

Research Article

Nonlinear Structural Fusion for Multiplex Network

Nianwen Ning¹, Feiyu Long¹, Chunchun Wang², Youjie Zhang¹, Yilin Yang¹, and Bin Wu¹

¹Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Beijing 100876, China

²China United Network Communication Co., Ltd., No. 222, Zhongzhou Middle Road, Luoyang 471000, China

Correspondence should be addressed to Bin Wu; wubin@bupt.edu.cn

Received 20 February 2020; Accepted 11 June 2020; Published 14 July 2020

Academic Editor: Giulio Cimini

Copyright © 2020 Nianwen Ning et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many real-world complex systems have multiple types of relations between their components, and they are popularly modeled as multiplex networks with each type of relation as one layer. Since the fusion analysis of multiplex networks can provide a comprehensive insight, the structural information fusion of multiplex networks has become a crucial issue. However, most of these existing data fusion methods are inappropriate for researchers to apply to complex network analysis directly. The feature-based fusion methods ignore the sharing and complementarity of interlayer structural information. To tackle this problem, we propose a multiplex network structural fusion (MNSF) model, which can construct a network with comprehensive information. It is composed of two modules: the network feature extraction (NFE) module and the network structural fusion (NSF) module. (1) In NFE, MNSF first extracts a low-dimensional vector representation of a node from each layer. Then, we construct a node similarity network based on embedding matrices and K-D tree algorithm. (2) In NSF, we present a nonlinear enhanced iterative fusion (EIF) strategy. EIF can strengthen high-weight edges presented in one (i.e., complementary information) or more (i.e., shared information) networks and weaken low-weight edges (i.e., redundant information). The retention of low-weight edges shared by all layers depends on the tightness of connections of their K-order proximity. The usage of higher-order proximity in EIF alleviates the dependence on the quality of node embedding. Besides, the fused network can be easily exploited by traditional single-layer network analysis methods. Experiments on real-world networks demonstrate that MNSF outperforms the state-of-the-art methods in tasks link prediction and shared community detection.

1. Introduction

The abundant relation data between entities can be collected from various sources or scenarios, allowing a slew of problems to be better solved in different application domains, e.g., information retrieval, cross-media computing, science and technology management, business intelligence, biomedicine, and ecology [1, 2]. Taking together these types and sources of data may be able to give a more accurate and nuanced picture of network structure than individual network alone [3].

We believe that the joint analysis of multiple sources/types of network data can provide a more accurate and comprehensive perspective. (1) In relation extraction tasks of multisource and multimodal data [4–6], networks can be

extracted from video, text, and audio, respectively. Each network only reflects the connectivities among nodes in a single view. Therefore, data analysis results can be easily misinterpreted if we only rely on data from a single source or modal. (2) In the process of knowledge graph fusion [7], to obtain a more informative knowledge graph, we need to integrate the existing knowledge graph with other specialized knowledge graphs. (3) In social network data analysis [8, 9], lots of new online social networks have emerged and started to provide services, and the information available for the users in these emerging networks tends to be limited. The abundant information available in mature networks can be quite useful for link prediction and community detection in the emerging networks. (4) In biological multiomic data research studies [10, 11], by using the individual's expression

in each omic, researchers can construct networks of different omics and then fuse these networks into one comprehensive network to achieve more accurate prediction and analysis. In a word, the information fusion of multiplex networks in different application scenarios is a crucial issue and should be paid more attention. To model such networks, we represent such kind of networks as multiplex networks [12]. A multiplex network is one of the multilayer networks [13] in which the same set of nodes are connected by different types of relationships. Multiplex networks can not only present the intralayer links between nodes but also model the interlayer dependencies and interactions well. However, the latter is ignored by heterogeneous information network models [14]. On the left side of Figure 1, three layers of this multiplex network, respectively, are derived from three types of data, in which each layer has the same number of nodes. Nodes connected by dotted lines represent the same entity or user.

In order to fuse the multiple network data, inspired by the idea of data fusion, we divide the existing methods into four categories: (1) network structural (data-level) fusion methods [15, 16]; (2) network feature (feature-level) fusion methods [17, 18]; (3) network analysis model (strategy-level) fusion methods [19, 20]; and (4) hybrid fusion methods. In this paper, we mainly focus on the former two methods. The first category mainly utilizes a network embedding method for multiplex networks to fuse the multiple features of each node into a comprehensive feature/tensor. The second category mainly aims to preserve the sharing and complementary information of the network and fuse the structural information of multiplex networks. Despite the fact that the forceful aggregation of multiplex networks may result in loss of information [21], if the lost information is shared by some layers, it will not affect actual network information due to redundancy. If the loss information is complementary, it actually causes the loss of information.

The existing feature-level fusion based on multiplex network embedding methods cannot clearly distinguish shared and complementary information of network structure. The similarity network fusion (SNF) [22] can fuse structural information of networks. Nevertheless, SNF operates on feature matrices, which is easy to compute in the structured data but is not directly applicable to the graph domain anymore. Furthermore, the existing structural fusion methods based on SNF only consider adjacency information and fail to notice higher-order proximity information. Overall, these problems severely limit the effectiveness of conjoint analysis and mining of multiple network data with heterogeneous information. In order to solve the existing research problems, we have the following challenges. (1) How to alleviate the dramatic increase in computational complexity with the increase of network layers for filtering redundant or uninformative information of multiplex networks? (2) How to effectively fuse the information of different layers, taking advantage of the complementarity in the network data? (3) How to solve the sensitivity and dependence of the model on parameters and network embedding quality?

Aiming at these challenges, we model multiple network data as multiplex networks. We propose a multiplex network

deep structural fusion model (MNSF) to construct a single-layer network with comprehensive information. The model can not only extract structural features but also filter redundant information and fuse complementary information. We present a method to construct different higher-order proximity networks based on network embedding matrices. The network constructed by our method can provide more abundant information and improve the robustness of MNSF. We also design a nonlinear enhanced iterative fusion (EIF) strategy. To make each layer more similar to the others, EIF utilizes the higher-order proximity and message-passing theory to iteratively update the similarity matrix of each layer. We perform link prediction and shared community detection tasks on a variety of real-world multiplex network datasets. The results show that MNSF outperforms other state-of-the-art algorithms in terms of performance and efficiency. According to [23], the feature vectors of the benchmark datasets already provide much useful information, and the graph structure only provides a method for the data denoising. Therefore, MNSF achieves the goal of noise reduction of network features using the similarity network constructed based on the embedding of nodes to refine features of original networks.

The rest of the paper is organized as follows. Section 2 presents some feature-level and structure-level fusion methods. Section 3 introduces related definitions of the data model we use and problem formulations. Section 4 presents our model and core algorithm. Section 5 shows the experiment results. Finally, the summary and outlook are described in Section 6.

2. Related Work

Network fusion is the process of integrating multiple network structures and additional information to produce more comprehensive, accurate, and useful information than that provided by any individual network data. We mainly focus on two categories of network fusion methods: (1) network feature fusion and (2) network structural fusion. In this section, we introduce and summarize the related work from these two aspects.

2.1. Network Feature Fusion. The idea of network feature fusion is mainly based on network representation learning, aiming at mapping the multiple features of nodes in different layers into low-dimensional representation spaces. The goal of network feature fusion methods is to achieve the information fusion of multiple networks' features, in which these methods can be divided into coordinated representation fusion and joint representation fusion [24], as shown in Figure 2.

2.1.1. Coordinated Representation Fusion. Liu et al. [25] proposed a new multiview multigraph embedding method (M2E). The method superimposes multiple networks into multiple partial symmetric tensors and utilizes the tensor technique to simultaneously use the dependencies and correlations between multiview multigraph brain networks. Ma et al. [26] also proposed a multiview network embedding framework containing central node detection for brain

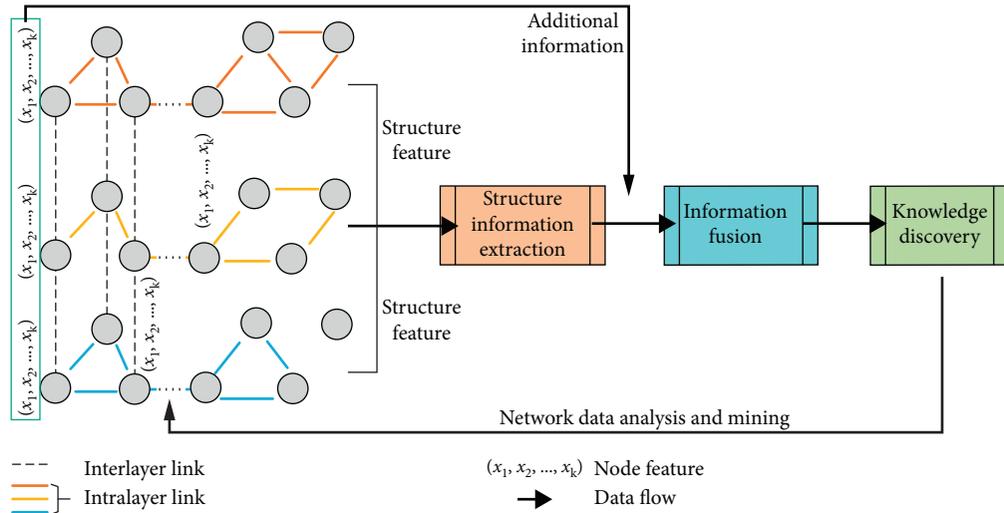


FIGURE 1: A network fusion architecture of undirected multiplex networks. On the left side of the figure, this undirected multiplex network has three layers, representing three types of relations, respectively. Each layer contains seven entities. Entities with the same name in different layers represent the same node. On the right side, three critical components of multiplex network fusion are shown.

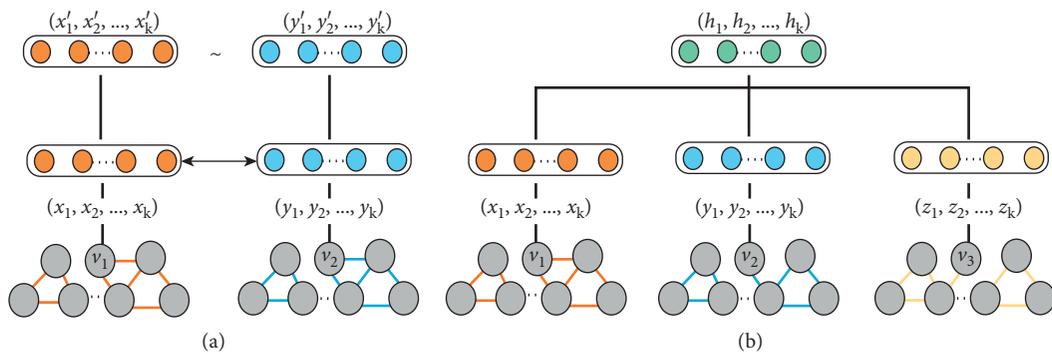


FIGURE 2: Structure of coordinated and joint representation fusion. Coordinated representation fusion achieves information interaction through the middle layers. Joint representation fusion directly fuses multiple information of nodes.

network analysis. Zhang et al. [27] proposed a scalable multiplex network embedding method, which assumes that the same node in multiple networks preserves certain common features and unique features of each layer. Thus, the common and unique embedding of nodes in each layer is learned by the DeepWalk algorithm separately. Ma et al. [28] implemented node embedding for multidimensional networks with a hierarchical structure. The method adds up node embedding in multiple dimensions as the fusion feature of nodes in multiple networks. Based on the simultaneous modeling of two properties of multiview networks identified in the real world (preservation and cooperation), Shi et al. [29] proposed a feasible multiview network embedding algorithm MVN2VEC. Matsuno et al. [30] presented a multilayer network embedding method that captures and characterizes each layer's connectivity. The method utilizes the overall structure to consider sharing or complementarity of the layer structure. The fusion feature of nodes in a multiplex network is obtained by considering the combination of node embedding in each layer with layer vectors. In order to improve the performance of the existing embedding algorithms, Al-Sayouri et al. [31] proposed a

tensor-based node embedding method, which constructed an explicit view based on a connection matrix and an implicit view through the nearest neighbors. For the multi-network embedding task, DMNE [32] is very flexible and can be embedded into low-dimensional space for different scales and weighted (unweighted) and directed (undirected) networks. DMNE embeds a single network independently and then uses a joint regularization method to achieve the fusion of multiple features. Although the methods mentioned above implement the fusion of network information by shared/common embedding methods, the final output is the embedding of nodes in different layers, rather than a fusion representation (i.e., a fusion vector/tensor).

2.1.2. Joint Representation Fusion. Ohmnet framework was used to learn features of proteins of different tissues in [15]. They represented each tissue as a network, where nodes represent proteins. Individual tissue networks act as layers in a multilayer network, where they use a hierarchy to model dependencies between the layers (i.e., tissues). Recently, Liu et al. [16] extended a standard graph mining into the area of

multilayer networks. The proposed methods (“PMNE(n),” “PMNE(r),” and “PMNE(c)”) can project a multilayer network on a continuous vector space. On the one hand, without leveraging interactions among layers, “PMNE(n)” and “PMNE(r)” apply the standard network embedding method on the merged graph or each layer to find a vector space for multilayer network. On the other hand, in order to consider the influence of interactions among layers, “PMNE(c)” expands arbitrary single-layer network embedding method to a multilayer network.

2.2. Network Structural Fusion. The main idea of network structural fusion is to follow the principle of sharing and complementarity for carrying out the structural fusion of the network. Hristova et al. [33] studied the geo-social properties of multiplex links for spanning more than one social networks. They applied structural and features of interaction to the problem of link prediction across social networking services. Deng et al. [34] analyzed that social network is multisource and heterogeneous. Each network represents a specific relationship, and each node has different roles in different relationships. Therefore, they proposed an optimal linear combination method for learning multiple relationships and extracted the connectivity of multisource social networks. Michele et al. [35] proposed a method which can preserve the information of the network in each dimension as much as possible and mapped the network into a single-layer network. Tang et al. [36] proposed a multidimension network fusion method based on structural features and conducted community detection task in multisource networks. They also analyzed four possible network integration strategies: network integration, benefit integration, feature integration, and partition integration.

Wang et al. [22] proposed an important similarity network fusion method. Based on the information propagation theory, the method iteratively updates each network so that multiple networks are obtained as similar as possible. The resulting network contains structural information of multiple networks. Manlio et al. [37] presented a dimension reduction method for reducing the number of layers in multilayer networks based on quantum theory. This method can maximize the resolution of original and aggregated networks. It quantifies the information loss as a consequence of dimensionality reduction and calculates a threshold for the fusion. Xu et al. [38] proposed a weighted similarity network fusion method for the identification of cancer subtypes. Ma et al. [18] extended an upgraded version of ANF for similarity network fusion. ANF can reduce the computational complexity of similarity network fusion, and the structural information of multiplex networks is well preserved. Cowen et al. [39] reviewed the fusion analysis method of the propagation characteristics of biological data into a network. Ruan et al. [17] proposed an enhanced similarity network fusion method (ab-SNF) for associated signal annotation. The ab-SNF method adds feature-level correlation signal annotation as a weight when constructing a topic similarity network, aiming to increase signal features and reduce noise features for improving the performance of disease subtypes. Pai et al. [40] reviewed some of the latest approaches for patient similarity networks

and looked forward to the widespread usage of network fusion-based approaches in medical and genomic data. Subsequently, they proposed a novel supervised classification framework (netDx) [41] for patient classification problems. The framework has high accuracy and scalability, which is able to integrate multiple types of data and handle sparse data well. Most of these methods are for multisource Euclidean data fusion, but it is difficult to expand to non-Euclidean graph data fusion. Therefore, the study of this paper fills the gap. Ghavasieh et al. [42] introduced a framework for functional reducibility which allows enhancing transport phenomena in multilayer systems by coupling layers together with respect to dynamics rather than structure.

3. Data Model and Problem Formulation

In this section, we describe related symbols, concepts, and definitions in detail. We first describe a multiplex network to show the basic concepts. Next, we give the complementary structural information concepts of multiplex networks and define the problem of network feature fusion based on network embedding of multiplex networks. Finally, we formalize a generalized structural fusion problem of multiplex networks.

3.1. Data Model. In terms of network data of multiple types and sources, it is more appropriate to represent such kind of networks as multiplex networks. As shown in Figure 3, three layers of this multiplex network are derived from three modal data, such as co-author network, semantic relation network, and social network. Multiplex networks can not only express the intralayer link but also model the dependencies and interactions between networks well. The detailed definitions of multiplex networks are as follows.

Definition 1 (multiplex networks). Consider L -layer multiplex networks of N nodes, in which each node can interact with the other ones through L kinds of relationships. An aligned multiplex network $G = \{G^\alpha(V, E^\alpha), \alpha \in [1, L]\}$ is made up of L layers with $N = |V|$ nodes and $M = |\sum_{\alpha=1}^L E^\alpha|$ edges.

Let v_i^α denote node v_i at layer $\alpha \in [1, L]$ and $e_{i,j}^\alpha$ denote the edge to link node v_i^α and node v_j^α in layer α . $e_{i,j}^{\alpha,\beta}$ denotes an anchor link. v_i^α and v_i^β are the duplicates of same node v_i in different layers. We assume that nodes v_i and v_j can be linked by $e_{i,j}^{\alpha,\beta}$ and $e_{i,j}^\beta$ cross layers α and β . Take Figure 3 as an example; the node set $V = \{\text{Bob, Brian, Danny, Adam, Losa, Carlo, Chris}\}$, co-author network, semantic relation network, and social network can be abbreviated as $\{\alpha_1, \alpha_2, \alpha_3\}$. $G = \{G^{\alpha_1}(V, E), G^{\alpha_2}(V, E), G^{\alpha_3}(V, E)\}$. $e_{\text{Bob, Brian}}^{\alpha_1}$ denotes Bob ^{α_1} and Brian ^{α_1} is connected in layer α_1 . $e_{\text{Danny, Adam}}^{\alpha_2, \alpha_3}$ denotes Danny ^{α_2} and Adam ^{α_3} is connected by duplicates Danny ^{α_2} cross layers α_2 and α_3 .

3.2. Problem Formulation

Definition 2 (shared and complementarity structural information of multiplex networks). Given a two-layer

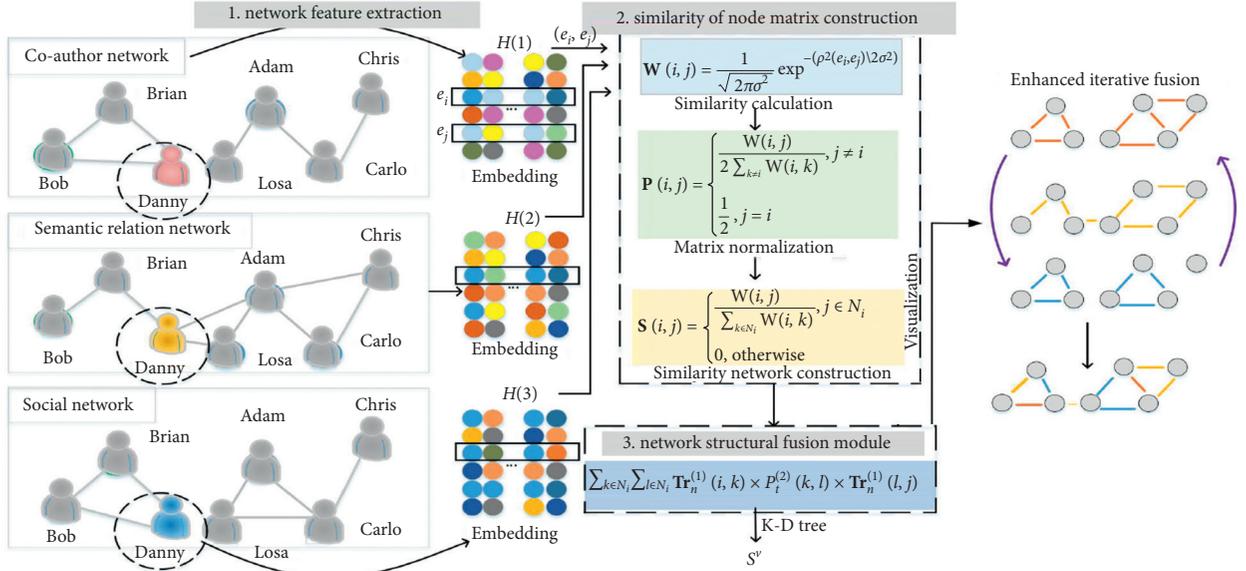


FIGURE 3: The architecture of MNSF for three networks as an example. Network representation learning is a node embedding process. The embedded matrices contain vectors that represent the latent representations of all nodes. Enhanced iterative fusion is a process of multiplex network fusion.

multiplex network $G = \{G^1, G^2\}$, where the structural information of G^1 layer can be denoted as I^1 and the structural information of G^2 layer can be denoted as I^2 . Then, $I_s = I^1 \cap I^2$ denotes shared information, also known as consistency information. $I_c = (I^1 - I^1 \cap I^2) \cup (I^2 - I^1 \cap I^2) = I^1 + I^2 - 2(I^1 \cap I^2)$ denotes complementary information, also known as unique information.

It is worth noting that shared information does not refer only to the edges shared by multiple networks. On the

mesoscales and macroscales, a group of nodes always belongs to the same community in different layers, while the edges among these nodes are more likely to be different in each layer. Inspired by the idea [33], we assume the similarity (dependence) of different layers is a quantification of consistency information between layers. Furthermore, as shown in Figure 4, we visualize the distribution of shared and complementary information among all layers in the datasets of this paper according to the following formula:

$$\text{sim}(s, t) = \frac{\sum_{v_i}^V |\text{ego}(s, v_i) \cap \text{ego}(t, v_i)| / \min(|\text{ego}(s, v_i)|, |\text{ego}(t, v_i)|)}{N}, \quad (1)$$

where s and t denote source and target layers, respectively. V is a set of nodes of a multiplex network. $\text{ego}(s, v_i)$ denotes the set of nodes of ego network with respect to node v_i in global network s . The detailed information of these datasets is presented in Section 5. It can be seen from Figure 4 that the similarity between different layers is obviously different. The lighter the color of a block, the greater the similarity of a local structure between the corresponding pair of layers and the more the information they share. On the contrary, the darker the color, the more the complementary information between the corresponding pair of layers.

Definition 3 (network feature fusion; multiplex network embedding). Suppose the methods make use of a real-valued superadjacency matrix A , $A \in \mathbb{R}^{N \times L, N \times L}$ (e.g., representing text or metadata associated with nodes). Node embedding aims at learning a map function $f: A_{v_i} \rightarrow H_{v_i} \in \mathbb{R}^{N, d}$, $d \ll N$.

f is a function, which maps $A_{v_i} = \{A_{v_i}^1, A_{v_i}^2, \dots, A_{v_i}^L\} \in \mathbb{R}^{N, L}$ to a d -dimensional representation of node v_i , and A_{v_i} is a group of vectors of node v_i in superadjacency matrix of G , and it can also be understood that it is composed of adjacency matrices of multiple layers [43]. H_{v_i} is the d -dimensional vector and $d \ll N < N \times L$. Notice that all definitions above can be easily extended to the case of weighted networks. We embody this definition and related symbols in Figure 5.

Definition 4 (network structural fusion). A multiplex network $G = \{G^1, \dots, G^L\}$, where each layer should have the same set of nodes V and the number of nodes in the multiplex networks is $N = |V|$. Network structural fusion aims to find a joint structural representation that can describe the same node in a multiplex network based on the consistency and complementary information between multiple layers. This joint representation can reflect the properties of each node in different layers. We define the

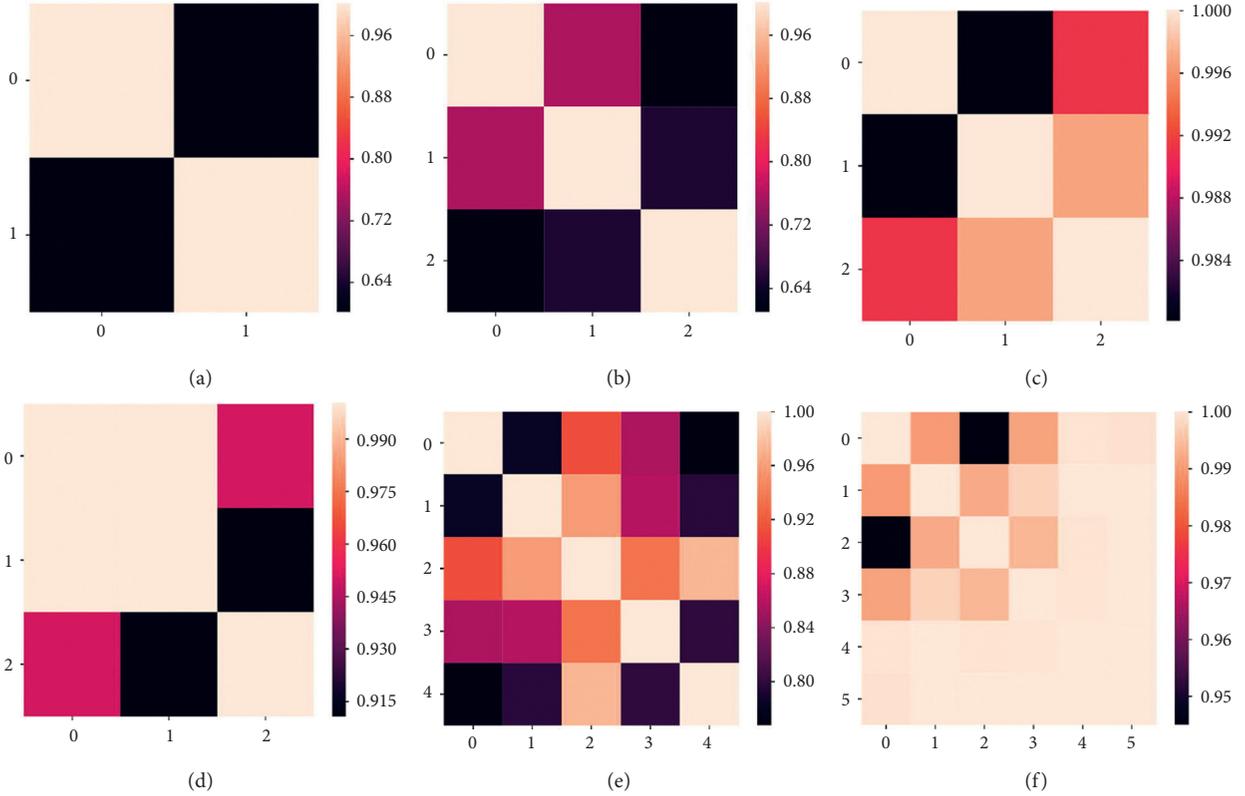


FIGURE 4: The interlayer similarity illustration of multiplex networks in 6 real-world datasets. The distribution visualization of shared and complementary information among all layers in different multiplex network datasets contains Ack-Co-author multiplex network, CKM physicians innovation multiplex network, London multiplex transport network, VICKERS classroom social multiplex networks, CS-Aarhus social multiplex networks, and Celegans multiplex gpi network. (a) Ack-Co-author. (b) CKM. (c) London. (d) VICKERS. (e) CS-Aarhus. (f) Celegans.

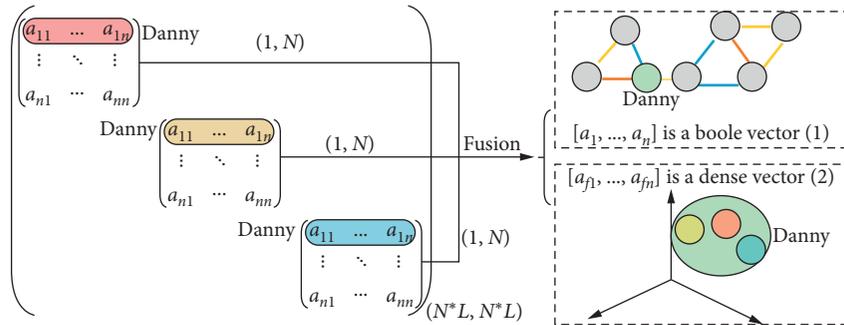


FIGURE 5: Superadjacency matrix, network feature-level fusion, and network structural-level fusion illustration. Boole vector denotes the result of structural-level fusion and dense vector denotes the result of feature-level fusion.

fusion function as $F: G \rightarrow g$, where G represents a multiplex network and g represents a single-layer network after fusion. The number of nodes in the network g is N .

In this study, the structural information of networks can be divided into three types: (1) macroscale information, such as small-world property and power-law distribution; (2) mesoscale information, such as community and motif; and (3) microscale information, such as node adjacency information, node degree, and the centrality of nodes. Therefore, some network properties can be captured by network fusion methods, such as the rich club property, the calculation of

edge betweenness, and k -core index, the synchronization analysis of the complex network. However, these properties cannot be captured by network feature fusion methods. The intuitive understanding of this definition and related mathematical symbols are shown in Figure 5.

In summary, the fusion of multiplex networks can provide more abundant information than a single-layer network. Yet, this abundant information is reflected by the complementary information. It is worth noting that the fusion of multiplex networks needs to not only consider the edges of networks (microscale) but also deeply consider the

community (mesoscale) of the network and the degree distribution of nodes (macroscale). For preserving the node's adjacency information, we simply integrate these edges existing in a multiplex network and name it as MerNet. It does not guarantee the preservation of mesoscale and macroscale information [13]. It is also verified in our subsequent experiments. In addition, as shown in Figure 5, one form of network structure fusion result is boole vectors with N dimension. One form of network structure fusion result is a dense vector with d dimension. Therefore, there is an essential difference between structural fusion and feature fusion of network.

4. Multiplex Network Structural Fusion

In this section, we introduce a deep structural fusion model for multiplex networks (MNSF). MNSF incorporates network feature extraction (NFE) and network structural fusion (NSF). Figure 3 illustrates the steps of MNSF by three networks as an example. The gray section shows the name of the module. The right side of the figure visualizes the process of enhanced iterative fusion.

The first module is the network feature extraction module, which performs representation learning for each layer. In the process of acquiring network embedding, the structural information of the original network is comprehensively exploited. The second module is the construction of node similarity matrices module, which takes the result of the network representation learning module as input. For each layer of a multiplex network, the similarities between nodes and its neighbors are calculated based on K-D tree algorithm. The third module is the network structural fusion module. In the process of algorithm iteration, the similarities between nodes and their neighbors are used to update the global similarity between nodes. The similarity information can be propagated among layers. After a few iterations, until the algorithm converges, the similarity matrix of each layer network is averaged at the layer as the final output to construct the final network. We mainly focus on the two modules: the network feature extraction and enhanced iterative fusion.

4.1. Network Feature Extraction. In network feature extraction module, we conduct a biased random walk according to [44] to generate the sequence of nodes. Then, we perform Skip-gram over the sequences to learn the node embedding with a given dimension d . For a node v_j , it appears in position j , and we define $C(v_j) = \{v_{j-c}, \dots, v_{j+c}\}$ as the context of v_j , where $2c$ is the window size. We optimize the following objective function, which minimizes the log-probability of neighborhoods conditioned on its feature representation.

$$-\log P(v_{j-c}, \dots, v_{j-1}, v_{j+1}, \dots, v_{j+c} | v_j). \quad (2)$$

Hence, we need to minimize the following equation.

$$-\log \prod_{v_u \in C(v_j)} P(v_u | v_j) = - \sum_{v_u \in C(v_j)} \log P(v_u | v_j). \quad (3)$$

For each $P(v_u | v_j)$, a softmax function is used to define the probability:

$$P(v_u | v_j) = \frac{\exp(X_{v_u}^T X_{v_j})}{\sum_{v_s \in V} \exp(X_{v_s}^T X_{v_j})}, \quad (4)$$

where X_{v_j} denotes the word embedding of node v_j and X_{v_u} and X_{v_s} denote the local context vectors and global node vectors. To solve the computation problem, we adopt the negative sampling approach proposed in [45], which samples multiple negative edges according to some noisy distribution for each edge. We replace each $\log P(v_u | v_j)$ with

$$O = -\log \sigma(X_{v_u}^T X_{v_j}) - \sum_{v_n \in C_n} \sigma(-X_{v_n}^T X_{v_j}), \quad (5)$$

where $\sigma = 1/1 + \exp(-x)$ is the sigmoid function and C_n is the negative sample node set. We employ stochastic gradient descent (SGD) to optimize objective function equation (5).

4.2. Enhanced Iterative Fusion

4.2.1. Network Structural Fusion. The network structural fusion module takes the result of the network representation learning as input. A similarity network $g = (V, E)$ has $N = |V|$ nodes. The distance between two nodes can be measured by embeddings of two nodes. Using the square of the Euclidean distance as a measure of similarity, $\rho(v_i, v_j)$ represents the distance between nodes v_i and v_j . The similarity matrix \mathbf{W} is constructed according to the embedding of nodes, where $\mathbf{W}(v_i, v_j)$ represents the similarity between the nodes v_i and v_j , and the calculation formula is

$$\mathbf{W}(v_i, v_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\rho^2((v_i, v_j)/2\sigma^2)}, \quad (6)$$

$$\sigma = \mu \frac{\bar{\rho}(v_i, N_i) + \bar{\rho}(v_j, N_j) + \rho(v_i, v_j)}{3},$$

where $\bar{\rho}(\cdot, N_i)$ is the average distance between the node v_i and the set of its neighbors N_i . In this step, to increase the computational efficiency of the nearest neighbors, we use K-D tree algorithm with time complexity $O(\log N)$ to calculate $\rho(\cdot, \cdot)$. In order to eliminate the influence of node autocorrelation on the similarity matrix (that is, the elements on the main diagonal of the similarity matrix have larger values), the similarity matrix \mathbf{W} needs to be normalized:

$$\mathbf{P}(v_i, v_j) = \begin{cases} \frac{\mathbf{W}(v_i, v_j)}{2 \sum_{v_k \neq v_i} \mathbf{W}(v_i, v_k)}, & v_j \neq v_i, \\ \frac{1}{2}, & v_j = v_i. \end{cases} \quad (7)$$

Let N_i denote the neighbor of node v_i . For the given network G , we use K nearest neighbors to measure local affinity of node v_i :

$$\mathbf{S}(v_i, v_j) = \begin{cases} \frac{\mathbf{W}(v_i, v_j)}{\sum_{v_k \in N_i} \mathbf{W}(v_i, v_k)}, & v_j \in N_i, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

It is generally believed that the affinity of a node and its neighbors is higher than that of the node and other further nodes, so this equation constructs a local affinity matrix \mathbf{S} . The matrix \mathbf{P} contains the global similarities between the node and all other nodes in the fusion network, while matrix \mathbf{S} contains only the similarities between a node and its K -order nearest neighbors. Furthermore, the correlation between \mathbf{P} and \mathbf{S} is nonlinear. Based on the information propagation theory, the local affinity matrix of a layer of multiplex networks is exchanged with the global similarity matrix of another layer of multiplex networks, and the structural information fusion of the multiplex networks can be completed by iterative updation.

$$\mathbf{P}^{(l)} = \mathbf{S}^{(l)} \times \left(\frac{\sum_{k \neq l} \mathbf{P}^{(k)}}{L-1} \right) \times (\mathbf{S}^{(l)})^T, \quad l = 1, \dots, L, \quad (9)$$

where l denotes current layer and L denotes the layer index of multiplex networks. In this module, due to the construction of the affinity network, this module reconstructs the direct and indirect relations between nodes. Through the selection of nearest neighbors, K nodes with the strongest correlation are used as direct neighbors to achieve information purification. For example, if certain 2-order neighbors v_j of node v_i have higher similarity than some direct neighbors, v_j can be served as a direct neighbor of v_i with high probability in affinity network \mathbf{S} . In iteration fusion process of computing \mathbf{P} , we consider the weight of interlayer and intralayer edges at the same time. Take edge $e(v_i, v_j)$ in affinity network $\mathbf{S}^{(l)}$ as an example. By interaction of $e(v_i, v_j)$ and $\mathbf{P}^{(k)}(v_i, v_j)$ in other network k , we can determine complementarity and consistency of $e(v_i, v_j)$ in the whole multiplex network. To summarize, the result of this module is that (1) the disappearance of weak similarities (low-weight edges) helps to reduce the noise; (2) strong similarities (high-weight edges) in one or more networks are added to the others; and (3) low-weight edges supported by all networks are retained depending on how tightly connected their neighborhoods are across networks, which brings about reducing redundant information.

4.2.2. Higher-Order Proximity. In equation (9), the result of iteration may lose higher-order proximity information and slow convergence rate if and only if \mathbf{S} matrix with first-order proximity information is considered in the iteration update process. Moreover, higher-order proximity can avoid the model's strong dependence on the accuracy of similarity calculation based on embedding vectors and alleviate the loss of structural information [46]. On the basis of introducing higher-order proximity, we propose a nonlinear enhanced iterative fusion strategy to improve equation (9). Taking a paired network as an example, the equation for iterating using higher-order proximity is

$$P_{t+1}^{(1)}(v_i, v_j) = \sum_{v_u \in N_i} \sum_{v_l \in N_j} Tr_n^{(1)}(v_i, v_u) \times P_t^{(2)}(v_u, v_l) \times Tr_n^{(1)}(v_l, v_j), \quad (10)$$

where $Tr_n^{(1)}$ denotes the measure of n -order proximity in layer 1 and t denotes the current time of iteration. The final formula of enhanced iterative fusion is

$$P^{(l)} = \alpha_0 \times S^{(l)} \times \left[\frac{\sum_{k \neq l} P^{(k)}}{L-1} \right] + \alpha_1 \times S^{(l)} \times \left[\frac{\sum_{k \neq l} P^{(k)}}{L-1} \right] \times S^{(l)T} \\ + \alpha_2 \times Tr_2^{(l)} \times \left[\frac{\sum_{k \neq l} P^{(k)}}{L-1} \right] \times Tr_2^{(l)T} + \dots + \alpha_n \\ \times Tr_n^{(l)} \times \left[\frac{\sum_{k \neq l} P^{(k)}}{L-1} \right] \times Tr_n^{(l)T}, \quad (11)$$

where $\alpha \in [0, 1]$ is a parameter to control the importance of proximity and $\alpha_1 + \alpha_2 + \dots + \alpha_k = 1$, $l, k = 1, 2, \dots, n$. The global similarity matrix \mathbf{P} of each layer of the network is averaged to obtain the similarity matrix of the fused network as the final output of the algorithm. It should be noted that the studies in [47] have empirically demonstrated that network features extracted based on three-hop neighborhoods contain the most useful information, so the choice of K is not the bigger the better in Tr_n . Therefore, in this paper, we use the 1st, 2nd, and 3rd order proximity to implement the enhanced iterative fusion module, and we have verified through many experiments that important parameters of these similarities are set to $\alpha_1 = 0.3$, $\alpha_2 = 0.4$, and $\alpha_3 = 0.3$ to obtain trade-off between precision and efficiency. We use the nearest neighbors to measure local affinity. Then, the corresponding localized network can be constructed from the original weighted network using the following equation:

$$Tr_2(v_i, v_j) = \sum_{v_u \in N_i \cap N_j} \frac{S_{v_i, v_u} S_{v_j, v_u}}{\sum_{v_l \in N_k} S_{v_l, v_u}}. \quad (12)$$

Equation (12) can be understood as constructing a new network only based on second-order proximity. As shown in Figure 6, to avoid repeatedly calculating low-order proximity problems when constructing a high-order affinity network, we need to remove some edges existing in low-order networks from high-order proximity networks. In fact, equation (12) is a second-order instance of $Tr_n^{(1)}$ in equation (10).

$$P^{(v)} = \alpha_0 \times S^{(v)} \times \left[\frac{\sum_{k \neq v} P^{(k)}}{m-1} \right] + \alpha_1 S^{(v)} \times \left[\frac{\sum_{k \neq v} P^{(k)}}{m-1} \right] \times S^{(v)T} \\ + \alpha_2 Tr_2 \times \left[\frac{\sum_{k \neq v} P^{(k)}}{m-1} \right] \times Tr_2^T. \quad (13)$$

According to equation (13), through the network structure \mathbf{S} of a layer in the iterative process, the average value of the similarity matrix \mathbf{P} of the remaining other

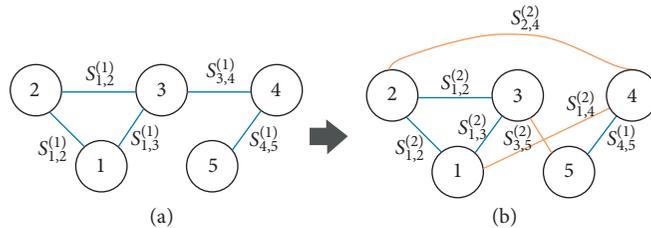


FIGURE 6: Construction of first- and second-order weight proximity networks. (a) First-order proximity network (original network). (b) Second-order proximity network. The latter is constructed on the basis of the former network. In (b), blue edges denote edges common to first-order networks, and yellow edges denote edges unique to second-order networks.

network is iteratively updated, which can gradually converge to each other after updating each layer of the multiplex networks. This similarity network contains its shared information and complementary information and finally achieves the fusion of multiplex networks. Taking a node pair (1, 2) as an example, according to equation (13), we elaborate a computation process of the proximity between nodes 1 and 2 in Figure 7. α controls the contribution of different proximities to the result. Equation (13) is an implementation of equation (11) based on 1st, 2nd, and 3rd order proximity.

This module can strengthen high-weight edges presented in one (i.e., complementary information) or more (i.e., complementary information) networks and weaken low-weight edges (i.e., complementary information). The retention of low-weight edges shared by all layers depends on the tightness of connections of their K -order proximity. The pseudocode of MNSF is shown in Algorithm 1. In this pseudocode, lines 2–4 correspond to the network feature extraction module. Lines 5–11 implement the similarity of node matrices construction. Lines 12–15 indicate the network structural fusion module, in which line 13 achieves the calculation of high-order proximity and enhanced iterative fusion is implemented in line 14. We can obtain a matrix of the average global similarity network in line 16. The resulting network graph is calculated by K -D tree algorithm. Using this algorithm, we construct neighbors of each node.

5. Experiment Analysis

In this section, we study the performance of MNSF with high-order proximity iteration in different real-world datasets. We use link prediction and shared community detection tasks to verify the performance of MNSF.

5.1. Datasets. For our experiments, we run MNSF and compare baseline methods on each of the following multiplex networks. These datasets contain two categories: Public datasets and Private dataset. Public datasets are composed of five benchmark multiplex network datasets involving social, biological, genetic, and transportation. The specific information on these public datasets is shown in Table 1. Private dataset is an interesting semantic network dataset that we construct. This dataset is a network of acknowledgment relationships extracted from the

acknowledgment part of dissertation data and the collaboration network of corresponding entities from AMiner.

5.1.1. Public Datasets. VICKERS classroom social multiplex networks: this dataset was collected by Vickers from 29 seventh grade students in a school in Victoria, Australia. Students were asked to nominate their classmates on a number of relations.

CS-Aarhus social multiplex networks [48]: this dataset consists of five kinds of online and offline relationships (Facebook, Leisure, Work, Co-authorship, and Lunch) between the employees of Computer Science department at Aarhus. These variables cover different types of relations between the actors based on their interactions.

CKM physicians innovation multiplex network [37]: this dataset was collected by Coleman, Katz, and Menzel on medical innovation, considering physicians in four towns in Illinois, Peoria, Bloomington, Quincy, and Galesburg. They were concerned with the impact of network ties on the physicians' adoption of a new drug, tetracycline.

London multiplex transport network [49]: this dataset was collected in 2013 from the official website of Transport for London and was manually cross-checked. Nodes are train stations in London and edges encode existing routes between stations. Underground, overground, and DLR stations are considered.

Celegans multiplex gpi network [37]: this dataset considered different types of genetic interactions for organisms in the Biological General Repository for Interaction Datasets (BioGRID, thebiogrid.org), a public database that archives and disseminates genetic and protein interaction data from humans and model organisms.

These networks have been used as benchmark datasets for evaluating multiplex network analysis methods. In addition, the CKM dataset has ground-truth information about the community label of nodes. Therefore, we perform performance testing of link prediction task on all datasets and perform performance testing of shared community detection task on CKM dataset.

5.1.2. Private Dataset. We first introduce the acknowledgment data from the dissertation. The acknowledgment chapter is the most emotional part in a dissertation, which can truly reflect the author's research and social interaction. There are many entities in this chapter, including the

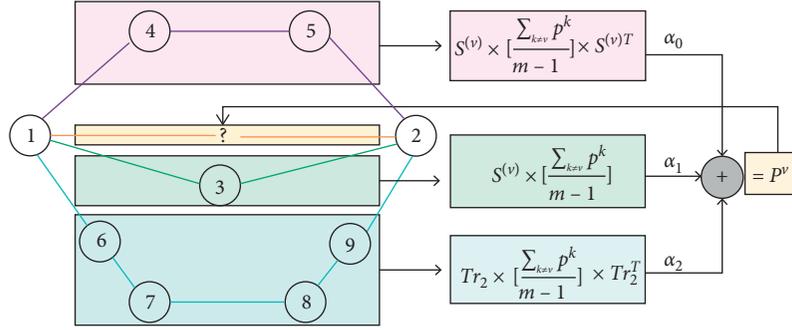


FIGURE 7: A toy example illustration. Green part is an iterative process that directly uses adjacent information. Purple part is an iterative process that uses first-order neighbor similarity information. Blue part is an iterative process that uses second-order neighbor similarity information. The final summation is the approximate similarity between nodes 1 and 2.

Input: Graph $G = \langle V, E, L \rangle$, K is the number of nearest neighbors, μ is a hyperparameter, and α is a parameter in equation (11). $N2V$ is the Node2Vec method. $Nodes_vectors$ are the embedding matrices of node with size N . ζ is the order of enhanced iterative fusion.

Output: *Graph*: the similarity matrix of fusion network

```

(1) begin
(2)   for  $g$  in  $G$  do
(3)      $Node\_vectors.append(N2V(g))$ 
(4)   end
(5)   for  $l$  in  $L$  do
(6)     for  $(v_i, v_j)$  in  $E^l$  do
(7)        $N_i = K\text{-D Tree}(v_i, Node\_vectors, K)$ ;
(8)        $N_j = K\text{-D Tree}(v_j, Node\_vectors, K)$ 
(9)        $W^{(l)} = \text{equation (6)}$ ;  $P^{(l)} = \text{equation (7)}$ ;  $S^{(l)} = \text{equation (8)}$ .
(10)    end
(11)  end
(12)  for  $l$  in  $L$  do
(13)    Calculate 3-order proximity by equation (12).
(14)    Update  $P^{(l)}$  according to equation (13).
(15)  end
(16)   $P = (1/L) \sum_l P^{(l)}$ 
(17)   $Graph = \text{initialize\_graph}()$ 
(18)  for  $v$  in  $V$  do
(19)     $N_v = K\text{-D Tree}(v, P, K)$ 
(20)     $Graph.addedges(v, N_v)$ 
(21)  end
(22)  Return  $Graph$ 
(23) end

```

ALGORITHM 1: MNSF model.

TABLE 1: Basic statistics about different multiplex networks used in this study.

| Name | Nodes | Edges | Layers | Description |
|---------------|-------|-------|--------|---------------|
| VICKERS | 29 | 740 | 3 | Social |
| CS-Aarhus | 61 | 620 | 5 | Social |
| London | 369 | 441 | 3 | Transport |
| CKM | 246 | 1551 | 3 | Social |
| Celegans | 3879 | 8181 | 6 | Genetic |
| Ack-Co-author | 3383 | 29128 | 2 | Co-authorship |

author's mentor, teachers, fellow students in laboratory, classmates or family members, etc. We construct acknowledgment network in the acknowledgment of all

dissertations from 1997 to 2015 in several Chinese universities. In addition, we also construct the co-authorship network by an open academic graph: ArnetMiner,

containing 154,771,162 papers from AMiner. Based on these two datasets, we construct a private Ack-Co-author multiplex network which contains explicit relation (co-author) and implicit semantic relation (acknowledgment network). We also perform algorithm verification on this dataset.

5.2. Experimental Setup. For implementing the network feature extraction module, we use representation learning of nodes to extract the feature of each layer. We set $p=2$ and $q=1$ as default parameters in the biased sample process of the Node2Vec method. We set the number of walks to 20, walk length to 30, and the dimension of vectors to 128. In MNSF, the detailed parameters in enhanced iterative fusion process are set as $K=20$, $\mu=0.4$, and $\alpha_0, \alpha_1, \alpha_2 = \{0.3, 0.4, 0.3\}$. In the construction process of the fused network, the K value of K-D tree is also 20. According to a large number of experiments, the settings of the above parameters are designed for the performance trade-off of our model on different datasets.

5.3. Baseline Method. In these experiments, we test 11 baseline methods with the same parameters and dimensions. The explanations of these baseline methods are as follows. Some of these methods can be used to test two tasks simultaneously. Other methods can only be suited to one of the two tasks. Details of baseline methods are as follows:

- (i) CN (common neighbor) captures the notion that two nodes that have a common neighbor may be introduced by that neighbor. It has the effect of “closing a triangle” in the graph and likes a common mechanism in real life.
- (ii) JC (Jaccard coefficient) is a measure used for gauging the similarity and diversity of sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets.
- (iii) AA (Adamic/Adar) is a measure to predict links, according to the number of shared links between two nodes. It is defined as the sum of the inverse logarithmic degree centrality of the neighbors shared by the two nodes.
- (iv) AAMT [33] is a link prediction method for multiplex networks based on the Adamic/Adar coefficient neighbor similarity, which considers the intensity and structural overlap of multiplex links simultaneously.
- (v) Node2Vec [44] adds a pair of parameters to achieve BFS and DFS sampling process on the single-layer network. It makes it better for capturing the role of nodes, such as hubs or tail users.
- (vi) Ohmnet [15] is a node embedding method for multiplex networks, where hierarchy information is used to model dependencies between the layers.
- (vii) PMNE [16] has three methods of node embedding, each of which generates a common embedding of each node by merging multiple networks. We

compare these three models with other baseline methods. We denote their “network aggregation,” “results aggregation,” and “co-analysis model” as PMNE(n), PMNE(r), and PMNE(c), respectively.

- (viii) MNE [27] is a scalable multiplex network embedding. It contains one high-dimensional common embedding and a low-dimensional additional embedding for each type of relations. Then, multiple relations can be learned jointly based on a unified network embedding model.
- (ix) MELL [30] is a novel embedding method for multiplex networks, which incorporates an idea of layer vector that captures and characterizes each layer’s connectivity. This method exploits the overall structure effectively and embeds both directed and undirected multiplex networks, whether their layer structures are similar or complementary.
- (x) GraphSAGE [50] is a graph neural network framework for inductive representation learning on large graphs. GraphSAGE is used to generate low-dimensional vector representations for nodes and is especially useful for graphs that have rich node attribute information. We use an unsupervised learning version of GraphSAGE to serve as a baseline method of the link prediction task.
- (xi) GenLouvain [51] is a modularity-based multiplex network community detection algorithm. The algorithm not only considers the modularity within the layer but also considers the modularity between layers. By maximizing the modularity metrics, the algorithm completes the community detection task. We only use this algorithm as a baseline method for the node clustering task.

In this paper, we only apply CN, JC, AA, Node2Vec, and GraphSAGE to link prediction tasks on a single layer where the test edge is located. For the Ohmnet algorithm, we construct a hierarchy describing relationships between different layers randomly. We regard the common embedding in the MNE algorithm as the node global embedding. AAMT uses the multiplexity property of nodes (interlayer information) and similarity between nodes (intralayer information) to predict the probability of link. Besides the same walk length, walk times and embedded dimensions are set as the same parameters of MNSF, and we also set other experimental baseline methods using the default parameters, such as PMNE and MELL.

5.4. Link Prediction. In this section, we perform the link prediction task on these multiplex networks. We refer to the experimental settings of the multiplex networks of literature [52]. For the link prediction task, we remove 20% edges of each layer in the original network and use area under the curve (AUC) scores to evaluate the performance of these algorithms for predicting missing edges in each layer. In this paper, we use the residual (80%) edges of each layer for training, and 20% edges are randomly selected from each

layer for testing. These node pairs in edge sets of the test set are regarded as positive examples. Then, we randomly sample an equal number of node pairs from the test set, in which no edge connecting node pairs is served as negative examples. AUC is the area under the receiver operating characteristic (ROC) curve. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. With positive examples (Pos) and negative examples (Neg), AUC can be calculated by

$$\text{AUC} = \frac{\sum_{i \in \epsilon_+} \text{rank}_i - \text{Pos}(1 + \text{Pos})/2}{\text{Pos} \times \text{Neg}}. \quad (14)$$

We calculate the similarity between nodes by AA (Adamic/Adar) metric based on the fused network. In terms of node embedding methods, we use the cosine function of vectors as a similarity metric. The larger the similarity scores are, the more likely there exists a link between them. For other single-layer network methods, we train a separate embedding for each relation type of the network to predict links on the corresponding relation type. It means that they do not have information from other relation types of the network.

From Table 2, we can know that MNSF is significantly better than other comparison algorithms. Our model shows better performance on multiplex network dataset method than single-layer methods such as CN, JC, AA, Node2Vec and GraphSAGE, which directly proves that structural information fusion can improve the accuracy of link prediction. We perform the Node2Vec method on a single target layer of multiplex networks. Under the condition of having the same network feature extraction module, the results of the comparison experiment also indirectly prove the effectiveness of the enhanced iterative fusion module. Similarity network construction can refine the original network's information and filter the noise information and redundant information. The iterative fusion process can capture structural information from other networks. This result also validates the point of view of literature [23]. We regard Ohmnet, PMNE, MNE, MELL, and MNSF as comparative experimental groups. The first four algorithms are the latest multiplex network representation learning methods to realize network feature fusion. Ohmnet and PMNE are extensions of the traditional single-layer network embedding method (Node2Vec), but there is no direct or indirect consideration of interlayer correlation and dependency information in the fusion information. It leads to an inevitable loss of information in the fusion process, so the structural information fusion of multiplex networks cannot be well realized. Both MNE and MELL transfer between nodes in the layer from the perspective of consistency information (shared information) and complementary information (unique information). In these two algorithms, the common (or layer) embedding is considered, but these embedding methods ignore redundant and uninformative information in the network. This process of interlayer node embedding based on common vectors can lead to distortion and inaccuracy of information. On the Celegans dataset, the AAMT obtains outstanding results. We think that nodes

have strong shared information except layer 2. In the iteration process, MNSF makes structural information between each layer as similar as possible. The specificity of layer 2 leads to differences between the fused network and other layers. In fact, almost all the layers except the second are very similar, which is the reason for the unsatisfactory performance of our model. AAMT can consider the Adamic/Adar index and the multiplexity of nodes of each layer comprehensively, so this influence is weaker than that of MNSF.

5.5. Shared Community Detection. Community detection aims to group similar nodes so that nodes in the same group are more similar to each other than those in different groups. In CKM dataset, nodes have the global community label. In other words, each node in a multiplex network has different relation types but only belongs to a unique community. For this dataset, this task usually is called a shared community detection task, which is a significant mining task in multiplex network analysis. So, we use CKM dataset as the benchmark dataset of the shared community detection task. After fused multiplex networks, the traditional community detection algorithms can be applied to the fusion network and be treated as comparison methods of this paper. In this paper, since GN algorithm [53] simply obtains node partition with different number communities, experimental comparison can be conducted under the same number of communities. So, we use GN algorithm to conduct the shared community detection task in this paper.

5.5.1. Evaluation Metrics. Given the ground-truth community in the real-world datasets, we use normalized mutual information (NMI) [54] to evaluate the performance of the methods.

$$\text{NMI}(X|Y) = 1 - \frac{H(X|Y) + H(Y|X)}{2}, \quad (15)$$

where X and Y denote two partitions of the network and $H(X|Y)$ denotes the normalized conditional entropy of a partition X with respect to Y which can be expressed as follows:

$$H(X|Y) = \frac{1}{|C|} \sum_k \frac{H(X_k|Y)}{H(X_k)}, \quad (16)$$

where $|C|$ denotes the number of the community. The larger the NMI is, the better the result is. The value of NMI ranges from 0 to 1. It equals to 1 when two partitions match perfectly and equals to 0 on the contrary.

In the domain of node clustering, the chance-corrected version of this measure is the adjusted Rand index (ARI). It is known to be less sensitive to the number of parts. It is possible to say that two elements of \mathbf{Y} , i.e., (x, x') , are paired in \mathbf{P} if they belong to the same cluster. Let \mathbf{Q} and \mathbf{U} be two partitions of the object set \mathbf{Y} . The Hubert–Arabie formulation of the adjusted Rand index is

$$\text{ARI} = \frac{2(ad - bc)}{b^2 + c^2 + 2ad + (a + d)(c + b)}, \quad (17)$$

where a is the number of pairs $(y, y') \in \mathbf{Y}$ that are paired in \mathbf{Q} and in \mathbf{U} ; b is the number of pairs $(y, y') \in \mathbf{Y}$ that are paired in \mathbf{Q} but not paired in \mathbf{U} ; c is the number of pairs $(y, y') \in \mathbf{Y}$ that

TABLE 2: Link prediction task.

| Algorithm | Datasets | | | | | |
|-----------|---------------|---------------|---------------|---------------|---------------|---------------|
| | Celegans | CKM | CS-Aarhus | London | VICKERS | Ack-Co-author |
| Node2Vec | 0.7847 | 0.8021 | 0.8997 | 0.6816 | 0.6667 | 0.5097 |
| Ohmnet | 0.8427 | 0.8576 | 0.8826 | 0.3580 | 0.7841 | 0.8060 |
| PMNE(n) | 0.5012 | 0.4773 | 0.5154 | 0.4993 | 0.5013 | 0.4981 |
| PMNE(r) | 0.4945 | 0.5043 | 0.5205 | 0.4782 | 0.5002 | 0.5076 |
| PMNE(c) | 0.5003 | 0.4757 | 0.5047 | 0.5043 | 0.4955 | 0.4983 |
| MNE | 0.6313 | 0.7902 | 0.8842 | 0.4526 | 0.7048 | 0.7093 |
| MELL | 0.8085 | 0.7599 | 0.9014 | 0.4991 | 0.7923 | 0.7227 |
| GraphSAGE | 0.7629 | 0.8521 | 0.7023 | 0.5160 | 0.7571 | 0.3991 |
| CN | 0.7467 | 0.6517 | 0.8855 | 0.5054 | 0.7932 | 0.5104 |
| JC | 0.7330 | 0.6526 | 0.8883 | 0.5054 | 0.7864 | 0.5102 |
| AA | 0.7524 | 0.6523 | 0.8962 | 0.5054 | 0.8145 | 0.6968 |
| AAMT | 0.8604 | 0.8239 | 0.9232 | 0.5266 | 0.7389 | 0.6968 |
| MNSF | 0.8112 | 0.9181 | 0.9250 | 0.7751 | 0.8169 | 0.8105 |

All numbers are the averaged AUC scores based on 10 time calculations.

are not paired in \mathbf{Q} but paired in \mathbf{U} ; and d is the number of pairs $(y, y') \in \mathbf{Y}$ that are neither paired in \mathbf{Q} nor paired in \mathbf{U} . This index has an upper bound of 1 and takes the value 0 when the Rand index is equal to its expected value.

5.5.2. Result Analysis. As shown in Figure 8, MNSF also shows excellent performance in shared community detection task in general. Among them, MNSF has obtained the largest NMI and ARI scores. The main reason is that there are isolated nodes in the network, and the shared community detection result of the whole network is up to 0.9835. Compared with other comparison algorithms, it is verified in the shared community detection task that our model can preserve the global mesoscale information of the multiplex network more effectively. In particular, there is strong complementary information between layers in CKM dataset. MNSF further validates that the network fusion method can more fully consider the consistency and complementarity between networks. In terms of other methods, MELL learns a representation of each node separately in each layer. We sum the representations in different layers of nodes as the global embedding of nodes and compare them with our model. Therefore, the performance of MNE and MELL in this task shows that this kind of algorithm cannot well preserve the shared community information of nodes. Ohmnet and GenLouvain methods show competitive performance. They detect the network sharing community from a global perspective.

In general, the results of link prediction and shared community detection tasks prove the effectiveness of our model. Considering the 1st, 2nd, and 3rd order proximity of enhancement iteration process, MNSF can effectively fuse the shared and complementary structural information between layers and preserve more abundant network structural information such as microscale and mesoscale information.

5.6. Parameter Sensitivity. In this section, we test the parameter sensitivity of MNSF for the link prediction task. Based on the above experimental setup, we use the variable-controlling (adjust one parameter and fix other parameters)

strategy and the CS-Aarhus dataset to study parameter sensitivity. The detailed experimental parameters are (1) the nearest neighbor parameter K in proximity network construction and iteration process and (2) the hyperparameter μ in calculating the weight matrix process. As shown in Figure 9(a), with the parameter K gradually increasing, there is a significant change for AUC scores, which rises first and then declines. It shows that the performance of our model (blue line) is dependent on the selection of nearest neighbor parameters in the iteration fusion process. In the fusion network construction process, K can directly affect the quality of the fusion network and the performance of downstream tasks. As K gradually increases, the selected nearest neighbors result in a nonsimilar node pair being constructed as an edge and introduce noisy information. However, SNF is more sensitive to the selection of K values than MNSF. Compared with the original SNF model (orange line) in the literature [22], the proposed enhanced iterative fusion strategy can alleviate the sensitivity problem. According to Figure 9(b), the hyperparameter μ is suitable around 0.4 in CS-Aarhus and optimizes the performance of the test task. It is consistent with the recommended range of values given in [22].

5.7. Verification Experiment

5.7.1. Compared Methods. We use three comparison algorithms to verify the validity of each module of MNSF.

- (i) MerNet: this network construction method is to integrate edges of multiplex networks directly. In a multiplex network, if there are edges between node pairs, then there is an edge between the node pairs in the merge network.
- (ii) Net4Mnsf constructs a network by making each layer of multiplex networks participate directly in the iteration fusion process of the MNSF method.
- (iii) MMSF(DW) is constructed via DeepWalk, a node embedding method, to replace Node2Vec method in the network feature extraction module of MNSF.

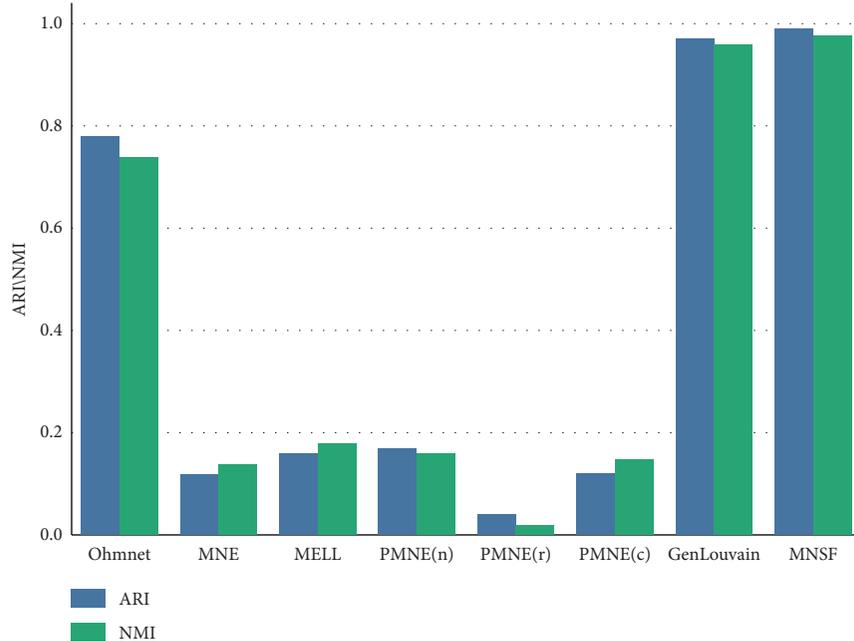


FIGURE 8: The ARI and NMI score performance of our model and baselines on the CKM multiplex network dataset.

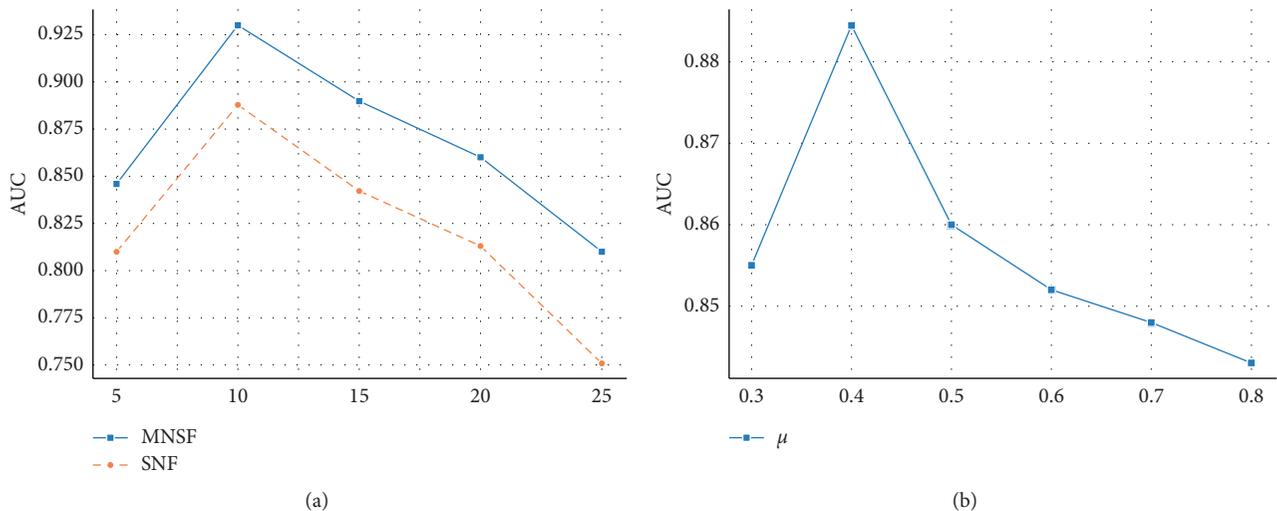


FIGURE 9: Impact of two parameters on area under curve (AUC) scores for models in link prediction task. (a) Impact of the nearest neighbor parameter K on MNSF and SNF. (b) Impact of the hyperparameter μ on MNSF.

5.7.2. Result Analysis. As can be seen from Figure 10, the fused network obtained by MNSF is superior to MerNet obtained by directly collecting the edges of the multiplex network (the flattening of a multiplex network). It shows that our model enhances the mesoscale structure of the network through nonlinear enhanced iterative instead of linearly merging the edges between nodes from a microscale perspective. Moreover, MNSF outperforms Net4Mnsf method that directly feeds the network to the MNSF iterative fusion process. It indicates that the network feature extraction module (representing learning method and similarity network construction) in our model can filter some noise, so the module is effective and meaningful. Both

MNSF(DW) and MNSF are different in the network feature extraction module. Both of them can achieve the best ARI and NMI scores, which also verify that the method of high-order similarity used in the enhanced iterative fusion process can effectively alleviate the dependence on the quality of the obtained node embedding.

5.8. Efficiency Experiment. From Figure 11, we can see that MNSF generally has relatively fast computational efficiency. Our model is based on K-D tree algorithm, and the average time complexity of K-D tree is $O(\log N)$. Therefore, MNSF has lower computational complexity than the original SNF

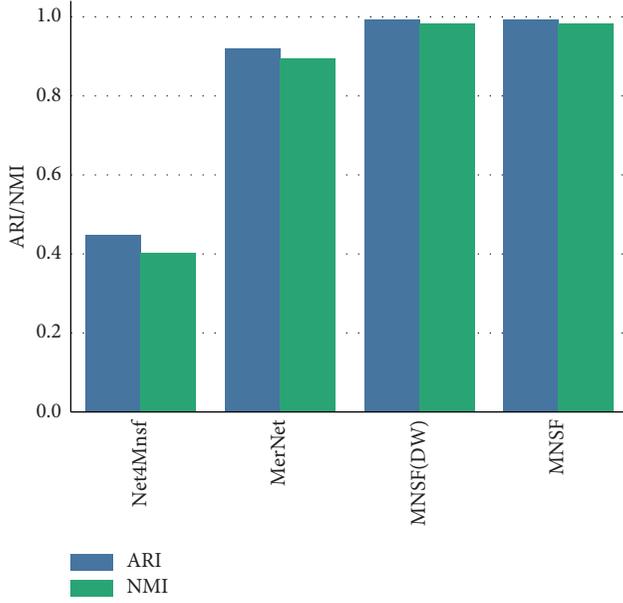


FIGURE 10: The ARI and NMI scores of different verification algorithms in CKM dataset.

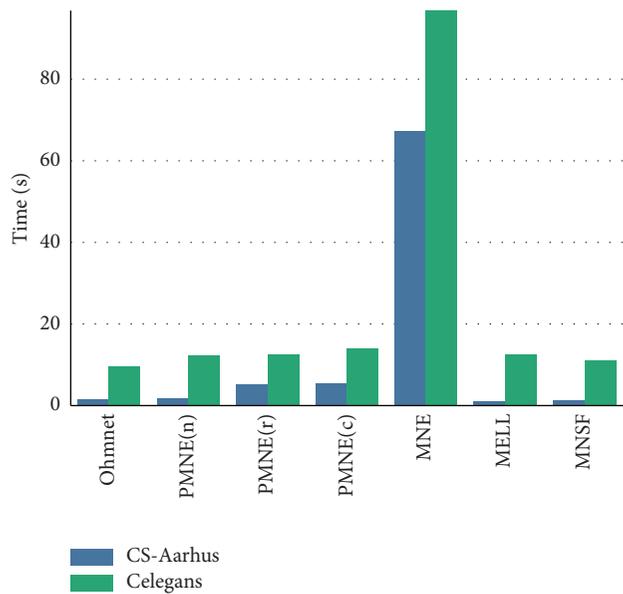


FIGURE 11: The run time of different verification algorithms in Celegans and CS-Aarhus datasets.

model (based on KNN). Besides, in the network feature extraction model, we use the Node2Vec algorithm, which has better scalability. The MNE and MELL algorithms result in computational complexity due to feature learning and backpropagation to optimize the model. For Ohmnet and PMNE algorithms, random walks at each layer of multiplex networks are required to reduce the computational efficiency of the algorithm. Therefore, MNSF also has satisfactory computational performance in real-world datasets with different scales.

6. Conclusions

In this paper, we propose a deep structural fusion framework of multiplex networks, named MNSF, which is based on network representation learning and enhanced iterative fusion (EIF). MNSF utilizes a network embedding method to generate a low-dimensional vector representation of nodes in each layer. Based on the node embedding matrices, MNSF constructs node similarity networks for each layer of a multiplex network. Considering the sharing and complementarity of multiplex networks, we also propose a non-linear enhanced iterative fusion strategy to fuse these similarity networks into a comprehensive single-layer network. Moreover, in the iteration process, EIF alleviates the dependence on the quality of node embedding and provides more abundant information by higher-order proximity. We evaluate MNSF for link prediction and shared community detection tasks on real datasets from different domains. The experimental results verify that our model outperforms baseline methods in general. It indicates that MNSF can fuse the structural information of multiplex networks more effectively than the existing methods. The structural fusion of multiplex networks has promising prospects. For future work, we will investigate the impact of different network representation learning methods on our model. Besides, we will try to apply it to other applications such as cross-domain retrieval and cross-network information propagation.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Key Research and Development Program of China (2018YFC0831500), the National Natural Science Foundation of China (grant no. 61972047), and the NSFC-General Technology Basic Research Joint Funds (grant no. U1936220).

References

- [1] M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, and D. Kim, "Methods of integrating data to uncover genotype-phenotype interactions," *Nature Reviews Genetics*, vol. 16, no. 2, pp. 85–97, 2015.
- [2] J. Zhang and S. Y. Philip, *Broad Learning Through Fusions*, Springer, Basel, Switzerland, 2019.
- [3] M. E. J. Newman, "Network structure from rich but noisy data," *Nature Physics*, vol. 14, no. 6, pp. 542–545, 2018.
- [4] H. Nassar and D. F. Gleich, "Multimodal network alignment," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, SIAM, Houston, TX, USA, pp. 615–623, April 2017.
- [5] A. Hu and S. Flaxman, "Multimodal sentiment analysis to explore the structure of emotions," in *Proceedings of the 24th*

- ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, London, UK, pp. 350–358, August 2018.
- [6] S. Xue, J. Lu, and G. Zhang, “Cross-domain network representations,” *Pattern Recognition*, vol. 94, pp. 135–148, 2019.
 - [7] P. Cudré-Mauroux and K. Janowicz, “Leveraging knowledge graphs for big data integration: the XI pipeline,” *Semantic Web*, vol. 11, no. 1, pp. 13–17, 2020.
 - [8] J. Zhang and S. Y. Philip, “Community detection for emerging networks,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*, SIAM, Vancouver, BC, Canada, pp. 127–135, April 2015.
 - [9] J. Zhang, C. Xia, C. Zhang, L. Cui, Y. Fu, and S. Y. Philip, “Bl-mne: emerging heterogeneous social network embedding through broad learning with aligned autoencoder,” in *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, New Orleans, LA, USA, pp. 605–614, November 2017.
 - [10] N. Malod-Dognin, P. Julia, S. F. L. Windels et al., “Towards a data-integrated cell,” *Nature Communications*, vol. 10, no. 1, p. 805, 2019.
 - [11] L. Liang, V. Chen, K. Zhu, X. Fan, X. Lu, and S. Lu, “Integrating data and knowledge to identify functional modules of genes: a multilayer approach,” *BMC Bioinformatics*, vol. 20, no. 1, p. 225, 2019.
 - [12] X. Zhu, X. Su, J. Ma, H. Tian, R. Liu, and S. Cai, “Evolutionary cooperation in networked public goods game with dependency groups,” *Complexity*, vol. 2019, Article ID 5920187, 8 pages, 2019.
 - [13] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, 2014.
 - [14] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, “A survey of heterogeneous information network analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.
 - [15] M. Leskovec and L. Jure, “Predicting multicellular function through multi-layer tissue networks,” *Bioinformatics*, vol. 33, no. 14, pp. i190–i198, 2017.
 - [16] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, and L. Chen, “Principled multilayer network embedding,” in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, New Orleans, LA, USA, pp. 134–141, November 2017.
 - [17] P. Ruan, Y. Wang, R. Shen, and S. Wang, “Using association signal annotations to boost similarity network fusion,” *Bioinformatics*, vol. 35, no. 19, pp. 3718–3726, 2019.
 - [18] T. Ma and A. Zhang, “Integrate multi-omic data using Affinity Network Fusion (ANF) for cancer patient clustering,” in *Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, Kansas City, MO, USA, pp. 398–403, November 2017.
 - [19] M. Burgess, E. Adar, and M. Cafarella, “Link-prediction enhanced consensus clustering for complex networks,” *PloS One*, vol. 11, no. 5, Article ID e0153384, 2016.
 - [20] E. Ivannikova, H. Park, T. Hämäläinen, and K. Lee, “Revealing community structures by ensemble clustering using group diffusion,” *Information Fusion*, vol. 42, pp. 24–36, 2018.
 - [21] C. De Bacco, A. P. Eleanor, D. B. Larremore, and C. Moore, “Community detection, link prediction, and layer interdependence in multilayer networks,” *Physical Review E*, vol. 95, no. 4, Article ID 042317, 2017.
 - [22] B. Wang, A. M. Mezlini, F. Demir et al., “Similarity network fusion for aggregating data types on a genomic scale,” *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.
 - [23] N. Hoang and T. Maehara, “Revisiting graph neural networks: all we have is low-pass filters,” 2019, <https://arxiv.org/abs/1905.09550>.
 - [24] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, “Multimodal machine learning: a survey and taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.
 - [25] Y. Liu, L. He, B. Cao, S. Y. Philip, A. B. Ragin, and A. D. Leow, “Multi-view multi-graph embedding for brain network clustering analysis,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 117–124, New Orleans, LA, USA, February 2018.
 - [26] G. Ma, C.-Ta Lu, L. He, S. Yu Philip, and A. B. Ragin, “Multi-view graph embedding with hub detection for brain network analysis,” in *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, New Orleans, LA, USA, pp. 967–972, November 2017.
 - [27] H. Zhang, L. Qiu, L. Yi, and Y. Song, “Scalable multiplex network embedding,” in *Proceedings of the International Joint Conferences on Artificial Intelligence*, pp. 3082–3088, London, UK, August 2018.
 - [28] Ma Yao, Z. Ren, Z. Jiang, J. Tang, and D. Yin, “Multi-dimensional network embedding with hierarchical structure,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ACM, Los Angeles, CA, USA, pp. 387–395, February 2018.
 - [29] Yu Shi, F. Han, X. He et al., “mvm2vec: preservation and collaboration in multi-view network embedding,” 2018, <http://arxiv.org/abs/1801.06597>.
 - [30] R. Matsuno and T. Murata, “Mell: effective embedding method for multiplex networks,” in *Proceedings of the Companion of the Web Conference*, International World Wide Web Conferences Steering Committee, Lyon France, pp. 1261–1268, April 2018.
 - [31] S. A. Al-Sayouri, E. Gujral, D. Koutra, E. E. Papalexakis, and S. S. Lam, “t-PNE: tensor-based predictable node embeddings,” in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, Barcelona, Spain, pp. 491–494, August 2018.
 - [32] J. Ni, S. Chang, X. Liu et al., “Co-regularized deep multi-network embedding,” in *Proceedings of the 2018 World Wide Web Conference*, International World Wide Web Conferences Steering Committee, Lyon, France, pp. 469–478, April 2018.
 - [33] D. Hristova, A. Noulas, C. Brown, M. Musolesi, and C. Mascolo, “A multilayer approach to multiplexity and link prediction in online geo-social networks,” *EPJ Data Science*, vol. 5, no. 1, p. 24, 2016.
 - [34] C. Deng, S. Zheng, X. He, X. Yan, and J. Han, “Mining hidden community in heterogeneous social networks,” in *Proceedings of the 3rd International Workshop on Link Discovery*, ACM, New York, NY, USA, pp. 58–65, August 2005.
 - [35] M. Berlingerio, M. Coscia, and F. Giannotti, “Finding and characterizing communities in multidimensional networks,” in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, IEEE, Sydney, Australia, pp. 490–494, July 2011.
 - [36] L. Tang, X. Wang, and H. Liu, “Community detection via heterogeneous interaction analysis,” *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

- [37] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Communications*, vol. 6, no. 1, p. 6864, 2015.
- [38] T. Xu, T. D. Le, L. Liu, R. Wang, B. Sun, and J. Li, "Identifying cancer subtypes from miRNA-TF-mRNA regulatory networks and expression data," *PloS One*, vol. 11, no. 4, Article ID e0152792, 2016.
- [39] L. Cowen, T. Ideker, B. J. Raphael, and R. Sharan, "Network propagation: a universal amplifier of genetic associations," *Nature Reviews Genetics*, vol. 18, no. 9, pp. 551–562, 2017.
- [40] S. Pai and G. D. Bader, "Patient similarity networks for precision medicine," *Journal of Molecular Biology*, vol. 430, no. 18, pp. 2924–2938, 2018.
- [41] S. Pai, S. Hui, I. Ruth, M. A. Shah, H. Kaka, and G. D. Bader, "Netdx: interpretable patient classification using integrated patient similarity networks," *Molecular Systems Biology*, vol. 15, no. 3, Article ID e8497, 2019.
- [42] A. Ghavasieh and M. De Domenico, "Enhancing transport properties in interconnected systems without altering their structure," *Physical Review Research*, vol. 2, no. 1, Article ID 013155, 2020.
- [43] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: a survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.
- [44] 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, San Francisco, CA, USA, pp. 855–864, 2016.
- [45] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196, Beijing, China, June 2014.
- [46] M. Zitnik, B. Wang, A. Pourshafeie et al., "Network enhancement: a general method to denoise weighted biological networks," *Nature Communications*, vol. 9, no. 1, pp. 855–864, 2018.
- [47] A. Goldenberg, Q. Morris, S. Mostafavi, and T. Ravasi, "Labeling nodes using three degrees of propagation," *PLoS One*, vol. 7, no. 6, Article ID e51947, 2012.
- [48] M. Magnani, B. Micenkova, and L. Rossi, "Combinatorial analysis of multiple networks," 2013, <http://arxiv.org/abs/1303.4986>.
- [49] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, "Navigability of interconnected networks under random failures," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8351–8356, 2014.
- [50] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 2017, pp. 1025–1035, 2017.
- [51] J. L. Jeub and M. Bazzi, "A generalized louvain method for community detection implemented," 2011, <https://github.com/GenLouvain/GenLouvain>.
- [52] S. Bai, L. Li, J. Cheng, S. Xu, and X. Chen, "Predicting missing links based on a new triangle structure," *Complexity*, vol. 2018, Article ID 7312603, 11 pages, 2018.
- [53] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [54] J. Cheng, X. Su, H. Yang et al., "Neighbor similarity based agglomerative method for community detection in networks," *Complexity*, vol. 2019, Article ID 8292485, 16 pages, 2019.