^3 = \mathbf{u}_4 + \mathbf{r}^3. \quad (3)$$

By learning the structural information in  $G^3$ ,  $\mathbf{v}_4^3$  can contain some common features similar to  $\mathbf{v}_1^1$  or  $\mathbf{v}_1^2$ , for example  $v_4$  also tends to interact with  $v_2$  and  $v_3$ . Owing to to other known anchor links between  $G^3$  and the other networks, the network vector  $\mathbf{r}^3$  can preserve the specific semantic features of  $G^3$ . Therefore, by peeling off the impact of network semantic difference via  $\mathbf{u}_4 = \mathbf{v}_4^3 - \mathbf{r}^3$ , the inter-vector  $\mathbf{u}_4$  can reflect much more similar features with  $\mathbf{u}_1$ , by which we can infer that  $v_4$  is the counterpart of  $v_1$ .

The meanings of the inter-vector and intra-vector are different, they can be in different vector spaces. Therefore, we propose a transformation matrix  $\mathbf{W}$  to align them with different dimensions and rewrite the Equation (1) as:

$$\mathbf{v}_i^k = \mathbf{W}\mathbf{u}_i + \mathbf{r}^k, \mathbf{u}_i \in \mathbb{U}^{d_1}, \mathbf{v}_i^k \in \mathbb{R}^{d_2}, \quad (4)$$

where  $\mathbb{U}^{d_1}$  and  $\mathbb{R}^{d_2}$  are two different vector spaces with dimension  $d_1$  and  $d_2$  respectively.  $\mathbf{W}$ ,  $\mathbf{u}_i$  and  $\mathbf{r}^k$  are the parameters need to learned in CrossMNA.

From the above discussion, we can find that theoretically the more related networks we considered, the more accurate the node matching is. Therefore, CrossMNA needs to jointly train all the related work, and we thus propose the total objective function as:

$$\mathcal{J} = \sum_k \mathcal{J}^k, \quad (5)$$

where  $\mathcal{J}^k$  is the objective function of each  $G^k$ , which tries to preserve the structural information of each node in  $G^k$ . Following [31], for each directed edge  $(v_i^k, v_j^k)$  in  $G^k$ , we define the conditional probability of  $v_j^k$  generated by  $v_i^k$  as:

$$\begin{aligned} p(v_j^k | v_i^k) &= \frac{\exp(\mathbf{v}_i^k \cdot \mathbf{v}_j^k)}{\sum_{v_z^k \in V^k} \exp(\mathbf{v}_i^k \cdot \mathbf{v}_z^k)} \\ &= \frac{\exp((\mathbf{W}\mathbf{u}_i + \mathbf{r}^k) \cdot (\mathbf{W}\mathbf{u}_j + \mathbf{r}^k))}{\sum_{v_z^k \in V^k} \exp((\mathbf{W}\mathbf{u}_i + \mathbf{r}^k) \cdot (\mathbf{W}\mathbf{u}_z + \mathbf{r}^k))} \end{aligned}, \quad (6)$$

and the objective of each network  $G^k$  can be defined as:

$$\mathcal{J}^k = \sum_{(v_i^k, v_j^k) \in E^k} p(v_j^k | v_i^k). \quad (7)$$

By jointly optimizing, the known anchor nodes will play their roles as much as possible to transmit the structural information through the shared inter-vector  $\mathbf{u}$ , making this vector contain the commonness among the anchor nodes. At the same time, through Equation (6), the intra-vector of each anchor node can preserve the diverse structural features in different networks, which makes the network vector  $\mathbf{r}^k$  extract the specific feature of its own global structure.

For each unknown anchor nodes  $v_i^k$ , although we cannot directly make connections to its counterparts, its combining-based intra-vector can preserve some similar features as its counterparts in other networks. By peeling off the impact of the network semantic meaning  $\mathbf{r}^k$ , the inter-vector  $\mathbf{u}_i$  will display the characteristic

features as its anchor nodes do, which can be effectively used for node matching.

To speed up the training process, following word2vec [21], we perform negative sampling to approximate the objective function (6) as:

$$E_{(v_i^k, v_j^k)} = -\log\sigma(\mathbf{v}_i^k \cdot \mathbf{v}_j^k) - \sum_{v_z^k \in P(z)} \log\sigma(-\mathbf{v}_i^k \cdot \mathbf{v}_z^k), \quad (8)$$

where  $\sigma(x) = 1/(1 + \exp(-x))$  is the sigmoid function and the distribution  $P(v) \propto d_v^{3/4}$ , where  $d_v$  is the degree of node  $v$  in the given network. Therefore, we rewrite the objective function of (5) as the total loss function of the whole multi-network:

$$\mathcal{L} = \sum_{k \in [1, N]} \sum_{(v_i^k, v_j^k) \in E^k} -\log\sigma(\mathbf{v}_i^k \cdot \mathbf{v}_j^k) - \sum_z \log\sigma(-\mathbf{v}_i^k \cdot \mathbf{v}_z^k). \quad (9)$$

We adopt Adam[11] to minimize the total loss function.

As we know, in real-world, it is ubiquitous to meet the data imbalance problem among multiple networks. Previous methods can easily suffer from the data imbalance problem as they under the pair-wise learning scheme. However, our CrossMNA can make up this problem, because CrossMNA jointly trains all the networks and transmits the complementary information among the known anchor nodes through the shared inter-vectors, which can alleviate the influence of the data imbalance problem. What's more, compared to existing embedding-based methods, CrossMNA is much more lightweight. Instead of generating embedding vector for each node in each pair of networks, CrossMNA shares the common intra-vector across the networks and uses a more flexible combining-based vector generation strategy as defined in Equation (4) to generate the intra-vector for each node in each network. No matter compared to the embedding-based network alignment methods or the existing multi-network embedding methods, CrossMNA can always significantly reduce the memory overhead, and we will compare the size of CrossMNA with other models in Section 4.7.

### 3.3 Multi-Network Mining

After training, we can get two types of embedding vectors for each node: the intra-vector reflects the structural information of a node in a selected network, which is suitable for the intra-network analysis tasks, while the inter-vector interprets the commonness of the anchor nodes in different networks, which can be effectively applied to multi-network alignment task.

A naive way to find the alignments for a node is to compute all pairs of similarities between the inter-vectors, which is time-consuming. However, in practice, we usually only need to find the soft alignments for each node by returning its top- $\alpha$  most likely nodes in other networks. So following [10], we use the k-d tree data structure to accelerate similarity search [4] in the matching process. We here use cosine similarity to define the distance of two nodes in vector space.

In some applications, we can get the complete groundtruth among multiple networks beforehand, which means all the aligned links are known. As our CrossMNA is a cross-network embedding approach, it can be easily exploited in these applications to make the data mining tasks on multiple networks more actionable. During

training, the inter-vector takes the role of transmitting the structural information among the anchor nodes. Therefore, CrossMNA can preserve the complementary information across multiple networks. At the same time, because we extract the specific features of each network into its network vector, the intra-vector which is combined by the inter-vector and the specific network vector, can both keep the complementary information of its counterparts in other networks and the distinctive properties of its selected network. This can be effective in many downstream multi-network analysis tasks, such as link prediction and vertex identity.

## 4 EXPERIMENTAL EVALUATION

In this section, we perform a set of experiments to validate our proposed CrossMNA. We first compare our method with several state-of-the-art methods on the multi-network alignment task. Then, to verify the effectiveness of CrossMNA in extracting useful features of the nodes, we also compare our CrossMNA with several multi-network embedding methods on link prediction tasks.

### 4.1 Dataset Description

We employ three real-world multi-network datasets from different fields, e.g., social platform, bioinformatics, and academics. To reflect the data imbalance problem in each dataset, we present the distribution of anchor links between each pair of networks in Figure 2(a) and size of each network in Figure 2(b). We can find that the scale of each network vary greatly in all datasets and distribution of anchor links can be uneven, especially in Twitter and SacchCere. We also present the global structural similarities between each pair of networks, by comparing the common neighbors reachable in three steps of the anchor nodes. From Figure 2(c) we can find that the related networks have quite diversity structures, especially in Twitter and SacchCere. We detail the datasets as follows:

**arXiv**<sup>1</sup> [6] consists of various networks in terms of different arXiv categories. There are 14,489 nodes, 59,026 coauthorship connections, and 13 networks. The number of anchor links among the networks is 23,626.

**Twitter**<sup>1</sup> [24] This dataset is a specific Twitter dataset which focused on People's Climate March in 2014. It makes use of 3 networks, corresponding to retweet, mentions, and replies. 102,439 nodes, 353,495 edges and 55,600 anchor links are included.

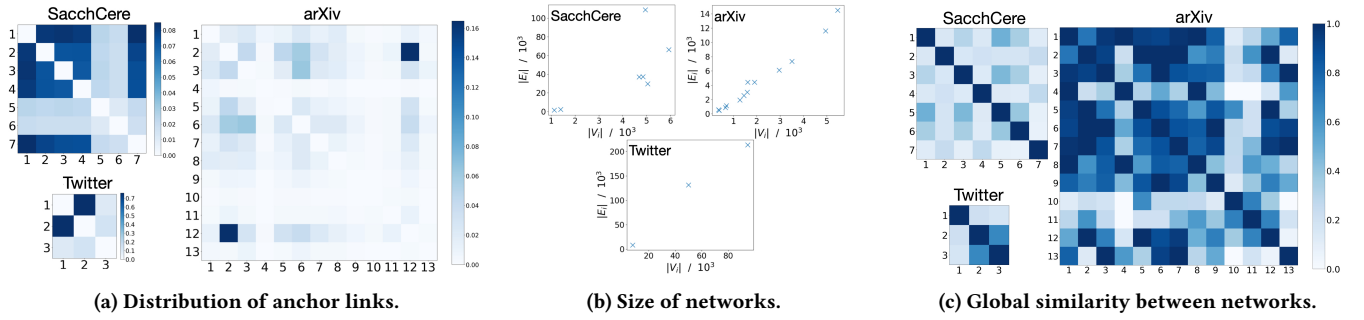
**SacchCere**<sup>1</sup> [7, 30] is a subset of BioGRID, which is a public database that archives and disseminates genetic and protein interaction data from humans and model organisms. It contains 6,570 nodes, 282,755 connections and 7 networks, each represents one type of genetic interaction. There are a total of 55,831 anchor links.

### 4.2 Baseline Methods

**Multi-Network Alignment.** To show the effectiveness of our method in addressing the multi-network alignment problem, we compare with four different baseline methods through only using network structure information.

*NetAlign* [3] proposes the message passing algorithms to match networks under unsupervised schemas.

<sup>1</sup><https://comunelab.fbk.eu/data.php>



**Figure 2: Detailed information of the datasets.** (a) Each grid represents  $|A^{(i,j)}|/|A|$ . (b) Each dot is a network, the abscissa/ordinate represents the number of nodes/edges. (c) Each grid reflects the global similarity between two networks.

*FINAL* [43] proposes a family of algorithms to align attributed networks. It formulates network alignment from an optimization perspective referring to the alignment consistency principle.

*REGAL* [10] is an unsupervised multi-network alignment method, which extracts similarity-based representations among graphs and infers the soft alignments by comparing the learnt embedding vectors.

*IONE* [15] is a state-of-the-art network embedding based method under semi-supervised schemes, which solves both the network embedding problem and the user alignment problem simultaneously with a unified optimization framework.

As *NetAlign* and *FINAL* both require the prior alignment information, following previous work we construct a degree similarity matrix and take the top- $\log|V|$  entries for each node.

**Intra-Link Prediction.** To verify the ability of *CrossMNA* on extracting effective features of the nodes among the networks. We also compare our proposed *CrossMNA* with five network embedding methods on link prediction task, which are divided into two categories in terms of single-network embedding or multi-network embedding:

*DeepWalk* [27] is a typical network embedding method that learns vertex representations based on single network structures. It performs a random walk on the network to obtain vertex sequences and conducts Skip-Gram model to train the sequences.

*LINE* [31] is another network embedding model which aims at learning node embeddings in a large-scale network. It minimizes a loss function to preserve both first-order and second-order proximity between nodes.

*node2vec* [8] defines a flexible notion of a node’s neighborhood and propose a biased random walk, which is a trade-off between Breadth First Search and Depth First Search.

*PMNE* [17] is a multi-network embedding method which proposes two simply merged approaches and a Co-analysis method to obtain one overall embedding for each node. In our experiments, we compare with its final Co-analysis method. The Co-analysis *PMNE* performs a biased random-walk across each network and conducts Skip-Gram model to train the node sequences.

*MELL* [19] is a multi-network embedding method which simultaneously learns the embedding vector of each node and a layer embedding of each single network, using all of the network structures.

### 4.3 Experiment Configuration

The parameters of the compared methods are set as follows: For *FINAL*, we follow the default setting, i.e.  $\{\alpha, t_{max}\} = \{0.3, 30\}$ . For *REGAL* we set the maximum hop distance  $K = 2$ ; For *Deepwalk*, we set walks per vertex as 20, window size as 5 and walk length as 80; For *LINE*, we employ both first-order and second-order proximity and obtain representations via concatenation, and we set the negative samples  $k = 5$ ; For *node2vec*, we empirically set  $p = 2$  and  $q = 0.5$ . For *PMNE* and *MELL*, we follow their default settings, i.e.,  $\{\alpha, p, q\} = \{0.5, 0.5, 0.5\}$ ,  $\{k, \lambda, \beta, \gamma\} = \{4, 1, 1, 1\}$ . For fair comparisons, we set the same node dimension  $d = 200$  for all embedding-based methods in network alignment, and  $d = 100$  for each network embedding model in link prediction. For our method *CrossMNA*, we thus set  $d_1 = 200$ ,  $d_2 = 100$  and the number of negative samples as 1 to speed up the training process. We use the inter-vector  $\mathbf{u}$  for network alignment task and the intra-vector  $\mathbf{v}$  for link prediction task.

For multi-network alignment task, we randomly remove part of the anchor links as the test set and the rest as the training set. We compare the precision of the soft alignment for each method. We first introduce the  $Precision^{(i,j)}@ \alpha$  as the evaluation metric of mapping nodes from  $G^i$  to  $G^j$ :

$$Precision^{(i,j)}@ \alpha = \frac{|CorrectNodes^{(i,j)}@ \alpha|}{|UnMappedAnchors^{(i,j)}|}, \quad (10)$$

where  $|UnMappedAnchors^{(i,j)}|$  is the number of unknown anchor links between  $G^i$  and  $G^j$  and  $|CorrectNodes^{(i,j)}@ \alpha|$  is the number of the correct alignments from  $G^i$  to  $G^j$  in top- $\alpha$  choices. Therefore, the evaluation metric on multi-network alignment can be defined as:

$$Precision@ \alpha = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} Precision^{(i,j)}@ \alpha, \quad (11)$$

where  $N$  is the number of networks.

For intra-link prediction task, we randomly split all edges into two sets for training and testing respectively. We also randomly sample an unconnected node pair as a negative edge for each positive edge in the test set and use both the positive and negative edges for testing. We here adopt a standard evaluation metric ROC-AUC [9] in our experiments.

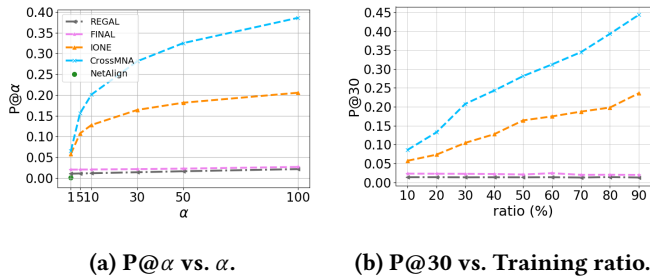


Figure 3: Performance with Twitter.

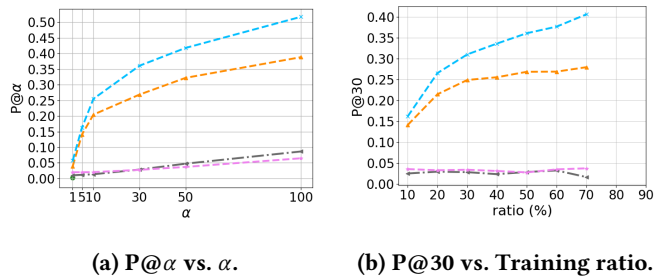


Figure 4: Performance with SacchCere.

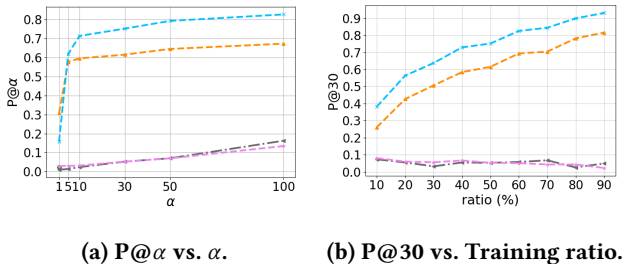


Figure 5: Performance with arXiv.

#### 4.4 Performance Evaluation

**Multi-Network Alignment.** The experimental results are presented in Figure 3-5. We first keep the training ratios to 0.5 and give the results of different  $@\alpha$  settings, which is the subfigure (a) for each dataset. Then we vary the training ratio from 0.1 to 0.9 with an interval of 0.1 and evaluate the  $Precision@30$  of each method, which is the subfigure (b) for each dataset. We remove the results of training ratio 0.8 and 0.9 on SacchCere, because most networks have few anchor nodes in the test set, leading to the unstable performance of each method. We only give  $P@1$  for NetAlign as it aims to find the hard matching for the node. From the results, we have the following observations:

(i) CrossMNA can outperform all the baselines in each dataset on almost all training ratios. We can gain 10% to 20% improvement on Twitter dataset than the state-of-the-art method IONE, and 5% to 12% improvement on SacchCere. In arXiv dataset, CrossMNA can also achieve about 10% improvement on different training ratios. Furtherly, even when the training ratio is very small, such as 10% and 20%, CrossMNA can still achieve the best performance and get 5% to 10% improvement. This result demonstrates that our method is still effective when the known anchor links are rare, which is a very common problem in many real-world applications.

(ii) The unsupervised methods NetAlign, REGAL, and FINAL show poor performance on all datasets, because the structures of each network do not follow the topology consistent assumption they hold, which leads to the misleading alignments in all datasets. The state-of-the-art embedding-based method IONE, however, can achieve good performance on each dataset. However, it only considers the pair-wise learning, thus it ignores many useful complementary information across different networks. What's more, it does not explicitly consider the problems of distinctive network

$P^{(i,j)}@30$	1→2	2→1	1→3	3→1	2→3	3→2
IONE	0.056	0.054	0.035	0.032	0.4601	0.4589
CrossMNA	0.1327	0.1106	0.2817	0.1676	0.5863	0.4210

Table 1: Results of alignment between each pair of networks in Twitter dataset. The training ratio is 50% and 1→2 denotes matching nodes from the 1st network to the 2nd network.

structures, resulting in the mismatching of some anchor nodes with different local structures. Therefore, it is hard to achieve better performance when the networks have quite different structures.

(iii) Combined with Figure 2, we can find that most networks in arXiv dataset tend to have similar global structure, and the anchor nodes are more evenly distributed across networks. Therefore, IONE can achieve good performance on this dataset, e.g., IONE can even gain better performance on  $P@1$  when the training ratio is 50% than CrossMNA. However, CrossMNA can take advantage of the jointly learning all the networks, so our CrossMNA can have a stable improvement of about 10% on different ratios. For SacchCere dataset, as the semantic meanings are different among the networks, the structure of each network is diverse, which prevents the baselines from achieving good results. However, as we jointly learn from all the networks, the abundant complementary information among the networks can improve the quality of learnt embeddings, alleviating the problem above. For Twitter dataset, the state-of-the-art method IONE cannot achieve satisfactory result due to two reasons: (i) the sizes of networks vary a lot and the distribution of anchor nodes is uneven in this dataset, e.g., the 1st network has nearly 100,000 nodes but 3rd network has only about 8,000 nodes and most anchor links are between the 1st and 2nd networks; (ii) the networks show diverse structures, e.g., the 1st network shows quite different topology structure compared to the other two networks. For better understanding, we present the performance of alignment between each pair of networks on Twitter in Table 1. We can observe that IONE can gain a comparable result with our CrossMNA in alignment between the 2nd and 3rd networks, which have the similar topology structure. However, for alignment between the 1st-2nd networks and the 1st-3rd networks, IONE cannot achieve satisfactory results because the size of the networks vary a lot and they show quite diversity global structures, which we can find in Figure 2(c). However, as our CrossMNA extracts an extra network vector to peel off the influences of network semantic on node's local connections, the inter-vector can preserve

Dataset	arXiv			SacchCere			Twitter		
	30%	50%	80%	30%	50%	80%	30%	50%	80%
DeepWalk	87.86	94.41	98.12	69.20	73.96	78.29	57.09	59.96	63.74
LINE	75.05	85.73	94.75	60.54	65.05	68.87	53.12	52.97	53.22
node2vec	88.06	94.57	97.11	71.15	76.33	80.16	56.84	61.22	65.93
PMNE	90.12	94.47	95.24	77.61	79.85	81.35	61.12	70.72	75.91
MELL	93.51	96.30	98.84	76.18	79.92	81.21	70.64	75.89	79.84
CrossMNA	<b>96.46</b>	<b>97.53</b>	<b>99.19</b>	76.88	<b>81.12</b>	<b>82.59</b>	<b>75.85</b>	<b>80.48</b>	<b>85.29</b>

Table 2: AUC score of link prediction task.

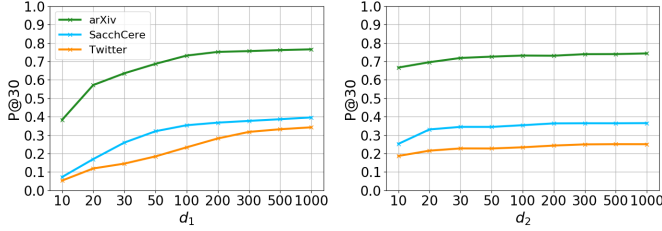


Figure 6: Parameter study on multi-network alignment.

the common features among the anchor nodes as much as possible, which shows significant improvement on this dataset.

To summarize, the above observations illustrate our method is significantly effective and appropriate to address multi-network alignment problem.

**Intra-Link Prediction.** Besides applied on network alignment, the node embedding learnt by our CrossMNA can also be adopted in other downstream network analysis tasks, e.g., link prediction and vertex identity. We take the link prediction task as an example to evaluate the quality of the learnt features. We compare with five state-of-the-art methods, three for the single-layer network and two for multiple networks. For the models designed for the single-layer network, we train a separate embedding for each network and use that to predict the links of the corresponding networks. We evaluate the AUC values of different models with the training ratio 30%, 50%, and 80% and propose the results in Table 2.

Upon the experimental results, we find that for all the datasets, our CrossMNA can achieve comparable performance or even significantly outperform all the baselines. What calls for special attention is that CrossMNA can achieve a nearly 5% improvement on Twitter dataset on all the training ratios. This is because the inter-vector in CrossMNA plays a role to transmit the complementary information across the networks while the network vectors can store the distinctive properties of each network, which make the combining-based intra-vector integrate the cross-network information without sacrificing the distinctive properties in its selected network. To summarize, the above results indicate the effectiveness of features refined by our method in multi-network mining tasks.

#### 4.5 Parameter Sensitivity

There are two parameters in our method: the dimension of inter-vector  $d_1$  and intra-vector  $d_2$ . To explore the influence of the parameters on the model performance, we set  $d_1 = d_2 = 100$  as default, pick up one parameter each time and vary that parameter while

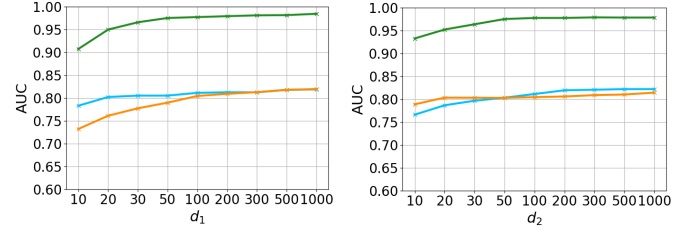


Figure 7: Parameter study on link prediction.

fixing another to check this parameter’s impact on the tasks of network alignment and link prediction. For network alignment task, we keep the training ratio to 50% and use the  $P@30$  as the metric. For link prediction task, we also keep the training ratio to 50%.

From Figures 6 and 7, we can observe that  $d_1$  has a great influence on both tasks, and as  $d_1$  grows the performance is significantly improved. While  $d_1$  is larger than a threshold, the performance will be stable, and the threshold is related to the size of the network. When  $d_1$  is larger than 200 on arXiv and SacchCere, the performance will shift towards a steady distribution. However,  $d_1 = 300$  has a significant improvement than  $d_1 = 200$  on Twitter dataset in alignment task, as the scale of this dataset is very large. We also find that  $d_2$  has a relatively smaller effect on the performance if  $d_1$  chooses a suitable number. Upon these observations, we are motivated to set a relative larger  $d_1$ , such as 200, 300 and a very small  $d_2$ , such as 30, 50 to save the memory in practice.

#### 4.6 Case Study

In this section, we perform a set of experiments to explore whether CrossMNA can gain better performance with more related networks taken into account. To make the experimental results more convincing, we choose the arXiv dataset which has the most number of networks. This means the relationships among the networks are very complicated.

We vary the number of networks from 2 to 13 and set the training ratio to 50%. We perform this experiment 10 times, and at each time we shuffle the order of the networks which are taken into consideration. We calculate the average score and present the result of  $P@30$  in Figure 8. We can observe that as the number of networks increases, the overall trend of precision also goes up. These results indicate that the complementary information among the networks benefits the result of alignment, and the more related networks considered, the better performance our CrossMNA can achieve.



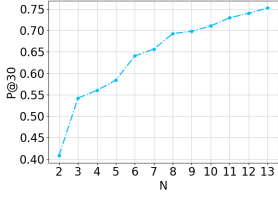


Figure 8: Precision w.r.t.  $N$ .

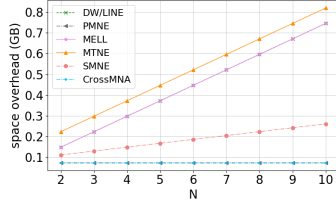
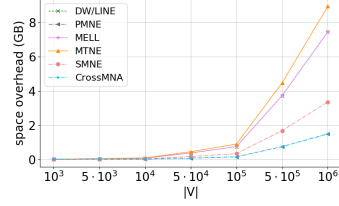


Figure 9: Memory use comparisons among multi-network embedding methods.



model	number of parameters	$p=0$	$p=0.1$	$p=0.2$
IONE	$d \cdot (N - 1)N V $		3051.75 (MB)	
REGAL	$d \cdot N V $		762.93	
CrossMNA	$d_1 \cdot  V  (N - Np + p) + d_2 \cdot N + d_1 \cdot d_2$	763.01	701.21	640.94

**Notes:**

In practice, we usually set  $d_2 \ll d_1 \approx d$ .  
 $p$  denotes the ratio of known anchor links.

Table 3: Space complexity analysis among embedding-based network alignment methods.

model	number of parameters
DeepWalk/LINE	$d \cdot N V $
PMNE	$d \cdot  V $
MELL	$d \cdot N( V  + 1)$
MTNE [35]	$d \cdot (N + 1) V $
SMNE [38]	$d \cdot  V  + s \cdot N V  + d \cdot s \cdot N$
CrossMNA	$d_1 \cdot  V  + d_2 \cdot N + d_1 \cdot d_2$

**Notes:**

$s$  is the dimension of typed relation vector in SMNE and  $s \ll d$ .

Table 4: Space complexity analysis among multi-network embedding methods.

### 4.7 Scalability

In this section, we explore the scalability of our CrossMNA. Concretely, we analyze the space complexity and time complexity of our method respectively.

**4.7.1 Space Complexity Analysis.** As we know, in real life the networks could be very huge. In large-scale multi-network analysis tasks, if we learn embeddings for all nodes in each network, the storage issue could be a big challenge. Therefore, we ought to analyze the space complexity of the embedding-based models. We first analyze the number of parameters in each model. Then we perform a set of experiments to show the actual space overhead with regard to the size of the network. For convenience, we assume that each network shares the same set of nodes. We here use a subgraph of Youtube [32] and adopt randomized permutation to generate multiple networks. We analyze the embedding-based network alignment methods and the multi-network embedding methods respectively.

From Table 3, we observe that IONE inevitably takes too much memory especially when many related networks exist. Given IONE belongs to pair-wise learning, it needs to generate embeddings for each pair of networks. Unlike IONE, REGAL saves much space as it can directly train multiple networks together owing to the randomly sampling landmarks strategy to approximate the similarities among nodes. For our CrossMNA, because of jointly training multiple networks and the combination strategy, it only generates the inter-vector for the node and the network vector for each network. It is noteworthy that the inter-vector is shared among the known anchor nodes, so the more the known anchor nodes, the less space our CrossMNA takes. For easy understanding, we give an example with five networks and 100,000 nodes and present the space overhead with different ratios of the known anchor links. We set  $d = d_1 = 200, d_2 = 50$ . We can find that even in the extreme case where

there is no known anchor link, our method can still take up nearly the same amount of space compared to REGAL. When there are a few known anchor links, our CrossMNA can take advantage of its combination strategy and show less space overhead.

Then we compare CrossMNA with existing multi-network embedding methods. From Table 4, we can find that PMNE has the least number of parameters among all the methods. Compared to other methods, the size of SMNE is relatively smaller as its hyper-parameter  $s$  can be set as a small number. However, our CrossMNA can take the smallest memory overhead in most cases, because we do not need to generate the embedding vector for each node in each network, which requires  $dN|V|$  parameters. We use the strategy of combining the common inter-vector and the network vector via a transformation matrix to generate the vector for each node in each network. Therefore CrossMNA only generates  $d_1|V| + d_2N + d_1d_2$  parameters, where  $d_1 \approx d$  and  $d_2$  can be a very small number. To show the difference more clearly, we extend the size of a multi-network in the number of networks and nodes respectively, and present the memory overhead of each model in Figure 9. We can observe that CrossMNA and PMNE are significantly lighter than other models, and as the size of the network increases, the space overhead of CrossMNA and PMNE grows very slowly. However, as PMNE merges the information from all networks into one type of embedding, it inevitably loses the distinctive information in each network as shown in our previous experiments. Thus our CrossMNA is dramatically space efficient for large-scale multi-network applications compared to existing methods.

**4.7.2 Time Complexity Analysis.** The runtime of CrossMNA is in two parts: learning cross-network embedding and matching nodes through vectors comparison. The total time complexity of learning

embeddings is approximately  $O(tN(d_1d_2|V| + d_2|E|))$ , where  $t$  is the number of iterations,  $N$  denotes the number of networks, and  $|V|, |E|$  denote the number of nodes and edges in each network respectively. The time complexity of finding soft alignment between each two networks is  $O(|V|\log|V|)$ .

## 5 RELATED WORK

### 5.1 Network Alignment

Network alignment is a fundamental problem for cross-network mining and many pieces of literature have been proposed to handle this problem. Most previous works make full use of the attribute information of nodes, e.g., username, gender, etc. The username is the most commonly used feature in almost all these works. Perito et al. [26] employ binary classifiers to determine if the cross-platform user with the same name is the same person. Vosecky et al. [34] extracts distance-based profile features and build classifiers to match users in multiple networks, and similar works can also be found in [18, 25]. In some scenarios, we can get some special contents information of the nodes. Therefore, there also exist some methods to take account of these useful contents for more precision node matching. Riederer et al. [28] links users by considering their trajectory-based content features. Almishari et al. [22] aligns users by exploiting their writing style.

The connection relationship between the nodes is another common and vital feature. Thus many works have been proposed to consider both the content information and the network structure. COSNET [44] uses an energy-based model to link users by considering both local and global consistency. HYDRA [16] presents a multi-objective framework to model heterogeneous behaviors and structure consistency simultaneously.

However, as the attribute information of the nodes is usually missing, unreliable or even unavailable in real-life applications, some methods have been proposed to align nodes only based on the structural information. BigAlign [13] proposes to align two bipartite graphs with a fast alignment algorithm. UMA [41] jointly optimizes multiple anonymized social networks in unsupervised schemes, under the constraints of transitivity law and one-to-one property. IONE [15] learns the follower-ship/followee-ship of each user under the framework of network embeddings and utilizes the embedding vectors to match unknown anchor users.

People nowadays usually participate in multiple diversities of networks simultaneously, and jointly learning from these related networks can get much more useful complementary information for alignment. UMA [41] and REGAL [10] have been proposed to optimize multiple networks together only considering structural information. However, the assumption of topology consistency makes them fail to deal with networks with distinct structures. Considering the problems above, we propose CrossMNA, a light-weight cross-network embedding based network alignment method, to jointly learn structural information across diversities of networks.

### 5.2 Network Embedding

Network embedding has been emerged as an effective and efficient approach for learning low-dimension distributed representations for the nodes in networks and is researched intensively. Previous

works can be roughly divided into two categories: single-network embedding and multi-network embedding.

Most of the previous works focus on the single-layer network. Inspired by the distributed representation learning of words in NLP [20], DeepWalk [27] performs random walk over networks to generate vertex sequences and conducts Skip-Gram to obtain node embeddings. On top of DeepWalk, Node2vec [8] modifies the random walk strategy into a biased random walk to explore network structure more efficiently. Unlike the random walk based methods above, LINE [31] optimizes two objective functions to separately approximate first-order and second-order proximity in the large-scale networks. Another general approach for obtaining node embeddings is matrix factorization. GreRap [5] proposed a matrix factorization based methods to encode k-step representations, where each step reflects different local information. TADW [36] incorporates text contents into network embedding under the framework of matrix factorization. Although these methods have achieved satisfactory performance on many single-layer network mining tasks, they ignore the multiple networks scenario.

To make the data mining task on the multiple networks more actionable, recently several works have been proposed to transform the multiple networks into low-dimension vector space. PMNE [17] proposes two simple merge-based methods which only consider inter-layer edges or intra-layer edges, and one cross-layer method which performs a biased random-walk across each layer, to obtain one overall embedding for each node. MTNE [35] and SMNE [38] builds a bridge among different layers by sharing a common embedding across each layer of the multiple networks. MELL [19] proposes the method of simultaneously learning node embeddings and layer embeddings using all of the network structures.

However, most of these multi-network embedding models ignore the problem of space overhead, which makes them hard to be applied in large-scale multi-network applications. Unlike these methods, our CrossMNA occupies much less physical without sacrificing the performance, owing to the shared inter-vectors and the flexible combination strategy.

## 6 CONCLUSION

We have studied the problem of multi-network alignment and propose CrossMNA, a light-weight cross-network embedding based network alignment method for tackling today's large-scale multi-network applications. CrossMNA novelly defines two categories of embedding vectors for each node, i.e. inter-vector and intra-vector. The commonness of anchor nodes in different networks is represented by the inter-vector, while the specific structural feature is interpreted by the intra-vector. In addition, the coordination of inter-vector and intra-vector dramatically decline the space overhead while reserving the adequate performance. Extensive experiments showed analytically our CrossMNA can significantly outperform other currently existing baselines and multi-network embedding methods.

## ACKNOWLEDGMENTS

This work has been supported by the National Natural Science Foundation of China under Grant No.: 61702470, 61472403, and Beijing Natural Science Foundation under Grant No.: 4182062.

## REFERENCES

- [1] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *International Conference on World Wide Web*. 181–190.
- [2] Mohsen Bayati, Margot Gerritsen, David F. Gleich, Amin Saberi, and Ying Wang. 2009. Algorithms for Large, Sparse Network Alignment Problems. In *Proceedings of the 9th ICDM IEEE International Conference on Data Mining, ICDM '09, 2009*. 705–710.
- [3] Mohsen Bayati, David F. Gleich, Amin Saberi, and Ying Wang. 2013. Message-Passing Algorithms for Sparse Network Alignment. *ACM Transactions on Knowledge Discovery from Data* 7, 1 (2013), 1–31.
- [4] Nitin Bhatia and Vandana. 2010. Survey of Nearest Neighbor Techniques. *CoRR* abs/1007.0085 (2010).
- [5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM '15, 2015*. 891–900.
- [6] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. 2015. Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems. *Physical Review X* 5, 1 (2015), 011027.
- [7] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. 2015. Structural Reducibility of Multilayer Networks. *Nature Communications* 6 (2015), 6864.
- [8] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, 2016*. 855–864.
- [9] J. A. Hanley and B. J. Mcneil. 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* 143, 1 (1982), 29.
- [10] Mark Heimann, Haoming Shen, Tara Safavi, and Danaï Koutra. 2018. REGAL: Representation Learning-based Graph Alignment. In *Proceedings of the 27th CIKM ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22–26, 2018*. 117–126.
- [11] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).
- [12] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. 2013. Inferring Anchor Links across Multiple Heterogeneous Social Networks. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, 2013*. 179–188.
- [13] Danaï Koutra, Hanghang Tong, and David Lubensky. 2013. BIG-ALIGN: Fast Bipartite Graph Alignment. In *Proceedings of the 13th ICDM IEEE International Conference on Data Mining, ICDM '13, 2013*. 389–398.
- [14] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. 2009. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25, 12 (2009), 253–8.
- [15] Li Liu, William K. Cheung, Xin Li, and Lejian Liao. 2016. Aligning Users across Social Networks Using Network Embedding. In *Proceedings of the 25th IJCAI International Joint Conference on Artificial Intelligence, IJCAI '16, 2016*. 1774–1780.
- [16] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. 2014. HYDRA: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling. In *Proceedings of the 41th International Conference on Management of Data, SIGMOD '14, 2014*. 51–62.
- [17] Weiyi Liu, Pin-Yu Chen, Sailung Yeung, Toyotaro Suzumura, and Lingli Chen. 2017. Principled Multilayer Network Embedding. In *Proceedings of the 17th ICDM IEEE International Conference on Data Mining Workshops, ICDM Workshops '17, 2017*. 134–141.
- [18] Anshu Malhotra, Luam C. Totti, Wagner Meira Jr., Ponnurangam Kumaraguru, and Virgilio A. F. Almeida. 2012. Studying User Footprints in Different Online Social Networks. In *Proceedings of the 3rd ASONAM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '12, 2012*. 1065–1070.
- [19] Ryuta Matsuno and Tsuyoshi Murata. 2018. MELL: Effective Embedding Method for Multiplex Networks. In *Proceedings of the 27th World Wide Web Conference on World Wide Web, WWW '18, 2018*. 1261–1268.
- [20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems, NIPS '13, 2013*. 3111–3119.
- [22] Mishari Al Mishari and Gene Tsudik. 2012. Exploring Linkability of User Reviews. In *Proceedings of the 17th ESORICS European Symposium on Research in Computer Security, ESORICS '12, 2012*. 307–324.
- [23] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the Atoms: Ranking on a Network of Networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, 2014*. 1356–1365.
- [24] Elisa Omodei, Manlio De Domenico, and Alex Arenas. 2015. Characterizing Interactions in Online Social Networks during Exceptional Events. *Frontiers in Physics* 3 (2015), 59.
- [25] Olga Peled, Michael Fire, Lior Rokach, and Yuval Elovici. 2013. Entity Matching in Online Social Networks. In *Proceedings of the 5th SocialCom International Conference on Social Computing, SocialCom '13, 2013*. 339–344.
- [26] Daniele Perito, Claude Castelluccia, Mohamed Ali Kâafar, and Pere Manils. 2011. How Unique and Traceable Are Usernames?. In *Proceedings of the 11th PETS International Symposium on Privacy Enhancing Technologies, PETS '11, 2011*. 1–17.
- [27] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, 2014*. 701–710.
- [28] Christopher J. Riederer, Yunsung Kim, Augustin Chaintreau, Nitish Korula, and Silvio Lattanzi. 2016. Linking Users Across Domains with Location Data: Theory and Validation. In *Proceedings of the 25th WWW International Conference on World Wide Web, WWW '16, 2016*. 707–719.
- [29] Rohit Singh, Jinbo Xu, and Bonnie Berger. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences of the United States of America* 105, 35 (2008), 12763–12768.
- [30] Chris Stark, Bobby-Joe Breittkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breittkreutz, and Mike Tyers. 2006. BioGRID: a General Repository for Interaction Datasets. *Nucleic Acids Research* 34, suppl\_1 (2006), 535–539.
- [31] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15, 2015*. 1067–1077.
- [32] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, 2009*. 817–826.
- [33] Marc A van Driel, Jorn Bruggeman, Gert Vriend, Han G Brunner, and Jack A M Leunissen. 2006. A text-mining analysis of the human genome. *European Journal of Human Genetics* 14, 5 (2006), 535–542.
- [34] Jan Vosecky, Dan Hong, and Vincent Y. Shen. 2010. User identification across multiple social networks. In *Proceedings of the 1st NDT International Conference on Networked Digital Technologies, NDT '10, 2010*. 360–365.
- [35] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. 2017. Multi-task Network Embedding. In *Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics, DSAA '17, 2017*. 571–580.
- [36] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network Representation Learning with Rich Text Information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI '15, 2015*. 2111–2117.
- [37] Reza Zafarani and Huan Liu. 2013. Connecting Users Across Social Media Sites: a Behavioral-modeling Approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, 2013*. 41–49.
- [38] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, 2018*. 3082–3088.
- [39] Jiawei Zhang and Philip S. Yu. 2015. Community Detection for Emerging Networks. In *Proceedings of the 15th SIAM International Conference on Data Mining, SIAM '15, 2015*. 127–135.
- [40] Jiawei Zhang and Philip S. Yu. 2015. Integrated Anchor and Social Link Predictions across Social Networks. In *Proceedings of the 24th IJCAI International Joint Conference on Artificial Intelligence, IJCAI '15, 2015*. 2125–2132.
- [41] Jiawei Zhang and Philip S. Yu. 2015. Multiple Anonymized Social Networks Alignment. In *Proceedings of 15th ICDM IEEE International Conference on Data Mining, ICDM '15, 2015*. 599–608.
- [42] Jiawei Zhang, Philip S. Yu, and Zhi-Hua Zhou. 2014. Meta-path based Multinetwork Collective Link Prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, 2014*. 1286–1295.
- [43] Si Zhang and Hanghang Tong. 2016. FINAL: Fast Attributed Network Alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, 2016*. 1345–1354.
- [44] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S. Yu. 2015. COSNET: Connecting Heterogeneous Social Networks with Local and Global Consistency. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, 2015*. 1485–1494.
- [45] Xiaoping Zhou, Xun Liang, Xiaoyong Du, and Jichao Zhao. 2018. Structure Based User Identification across Social Networks. *IEEE Trans. Knowl. Data Eng.* 30, 6 (2018), 1178–1191.