Applied Intelligence (2024) 54:592–613
https://doi.org/10.1007/s10489-023-05211-1

Deep non-negative matrix factorization with edge generator for link prediction in complex networks

Yabing Yao[1] · Yangyang He[1] · Zhentian Huang[1] · Zhipeng Xu[1] · Fan Yang[2] · Jianxin Tang[1] · Kai Gao[1]

Accepted: 1 December 2023 / Published online: 15 December 2023

Abstract
Link prediction aims to infer missing links or predict future links based on observed topology or attribute information in the network. Many link prediction methods based on non-negative matrix factorization (NMF) have been proposed to solve prediction problem. However, due to the sparsity of real networks, the observed topology information is probably very limited, which affects the performance of existing link prediction methods. In this paper, we utilize Deep Non-negative Matrix Factorization (DNMF) models with Edge Generator to address the network sparsity problem and propose link prediction methods EG-DNMF and EG-FDNMF. Under the framework of DNMF, several representative potential edges are incorporated so as to reconstruct the original network for link prediction. Specifically, in order to explore the potential structural features of the network in a more fine-grained manner, we first divide the original network into three sub-networks. Then, the DNMF models are employed to mine complex and nonlinear interaction relationships in sub-networks, thereby guiding the network reconstruction process. Finally, the NMF algorithm is applied on the reconstructed original network for link prediction. Experiment results on 12 different networks show that our methods have comparable performance with respect to 13 representative link prediction methods which include 6 NMF/DNMF-based approaches and 7 heuristic-based approaches. In addition, experiments also show that the sub-networks after partitioning are beneficial for capturing the underlying features of the network. Codes are available at https://github.com/yabingyao/EGDNMF4LinkPrediction

Keywords Link prediction · Network reconstruction · Sub-network · Deep non-negative matrix factorization

1 Introduction

Network science offers a novel approach to studying real-world complex system. In this way, complex systems are modeled as networks where nodes denote entities and links represent interactions between two entities [1, 2]. Given the presence of missing links during network construction and the potential emergence of new interactions in the future, the task of identifying these missing or potential links becomes a pivotal concern. Link prediction as a tool can effectively solve this problem, which aims to infer missing links or predict future links based on observed topology or attribute information in the network [3, 4]. It is widely used in various scenarios such as social recommendation, e-commerce and biological networks, etc. For instance, in friendship or co-author networks, link prediction can recommend potential friends or seek possible partnerships for people [5, 6]. In e-commerce platforms, link prediction is often used for product recommendations [7]. To reduce the expensive cost of biological experiments, link prediction is usually applied to the preprocessing stage [8, 9].

Many link prediction methods have been proposed over the past few decades. They can be roughly grouped into three categories [10, 11]: heuristic-based approaches, probabilistic models and dimensionality reduction-based approaches. The simplest methods used for link prediction are heuristic-based approaches, which predict links by calculating the similarity score between two nodes with the assistance of heuristic hypotheses. Specifically, the node similarity can be calculated by various indexes, including local indexes, global indexes and quasi-local indexes [10, 12]. The local indexes

✉ Fan Yang
100002022@gxust.edu.cn

[1] School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

[2] School of Computer Science and Technology, Guangxi University of Science and Technology, Liuzhou 545006, China

calculate similarity scores based on the local topology around nodes, such as Common Neighbors (CN) [13], Adamic-Adar (AA) [14] and Resource Allocation (RA) [15], etc. The global indexes consider the overall topology of the network, for example, Katz index (Katz) [16], Average Commute Time (ACT) [17] and Random Walk with Restart (RWR) [18], etc. Quasi-local indexes have better predictive ability than local indexes, but are less complex than global indexes, representative approaches include Local Random Walk (LRW) [17], Local Path (LP) [19] and CNDP [20], etc. Probabilistic models use the likelihood estimation of model parameters to predict connection probabilities [10]. Hierarchical Structure Models (HSM) [21] and Stochastic Block Models (SBM) [22] are representative approaches based on this idea.

Dimensionality reduction-based methods can be divided into network embedding and non-negative matrix factorization (NMF)-based approaches [11]. The former maps nodes in the network to a low-dimensional vector space, while preserving the network structure information [23]. Representative methods include DeepWalk [24], node2vec [25], GCN-GAN [26], DEAL [27], Graph2Feat [28], RelpNet [29], LLP [30], etc. The NMF-based approaches explore the potential links by approximating the adjacency matrix as the product of two low-dimensional factor matrices, which have strong interpretability [31], such as NMF-A1 [32], MS-RNMF [33], AM-NMF [34] and ICP [31], etc. However, NMF is a shallow model that cannot capture complex and non-linear topological features in real networks [35, 36]. Therefore, deep non-negative matrix factorization (DNMF) model is proposed to cope with this problem, which can explore the non-linear relationships in the data by multi-layer mapping. Representative approaches include DANMF [37] and FSSDNMF [11], etc.

Although existing link prediction methods have achieved good performance in many fields, they are still susceptible to sparse data, which limits their predictive ability [31, 38]. Moreover, in the process of data collection and network evolution, it is unavoidable to generate random noise [32]. Specifically, these issues are described in detail as follows:

- Sparse data. During the data collection process from various system platforms, critical topology information may not be available for various reasons, such as privacy protection policies or challenging data access, leading to the formation of sparse networks [39, 40]. For instance, in some social networks such as Facebook network, which has 3097165 nodes and 23667394 links, but its density is only 4.9346×10^{-6}.
- Random noise. The network topology we observed may be an artifact caused by random noise [32]. For instance, it is possible that two individuals without common friends are connected by accident in social networks. However, they are less likely to interact depending on the sociological mechanisms. If the prediction models excessively emphasize these instances of noise, it can negatively impact their ability to make accurate predictions [33].

To address the aforementioned issues, various methods have been proposed, such as NMF-A1 [32], to mitigate the impact of random noise and sparse data in real networks by introducing random edges into the network. However, due to the stochastic nature of this process, although it addresses the sparsity issues, the randomly added edges may introduce additional noise to the network. In essence, because the newly added edges are generated randomly, there is no guarantee that they will contribute valuable information to link prediction. Consequently, the challenge of selectively adding meaningful edges to the network becomes an intriguing problem.

In this paper, our motivation is to introduce a network reconstruction approach with the aim of enhancing the performance of existing link prediction methods. Leveraging the demonstrated effectiveness of the deep non-negative matrix factorization (DNMF) in uncovering non-linear features in complex systems [36, 37], we employ DNMF models as edge generator to steer network reconstruction. This approach leads to the development of the methods EG-DNMF and EG-FDNMF. Our methodology commences with the division of the original network into three sub-networks, enabling the capture of topological characteristics in a more refined manner [41, 42]. Furthermore, the divide-and-conquer strategy we introduced for managing complex correlated information, can also be extended to diverse domains, including the Internet of Things and multi-agent systems [43, 44], etc. Subsequently, the DNMF algorithm is applied to unearth concealed features within the sub-networks from a localized perspective and effectively guides the network reconstruction process. The final step involves the utilization of the NMF algorithm within the reconstructed network to address the link prediction task. To demonstrate the performance of EG-DNMF and EG-FDNMF, we selected 13 representative link prediction methods, including 6 NMF/DNMF-based approaches and 7 heuristic-based approaches as baselines. The results clearly show that EG-DNMF and EG-FDNMF consistently deliver robust predictive performance across 12 distinct real networks. Furthermore, we highlight the benefits of chunking processing in improving network reconstruction and enhancing prediction capabilities. In summary, our contribution are as follows:

- Our proposed methods enhance available information by selectively introducing additional links to the network, thereby improving the performance of existing link prediction methods.

- We adopt the divide-and-conquer approach to partition the original network into three sub-networks, allow us to explore the structure characteristics of the network in a more detailed manner.
- We conduct experiments on 12 real networks to verify the performance of the proposed methods EG-DNMF and EG-FDNMF. The experimental results demonstrate their outstanding performance.

The rest of the paper is organized in the following manner. Section 2 provides an overview of related work on link prediction. In Section 3, we provide a detail description of our methods. Section 4 introduces the baselines and evaluation metrics used. Section 5 discusses the experimental results, and Section 6 concludes the paper.

2 Related work

Over the past decade, various link prediction methods have been proposed to predict the connection probability between two nodes. Broadly speaking, these methods can be grouped into three categories [10, 11]: heuristic-based approaches, probabilistic models and dimensionality reduction-based approaches. In this section, we briefly introduce some representative methods and mainly focus on NMF-based approaches. In addition, we summarize the advantages, disadvantages, and representative approaches of each category in Table 1.

2.1 Heuristic-based methods

Heuristic-based methods can be divided into local indexes, global indexes and quasi-local indexes. We introduce some classic approaches as baselines for comparison with our proposed methods.

2.1.1 Common Neighbors

Common Neighbors (CN) [13] index assumes that the similarity between two nodes can be measured by the number of their common neighbors. It is defined by:

$$S_{xy}^{CN} = \mid \Gamma(x) \cap \Gamma(y) \mid \tag{1}$$

where $\Gamma(x)$ and $\Gamma(y)$ represents the neighbor set of nodes x and y, respectively.

2.1.2 Adamic-Adar

Adamic-Adar (AA) [14] index uses the degree of common neighbor to measure the contribution of different neighbors to similarity scores, and assumes that the neighbors with smaller degrees contribute more to the connection probability, which is defined by:

$$S_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{logk_z} \tag{2}$$

where z is the common neighbor between nodes x and y, k_z denotes the degree of node z.

2.1.3 Resource allocation

Resource Allocation (RA) [15] index assumes that each node in the network has a unit of resource and distributes it equally to its neighbors. The connection probability between two nodes is determined by the amount of resources they receive.

Table 1 Comparison of existing link prediction methods

Category	Subcategory	Advantages	Disadvantages	Methods
Heuristic-based	Local	Simple and efficient	Inaccurate prediction.	CN [13] RA [15] AA [20]
	Global	Higher prediction performance than local-based	Higher complexity than local-based.	Katz [16] ACT [12] RWR [18]
	Quasi-local	Lower complexity than global-based.	Weak generalization.	DGLP [45] CNDP [20]
Probabilistic models	Hierarchical structure model	Predicting links through probability model	More complex than similarity-based.	HSM [21] NARM [48]
	Stochastic block model	Easy to capture local structure of nodes.	Complex and not suitable for large-scale networks.	SBM [22] Attribute-SBM [46]
Dimensionality reduction-based	NMF-based	Effectively reduce feature dimension.	Over fitting for sparse networks.	NMF-A1 [32] NMF-D1 [32] ICP [31]
	Network embedding	Could utilize attribute information for prediction.	Poor interpretability and time consuming.	node2vec [25] WLNM [51] RelpNet [29]

It is defined by:

$$S_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z} \tag{3}$$

2.1.4 Degree of Gravity for Link Prediction

Degree of Gravity for Link Prediction (DGLP) [45] combines degree centrality, common neighbors, and distance between candidate node pairs to mitigate the cold start problem for link prediction. It is defined by:

$$S_{xy}^{DGLP} = \frac{\mid \Gamma(x) \mid + \mid \Gamma(y) \mid}{d_{xy} + 1} + \sum_{z \in \Gamma(x) \cap \Gamma(y)} k_z \tag{4}$$

where d_{xy} is the shortest distance between nodes x and y.

2.1.5 Local path

In addition to considering the common neighbors between nodes x and y, i.e., the second-order paths, the Local Path (LP) [19] index simultaneously takes into account the third-order paths between them. It is defined by:

$$S_{xy}^{LP} = (A)_{xy}^2 + \alpha \cdot (A)_{xy}^3 \tag{5}$$

where A_{ij}^2 and A_{ij}^3 represents the number of second-order paths and the third-order paths between nodes x and y, respectively. α is a hyperparameter to control the contribute of the third-order path.

2.1.6 Katz

Katz [16] index takes into account all paths between node pairs and assumes that the shorter paths contribute more to the connection probability, which is defined by:

$$S_{xy}^{Katz} = \sum_{z=1}^{\infty} \beta^z \cdot (A)_{xy}^z \tag{6}$$

where A^z represents the z-order paths, β is a hyperparameter.

2.1.7 Cosine similarity

Cosine similarity [10] combines the ideas of cosine similarity and random walk, which is defined by:

$$S_{xy}^{Cos} = \frac{l_{xy}^+}{\sqrt{l_{xx}^+ \cdot l_{yy}^+}} \tag{7}$$

where l_{xy}^+ represents the inner product of vector v_x and v_y. Here, $v_x = \Lambda^{1/2} U^T e_x$, U is a standard orthogonal matrix, Λ is a diagonal matrix, and e_x represents a one-dimensional vector where only the x-th element being 1 and others being 0.

2.2 Probabilistic models

The most commonly used probabilistic models include hierarchical structure models (HSM) [21] and stochastic block models (SBM) [22]. These methods view the network as a probability model, and predict links by maximum likelihood estimation of model parameters. Moreover, Stanley et al. [46] propose a probability model Attribute SBM, where node is associated with a continuous attribute vector. In order to discover latent links in multilayer networks, Kuang et al. [47] propose a link prediction method based on maximum a posteriori (MAP), which aims to reconstruct network hierarchy and infer missing links. Zhao et al. [48] propose a Bayesian probabilistic model NARM, which incorporates diverse node attribute encodings and exhibits strong performance in both directed and undirected networks.

2.3 Dimensionality reduction-based methods

Dimensionality reduction-based methods can be divided into network embedding and non-negative matrix factorization (NMF)-based approaches.

The network embedding methods maps nodes to a low-dimensional vector space, while preserving the topology information in the network [23, 49]. Inspired by word2vec [50], Perozzi et al. [24] propose the DeepWalk method, opening up the research on network embedding. Subsequently, Grover et al. [25] improve the random walk strategy in DeepWalk and propose the node2vec method. Weisfeiler-Lehman Neural Machine (WLNM) [51] considers the closed subgraph around the target nodes and predicts links by neural network model. With the development of research, graph neural networks (GNNs) have been proposed for link prediction. SEAL [52] utilizes GNN to automatically learn the features of subgraph around the node pairs. NIAN [53] and HalpNet [54] use attention mechanism to obtain the importance of each node in subgraph. In addition, network embedding can also be used to inductive link prediction and temporal link prediction. For example, DEAL [27] considers two node embedding encoders and one alignment mechanism for inductive link prediction. Graph2Feat [28] conducts inductive link prediction by knowledge distillation model. Lei et al. [26] propose a non-linear model GCN-GAN to tackle the temporal link prediction task. Qin et al. [55] present a temporal link prediction model IDEA in weighted dynamic networks.

The matrix factorization (MF) decomposes the adjacency matrix into the product of multiple low-dimensional matrices to explore hidden information. From another perspective, this approach maps node in high-dimensional space to low-dimensional feature space, the proximity between two nodes is determined by their position in feature space [12]. Koren et al. [56] propose a gradient descent-based algorithm for MF, which has been widely applied in recommendation systems. Non-negative Matrix Factorization (NMF) decomposes the adjacency matrix into the product of two non-negative matrices, which offers high interpretability. However, it faces problems related to overfitting and cold start in sparse networks. To address these issues, several representative NMF-based methods have been proposed, for example, NMF-A1 and NMF-D1 [32], MS-RNMF [33], ICP [31], and so on. NMF-A1 and NMF-D1 mitigate the impact of sparsity and random noise by introducing randomly perturbations into the network. MS-RNMF utilizes $\ell_{2,1}$-norm to effectively remove random noise and pseudo-links. ICP improves prediction performance by combining topology and attribute information. In addition, NMF can also be used for dynamic networks. For example, Chen et al. [57] design a novel iterative rule and propose DeepEye method. Lei et al. [34] present AM-NMF based on the NMF framework, which can effectively predict the topology of dynamic networks. Generally, real world networks typically exhibit a diverse organizational structure that contains nonlinear structural information [11]. However, NMF is a shallow model which only one layer mapping the adjacency matrix to the feature space, as a result, this nonlinear relationship is not well captured by NMF [35, 36]. To address the aforementioned shortcomings, deep non-negative matrix decomposition (DNMF) has been proposed. DNMF tackles this issue by mapping the adjacency matrix to the feature space across multiple layers, representative methods include FSSDNMF [11], DNBMF [35] and DANMF [37]. FSSDNMF combines the observed link information and topological features to mine underlying characteristic and utilizes $\ell_{2,1}$-norm to remove random noise. DANMF has achieved good results in community detection by using deep NMF models.

3 Proposed method

In this section, we first give the problem description, then introduce the proposed methods EG-DNMF and EG-FDNMF in detail, which mainly consists of the following three parts:

1) Divide the network. For a given network, we first divide it into three sub-networks, which preserve the local topology information of the network.
2) Reconstruct the network through edge generator. For each sub-network, deep non-negative matrix factorization (DNMF) models are used to explore potential information and guide the process of network reconstruction.
3) NMF algorithm for link prediction. For the reconstructed network, we use the non-negative matrix factorization (NMF) algorithm to complete the link prediction task.

3.1 Problem description

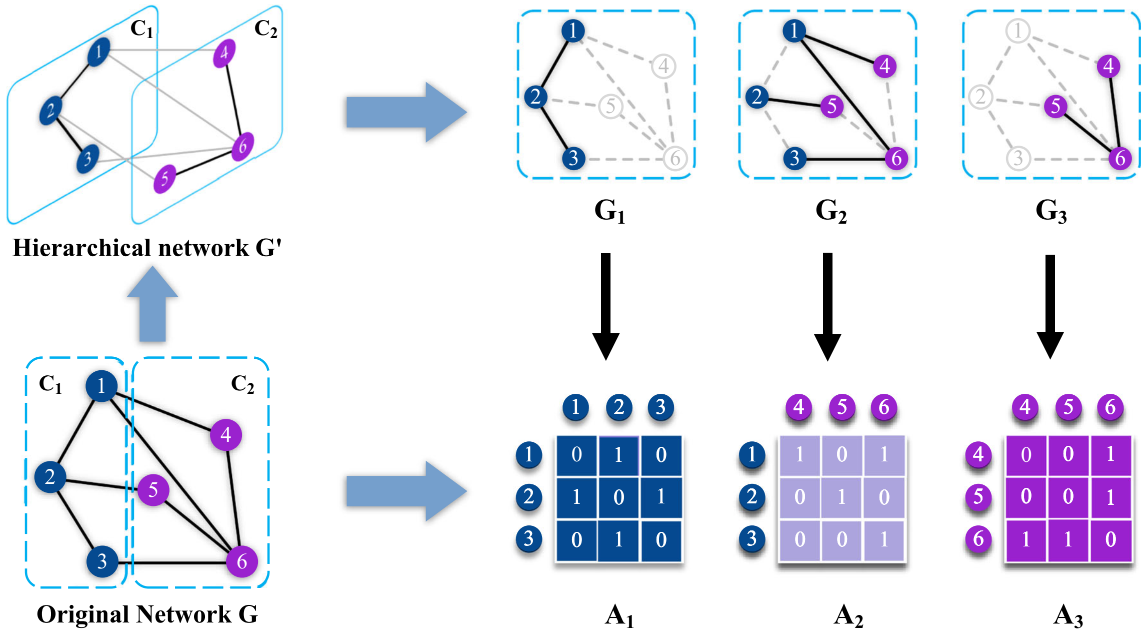
Given an undirected and unweighted network $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the nodes set and $E = \{(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ denotes the set of links. We use A to represent the adjacency matrix of G, where $A_{ij} = A_{ji} = 1$ can be interpreted as existing a link between nodes i and j, otherwise $A_{ij} = A_{ji} = 0$. Further, all possible links in the network can be represented by U, which can be defined as $\frac{|V|(|V|-1)}{2}$, and the goal of link prediction is to find missing links from $U - E$. To verify the performance of different algorithms, we randomly divide the network into a training set E^T and a testing set E^P. Specifically, $E^T \cap E^P = \varnothing$ and $E^T \cup E^P = E$.

Various link prediction methods have been proposed, and they mainly rely on the observed network structure to infer new links. However, real-networks are always sparse and some important topology information may be missing, which limits the performance of existing algorithms. In this paper, we reconstruct network by edge generator to cope with this problem.

3.2 Divide the network

In network representation learning, it has been proven that hierarchies can effectively preserve network structural information at different levels of granularity. For example, methods like MILE [58] and LouvainNE [41] adopt a hierarchical framework to obtain the global or local topology information of the network. Drawing inspiration from this concept, we randomly partition the nodes within a network into two distinct groups and build a hierarchical network by taking into account the intra-group and inter-group interactions. Note that this hierarchical network is a two-layer network. In order to capture the topological characteristics in a more fine-grained manner, we further construct three sub-networks based on the hierarchical relationship within the two-layer network. All three sub-networks contain local structure information of the network. The process of dividing network is illustrated in Fig. 1.

Specifically, given an original network G which has n nodes, we first split the nodes of G into two groups C_1 and C_2 with similar size, where $C_1 = \{v_1, v_2, ..., v_m\}$ and $C_2 = \{v_{m+1}, v_{m+2}, ..., v_n\}$. It is worth noting that in order to reduce complexity, this node division process is random. For example in Fig. 1, we divide the original network G into two groups, which include nodes {1, 2, 3} and {4, 5, 6}, respectively. Then, based on the interaction between two groups, we can get a hierarchical network G' with a two-layer structure,



$$A_1 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 0 & 1 & 0 \end{array} \qquad A_2 = \begin{array}{c|ccc} & 4 & 5 & 6 \\ \hline 1 & 1 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{array} \qquad A_3 = \begin{array}{c|ccc} & 4 & 5 & 6 \\ \hline 4 & 0 & 0 & 1 \\ 5 & 0 & 0 & 1 \\ 6 & 1 & 1 & 0 \end{array}$$

Fig. 1 The flowchart of partitioning network

which contains the intra-group and inter-group connections within these two groups.

To analyze the network in a more fine-grained manner, we further divide G' into three sub-networks G_1, G_2 and G_3. Each sub-network reflects the local structure of the hierarchical network G'. Clearly, $G_1 = \{(v_i, v_j)|v_i, v_j \in C_1\}$, $G_2 = \{(v_i, v_j)|v_i \in C_1, v_j \in C_2\}$ and $G_3 = \{(v_i, v_j)|v_i, v_j \in C_2\}$. In this way, $E(G_1) \cap E(G_2) \cap E(G_3) = \varnothing$ and $E(G) = E(G_1) \cup E(G_2) \cup E(G_3)$. For example in Fig. 1, the intra-group sub-network G_1 can be obtained according to the relationship between nodes 1, 2 and 3 in group C_1, similarly, we can get G_3 in the same way. G_2 reflects the connection between C_1 and C_2, which can be seen as an inter-group sub-network. Moreover, we use A_1, A_2 and A_3 to indicate the matrix representations of the three sub-networks, respectively. It should be noted that A_2 is a asymmetric association matrix, A_1 and A_3 are adjacency matrices.

3.3 Reconstruct the network through edge generator

DNMF models Existing NMF algorithms and their variant methods find extensive applications across various domains, including image recognition, text analysis, and more [59, 60]. They extract features from samples by decomposing an original non-negative matrices into the product of two non-negative matrices: the basis matrix and the coefficient matrix. In essence, these methods are rooted in a linear structure and can only map samples in a single layer [36]. Consequently, some researchers argue that this shallow linear structure may not effectively capture the non-linear underlying features in the data, leading to a growing interest in deep NMF (DNMF) [35]. Most DNMF-based approaches explore the underlying features of samples through iterative decomposition of the coefficient matrix. However, some studies have pointed out that the coefficient matrix of each layer is essentially a weighted coefficient in the process of sample reconstruction, lacking clear interpretability and purpose [35]. Actually, the extracted features are closely related to the basis matrix, so deep factorization of the basis matrix has stronger interpretability and can better obtain the underlying features [35]. Based on the above discussion, in this paper, we use the DNMF algorithm as edge generator to explore hidden links in sub-networks G_1, G_2, G_3 and propose method EG-DNMF. The process of decomposition consist of l layers. DNMF mainly includes two stages: the pre-training stage and the fine-tuning stage.

In the pre-training stage, the basis matrix and coefficient matrix of each layer are obtained by NMF algorithm, as follows:

$$A \approx W_1 H_1, \tag{8}$$

$$W_1 \approx W_2 H_2, \tag{9}$$

$$\vdots \tag{10}$$

$$W_{l-1} \approx W_l H_l \tag{11}$$

where A represents the adjacency matrix of a network, W_i and H_i, $i = 1, 2, 3, \ldots, l$ are basis matrix and coefficient matrix of the ith layer, respectively. Specifically, formula (8) represents the first layer decomposition, formula (9) indicates the basis matrix W_1 from the first layer is decomposed again into a new basis matrix W_2 and a coefficient matrix H_2, and so on. With the above factorization, A can be expressed as:

$$A \approx W_l H_l H_{l-1} \ldots . H_2 H_1 \tag{12}$$

In order to extract the underlying features in the network, the following optimization problems need to be solved:

$$\underset{W_i \geq 0, H_i \geq 0}{min\ O} \quad \| A - W_i H_i \Lambda_{i-1} \|_F^2 \tag{13}$$

where $\Lambda_{i-1} = H_{i-1} \ldots H_1$, and $\Lambda_0 = I_0$ is the identity matrix when $i = 1$.

In fine-tuning stage, we use the gradient descent method to solve the basis matrix and coefficient matrix for each layer, the specific solution process can be found at Ref. [35]. The final update formula is as follows:

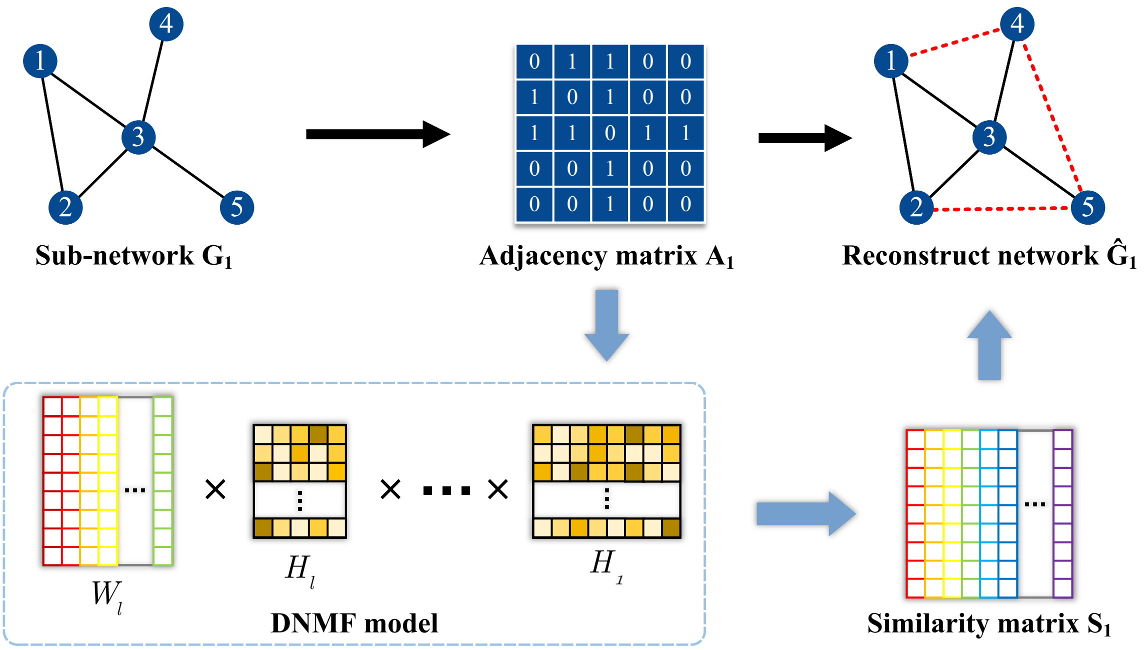
$$W_i = W_i \otimes \frac{A \Lambda_{i-1}^T H_i^T}{W_i H_i \Lambda_{i-1} \Lambda_{i-1}^T H_i^T} \tag{14}$$

$$H_i = H_i \otimes \frac{W_i^T A \Lambda_{i-1}^T}{W_i^T W_i H_i \Lambda_{i-1} \Lambda_{i-1}^T} \tag{15}$$

where $\otimes$ represents the point multiplication operation and / is the matrix division by elements.

In addition, in order to investigate the ability of other DNMF-based models to explore the underlying features, we also use the existing model FSSDNMF [11] as edge generator, and propose the method EG-FDNMF. Unlike the decomposition basis matrix, FSSDNMF explore hidden information by decomposing the coefficient matrix and combing observed links and topological features. The specific solution process of FSSDNMF can be found at Ref. [11].

Reconstruct the network. When the basis matrix and coefficient matrix of each layer are solved, DNMF models unearth concealed information within the network by calculating a similarity matrix S, where S_{ij} denotes the similarity score between nodes i and j, and a higher score indicates a stronger connection between two nodes. We assume that if there is a high similarity score between two unconnected nodes, a missing link exists for that node pair. By introducing this missing link to the original network, we can provide additional valuable topology information to address the sparsity and random noise issues in link prediction. Based on the above ideas, we first apply DNMF or FSSDNMF model in three sub-networks to unearth concealed features from a localized perspective and obtain the similarity matrix S. Then a certain proportion of unconnected node pairs with high similarity scores within S are selected as missing edges. Finally we reconstruct the original network through adding these missing edges to the network, so as to provide additional useful information for the link prediction task. More detailed, the similarity matrix S can be defined as $S = W_l H_l \ldots . H_2 H_1$ if the DNMF model is applied in three sub-networks, and $S = W_1 W_2 \ldots . W_l H_l$ when use the FSSDNMF model in G_1, G_2 and G_3. The process of reconstructing the sub-network G_1 is shown in Fig. 2.



$$\begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}$$

Fig. 2 The flowchart of edge generator

Specifically, for sub-network G_1, we first obtain the similarity matrix S_1 by applying DNMF/FSSDNMF model in G_1, and then reconstruct it in the following steps:

Step 1 Select $\gamma|E_1|$ links from $\{U_1 - E_1\}$ as the candidate edges set $E_1^{'}$, it is important to note that these selected links have the highest similarity scores in S_1. U_1 represents all possible links and E_1 is the observed edges in G_1, γ is a hyperparameter to control the size of the candidate edges set $E_1^{'}$.

Step 2 Randomly select $\eta|E_1^{'}|$ links as the final edge set to add to the G_1, η is also a hyperparameter. Then we can get the reconstructed sub-network $\hat{G}_1$, its corresponding matrix representation is $\hat{A}_1^r$.

Step 3 In order to reduce the impact of randomness, step 2 is repeated R times and take the average to get the final reconstructed matrix $\hat{A}_1$, which is defined by $\hat{A}_1 = \frac{1}{R}\sum_{r=1}^{r=R}\hat{A}_1^r$.

Applying the above reconstruction process to each sub-networks separately, we can obtain three reconstructed sub-networks $\hat{G}_1$, $\hat{G}_2$ and $\hat{G}_3$, their corresponding matrices are $\hat{A}_1$, $\hat{A}_2$ and $\hat{A}_3$. The reconstructed original network $\hat{G} = \hat{G}_1 + \hat{G}_2 + \hat{G}_3$.

3.4 NMF algorithm for link prediction

After getting the reconstructed original network $\hat{G}$, we can use existing link prediction algorithms for link prediction tasks, such as deep learning or NMF-based approaches. In this paper, in order to reduce the overall complexity of the model, we use a shallow NMF algorithm to solve the downstream link prediction task. NMF is a dimensionality reduction-based method that approximates the adjacency matrix of the reconstructed network $\hat{G}$ as the product of two non-negative matrices to reveal potential links. In other words, NMF mines hidden information in a network by mapping the original high-dimensional features to a low-dimensional latent space.

More specifically, let $\hat{A}_{n*n}$ represents the adjacency matrix of reconstructed original network $\hat{G}$, U_{n*k} and V_{k*n} are the basis matrix and coefficient matrix obtained by linear decomposition, respectively. k is the dimension of the latent space, which is a hyperparameter. We learn U_{n*k} and V_{k*n} through multiplicative update and, in the end, ensure that $\hat{A}_{n*n}$, U_{n*k} and V_{k*n} satisfy the following formula:

$$\hat{A}_{n*n} \approx U_{n*k}V_{k*n} \tag{16}$$

Euclidean distance is used as loss function to update U_{n*k} and V_{k*n}, and the corresponding formula is as follows:

$$\underset{U\geq 0, V\geq 0}{min\ O} = ||\ \hat{A}_{n*n} - U_{n*k}V_{k*n}\ ||_F^2 \tag{17}$$

The finally update formula is:

$$U \leftarrow U \otimes \frac{AV^T}{UVV^T} \tag{18}$$

$$V \leftarrow V \otimes \frac{U^TA}{U^TUV} \tag{19}$$

We use the obtained similarity matrix $\hat{S} = U_{n*k}V_{k*n}$ to complete the link prediction task. The pseudocode of the overall process is showed in Algorithm 1.

Algorithm 1 Method of EG-DNMF/EG-FDNMF for link predication.

```
Input: Original network $G$; The network division ratio p; Hyperparameter $\gamma$, $\eta$ and $k$
Output: The similarity matrix $\hat{S}$ for link prediction
function EG- DNMF/EG- FDNMF
    divide $G$ into $E^T$ and $E^P$ based on $p$
    divide $E^P$ into $G_1$, $G_2$ and $G_3$
    for i=1 : 3 do
        get similarity matrix $S_i$ by apply DNMF/FSSDNMF on $G_i$
        reconstructed network $\hat{G}_i$ obtained by $S_i$ with $\gamma$ and $\eta$
    end for
    reconstructed original network $\hat{G} = \hat{G}_1 + \hat{G}_2 + \hat{G}_3$
    apply NMF algorithm on $\hat{G}$ and update $U$, $V$ by formula (18), (19)
    similarity matrix $\hat{S} = UV$ for link prediction
end function
```

3.5 Complexity analysis

For the computational complexity of EG-DNMF/EG-FDNMF, it is mainly composed of the following parts. Firstly, the non original network are divided into two groups and obtained three sub-networks, which complexity is $\mathcal{O}(n + n^2)$. Then apply the DNMF/FSSDNMF model on three sub-networks, the complexity is $\mathcal{O}(l(t_p + t_f)(n^2r + nr^2))$, where l denotes the number of layers, t_p is the number of iterations to achieve convergence in the pre-training process and t_f is the number of iterations in the fine-tuning process, r represents the maximal layer size out of all layers. Finally, the complexity of reconstructing the original network and applying the NMF algorithm is $\mathcal{O}(n^2 + t_pn^2r)$. Therefore, the overall time complexity is $\mathcal{O}(n(n+1) + l(t_p + t_f)(n^2r + nr^2) + t_pn^2r)$.

4 Baselines, metrics, datasets and parameter setting

4.1 Baselines

In this paper, a total of 13 representative link prediction methods are selected as baselines, including 7 heuristic-based approaches, 3 NMF-based approaches and 3 DNMF-based methods. The heuristic-based approaches contains CN [13], AA [14], RA [15], DGLP [45], LP [19], Katz [16] and Cosine similarity [10]. The NMF-based approaches include NMF [35], NMF-A1 [32] and NMF-D1 [32]. The DNMF-based methods are DNMF [35], DANMF [37] and FSSDNMF [11].

4.2 Metrics

AUC and Precision are used to measure the performance of different link prediction algorithms. AUC evaluates the overall performance of the model, while Precision only focuses on the proportion of links that are correctly predicted.

(1) AUC

The AUC (Area Under the ROC Curve) [61] can be interpreted as the probability that randomly selected links from E^P have higher similarity scores than links randomly selected from $\{U - E\}$. It can be calculated as follows:

$$AUC = \frac{n' + 0.5n''}{n} \tag{20}$$

specifically, the similarity scores from E^P and $\{U - E\}$ are calculated by different models and we compare them n times in total. n' represents the number of times that the links from E^P have higher similarity scores than links from $\{U - E\}$, n'' denotes the number of times that two links have the same score.

(2) Precision

Compared to AUC, Precision [62] measures the accuracy of a link prediction model in correctly identifying relevant or true positive links among the links it predicts. Specifically, by sorting the similarity scores from large to small and selecting the top L links as missing links, if there are m links present in E^P, then we can define Precision as follows:

$$Precision = \frac{m}{L} \tag{21}$$

4.3 Datasets

To verify the performance of different algorithms, we selected 12 real networks from various fields as datasets. Table 2 shows the basic topological features of these datasets, where $|V|$ and $|E|$ denote the number of nodes and edges in a network, respectively; $\langle k \rangle$ is the average degree; CC is the clustering coefficient; $\langle d \rangle$ represents the average distance between two nodes.

Table 2 The basic topological features of 12 real-networks

Network	$\|V\|$	$\|E\|$	$\langle k \rangle$	CC	$\langle d \rangle$
FWB	128	2106	32.91	0.33	1.77
Email	167	3250	38.92	0.59	1.97
Celegans	297	2345	14.46	0.29	2.46
UTM	300	2191	14.61	0.57	3.76
SmaGri	1024	4916	9.60	0.31	2.98
PB	1222	16714	27.36	0.32	2.74
Friend	1788	12476	13.96	0.14	3.45
YeastL	2224	6609	5.94	0.14	4.38
EPA	4253	8897	4.18	0.07	4.50
Router	5022	6258	2.49	0.01	6.45
Reactome	5973	145778	48.81	0.61	4.21
HTC	7610	15751	4.14	0.49	5.68

The specific description of these networks are follows: (a) FWB [63] is an ecological network used to describe predation relationships between species in Florida Bay, where nodes characterize the species and directed edges denote the predation relationships between them. (b) Email [64] is an electronic mail communication network among employees. (c) Celegans [65] is a neural network of C.elegans, where each node represents a neuron and edges denote the synaptic connection between neurons. (d) UTM [66] is a uedge test matrix. (e) SmaGri [63] is a citation network. (f) PB [67] is a directed network about American political blogs. (g) Friend [68] is a friendship network of pet hamster owners. (h) YeastL [63] is a biological network representing the protein-protein interactions in yeast. (i) EPA [63] is constructed by extending the 200 page response set to search engine queries. (j) Router [69] is an internet routing network, where nodes depict routers and edges symbolize the physical or logical connections between them. (k) Reactome [66] is a biological network that describe the metabolic relationships between proteins. (l) HTC [70] is a co-authorship network.

4.4 Parameter setting

Our methods mainly contain hyperparameters γ and η, which we set to 0.6 and 0.1 by default. It should be noted that these hyperparameters are not optimal. In addition, we also analyze the sensitivity of our methods to these two hyperparameters in the experimental section. The parameter R is set to 20.

As the NMF-based methods heavily relevant to the dimension k, we set the same k for NMF-based methods based on the size of networks. For DNMF-based approaches, the number of layers l and the dimension of each layer also need to be set. Specifically, the l is set to 3 and the dimensions of each layer we present in the following Table 3.

For LP and Katz baselines, we set the parameters α and β to 0.25 and 0.5, respectively.

Table 3 Parameter values about NMF/DNMF

Network	FWB	Email	Celegans	UTM	SmaGri	PB
l	64-32-16	64-32-16	64-32-16	64-32-16	64-32-16	64-32-16
k	25	25	25	25	25	25
Network	Friend	YeastL	EPA	Router	Reactome	HTC
l	64-32-32	64-32-32	64-32-32	64-32-32	64-32-32	64-32-32
k	70	70	70	70	70	70

5 Experiment

In this section, we comprehensively evaluate the performance of our methods and baselines by answering the following questions:

- **Q1** How do our methods compare to the baselines?
- **Q2** How do our methods and baselines perform in sparse data?
- **Q3** How sensitive are our methods to parameter γ, η and k?
- **Q4** whether the sub-networks are beneficial for capturing potential features?

5.1 Comparison with baselines

We compare the predictive performance of various methods by using AUC and Precision evaluation metrics on 12 networks. Tables 4 and 5 show the AUC and Precision results, respectively. Note that we partitioned 90% of the network as the training set, while the remaining 10% is used as the testing set. The best results are highlighted in bold, and the second-best are underlined. In addition, we also compare the relationship between mode performance and actual running time, and we have similar conclusions across all networks. To illustrate the problem, Fig. 3 only show the trade-off between predictive ability and actual running time on EPA network.

Overall, NMF-based and DNMF-based methods have better predictive performance than heuristic-based approaches. For NMF-A1 and NMF-D1, which eliminate noise in the network by randomly adding or removing some edges, their AUC or Precision results in some networks are superior to NMF, such as Email, EPA and Router. However, in most networks its predictive performance is not as good as NMF, because the added or removed edges are random, which may add new noise to the network. Since our methods, EG-DNMF and EG-FDNMF, selectively add edges to the network, they help alleviate the problem of introducing new noise. Based on the AUC results, our methods achieve the best or second-best performance on 11 networks, and they attain both the best and

Table 4 The AUC results on 12 networks

Network	FWB	Email	Celegans	UTM	SmaGri	PB
EG-DNMF	0.8967	0.9156	0.8846	0.9785	0.9706	0.9423
EG-FDNMF	0.9322	0.9232	0.8849	0.9770	0.8770	0.9420
DNMF	0.8944	0.8923	0.7979	0.9756	0.8201	0.9053
DANMF	0.7712	0.8284	0.7994	0.9558	0.8028	0.8996
FSSDNMF	0.8985	0.9184	0.8431	0.9645	0.8695	0.9359
NMF	0.9387	0.9081	0.8482	0.9778	0.8523	0.9279
NMF-A1	0.8354	0.8591	0.8065	0.9657	0.8257	0.9232
NMF-D1	0.8761	0.8852	0.8100	0.9621	0.8451	0.9220
CN	0.6033	0.9138	0.8425	0.9735	0.8342	0.9196
AA	0.6048	0.9159	0.8588	0.9765	0.8441	0.9226
RA	0.6081	0.9198	0.8630	0.9784	0.8444	0.9239
LP	0.7488	0.9044	0.7594	0.9167	0.8281	0.9129
Katz	0.6736	0.9118	0.8567	0.9731	0.8818	0.9239
Cos	0.6516	0.9012	0.8541	0.9691	0.8636	0.9175
DGLP	0.6040	0.9041	0.7663	0.9633	0.8429	0.9123
Network	Friend	YeastL	EPA	Router	Reactome	HTC
EG-DNMF	0.9130	0.8163	0.9072	0.5726	0.9913	0.9016
EG-FDNMF	0.9116	0.8189	0.9106	0.5732	0.9921	0.9023

Table 4 continued

DNMF	0.9103	0.7616	0.8770	0.5826	0.9916	0.8812
DANMF	0.8704	0.7660	0.8645	0.5324	0.9725	0.8726
FSSDNMF	0.8743	0.7661	0.8954	0.5734	0.9903	0.9003
NMF	0.9117	0.7840	0.8943	0.6042	0.9832	0.8990
NMF-A1	0.9099	0.8006	0.8872	0.5776	0.9820	0.9012
NMF-D1	0.8986	0.7721	0.8986	0.5803	0.9906	0.8998
CN	0.8049	0.6913	0.6073	0.5578	0.9912	0.8887
AA	0.8075	0.6917	0.6088	0.5580	0.9919	0.8892
RA	0.8082	0.6915	0.6087	0.5582	0.9924	0.8893
LP	0.8755	0.7902	0.8792	0.6298	0.9904	0.9042
Katz	0.9118	0.7938	0.8996	0.3984	0.5026	0.8830
Cos	0.9481	0.8169	0.8892	0.7129	0.9920	0.7630
DGLP	0.8628	0.7884	0.6128	0.5602	0.9913	0.8841

Table 5 The Precision results on 12 networks

Network	FWB	Email	Celegans	UTM	SmaGri	PB
EG-DNMF	0.3045	0.4334	0.1507	0.3474	0.1712	0.1873
EG-FDNMF	0.4489	0.4415	0.1505	0.2527	0.1745	0.2191
DNMF	0.4351	0.3786	0.1332	0.2340	0.0996	0.1586
DANMF	0.1865	0.2269	0.0886	0.1497	0.0772	0.1688
FSSDNMF	0.4469	0.4126	0.1272	0.1994	0.1115	0.1909
NMF	0.5325	0.3890	0.1344	0.2440	0.1280	0.1939
NMF-A1	0.2704	0.3002	0.1293	0.2259	0.0996	0.1969
NMF-D1	0.3306	0.2132	0.0847	0.1529	0.0467	0.1652
CN	0.0813	0.4165	0.1114	0.2448	0.1504	0.1712
AA	0.0777	0.4068	0.1021	0.3016	0.1240	0.1598
RA	0.0768	0.4143	0.1237	0.3457	0.1138	0.1430
LP	0.2187	0.3562	0.0951	0.1118	0.0772	0.1496
Katz	0.1047	0.3823	0.0965	0.2652	0.1718	0.1628
Cos	0.0299	0.3446	0.0635	0.1741	0.0183	0.1053
DGLP	0.0531	0.3689	0.0266	0.2172	0.0139	0.0879
Network	Friend	YeastL	EPA	Router	Reactome	HTC
EG-DNMF	0.3139	0.1278	0.1041	0.1604	0.8130	0.1413
EG-FDNMF	0.3172	0.1108	0.1043	0.1606	0.8131	0.1401
DNMF	0.3181	0.0953	0.0918	0.1492	0.8012	0.1302
DANMF	0.1554	0.1135	0.0890	0.1487	0.8007	0.1185
FSSDNMF	0.1731	0.0893	0.0972	0.1506	0.8098	0.1432
NMF	0.2965	0.1180	0.0916	0.1499	0.8087	0.1103
NMF-A1	0.2981	0.1165	0.1012	0.1512	0.8125	0.1308
NMF-D1	0.2340	0.0802	0.1008	0.1506	0.8112	0.1201
CN	0.0465	0.1286	0.0117	0.0182	0.2479	0.0502
AA	0.0377	0.1001	0.0204	0.0168	0.2587	0.0624
RA	0.0313	0.0787	0.0246	0.0082	0.3997	0.0618
LP	0.0801	0.0681	0.0928	0.1249	0.4481	0.0486
Katz	0.0609	0.0999	0.0152	0.0256	0.0670	0.0487
Cos	0.0024	0.1082	0.0051	0.0002	0.1598	0.0002
DGLP	0.0096	0.0193	0.0120	0.0193	0.2598	0.0510

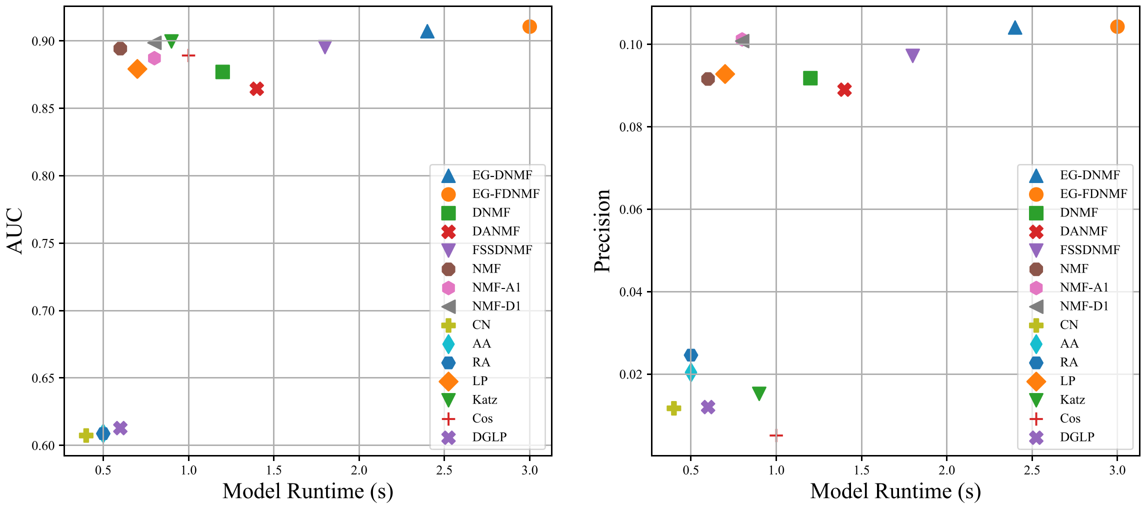


Fig. 3 Predict performance vs. Actual runtime on EPA network

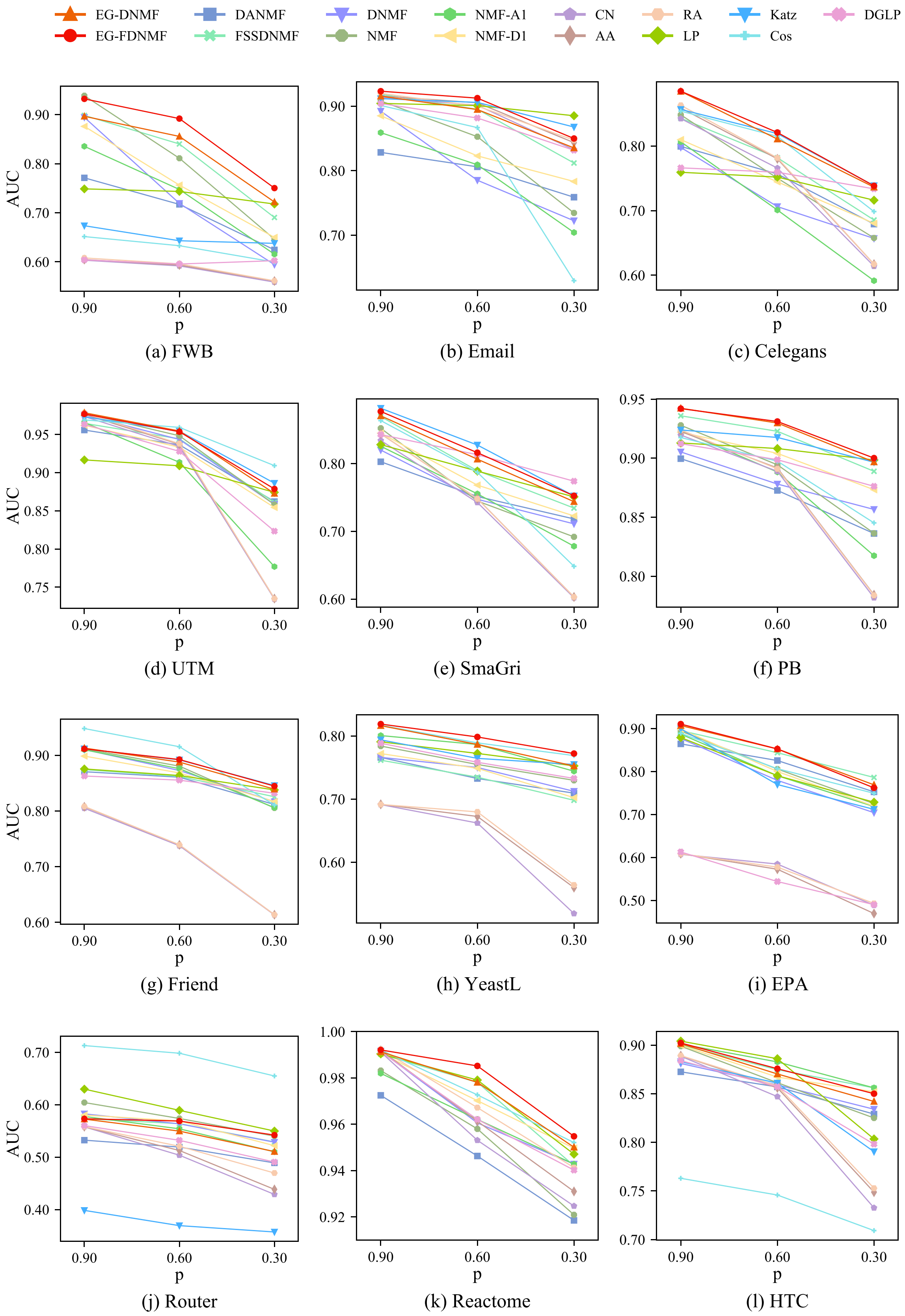
second-best results on Celegans and EPA. This demonstrates that our methods effectively uncover potential topology to enhance the network, thereby supplying valuable information for downstream link prediction tasks. For DNMF-based methods, because FSSDNMF can combine topological information and observed links to explore the underlying features in the network, its AUC and Precision results are superior to DNMF in most networks. Considering that EG-FDNMF utilize the feature matrix obtained through the FSSDNMF method, it outperforms EG-DNMF with four best and five second-best AUC results, while EG-DNMF only achieves two best and three second-best AUC results. On the other hand, the DANMF, primarily designed for community detection, does not perform well in link prediction tasks. These observations are consistent with the Precision results, which focus on prediction accuracy. In the case of Prediction, both EG-DNMF and EG-FDNMF secure the best or second-best results across all networks. This showcases the ability of our model to introduce valuable links into the network, thereby enhancing predictive accuracy.

The predictive performance of heuristic-based methods may not be strong as matrix factorization-based approaches in general. However, they still demonstrate competitive results in certain networks. For example, Katz achieves the highest AUC result of 0.8818 on SmaGri, while RA secures the second-best Precision result of 0.3457 on UTM, highlighting their effectiveness in special contexts. It is evident that the global-based methods, namely Katz and Cos, perform well by achieving two best results and two second-best result in terms of the AUC metric. On the other hand, the quasi-based methods have less widespread success, with only LP obtaining the best AUC result on HTC and the second-best result on Router. The local-based approaches, including CN, AA, RA, manage to secure only two second-best AUC results on Email and UTM. This discrepancy arises from the fact that Katz and Cos utilize the complete network topology to predict missing links, while CN, AA and RA rely solely on local information surrounding the target node pairs. In contrast, LP and DGLP leverage structural information encompassing both the entire network and local topology. Furthermore, it is worth noting that, under the Prediction metric, CN and RA achieve the top results in YeastL and HTC, respectively.

The actual running time of each method on EPA in Fig. 3 reveals that heuristic methods have minimal time consumption as they calculate similarity scores between node pairs based on empirical assumption. Among these methods, CN, AA, and RA, which focus on the local structure around nodes, require the least amount of time to compute. Quasi-based methods follow in terms of time consumption. Katz and Cos, on the other hand, need to consider the global topology of the network, leading to relatively longer times. Compared to heuristics-based approaches, matrix factorization-based methods require more running time because they need to update parameters iteratively. Particularly, DNMF-based approaches involve multiple layers of parameter, whereas NMF only requires updating a single layer of parameters, resulting in significantly reduced time consumption. Our methods are more complex compared to other baselines. This complexity arises from the necessity to partition the network into three sub-networks and apply the DNMF models to explore underlying features within each sub-network. Reducing this complexity will be the focus of our future work.

5.2 Robustness analysis

In order to checking the robustness of the these methods, we set the proportion of training set p to 0.9, 0.6 and 0.3. The results are shown in Figs. 4 and 5. Overall, as the proportion



(a) FWB

(b) Email

(c) Celegans

(d) UTM

(e) SmaGri

(f) PB

(g) Friend

(h) YeastL

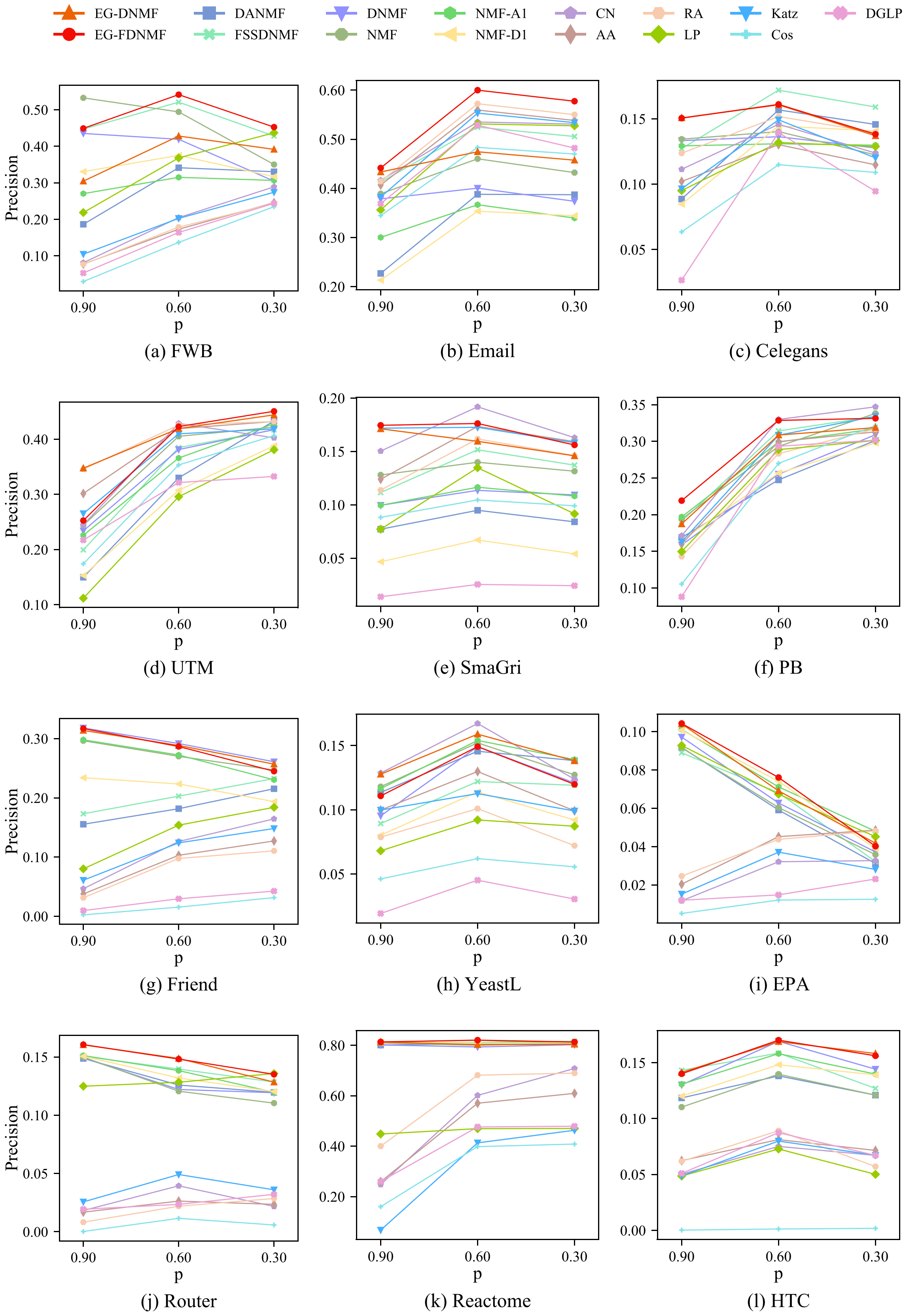
(i) EPA

(j) Router

(k) Reactome

(l) HTC

Fig. 4 AUC results under different training set ratios *p*



(a) FWB

(b) Email

(c) Celegans

(d) UTM

(e) SmaGri

(f) PB

(g) Friend

(h) YeastL

(i) EPA

(j) Router

(k) Reactome

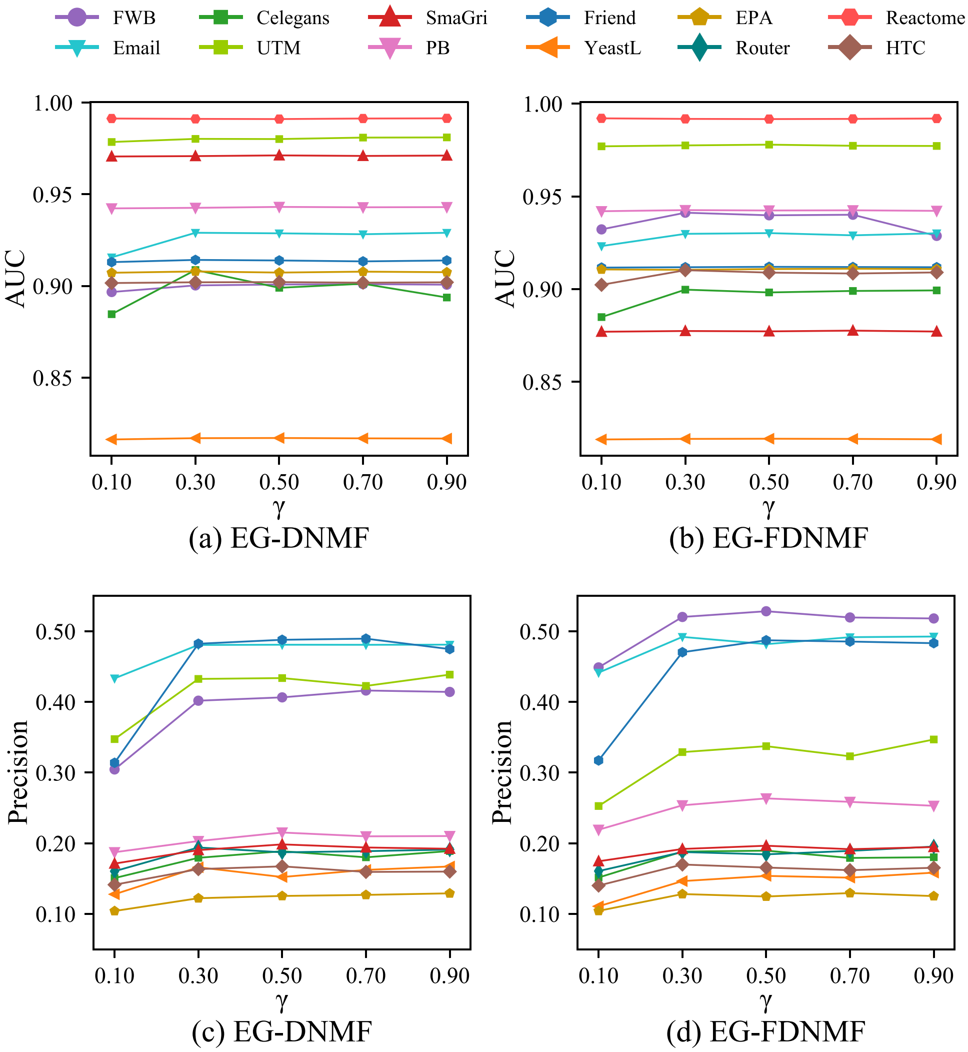
(l) HTC

Fig. 5 Precision results under different training set ratios p

of training set p decreased, the AUC performance of all methods shows a significant decrease, and the trend is particularly noticeable in Celegans, UTM, and Reactome. Specifically, this downward trend is most pronounced in heuristic-based methods, especially when p drops from 0.6 to 0.3, such as in networks Friend, YeastL and EPA, etc. This is because they rely on the network's topology to compute similarity scores. Therefore, when the network's structure is compromised, the available information becomes quiet limited. For matrix factorization methods, they can learn the underlying connection mechanisms in a network, although the AUC results are degraded with the reduction of training samples, they outperform heuristics-based methods in most networks. Furthermore, it is worth noting that FSSDNMF performs the best among these methods. For example, on FWB, EPA and PB, it excels because it incorporates observed links and topological features. Our methods, EG-DNMF and EG-FDNMF, are capable of adding valuable links to the network and enhancing the available topology information. This, in turn, effectively mitigates the network's sparsity issue. It can be seen that in almost all networks, our methods show excellent performance and achieve the best AUC result in FWB, Celegans, PB, YeastL and Reactome. However, for the Precision metric, when the value of p drops to 0.6, the prediction accuracy of most methods increases and then decreases, such as on network YeastL and SmaGri, etc. Compared to other methods, although the Precision results of EG-DNMF and EG-FDNMF also decrease, their prediction performance surpasses that of other baselines in most networks, such as on FWB, Email, UTM, and EPA, etc.

5.3 Parameter sensitivity analysis

As EG-DNMF and EG-FDNMF require three hyperparameters (γ, η and the dimension of latent space k), we explore the effect of different hyperparameters to our model performance including Precision and AUC. Section 5.3.1 demonstrates the



(a) EG-DNMF

(b) EG-FDNMF

(c) EG-DNMF

(d) EG-FDNMF

Fig. 6 The impact of different γ to the EG-DNMF and EG-FDNMF methods on 12 networks

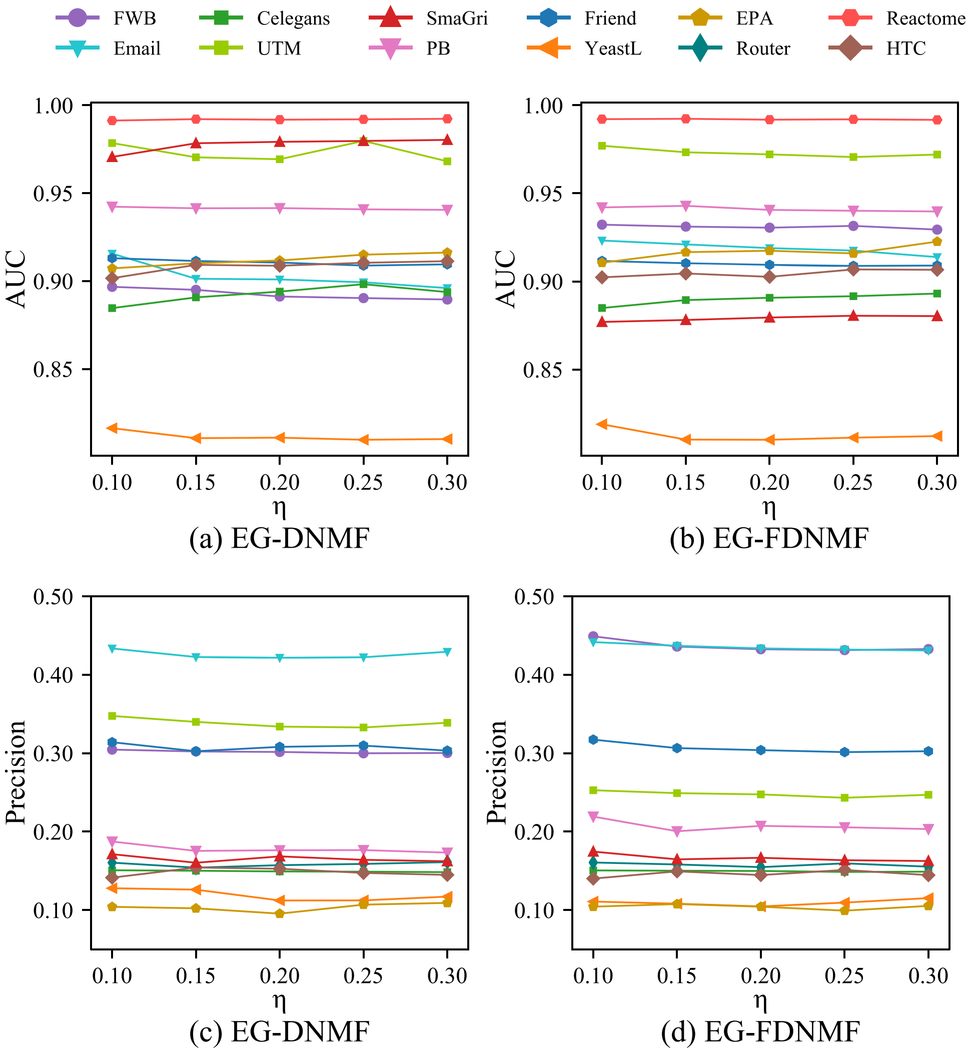
impact of γ and the result of η is in Sections 5.3.2 and 5.3.3 demonstrates the impact of k.

5.3.1 Impact of γ

Figure 6 illustrates the effect of the γ parameter on the performance of EG-DNMF and EG-FDNMF. In the context of network reconstruction, the γ is utilized to regulate the number of edges within the candidate edges set. Here, we set the proportion γ to the values $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ while maintaining a constant η of 0.1. We have omitted the AUC result for the Router dataset as it falls below 0.6 and does not contribute positively to the overall presentation of the results, similarly, the Precision result for Reactome dataset is not shown. From the Precision results, the performance of EG-DNMF and EG-FDNMF improve slightly as γ increases. Specifically, when γ is set to 0.3, the performance of our methods improve rapidly, with Friend, UTM, FWB and Email showing the most significant improvements. The reason is that when $\gamma < 0.3$, the proportion of useful edges is small, which can significantly reduce the performance of the link prediction algorithm. We can see that when γ is greater than 0.3, our methods tend to be stable. Similar conclusions were also observed in the AUC results, indicating that the optimal γ for our methods is 0.3 in most networks. However, it should be noted that our method is insensitive to the γ parameter on EPA and PB, which performance remain stable for different γ values.

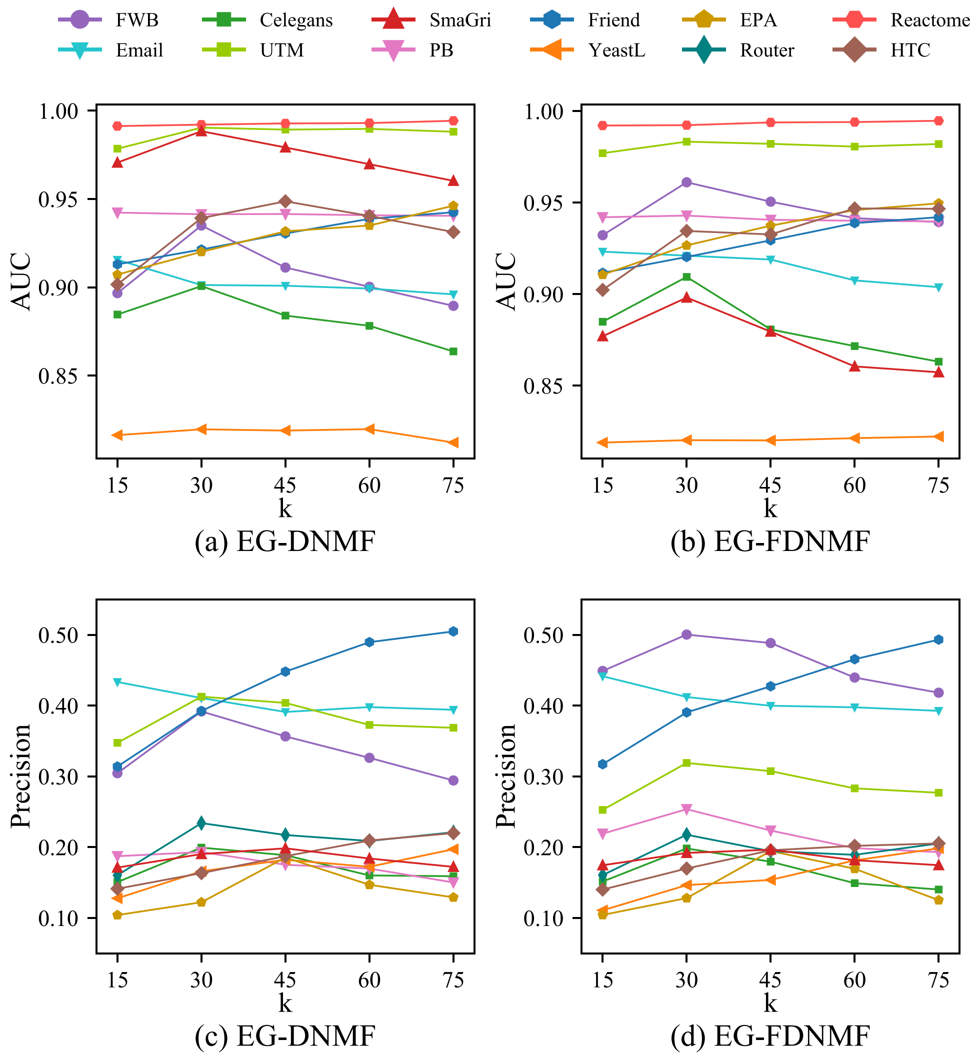
5.3.2 Impact of η

The results of η are presented in Fig. 7, indicating that we utilize the η parameter to regulate the number of edges added to the network, with the ultimate selection being made from the candidate edges set. In the previous section, it was observed that our methods stabilize when γ exceeds 0.3. Therefore, in this section, we keep γ fixed at 0.6 and vary the proportion of η, using values of $\{0.10, 0.15, 0.20, 0.25, 0.30\}$. We have



(a) EG-DNMF
(b) EG-FDNMF
(c) EG-DNMF
(d) EG-FDNMF

Fig. 7 The impact of different η to the EG-DNMF and EG-FDNMF methods on 12 networks



(a) EG-DNMF

(b) EG-FDNMF

(c) EG-DNMF

(d) EG-FDNMF

Fig. 8 The impact of different *k* to the EG-DNMF and EG-FDNMF methods on 12 networks

chosen not to display the AUC result for the Router dataset, as it falls below 0.6 and does not contribute positively to the overall presentation of the results, similarly, the Precision result of Reactome is not shown.

In general, increasing the value of η tend to result in a decrease in Precision results for both EG-DNMF and EG-FDNMF. Furthermore, concerning AUC results, when η is increased, indicating the addition of more edges to the network, EG-DNMF exhibits instability, especially in Email, UTM, and Celegans networks. This phenomenon occurs because an excessive addition of edges to the network can disrupt its structural integrity.

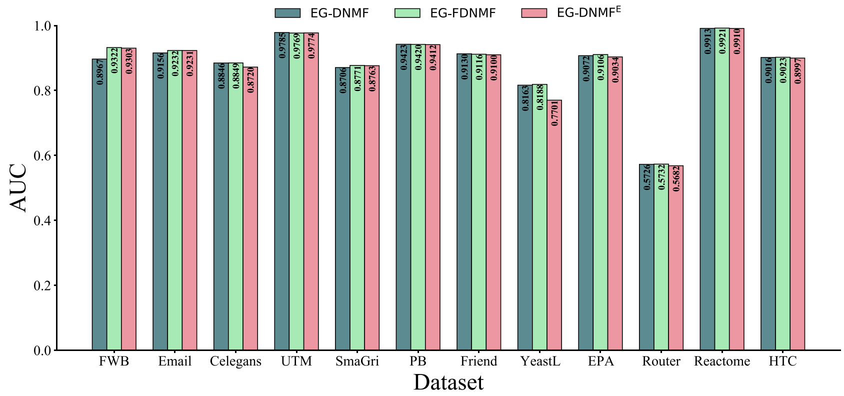
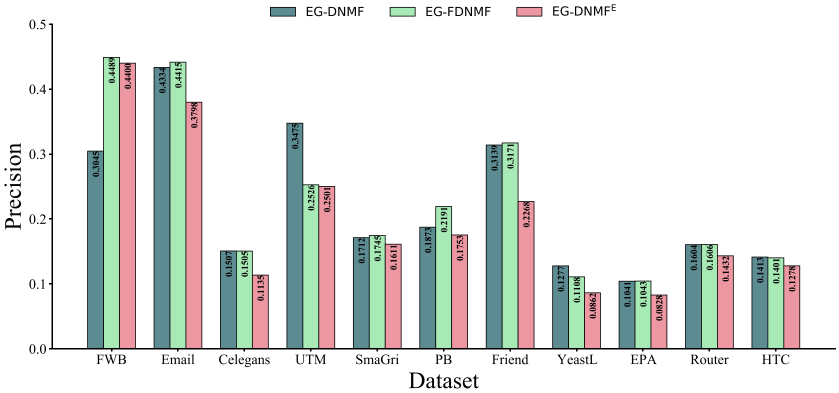


Fig. 9 AUC results under different ways of reconstructing the network

Fig. 10 Precision results under different ways of reconstructing the network



In contrast, EG-FDNMF remains notably stable across most networks because it effectively combines the network's topological characteristics, mitigating the disruptive effects of excessive edge additions.

5.3.3 Impact of *k*

We conducted an extensive series of experiments on 12 networks to assess the performance of our method under various k value. In this experiment, we maintained γ at 0.6, η at 0.1, and varied k within the range of $\{15, 30, 45, 60, 75\}$ to examine the influence of k. The results are presented in Fig. 8. We also did not show the AUC result for Router dataset and Precision result for Reactome dataset. Generally, the AUC performance of both EG-DNMF and EG-FDNMF on each network exhibits an initial increases with higher k values. However, as k reaches a certain point, the performance tends to plateau or even decline. This phenomenon is primarily attributed to overfitting, which occurs due to the increased dimensionality of the data. The optimal k value is proportionate to the size of the network. For instance, a network like UTM with 300 nodes achieves its optimal k at 30, with a larger network like EPA with 4253 nodes reaches its optimal k at 75. Nevertheless, it's worth noting that as k increases, the training time of the model also increase. Hence, the challenge lies in determining the optimal k that maintains high performance while keeping time complexity in check, which is an further research.

5.4 Sub-network or entire-network ?

To confirm whether dividing the original network into the three sub-networks enhances the capture of features in a more fine-grained manner, we reconstructed the original network using the DNMF models across the entire network (rather than sub-networks), and denoted this method as EG-DNMF^E. Figures 9 and 10 show the AUC and Precision results compared with our methods, respectively. We omitted the Precision results for the Reactome dataset as it higher than 0.8.

From the AUC results, which assess the overall performance of the model, although their results are very similar, it can be observed numerically that the performance of our model reconstructed within the sub-networks is slightly superior to the performance within the entire network. For Precision result, in general, the predictive performance of EG-DNMF^E is inferior to our methods in all networks. Specially, in the case of EG-DNMF and EG-FDNMF, the latter outperforms the former on FWB, Email, and PB. This is because the FSSDNMF method we employ demonstrates a superior capability to explore potential features in the network compared to the DNMF algorithm. However, the opposite scenario is also observed, with EG-DNMF achieving higher results on UTM and YeastL. Then for EG-DNMF^E, we can see that its predictive performance in almost all networks is not as good as our methods, especially in Email, Celegans, PB and Friend. This demonstrates the effectiveness of our approach in dividing the entire network into three sub-networks, allowing for the capture of more localized structural features. Consequently, the edges added to the network serve as valuable information for downstream link prediction tasks.

6 Conclusion

In this paper, we propose a novel link prediction methods EG-DNMF and EG-FDNMF based on DNMF. How to add some useful topology information to the network is the research motivation of this paper. Specially, we first divide the original network into three sub-networks, this divide-and-conquer idea can also be applied to other fields. Then use the DNMF algorithm to guide the process of network reconstruction. Experiments show that our methods can indeed add some useful link information to the network and improve the performance of link prediction. Furthermore, we have also verified that the division of sub-networks is beneficial for extracting hidden features within the network.

Nevertheless, the high complexity of this approach is a limitation that we aim to address in our future research. Our objectives include finding ways to reduce this complexity and exploring more effective methods for network reconstruction, which are interesting avenues for future work.

Acknowledgements This work was partly supported by the National Natural Science Foundation of China (No. 62366030, 62062010), the Gansu Provincial Natural Science Foundation (No. 23JRRA8222), the Science and Technology Planning Project of Guangxi (No. AD19245101), the Higher Education Innovation Fund project of Gansu (No. 2022A-022). Additionally, during the revision of this paper, we would like to express our sincere gratitude to Tongfeng Li and Ning Ma for their invaluable experimental support and writing guidance.

Author Contributions Yabing Yao: Methodology and Project administration. Yangyang He: Data analysis and Writing. Zhentian Huang: Software and Conducting experiment. Zhipeng Xu: Data curation and Graphing. Fan Yang: Conceptualization and Project administration. Jianxin Tang: Data collection. Kai Gao: Device support.

Funding This work is supported by the following funding sources: the National Natural Science Foundation of China (No. 62366030, 62062010), the Gansu Provincial Natural Science Foundation (No. 23JRRA8222), the Science and Technology Planning Project of Guangxi (No. AD19245101), the Higher Education Innovation Fund project of Gansu (No. 2022A-022).

Data availability and access The dataset analyzed during the experiments conducted in this paper can be obtained at the following URL: http://konect.uni-koblenz.de/ and http://networkrepository.com/networks.php.

Declarations

Ethical and informed consent for data used In our research, we emphasize the importance of ethical practices and informed consent in the acquisition and utilization of data. Firstly, we obtain the dataset used in our study following strict ethical guidelines. The data is sourced from http://konect.uni-koblenz.de/ and http://networkrepository.com/networks.php, and we ensure that the data is anonymized and stripped of any personally identifiable information to protect the privacy and confidentiality of the individuals or entities involved. Furthermore, we highlight the significance of informed consent in the process of data usage. As the data used in our study is pre-existing and publicly available, we adhere to the ethical standards and legal requirements set forth by the data source. It is important to emphasize that our research aligns with the principles of research ethics and integrity, complying with all regulations and guidelines related to data acquisition, storage, and usage. We acknowledge the importance of responsible data handling and are committed to maintaining the confidentiality and privacy of the data subjects.

Conflict of interest We declare that there are no conflicts of interest in this work. Throughout the research process, we have not encountered any conflicts of interest that could have influenced the research outcomes. We are committed to upholding the highest standards of research ethics and integrity.

References

1. Wahid-Ul-Ashraf A, Budka M, Musial K (2019) How to predict social relationships-physics-inspired approach to link prediction. Physica A 523:1110–1129
2. Yao Y, Cheng T, Li X, He Y, Yang F, Li T, Liu Z, Xu Z (2023) Link prediction based on the mutual information with high-order clustering structure of nodes in complex networks. Physica A 610:128428
3. Zhou T (2021) Progresses and challenges in link prediction. Iscience 24(11):103217
4. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. Physica A 390(6):1150–1170
5. Li S, Song X, Lu H, Zeng L, Shi M, Liu F (2020) Friend recommendation for cross marketing in online brand community based on intelligent attention allocation link prediction algorithm. Expert Syst Appl 139:112839
6. Su Z, Zheng X, Ai J, Shen Y, Zhang X (2020) Link prediction in recommender systems based on vector similarity. Physica A 560:125154
7. Liu G (2022) An ecommerce recommendation algorithm based on link prediction. Alex Eng J 61(1):905–910
8. Nasiri E, Berahmand K, Rostami M, Dabiri M (2021) A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. Comput Biol Med 137:104772
9. Li Z, Zhu S, Shao B, Zeng X, Wang T, Liu T-Y (2023) Dsn-ddi: an accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning. Briefings in Bioinformatics 24(1)
10. Kumar A, Singh SS, Singh K, Biswas B (2020) Link prediction techniques, applications, and performance: A survey. Physica A 553:124289
11. Chen G, Wang H, Fang Y, Jiang L (2022) Link prediction by deep non-negative matrix factorization. Expert Syst Appl 188:115991
12. Daud NN, Ab Hamid SH, Saadoon M, Sahran F, Anuar NB (2020) Applications of link prediction in social networks: A review. J Netw Comput Appl 166:102716
13. Newman ME (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64(2):025102
14. Adamic LA, Adar E (2003) Friends and neighbors on the web. Social networks 25(3):211–230
15. Liu S, Ji X, Liu C, Bai Y (2017) Extended resource allocation index for link prediction of complex network. Physica A 479:174–183
16. Vural H, Kaya M (2018) Prediction of new potential associations between lncrnas and environmental factors based on katz measure. Comput Biol Med 102:120–125
17. Liu W, Lü L (2010) Link prediction based on local random walk. Europhys Lett 89(5):58007
18. Zhou Y, Wu C, Tan L (2021) Biased random walk with restart for link prediction with graph embedding method. Physica A 570:125783
19. Aziz F, Gul H, Muhammad I, Uddin I (2020) Link prediction using node information on local paths. Physica A 557:124980
20. Rafiee S, Salavati C, Abdollahpouri A (2020) Cndp: Link prediction based on common neighbors degree penalization. Physica A 539:122950
21. Clauset A, Moore C, Newman ME (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453(7191):98–101
22. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. Proc Natl Acad Sci 106(52):22073–22078
23. Zhou J, Liu L, Wei W, Fan J (2022) Network representation learning: from preprocessing, feature extraction to node embedding. ACM Computing Surveys (CSUR) 55(2):1–35

24. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 701–710
25. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining, pp 855–864
26. Lei K, Qin M, Bai B, Zhang G, Yang M (2019) Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. In: IEEE INFOCOM 2019-IEEE conference on computer communications, IEEE pp 388–396
27. Hao Y, Cao X, Fang Y, Xie X, Wang S (2021) Inductive link prediction for nodes having only attribute information. In: Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pp 1209–1215
28. Samy AE, Kefato TZ, Girdzijauskas S (2023) Graph2feat: Inductive link prediction via knowledge distillation. Companion Proceedings of the ACM Web Conference 2023:805–812
29. Wu E, Cui H, Chen Z (2022) Relpnet: Relation-based link prediction neural network. In: Proceedings of the 31st ACM International conference on information & knowledge management, pp 2138–2147
30. Guo Z, Shiao W, Zhang S, Liu Y, Chawla NV, Shah N, Zhao T (2023) Linkless link prediction via relational distillation. In: International conference on machine learning, PMLR pp 12012–12033
31. Zhao Z, Gou Z, Du Y, Ma J, Li T, Zhang R (2022) A novel link prediction algorithm based on inductive matrix completion. Expert Syst Appl 188:116033
32. Wang W, Cai F, Jiao P, Pan L (2016) A perturbation-based framework for link prediction via non-negative matrix factorization. Sci Rep 6(1):1–11
33. Chen G, Xu C, Wang J, Feng J, Feng J (2020) Robust non-negative matrix factorization for link prediction in complex networks using manifold regularization and sparse learning. Physica A 539:122882
34. Lei K, Qin M, Bai B, Zhang G (2018) Adaptive multiple non-negative matrix factorization for temporal link prediction in dynamic networks. In: Proceedings of the 2018 workshop on network meets AI & ML, pp 28–34
35. Zhao Y, Wang H, Pei J (2019) Deep non-negative matrix factorization architecture based on underlying basis images learning. IEEE Trans Pattern Anal Mach Intell 43(6):1897–1913
36. Chen W-S, Zeng Q, Pan B (2022) A survey of deep nonnegative matrix factorization. Neurocomputing 491:305–320
37. Ye F, Chen C, Zheng Z (2018) Deep autoencoder-like nonnegative matrix factorization for community detection. In: Proceedings of the 27th ACM international conference on information and knowledge management, pp 1393–1402
38. Zhang W, Zhang X, Wang H, Chen D (2019) A deep variational matrix factorization method for recommendation on large scale sparse dataset. Neurocomputing 334:206–218
39. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on World Wide Web, pp 173–182
40. Luo L, Xie H, Rao Y, Wang FL (2019) Personalized recommendation by matrix co-factorization with tags and time information. expert systems with applications 119:311–321
41. Bhowmick AK, Meneni K, Danisch M, Guillaume J-L, Mitra B (2020) Louvainne: Hierarchical louvain method for high quality and scalable network embedding. In: Proceedings of the 13th international conference on web search and data mining, pp 43–51
42. Zhao S, Du Z, Chen J, Zhang Y, Tang J, Yu P (2021) Hierarchical representation learning for attributed networks. IEEE Transactions on Knowledge and Data Engineering
43. Wang Y, Zhao Y (2023) Arbitrary spatial trajectory reconstruction based on a single inertial sensor. IEEE Sensors Journal
44. Zhu Z, Huang G, Deng J, Ye Y, Huang J, Chen X, Zhu J, Yang T, Du D, Lu J et al (2022) Webface260m: A benchmark for million-scale deep face recognition. IEEE Trans Pattern Anal Mach Intell 45(2):2627–2644
45. Yuliansyah H, Othman Z, Bakar AA (2023) A new link prediction method to alleviate the cold-start problem based on extending common neighbor and degree centrality. Physica A 616:128546
46. Stanley N, Bonacci T, Kwitt R, Niethammer M, Mucha PJ (2019) Stochastic block models with multiple continuous attributes. Applied Netw Sci 4(1):1–22
47. Kuang J, Scoglio C (2021) Layer reconstruction and missing link prediction of a multilayer network with maximum a posteriori estimation. Phys Rev E 104(2):024301
48. Zhao H, Du L, Buntine W (2017) Leveraging node attributes for incomplete relational data. In: International Conference on Machine Learning, PMLR pp 4072–4081
49. Makarov I, Kiselev D, Nikitinsky N, Subelj L (2021) Survey on graph embeddings and their applications to machine learning problems on graphs. PeerJ Comput Sci 7:357
50. Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2023) Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Comput Surv 55(9):1–35
51. Zhang M, Chen Y (2017) Weisfeiler-lehman neural machine for link prediction. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 575–583
52. Zhang M, Chen Y (2018) Link prediction based on graph neural networks. Advances in neural information processing systems 31
53. Wang Z, Lei Y, Li W (2020) Neighborhood attention networks with adversarial learning for link prediction. IEEE Trans Neural Netw Learn Syst 32(8):3653–3663
54. Wang Z, Li W, Su H (2021) Hierarchical attention link prediction neural network. Knowl-Based Syst 232:107431
55. Qin M, Zhang C, Bai B, Zhang G, Yeung D-Y (2023) High-quality temporal link prediction for weighted dynamic graphs via inductive embedding aggregation. IEEE Transactions on Knowledge and Data Engineering
56. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):30–37
57. Ahmed NM, Chen L, Wang Y, Li B, Li Y, Liu W (2018) Deepeye: link prediction in dynamic networks based on non-negative matrix factorization. Big Data Mining and Analytics 1(1):19–33
58. Liang J, Gurukar S, Parthasarathy S (2021) Mile: A multi-level framework for scalable graph embedding. Proceedings of the International AAAI Conference on Web and Social Media 15:361-372
59. Chen Z, Shi Y, Qi Z (2019) Constrained matrix factorization for semi-weakly learning with label proportions. Pattern Recogn 91:13–24
60. Varikuti DP, Genon S, Sotiras A, Schwender H, Hoffstaedter F, Patil KR, Jockwitz C, Caspers S, Moebus S, Amunts K et al (2018) Evaluation of non-negative matrix factorization of grey matter in age prediction. Neuroimage 173:394–410
61. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143(1):29–36
62. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst (TOIS) 22(1):5–53
63. Batagelj V, Mrvar, A (2014) Pajek
64. De Winter S, Decuypere T, Mitrović S, Baesens B, De Weerdt J (2018) Combining temporal aspects of dynamic networks with node2vec for a more efficient dynamic link prediction. In: 2018 IEEE/ACM International conference on advances in social networks analysis and mining (ASONAM), IEEE pp 1234–1241

65. White JG, Southgate E, Thomson JN, Brenner S et al (1986) The structure of the nervous system of the nematode caenorhabditis elegans. Philos Trans R Soc Lond B Biol Sci 314(1165):1-340
66. Rossi R, Ahmed N (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the AAAI conference on artificial intelligence, vol 29
67. Adamic LA, Glance N (2005) The political blogosphere and the 2004 us election: divided they blog. In: Proceedings of the 3rd international workshop on link discovery, pp 36–43
68. Jorgensen Z, Yu T, Cormode G (2016) Publishing attributed social graphs with formal privacy guarantees. In: Proceedings of the 2016 international conference on management of data, pp 107–122
69. Spring N, Mahajan R, Wetherall D (2002) Measuring isp topologies with rocketfuel. ACM SIGCOMM Comput Commun Rev 32(4):133–145
70. Martinez V, Berzal F, Cubero J-C (2019) Noesis: a framework for complex network data analysis. Complexity 2019:1–14

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yabing Yao is currently an associate professor at School of computer and communication, Lanzhou University of Technology. He received the Ph.D. degree in the School of Information Science and Engineering at Lanzhou University in 2017. His work has focused on link prediction in complex networks. His research interests include machine learning on graphs, network science.

Yangyang He received the bachelor's degree from the Chaohu University, Hefei, China, in 2020. He is currently pursuing the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China. His research interests include complex network analysis, link prediction and higher-order link prediction.

Zhentian Huang is a undergraduate student of Lanzhou University of Technology, Lanzhou, China. His research interests include link prediction and graph neural network.

Zhipeng Xu received the bachelor's degree from the Lanzhou University of Arts and Science, Lanzhou, China, in 2021. He is currently pursuing the M.S. degree with the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China. His research interests include link prediction and higher-order link prediction.

Fan Yang is currently an associate professor at School of Computer Science and Technology, Guangxi University of Science and Technology. He completed his Ph.D. degree in the School of Information Science and Engineering at Lanzhou University in 2017. He received his B.Eng. degree and Master degree from the School of Computer and Communication at the Lanzhou University of Technology 2007 and 2011. His personal interests include graph neural networks and complex networks, etc. He is a member of IEEE.

Jianxin Tang received the Ph.D. degree from the School of Information Science & Engineering, Lanzhou University, Lanzhou, China, in 2019. Since 2012, he has been with the School of Computer Science and Communication Department, Lanzhou University of Technology, Lanzhou, China, where he became an Associate Professor in 2021. His current interests include intelligent algorithm & optimization, social network analysis, social computing.

Kai Gao is currently a teacher of Network and Information Center in Lanzhou University of Technology, Lanzhou, China. He received his B.S. degree in school of computer and software from Nanjing University of Information Science and Technology in 2017. He received his M.S. degree in school of computer and communication from Lanzhou University of Technology in 2020.