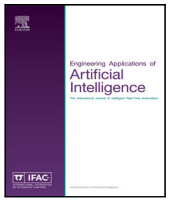
Engineering Applications of Artificial Intelligence 133 (2024) 108641

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Enhancing link prediction through adversarial training in deep Nonnegative Matrix Factorization

Reza Mahmoodi, Seyed Amjad Seyedi, Alireza Abdollahpouri, Fardin Akhlaghian Tab *

Computer Engineering Department, University of Kurdistan, Iran

ARTICLE INFO

Keywords:
Link prediction
Deep matrix factorization
Adversarial training
Generalization
Robustness

ABSTRACT

Link prediction is a fundamental problem in complex network analysis, aimed at predicting missing or forthcoming connections. Recent research investigates the potential of Nonnegative Matrix Factorization (NMF) models in reconstructing sparse networks and using deep NMF to uncover the hierarchical structure. Deep models have demonstrated remarkable performance in various domains, but they are susceptible to overfitting, especially on the limited training data. This paper proposes a novel Link Prediction using Adversarial Deep NMF (LPADNMF) to enhance the generalization of network reconstruction in sparse graphs. The main contribution is the introduction of an adversarial training that incorporates a bounded attack on the input, leveraging the $L_{2,1}$ norm to generate diverse perturbations. This adversarial training aims to improve the model's robustness and prevent overfitting, particularly in scenarios with limited training data. Additionally, the proposed method incorporates first and second-order affinities as input to capture higher-order dependencies and encourage the extraction of informative features from the network structure. To further mitigate overfitting, a smooth L_2 regularization is applied to the model parameters. To optimize the proposed model effectively, we utilize a majorization-minimization algorithm that efficiently updates the perturbation and latent factors in an iterative manner. Our findings demonstrate that the proposed model not only has the ability to uncover and learn complex structures but also possesses generalization capabilities. This method demonstrates superior performance compared to state-of-the-art methods across eight networks. In comparison to the best-performing approach, LPADNMF exhibited improvements of 2.36% in AUC, 4.67% in Precision, 2.21% in Recall, and 3.73% in F-Measure.

1. Introduction

Complex networks serve as a descriptive framework for a range of real-world complex interaction systems in nature and society, like social networks, biological neural networks, cooperation networks, citation networks, and protein interaction networks (Fortunato, 2010). Nodes in a complex network represent distinct entities, whereas connections between nodes signify their interactions or relationships. The study of network information mining and complex network analysis has grown in popularity as a result of the expansion of network data availability, and these techniques may delve into complicated systems and extract important information (Bai et al., 2019). Currently, link prediction stands as a popular area of research within the field of network structure, which aims to forecast the links that are not currently present or observable (Chai et al., 2022). Link prediction has many practical applications, such as recommending items in user-item networks (Xie et al., 2015), suggesting friends in social networks, detecting unknown interactions in protein–protein networks (Hu et al., 2017), and finding experts in academic networks (Pavlov and Ichise, 2007).

In the literature, several link prediction methods have been introduced (Lü and Zhou, 2011). Notably, similarity-based methods are more renowned among these algorithms, which predict links by adhering to the homophily principle, with assumption that nodes sharing similar properties are more inclined to be connected. These algorithms compute similarities between nodes that lack direct connections to make their predictions (Li et al., 2022). For this purpose, Common Neighbors (CN) (Newman, 2001) characterizes the similarity between two nodes by quantifying the number of common neighbors they share. Furthermore, Adamic-Adar (AA) (Adamic and Adar, 2003) and Resource Allocation (RA) (Ou et al., 2007) discern the varied contributions of distinct shared neighbors by assigning reduced weights to neighbors with larger degrees. There are also several path-based algorithms in addition to these neighbor-based ones, including the Relative-path-based algorithm (Li et al., 2020), Local Path index (Zhou

* Corresponding author.
E-mail addresses: reza.mahmoodi@uok.ac.ir (R. Mahmoodi), amjadseyedi@uok.ac.ir (S.A. Seyedi), abdollahpouri@uok.ac.ir (A. Abdollahpouri), f.akhlaghian@uok.ac.ir (F. Akhlaghian Tab).

https://doi.org/10.1016/j.engappai.2024.108641
Received 20 January 2024; Received in revised form 1 May 2024; Accepted 15 May 2024

et al., 2009), and Significance of Higher-Order Path Index (Kumar et al., 2020). Furthermore, other approaches have been developed to predict links, such as maximum likelihood and probabilistic modeling techniques. For example, probabilistic entity-relationship models (Getoor and Taskar, 2007), stochastic relational models (Yu et al., 2006), probabilistic relational models (Getoor et al., 2001), hierarchical structure models (Clauset et al., 2008), and stochastic block models (Guimerà et al., 2007) are employed to estimate network parameters. These inferred parameters can be subsequently employed to evaluate the probability of a connection between any given node pairs.

Due to the tight relationship of a network structure with a matrix (i.e., the adjacency matrix), link prediction research frequently uses matrix theory. In the initial discoveries of link prediction (Zhou et al., 2009), researchers utilize the quadratic power of the adjacency matrix or perform direct matrix operations to calculate the similarity scores of node pairs. Some researchers have reformulated link prediction as a matrix completion problem, which involves completing a matrix with additional information to enhance its coherence. This approach has gained popularity, and many scholars have combined matrix decomposition theory with link prediction, which has further advanced research in this area (Ma et al., 2017), Ahmed et al. (2018). An alternative method to integrate matrix theory into link prediction involves employing the low-rank representation of the adjacency matrix. To analyze the high-dimensional data, several low-rank approximation methods have been developed, including Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Nuclear Norm Minimization (NNM), Nonnegative Matrix Factorization (NMF) (Lee and Seung, 1999), and deep representation learning (LeCun et al., 2015).

However, link prediction encounters challenges attributed to the inherent sparsity of the adjacency matrix, where the fraction of observed edges is much smaller than the fraction of unobserved edges. In addition, the previous related algorithms have some noticeable weaknesses. Methods based on similarity only consider the network's topology and are unable to provide a detailed understanding of the network's organization (Wang et al., 2017). Hence, the development of an effective algorithm is essential to tackle the aforementioned issues. The utilization of matrix factorization models has become widespread in recommendation systems and matrix completion, because these techniques can uncover latent features or employ additional information to make a prediction. NMF uncovers two nonnegative matrices, namely the basis matrix $\boldsymbol{W}$ and the coefficient matrix $\boldsymbol{H}$. The reconstruction of the network with these matrices is employed to predict missing links (Lee and Seung, 1999). For instance, in the work by Menon and Elkan (2011), a latent feature learning approach using NMF was introduced for the purpose of network reconstruction. However, single-layer matrix approximations cannot capture multi-level features within intricate datasets.

Real-world networks have a diverse organizational and hierarchical structure (e.g., multi-layer), and the link prediction methods relying on shallow NMF may fail to capture the complexity of the network. Deep neural networks have gained popularity due to their performance in a variety of supervised classification tasks (LeCun et al., 2015; Marcus, 2018) and generative models (Goodfellow et al., 2014a). The main advantage of deep networks lies in their capacity to nonlinearly model complex relationships and hierarchical representations in data. Therefore, using deep learning approaches to more effectively uncover hierarchical clustering and network reconstruction is unavoidable (Chen et al., 2022). Recently, Deep Matrix Factorization (DMF) is introduced which can extract multi-layer structural information by hierarchical mapping the adjacency matrix into the low-dimensional latent space. DMF is halfway between deep neural networks and linear algebra models which inherits the deep learning ability, and integrates two characteristics, (1) Multiple hierarchical feature extraction, as made feasible by multi-layer architectures, and (2) Hierarchical interpretability, analogous to shallow matrix factorization (De Handschutter et al., 2021).

Previous shallow and deep network reconstruction methods primarily focus on preserving the structure relationship or minimizing the reconstruction error. However, they have often been plagued by issues such as overfitting and a limited ability to generalize to new or unseen data, especially when dealing with sparse data. To counteract these shortcomings, adversarial training techniques have emerged as a promising approach to amplify data robustness and enhance the overall performance of deep learning models. This approach gains particular significance in light of the findings (Goodfellow et al., 2014b) which reveal the susceptibility of deep learning methods to adversarial attacks on observations. To mitigate this susceptibility and enhance generalization abilities, adversarial training has gained popularity. Recently, there has been an expansion of generative adversarial-based techniques, as demonstrated by works such as Zhao et al. (2016) and Radford et al. (2015), aimed at acquiring robust representation. In more recent developments, some studies (Sinha et al., 2017; Farnia et al., 2018) have presented theoretical findings supporting the notion that adversarial training enhances generalization in diverse learning problems. A standard adversarial model follows empirical risk minimization training over the adversarially-perturbed samples by solving

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} l(f_{\boldsymbol{w}}(\boldsymbol{x}_i + \Delta_{\boldsymbol{w}}(\boldsymbol{x}_i)), \boldsymbol{y}_i) \tag{1}$$

where $l(\cdot,\cdot)$ is a loss function, $f_{\boldsymbol{w}}(\cdot)$ is the output function, $\Delta_{\boldsymbol{w}}(\boldsymbol{x}_i)$ is the adversarial additive perturbation for sample $\boldsymbol{x}_i$, and each $\boldsymbol{y}_i$ is a label vector. Some recent works explore using adversarial training and related techniques to improve robustness and performance in various applications (Qian et al., 2022; Xiao et al., 2024, 2023). For example, Chen et al. (2024) proposed using separate encoders for speech content, speaker identity, and pitch, along with a gradient reversal layer and noise discriminator. This aims to decouple these factors through adversarial training to enable high-quality speech synthesis. Zhang et al. (2024) focused on power quality disturbance classification using deep neural networks. It proposes an adversarial attack algorithm that can degrade the DNN's performance, as well as a defense algorithm that improves the model's robustness against such attacks. Nath et al. (2023) proposed using histogram equalization and adversarial training through neural structure learning for developing a robust surface defect classification framework.

Similarly, highly complex deep link prediction models (Chen et al., 2022; Li et al., 2024) often pose the risk of overfitting to the limited number of connections within the network, leading to diminished generalization capabilities. However, the integration of adversarial training into the DMF enables the factorization model to address various difficulties associated with predicting links in networks characterized by sparse connections. To elaborate further, adversarial training equips the deep model to gain robust latent factors that adeptly learn the intricate hierarchical structure, even when faced with incomplete or noisy data.

This research is motivated by acknowledging this gap that shallow NMF methods lack the capacity to capture the intricacies of real-world networks and current deep NMF methods have the potential to overfit limited observations. In this paper, we introduce an adversarial learning method to mitigate the sparsity problem in link prediction task. Specifically, we introduce an attacker matrix that modifies the input adjacency matrix by adding fake links and removing existing links in a bounded manner. This adversarial perturbation increases the number of training links and reduces sparsity, enabling the model to learn from richer connectivity patterns. Through this adversarial framework, the attacker iteratively manipulates the input graph structure to hinder the factorization from overfitting to the limited observations. Hence, through adversarial training, the deep model is encouraged to generate embeddings that exhibit robustness towards minor input variations, thereby enhancing its capacity to generalize effectively to new links. To be more specific, we introduce an adversarial variant of the Deep NMF to improve the prediction of DMF in the network reconstruction task. This approach utilizes Deep NMF to encode the network within a

low-dimensional space, and incorporates a modified adversarial training scheme to discover a robust graph representation. By imposing a sparse constraint to the attacker matrix, this model simultaneously minimize reconstruction errors in the topological structure and generate diverse adversary links, thereby improving generalization. In a unified majorization-minimization framework, the optimization of graph representation and adversarial learning is performed simultaneously, leveraging their mutual benefits to enhance graph reconstruction performance. In addition, to address the graph sparsity, the first and second-order affinities are combined as the input. Finally, to augment the generalization capacity and mitigate overfitting on the observed links, L_2 norm based smoothness regularizations are integrated into the proposed method. The primary contributions of this paper can be summarized as follows:

- To improve the accuracy of network reconstruction and extract more hidden information, deep matrix factorization with high learning capacity has been utilized to learn hierarchical network structure.
- Inspired by the generalization capabilities exhibited in adversarial training, we introduce a modified model for adversarial network reconstruction that incorporates $L_{2,1}$ regularization. In this model, the reconstruction performance is enhanced by introducing a diversity-aware adversary matrix. The goal of the learner is to minimize the reconstruction error while confronting the modifications made by the attacker.
- To tackle the issues arising from input sparsity and to maintain the local structure of the graph, we add common neighbor similarity into the adjacency matrix. This addition serves to compensate for the sparsity in the input and effectively preserves the relationships among neighboring nodes in the graph.
- To optimize our adversarial framework, we employ a majorization-minimization method incorporating L_2 norm smoothness regularization. We derive multiplicative updating rules to effectively learn the parameters of LPADNMF within this framework.
- We perform experiments on eight real-world networks, using two measurements, to assess the performance of LPADNMF. The experimental results indicate that LPADNMF surpasses state-of-the-art link prediction approaches, demonstrating its superior performance.

The remainder of this paper is organized into the following sections. Section 2 provides an overview of related works in this study. In Section 3, we detail the background of shallow and deep Nonnegative Matrix Factorizations. The proposed approach, adversarial link prediction based on the Deep NMF model, is presented in Section 4. Section 5 showcases the numerical and graphical results obtained from real-world datasets. Finally, in Section 6, we conclude the paper by discussing future research directions.

2. Related work

There are different approaches for link prediction available in the literature. In this section, we focus on the methods that are relevant to this approach. Some of the popular link prediction approaches include similarity-based methods, which leverage measures of similarity between nodes, and factorization-based methods, which use latent factor models to make predictions.

2.1. Similarity-based methods

Among the various link prediction techniques, similarity-based methods serve as a fundamental framework. These methods operate on the principle that two nodes are more likely to form a connection if they exhibit a higher level of similarity compared to other node pairs. One example of such a method is the Common Neighbors approach (Newman, 2001), which assumes that two nodes are considered similar if they share a significant number of common neighbors. The Jaccard index is another neighborhood-based method that normalizes the number of common neighbors by the total number of neighbors. A straightforward approach to quantify neighborhood overlap is through direct counting. Several similarity measures have been introduced that consider the impact of the degrees of pair nodes, such as the Salton Index (Salton and McGill, 1983), which is also referred to as cosine similarity. The Adamic-Adar Index (AA) (Adamic and Adar, 2003) is computed by summing the inverse logarithmic centrality degree of the shared neighbors between two nodes. This index assigns weights to nodes by considering the extent of their shared neighbors, with the principle that nodes with fewer common neighbors contribute more to the similarity measure than those with a higher number of common neighbors. On the other hand, the Resource Allocation Index (RA) (Ou et al., 2007) was developed to quantify resource allocation dynamics in complex networks. It measures the allocation of resources between nodes and serves as an indicator of their similarity. Preferential Attachment (PA) index (Martínez et al., 2016) has been proposed as a solution to model growth and relies on the assumption that the probability of a new link being connected to a node is proportional to the node degree. In spite of their simplicity, the similarity-based methods face difficulties in handling sparse data and sensitivity to the noisy links.

2.2. Factorization-based methods

Latent space algorithms present a promising alternative for efficiently analyzing both local and global structures within networks. The NMF model is one of the popular dimensionality-reduction techniques that provides a node representation learning for network reconstruction. In the NMF model, the adjacency matrix $\boldsymbol{A}$ is approximated by $\boldsymbol{W}\boldsymbol{H}$, where $\boldsymbol{W}$ is the mapping factor, and $\boldsymbol{H}$ is the representation factor (Lee and Seung, 1999). The reconstructed matrix $\boldsymbol{W}\boldsymbol{H}$ is then compared with the original input $\boldsymbol{A}$ to predict potential links.

A ground-breaking strategy for solving the link prediction problem was put forward in Zhao et al. (2015) as a neighborhood-based NMF model. This method aims to reveal latent features by considering both the global and local neighborhood structures of the network. NMFKL (Wang et al., 2017) is a kernel approach that get global and local network structure through kernel mapping. Two kernel functions with meaningful interpretations for network reconstruction are employed to map the adjacency matrix into a latent space. Wang et al. (2018) presented the SASNMF that considers the network structure, including node characteristics and latent structural elements extracted from the network, along with internal and external auxiliary information. Two possible pairings of internal and external data are proposed and incorporated into the factorization-based model, to handle the link prediction problem.

The GWNMF (Graph Regularized Weighted NMF) model, as introduced in the work by Chen et al. (2019), was developed for the purpose of incorporating local topology information and link weights into link prediction. In this model, the utilization of cosine similarity combines the link weights and local topological information, effectively calculating the weighted similarity between adjacent nodes. MS-RNMF (Chen et al., 2020b) fuses the k-medoids clustering with manifold learning to maintain the global and local topology information, respectively. Additionally, the MS-RNMF employs a $L_{2,1}$ regularization norm to ignore spurious and noisy connections. Chen et al. (2020a) employs the PageRank algorithm to ascertain the impact score of nodes and to gather information about the comprehensive network topology. Additionally, this model computes the link clustering coefficient score through the use of asymmetric link clustering and preserves information of the local structure of network. RC-NMF (Wang and Mu, 2021) is a regularized Convex NMF model for community detection and link prediction in the signed networks. This model uses a regularization to group positive-edge nodes into the same clusters and negative-edge ones into different clusters. More recently, LPANMF utilized an

adversarial training algorithm aimed at improving the generalization and robustness of a model. This enhancement involves incorporating an adversary matrix into the input. Additionally, a regularization term is employed to ensure that latent representations align with the local topological structure (Mahmoodi et al., 2023).

2.3. Deep factorization-based methods

The complexity of information within real-world input data matrices often poses a challenge in extracting high-level features using shallow NMF models. Trigeorgis et al. (2014) presented one of the first models based on the deep matrix factorization which is used for the face recognition task. The deep factorization model captures multiple layers of features, with each layer corresponding to a distinct interpretation, spanning from low-level features in the initial layer to high-level features in the final layer. In recent years, deep matrix factorization models are widely developed for different machine learning tasks such as data representation (Salahian et al., 2023; Faraji et al., 2024; Mozafari et al., 2024), matrix completion (Seyedi et al., 2019, 2023), recommender systems (Shajarian et al., 2017), hyperspectral unmixing (Li et al., 2021), graph clustering (Hajiveiseh et al., 2024; Ghodsi et al., 2024; Abdollahi et al., 2020), etc. However, there are a few deep matrix factorization models that are proposed for complex network analysis tasks that generally are developed for community detection and network embedding problems. For example, the deep factorization learning structure is used to develop the network embedding method known as Deep Robust NMF (DRNMF) (He et al., 2020). In this method, the combination of high-order matrices serves as the primary mapping matrix for the network reconstruction. The network embedding model's objective function was created using the $L_{2,1}$ norm to increase noise resistance. In order to discover communities, Huang et al. (2021) proposed a Modularized deep NMF approach that maintains both topological data and fundamental community structure properties. For community discovery in complicated networks, a new matrix factorization based on deep auto-encoder architecture has recently been introduced. This model has an encoder component and a decoder component, similar to the deep auto-encoder. By low-level to high-level network hidden features, this design enables matrix factorization to learn hierarchical mappings between the original network and the final community matrix (extracted in hidden layers) (Ye et al., 2018). In order to retain network structure, *Zhang* and *Zhou* created a deep auto-encoder-based NMF that combines first-order and second-order similarities as the input matrix (Zhang and Zhou, 2020). Chen et al. have proposed the first deep NMF model for the link prediction task called FSSDNMF. This model combines two deep NMF terms to learn a shared representation from the adjacency and similarity information. To capture the structural attributes of individual hidden layers, the FSSDNMF method employs common neighbor similarity, translating it into a multi-layer hidden space with reduced dimensionality. Additionally, an $L_{2,1}$ norm-based regularization is applied to each hidden layer to effectively eliminate random noises (Chen et al., 2022). More recently, Li et al. (2024) introduced DANMFL, a novel method for enhancing link prediction models, combining Deep Autoencoder-like NMF with the $L_{2,1}$ norm. DANMFL integrates encoder and decoder components to capture complex hierarchical patterns in data, while the $L_{2,1}$ norm inclusion aids in minimizing random noise.

To summarize, shallow NMF methods lack the capacity to capture the intricacies of real-world networks, even adversarial NMF (Mahmoodi et al., 2023) despite its generalization ability. On the other hand, deep NMF models have the potential to overfit limited observations. Therefore, there exists a research gap in understanding how to effectively enhance deep NMF models to prevent overfitting while retaining their expressive power. By acknowledging this gap, the adversarial deep NMF model's training process incorporates appropriate adversarial links to mitigate overfitting, thereby enhancing its generalization capabilities. This marks a significant advancement in the field, as it combines the strengths of deep NMF with the robustness of adversarial training, resulting in a novel approach to link prediction. Table 1 provides a summary of various link prediction methods along with their key characteristics.

3. Background

In this section, we begin by introducing essential notations to facilitate a clear understanding. Subsequently, we present an overview of both shallow and deep models of Nonnegative Matrix Factorization (NMF) that have garnered significant attention in various fields.

3.1. Notations

In this work, we use specific notations to represent different types of mathematical entities. Lowercase italic letters (e.g., i, j, n) represent scalars, vectors are represented by bold lowercase letters (e.g., $\boldsymbol{a}$, $\boldsymbol{x}$), and matrices are represented by bold uppercase letters (e.g., $\boldsymbol{W}$, $\boldsymbol{H}$). For matrix $\boldsymbol{A}$, $\boldsymbol{a}_i$ refers to the ith column, and A_{ij} refers to the element at the ith row and jth column. $\boldsymbol{A}^\top$ represents the transpose of $\boldsymbol{A}$, and trace of a matrix $\boldsymbol{A}$ is denoted as $\mathrm{Tr}(\boldsymbol{A})$. The Frobenius norm is defined as $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{A})}$.

3.2. Nonnegative matrix factorization

The matrix factorization method is a dimensionality reduction approach that has recently acquired a lot of attention for addressing various learning, recommendation, and prediction problems.

3.2.1. Shallow NMF

Given a d-dimensional variable $\boldsymbol{x}$ with nonnegative components, where n observations are represented as $\boldsymbol{x}_j$ for $j = \{1, 2, \ldots, n\}$, and let the input be $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}_{\geq 0}^{d \times n}$. NMF aims to decompose $\boldsymbol{X}$ into a nonnegative basis $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k] \in \mathbb{R}_{\geq 0}^{d \times k}$ and nonnegative coefficient $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n] \in \mathbb{R}_{\geq 0}^{k \times n}$, such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$ (Lee and Seung, 1999). With the input $\boldsymbol{x}_i \in \mathbb{R}_{\geq 0}^{d}$ representing the ith sample, NMF factorizes it into the basis $\boldsymbol{W} \in \mathbb{R}_{\geq 0}^{d \times k}$ and the corresponding representation $\boldsymbol{h}i \in \mathbb{R}_{\geq 0}^{k}$.

$$\min_{\boldsymbol{W},\boldsymbol{h} \geq 0} \sum_{i=1}^{n} z(\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{h}_i) \tag{2}$$

where $z(.)$ represents a cost function. Clearly, $\boldsymbol{h}_i$ is the weight coefficient indicating the influence of the observed entries $\boldsymbol{x}_i$ on the latent vectors within $\boldsymbol{X}$, corresponding to columns in $\boldsymbol{W}$. Consequently, Non-Negative Matrix Factorization (NMF) dissects the data into distinct components, utilizing the basis vectors in either a linear or nonlinear fashion. Given the precondition $k \ll \min(d, n)$, the resulting base vectors exhibit sparsity within the original space. In essence, achieving an exact factorization is only feasible when the inherent data characteristics are captured, as this method seeks to represent a high-dimensional pattern using significantly fewer bases. The least square error stands as the most prevalent loss function in NMF, expressed as follows:

$$z_2(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2 = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2, \tag{3}$$

3.2.2. Deep NMF

Real-world data often have intricate and varied organizational patterns. Therefore, there is a strong probability that the link between the original data and the latent space incorporates intricate hierarchical and structural intricacies, including hidden properties at lower levels that are implicitly encoded. Deep learning is well known for its ability to bridge the gap between lower-level abstraction and the higher level

Table 1
Summary of properties for the related learning-based methods and the proposed model.

Method	Properties					Publish year
	Factorization-based	Similarity-aware	Robust	Deep	Adversarial	
(Lee and Seung, 1999) NMF	✓					1999
(Wang et al., 2017) NMFKL	✓					2017
(Wang et al., 2018) SASNMF	✓	✓				2018
(Chen et al., 2019) GWNMF	✓	✓				2019
(Chen et al., 2020b) MS-RNMF	✓	✓	✓			2020
(Wang and Mu, 2021) RC-NMF	✓	✓				2021
(Chen et al., 2022) FSSDNMF	✓	✓		✓		2022
(Chai et al., 2023) LRNP			✓			2023
(Mahmoodi et al., 2023) LPANMF	✓	✓	✓		✓	2023
(Li et al., 2024) DANMFL	✓		✓	✓		2024
LPADNMF	✓	✓	✓	✓	✓	–

concepts of the original data (Seyedi et al., 2023). The deep NMF (Trigeorgis et al., 2014) is introduced, which factors a matrix $\boldsymbol{X}$ as input into $l+1$ factors as follows:

$$\boldsymbol{X} \approx \boldsymbol{W}_1 \boldsymbol{W}_2 \cdots \boldsymbol{W}_l \boldsymbol{H}_l \tag{4}$$

where $\boldsymbol{H}_l \in \mathbb{R}_{\geq 0}^{k \times n}$ is the representation in low dimensions. $\boldsymbol{W}_i \in \mathbb{R}_{\geq 0}^{r_{i-1} \times r_i} (1 \leq i \leq l)$ is the mapping variable of the ith layer. The factorization in (4) offers an implicit representation hierarchy of the input matrix with l levels, from which the following decomposition may be obtained:

$$\begin{aligned} \boldsymbol{H}_{l-1} &\approx \boldsymbol{W}_l \boldsymbol{H}_l \\ &\vdots \\ \boldsymbol{H}_2 &\approx \boldsymbol{W}_3 \cdots \boldsymbol{W}_l \boldsymbol{H}_l \\ \boldsymbol{H}_1 &\approx \boldsymbol{W}_2 \cdots \boldsymbol{W}_l \boldsymbol{H}_l \end{aligned} \tag{5}$$

where $\boldsymbol{W}_i$ is the basis factor of the ith layer and $\boldsymbol{H}_i$ represents the coefficient factor of the ith layer. The cost function for the Deep NMF model can be formulated as

$$\min_{\boldsymbol{W}_i, \boldsymbol{H}_l} \|\boldsymbol{X} - \boldsymbol{W}_1 \cdots \boldsymbol{W}_l \boldsymbol{H}_l\|_F^2, \quad \text{s.t.} \quad \boldsymbol{W}_i \geq 0, \boldsymbol{H}_l \geq 0, \forall i = 1, 2, \ldots, l. \tag{6}$$

4. Proposed model

This section introduces a novel link prediction method by formulating an efficient adversarial Deep NMF with enhanced generalization abilities. Four key elements contribute to its effectiveness: (1) It employs an efficient input matrix which is a combination of first-order and second-order similarities; (2) It leverages a Deep NMF structure to map the input data into the latent space to exploit hierarchical hidden structures; (3) A modified adversarial training method with a $L_{2,1}$ regularization is employed for generating diverse adversary links; (4) The LPADNMF model integrates the previously mentioned factors into a unified learning problem and employs a highly efficient alternating Majorization-Minimization technique for optimization. The model's structure is visually represented in Fig. 1. The details of our LPADNMF model are explained in 5 subsections. Section 4.1 defines the link prediction task and evaluation metrics. Section 4.2 explains how the model combines network structure information. Section 4.3 introduces the ADNMF model with an attacker and defender for adversarial training. Section 4.4 provides the optimization algorithm, including attacker-defender optimizations and pre-training. Finally, Section 4.5 analyzes computational complexity of the proposed model.

4.1. Problem description

A network is composed of nodes and edges, and the undirected input network is commonly represented as $G = (V, E)$, where V is the set of nodes and E is the set of links. Mathematically, the network can be represented by an adjacency matrix $\boldsymbol{A}_{n \times n}$ $(n = |V|)$, where $A_{ij} = 1$ if node i and node j are connected, and 0 otherwise. Link prediction seeks to identify links that are missing or do not currently exist. Our method and similar methods involve calculating scores for all unobserved edges and using these scores to estimate their existence probability in the reconstruction. We then rank the unobserved links according to their scores and display the links with the highest probabilities first. However, it is important to keep in mind that, various prediction algorithms could provide various ordered lists of unseen links. To evaluate the performance of approaches, random partitioning of the existing links into two sets is employed: a training set denoted as E^T and a test set denoted as E^P. This partition is determined by a specified proportion, such that E^T and E^P satisfy the conditions $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. The common criteria to evaluate link prediction performance are AUC (Hanley and McNeil, 1982) and Precision (Herlocker et al., 2004). While Precision solely assesses the accuracy of the prediction of the top-k links, AUC concentrates on analyzing the overall accuracy.

4.2. The construction of input matrix

This section defines an input matrix which is constructed from first and second-order affinities. Numerous techniques are available for extracting structural information, including pathway and neighborhood based methods (Newman, 2001; Zhao et al., 2015; Salton and McGill, 1983; Adamic and Adar, 2003; Ou et al., 2007; Martínez et al., 2016). In this paper, we use the common neighbors which is most effective and straightforward method to capture topological information. To accomplish this objective, the following formulation is utilized to extract the local structure matrix:

$$S_{i,j} = |\Gamma_i \cap \Gamma_j| = \boldsymbol{a}_i^\top \boldsymbol{a}_j \tag{7}$$

where Γ_i is the neighbors of node i and $\boldsymbol{a}_i$ is ith column of the adjacency matrix $\boldsymbol{A}$. The similarity matrix can be rewritten as:

$$\boldsymbol{S} = \boldsymbol{A}^\top \boldsymbol{A}. \tag{8}$$

Typically, the adjacency matrix is used to represent the first-order structure in a network, capturing direct connections between nodes. However, the network topology contains both local and global information. The local structure reflects relationships in a node's immediate neighborhood, such as shared neighbors. In contrast, the global structure indicates the significance of nodes in the overall network. To incorporate both aspects, some methods jointly consider first-order adjacency and second-order similarity during network analysis (Zhang and Zhou, 2020). Our approach also combines first-order and second-order information in an appropriate manner. This allows preserving both the global positioning of nodes as well as local topological structure. Specifically, we adopt the following formula to integrate the adjacency matrix $\boldsymbol{A}$ with the similarity matrix $\boldsymbol{S}$ for comprehensive network reconstruction.

$$\boldsymbol{A}_S = \boldsymbol{A} + \alpha \boldsymbol{S}, \tag{9}$$

where α is the compromise parameter between the local and global structures. Thus, we will utilize this new matrix $\boldsymbol{A}_S$ as an input matrix for adversarial deep nonnegative matrix factorization.

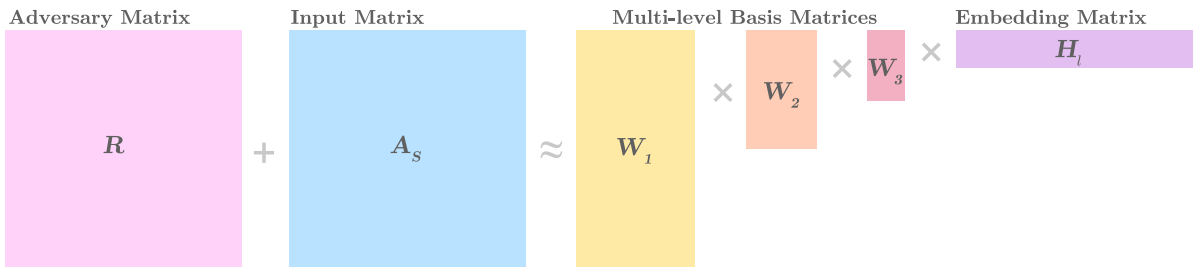


Fig. 1. The diagram of the proposed adversarial deep model.

4.3. Deep adversarial model

The majority of matrix factorization models are non-convex, making them prone to getting stuck in local minima, especially when outliers and missing data are present. Moreover, enhancing the capacity of a model by increasing its depth can also amplify its training capability, potentially resulting in overfitting. This makes DMFs more vulnerable to adversarial attacks. As a solution, we intend to use adversarial training to effectively tackle these issues. In this section, we introduce the Adversarial Deep NMF (ADNMF) model, which involves an attacker that introduces an arbitrary attack $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ by modifying the input matrix $\boldsymbol{A}_S$. This adversarial matrix $\boldsymbol{R}$ is designed to maximize the least square loss, which is the Frobenius norm between the input matrix $\boldsymbol{A}_S$ and the predicted matrix $\boldsymbol{W}_1\boldsymbol{W}_2\cdots\boldsymbol{W}_l\boldsymbol{H}_l$. We assume that $\boldsymbol{R}$ belongs to a bounded set since the attackers often have finite strength. Therefore, the expression for ADNMF is

$$\min_{\boldsymbol{W}_i,\boldsymbol{H}_l}\max_{\boldsymbol{R}\in\mathcal{R}}\|\boldsymbol{A}_S+\boldsymbol{R}-\boldsymbol{W}_1\cdots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2,\quad \text{s.t.}\quad \boldsymbol{W}_i\geq 0,\boldsymbol{H}_l\geq 0,\forall i=1,2,\ldots,l. \tag{10}$$

We can define bound constraint as

$$\mathcal{R}:=\{\boldsymbol{R}\in\mathbb{R}^{n\times n}:\|\boldsymbol{R}\|_{2,1}\leq\epsilon,\boldsymbol{A}_S+\boldsymbol{R}\geq 0\} \tag{11}$$

where $\|\boldsymbol{R}\|_{2,1}=\sum_{j=1}^{n}\|\boldsymbol{r}_j\|=\sum_{j=1}^{n}\sqrt{\sum_{i=1}^{n}R_{ij}^2}$ and $\epsilon>0$ is the strength of adversary; the larger ϵ corresponds to a strong adversary and vice versa. To adopt an optimization for the objective function (10) with an effective input matrix $\boldsymbol{A}_S+\boldsymbol{R}$, we assume that the matrix $\boldsymbol{R}$ is an adversarial matrix bounded by $L_{2,1}$ norm and $\boldsymbol{A}_S+\boldsymbol{R}$ is also restricted to be nonnegative. In such situations, using the sparse $L_{2,1}$ regularization for attacks would result in their dispersal. As a result, more diverse attacks would tend to target network links that are potentially not in harmony with the rest, such as outlier links and those suspected of containing noise. However, if the Frobenius norm is employed, the attacks would be almost uniformly applied to all the network links.

In general, solving the problem presented in Eqs. (10) and (11) is a challenging task. Therefore, we turn to a relaxation technique (Boyd et al., 2004) inspired by Lagrangian duality. We apply a Lagrange multiplier $\lambda>0$ to dualize the norm constraint $\|\boldsymbol{R}\|_{2,1}\leq\epsilon$ in (11), resulting in a simpler cost function:

$$\min_{\boldsymbol{W}_i,\boldsymbol{H}_l}\max_{\boldsymbol{R}}\|\boldsymbol{A}_S+\boldsymbol{R}-\boldsymbol{W}_1\cdots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2-\lambda\|\boldsymbol{R}\|_{2,1} \tag{12}$$
$$\text{s.t.}\quad \boldsymbol{A}_S+\boldsymbol{R}\geq 0,\boldsymbol{W}_i\geq 0,\boldsymbol{H}_l\geq 0,\forall i=1,2,\ldots,l.$$

The inner optimization involves maximization, which is the reason Eq. (12) includes a negative regularization term. The parameter λ can be viewed as the counterpart to ϵ and represents the adversary's strength. Specifically, as $\lambda\to 0^+$, the $\|\boldsymbol{R}\|_{2,1}$ term becomes unbounded, indicating a highly potent adversary. On the other hand, when $\lambda\to\infty$ increases, the influence of regularization decreases, and $\|\boldsymbol{R}\|_{2,1}$ approaches zero, indicating the absence of an adversary.

We can employ Majorization-Minimization (Hunter and Lange, 2000) to achieve solutions for (12). The optimization process described in Eq. (12) can be split into two distinct sections. The first section involves an inner maximization step to optimize the variable $\boldsymbol{R}$, while the second section involves an outer minimization step to optimize the variables $\boldsymbol{W}_1,\boldsymbol{W}_2,\ldots,\boldsymbol{W}_l$, and $\boldsymbol{H}_l$. The inner maximization may also be expressed as the following minimization:

$$\boldsymbol{R}^*=\arg\min_{\boldsymbol{R}}-\|\boldsymbol{A}_S+\boldsymbol{R}-\boldsymbol{W}_1\cdots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2+\lambda\|\boldsymbol{R}\|_{2,1},\ \text{s.t.}\quad \boldsymbol{A}_S+\boldsymbol{R}\geq 0. \tag{13}$$

The upcoming section illustrates that the solution to this minimization is explicitly introduced in a closed-form. After $\boldsymbol{R}^*$ has been obtained over $\boldsymbol{W}_1,\boldsymbol{W}_2,\ldots,\boldsymbol{W}_l,\boldsymbol{H}_l\geq 0$, the objective function can be optimized. By utilizing the updated input $\boldsymbol{A}_S+\boldsymbol{R}^*$, objective function (14) can be minimized to obtain solutions for $\boldsymbol{W}_1,\boldsymbol{W}_2,\ldots,\boldsymbol{W}_l$, and $\boldsymbol{H}_l$ as follows:

$$\min_{\boldsymbol{W}_i,\boldsymbol{H}_l}\|\boldsymbol{A}_S+\boldsymbol{R}^*-\boldsymbol{W}_1\cdots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2,\ \text{s.t.}\quad \boldsymbol{W}_i\geq 0,\boldsymbol{H}_l\geq 0,\forall i=1,2,\ldots,l. \tag{14}$$

Until now, we have created a deep adversarial trained factorization that is a perturbation-based model while using the least square loss (Frobenius norm). After completing the aforementioned preparations, we apply the Frobenius-norm soft regularization on the multiplication of the basis matrices $\prod_{p=1}^{l}\boldsymbol{W}_p$ and the matrix final coefficient matrix $\boldsymbol{H}_l$ to prevent overfitting the parameters of this deep factorization to the training graph. We are now ready to present the objective function for link prediction. The final objective function for the Link Prediction by Adversarial Deep NMF (LPADNMF) model is:

$$\min_{\boldsymbol{W},\boldsymbol{H}}\max_{\boldsymbol{R}\in\mathcal{R}}\|\boldsymbol{A}_S+\boldsymbol{R}-\boldsymbol{W}_1\cdots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2-\lambda\|\boldsymbol{R}\|_{2,1}+\beta(\|\boldsymbol{W}_1\ldots\boldsymbol{W}_l\|_F^2+\|\boldsymbol{H}_l^\top\|_F^2)$$
$$\text{s.t.}\quad \boldsymbol{W}_i\geq 0,\boldsymbol{H}_l\geq 0,\forall i=1,2,\ldots,l, \tag{15}$$

where β is the smoothness regularization hyperparameter. To sum up, the objective function (15) consists of three separate terms, each with a distinct objective. The first term aims to predict links in the network while incorporating the adversaries. In addition, the input matrix comprises the global and local structural similarities of the network. The second term is designed to manage the power of the attacker and utilize the properties of the $L_{2,1}$ norms on a loss function that can deal with random noises in the input. Lastly, the third term is introduced to enhance generalization and leverage the full capacity of the network across different layers.

In Fig. 2, the mechanism behind the proposed model for link prediction using deep NMF with adversarial perturbation is illustrated. The process initiates with an input graph, referred to as the "Whole Graph" (Fig. 2(a)). Subsequently, this graph undergoes modification by the removal of a random subset of links to create a "Probe Graph", as depicted in Fig. 2(b). The adversarial training model operates on this Probe Graph, employing a process that integrates Deep NMF with adversarial perturbation. This training approach dynamically targets and enhances edges most likely to form, assigning weights based on their predicted significance. The model incorporates adversarial perturbation to discern edges more likely to connect nodes with high

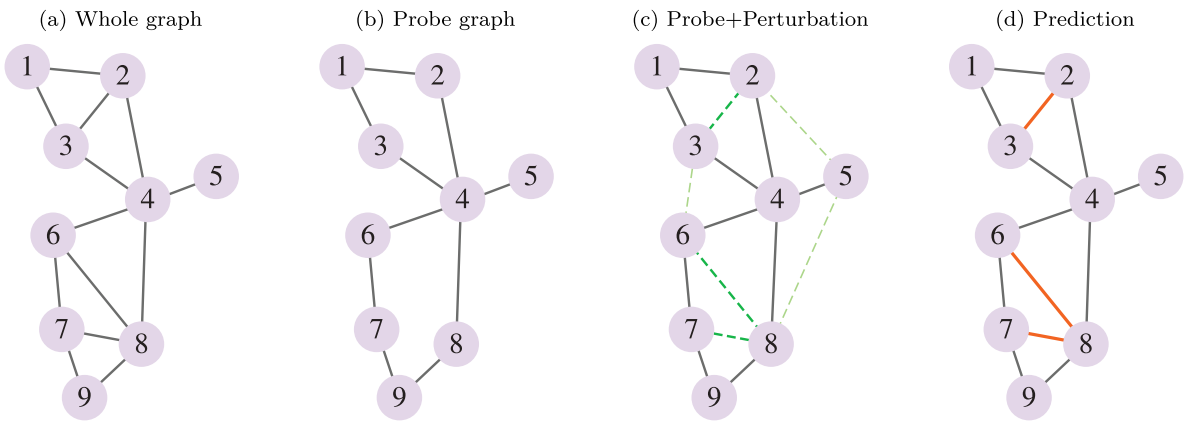


Fig. 2. The adversarial training mechanism of the proposed model (LPADNMF) for link prediction.

similarity. Such edges undergo increased attacks, leading to elevated weights. Edges with a lower likelihood of formation experience fewer attacks, resulting in diminished weights. In Fig. 2(c), the probability of forming links between nodes 6 and 8, as well as between nodes 2 and 3, is significantly higher than the probability of creating a link between nodes 5 and 8. Finally, from Fig. 2(d) we can derive that the model identifies edges in the Probe+Perturbation Graph, along with new highly ranked and scored edges, suggesting them as potential link predictions to prioritize the most probable connections.

4.4. Numerical solution

Algorithm 1, which is an alternating minimization method, is a suitable approach for solving problem (15) as it allows for the iterative updating of variables until a viable solution is obtained. This involves using gradient descent to update a single variable in each iteration while holding the remaining variables constant. By breaking down the overall optimization problem into a series of smaller sub-problems, we can tackle them more easily. The iterative optimization process contributes to the convergence of the objective function values through the use of the Multiplicative Update Rules (MUR) and the enforcement of the Karush-Kuhn–Tucker (KKT) conditions. The KKT conditions are necessary conditions for optimality in constrained optimization problems such as NMF. These conditions guarantee several key properties, including convergence to a stationary point satisfying the KKT conditions, monotonic descent of the objective function values, satisfaction of the non-negativity constraints, and convergence to a local minimum or stationary point (Lin, 2007).

4.4.1. Attacker optimization

This section elaborate on the methodology for obtaining $\boldsymbol{R}^*$ as defined in Eq. (13). Here, we treat the matrices $\boldsymbol{W}_i$ and $\boldsymbol{H}_i$ (for all $i \in 1, 2, \ldots, l$) as constants, and their product is represented as $\bar{\boldsymbol{A}}_S = \boldsymbol{W}_1 \cdots \boldsymbol{W}_l \boldsymbol{H}_l$. Therefore, the objective (13) can be reformulated as:

$$\begin{aligned} g(\boldsymbol{R}) :&= -\|\boldsymbol{A}_S + \boldsymbol{R} - \bar{\boldsymbol{A}}_S\|_F^2 + \lambda\|\boldsymbol{R}\|_{2,1} \\ &= \mathrm{Tr}(-\boldsymbol{A}_S\boldsymbol{A}_S^\top - \boldsymbol{R}\boldsymbol{R}^\top + \bar{\boldsymbol{A}}_S\bar{\boldsymbol{A}}_S{}^\top - 2\boldsymbol{A}_S\boldsymbol{R}^\top \\ &\quad + 2\boldsymbol{A}_S\bar{\boldsymbol{A}}_S{}^\top + 2\boldsymbol{R}\bar{\boldsymbol{A}}_S{}^\top) + \mathrm{Tr}(\lambda\boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^\top) \end{aligned} \tag{16}$$

where $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix denoted by:

$$D_{jj} = \frac{1}{\|\boldsymbol{r}_j\|}. \tag{17}$$

The convex cost function can be easily represented through direct differentiation:

$$\begin{aligned} &\frac{\partial g}{\partial \boldsymbol{R}} = -2\boldsymbol{R} - 2\boldsymbol{A}_S + 2\bar{\boldsymbol{A}}_S + 2\lambda\boldsymbol{R}\boldsymbol{D} = 0 \\ &\quad \Rightarrow \lambda\boldsymbol{R}\boldsymbol{D} - \boldsymbol{R} = \boldsymbol{A}_S - \bar{\boldsymbol{A}}_S \end{aligned} \tag{18}$$

Given the condition $\boldsymbol{A}_S + \boldsymbol{R} \geq 0$ as specified in Cai et al. (2021), the optimal solution for (16) is:

$$\boldsymbol{R} = \max\left\{(\boldsymbol{A}_S - \bar{\boldsymbol{A}}_S)\frac{\boldsymbol{I}}{\lambda\boldsymbol{D} - \boldsymbol{I}}, -\boldsymbol{A}_S\right\}, \tag{19}$$

where max operator is an element-wise maximum.

4.4.2. Learner optimization

To accelerate the optimization of factors in our model, we employ a hierarchical pre-training approach, initializing the matrices $\boldsymbol{W}_i$ and $\boldsymbol{H}_i$ in each layer. This substantial enhancement significantly reduces the model's training time (Hinton and Salakhutdinov, 2006). We initially factor the input matrix $\boldsymbol{A}_S \approx \boldsymbol{W}_1\boldsymbol{H}_1$ by minimizing $\|\boldsymbol{A}_S - \boldsymbol{W}_1\boldsymbol{H}_1\|_F^2$ where $\boldsymbol{W}_1 \in \mathbb{R}_+^{n\times r_1}$ and $\boldsymbol{H}_1 \in \mathbb{R}_+^{r_1\times n}$, then the matrix $\boldsymbol{H}_1$ is decomposed into $\boldsymbol{H}_1 \approx \boldsymbol{W}_2\boldsymbol{H}_2$ by minimizing the term $\|\boldsymbol{H}_1 - \boldsymbol{W}_2\boldsymbol{H}_2\|_F^2$. This procedure will be repeated until all of the layers have been initialized. The proposed model (15) is then alternately optimized to fine-tune each layer. The updating rules for the layers are derived as follows:

•*Updating basis matrix* $\boldsymbol{W}_i$

By keeping all factors constant (except $\boldsymbol{W}_i$) and introducing a Lagrangian multiplier $\boldsymbol{\Theta}_i$ to enforce nonnegativity on $\boldsymbol{W}_i$, the objective function presented in Eq. (15) becomes:

$$\begin{aligned} \min_{\boldsymbol{W}_i} &\|\boldsymbol{U} - \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\|_F^2 + \beta\|\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\|_F^2 - \mathrm{Tr}(\boldsymbol{\Theta}_i\boldsymbol{W}_i^\top) \\ &= \mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^\top + \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{W}_i^\top\boldsymbol{\Psi}_{i-1}^\top \\ &\quad - 2\boldsymbol{U}\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{W}_i^\top\boldsymbol{\Psi}_{i-1}^\top) + \beta\mathrm{Tr}(\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{W}_i^\top\boldsymbol{\Psi}_{i-1}^\top) \\ &\quad - \mathrm{Tr}(\boldsymbol{\Theta}_i\boldsymbol{W}_i^\top) \end{aligned} \tag{20}$$

where $\boldsymbol{U} = \boldsymbol{A}_S + \boldsymbol{R}$, $\boldsymbol{\Psi}_{i-1} = \boldsymbol{W}_1\boldsymbol{W}_2\cdots\boldsymbol{W}_{i-1}$ and $\boldsymbol{\Phi}_{i+1} = \boldsymbol{W}_{i+1}\boldsymbol{W}_{i+2}\cdots\boldsymbol{W}_l$. When $i = 1$, $\boldsymbol{\Psi}_0 = \boldsymbol{I}$ and when $i = l$ then $\boldsymbol{\Phi}_{i+1} = \boldsymbol{I}$.

By setting the partial derivative of (20) with respect to $\boldsymbol{W}_i$ to $\boldsymbol{0}$, we obtain:

$$\begin{aligned} \boldsymbol{\Theta}_i =& 2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top - 2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{U}\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top \\ &+ 2\beta\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \end{aligned} \tag{21}$$

Considering the complementary slackness condition of the KKT framework, we find:

$$\boldsymbol{\Theta}_i \odot \boldsymbol{W}_i = \boldsymbol{0}, \tag{22}$$

where the symbol $\odot$ denotes the element-wise product. Eq. (22) denotes the fixed point that the solution must meet to achieve convergence. Solving (22) yields the subsequent updating rule:

$$\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i \odot \frac{\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{U}\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top}{\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\boldsymbol{H}_l^\top\boldsymbol{\Phi}_{i+1}^\top + \beta\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top} \tag{23}$$

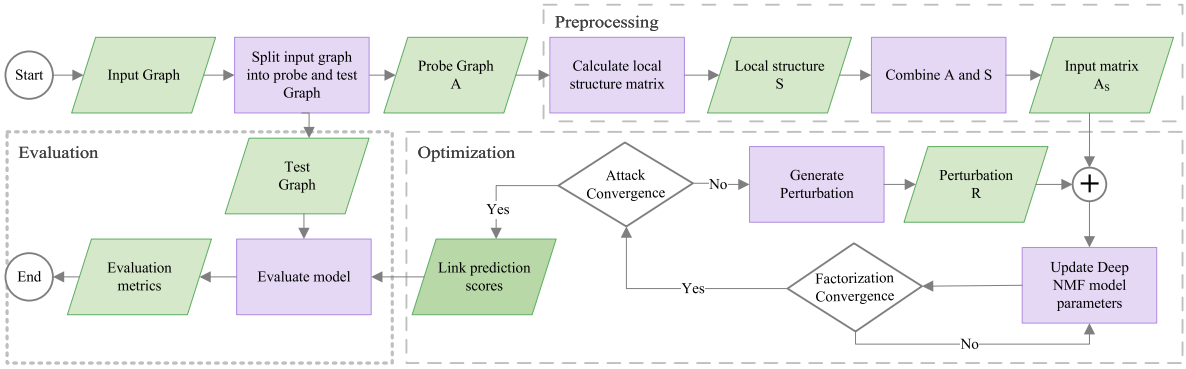


Fig. 3. Flowchart of the proposed link prediction model, encompassing preprocessing, optimization through adversarial training, and evaluation stages.

•*Updating latent matrix* $\boldsymbol{H}_l$

Similarly, we introduce a Lagrangian multiplier $\boldsymbol{\Xi}$ to enforce non-negativity on $\boldsymbol{H}_l$. By keeping all matrices fixed except for $\boldsymbol{H}_l$, the objective function in (15) can be expressed as:

$$\min_{\boldsymbol{H}_l} \|\boldsymbol{U} - \boldsymbol{\Psi}_l \boldsymbol{H}_l\|_F^2 + \beta \|\boldsymbol{H}_l\|_F^2 - \mathrm{Tr}(\boldsymbol{\Xi}_l \boldsymbol{H}_l^\top) \tag{24}$$
$$= \mathrm{Tr}(\boldsymbol{\Psi}_l \boldsymbol{H}_l \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top - 2\boldsymbol{U}\boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top) + 2\beta \mathrm{Tr}(\boldsymbol{H}_l \boldsymbol{H}_l^\top) - \mathrm{Tr}(\boldsymbol{\Xi}_l \boldsymbol{H}_l^\top)$$

By equating the partial derivative of (24) with respect to $\boldsymbol{H}_i$ to zero, we derive:

$$\boldsymbol{\Xi}_l = -2\boldsymbol{\Psi}_l^\top \boldsymbol{U} + 2\boldsymbol{\Psi}_l^\top \boldsymbol{\Psi}_l \boldsymbol{H}_l + 2\beta \boldsymbol{H}_l \tag{25}$$

Considering the complementary slackness condition of the KKT framework, we find:

$$\boldsymbol{\Xi}_l \odot \boldsymbol{H}_l = \boldsymbol{0}, \tag{26}$$

The update rule for $\boldsymbol{H}_l$ is obtained through a similar derivation process as the update rule for $\boldsymbol{H}_l$,

$$\boldsymbol{H}_l \leftarrow \boldsymbol{H}_l \odot \frac{\boldsymbol{\Psi}_l^\top \boldsymbol{U}}{\boldsymbol{\Psi}_l^\top \boldsymbol{\Psi}_l \boldsymbol{H}_l + \beta \boldsymbol{H}_l} \tag{27}$$

The optimization procedure for LPADNMF is outlined in Algorithm 1, with the *ShallowNMF* module performing the pre-training. In addition, the flowchart outlining the proposed link prediction model can be observed in Fig. 3. This schematic provides a visual representation of the sequential steps involved in our method to link prediction.

4.4.3. Termination conditions

The iterative minimization process for optimizing $\boldsymbol{W}$s and $\boldsymbol{H}_l$ is halted based on specific termination criteria. Initially, we focus on the criterion that determines the conclusion of the inner minimization process before updating a new $\boldsymbol{R}$. We compute the relative error of successive iterations, and the process concludes when the error falls below a predefined threshold ϵ_{in}. In other words, if $\boldsymbol{W}_i^{(out,in)}$ and $\boldsymbol{H}_l^{(out,in)}$ represent the iterate of $\boldsymbol{W}_i$ and $\boldsymbol{H}_l$ at the *out*-th outer iteration and *in*-th inner iteration, respectively, the process stops when the relative error satisfies the condition. The inner termination criterion is formally defined as follows,

$$\left\| \frac{\bar{\boldsymbol{A}}_S^{(out,in+1)} - \bar{\boldsymbol{A}}_S^{(out,in)}}{\bar{\boldsymbol{A}}_S^{(out,in)}} \right\|_F < \epsilon_{in} \tag{28}$$

where $\bar{\boldsymbol{A}}_S^{(out,in)} := \boldsymbol{W}_1^{(out,in)} \ldots \boldsymbol{W}_l^{(out,in)} \boldsymbol{H}_l^{(out,in)}$. Similarly, the outer stopping criterion can be defined as follows,

$$\left\| \frac{\bar{\boldsymbol{A}}_S^{(out+1,in)} - \bar{\boldsymbol{A}}_S^{(out,in)}}{\bar{\boldsymbol{A}}_S^{(out,in)}} \right\|_F < \epsilon_{out} \tag{29}$$

If the predefined stopping criteria are not met, the algorithm terminates both the inner and outer iterations when the specified practical iteration limits are reached.

Algorithm 1 Link prediction by Deep Adversarial NMF (LPDANMF)

Input: Train matrix $\boldsymbol{A}$, layers size $\boldsymbol{r}$, attack parameter λ, similarity parameter α, smoothness parameter β;
Output: $\boldsymbol{W}_i$ $(1 \le i < l)$, $\boldsymbol{H}_i$ $(1 \le i < l)$;

Construct the second-order matrix $\boldsymbol{S}$ according to (8);
Construct the input matrix $\boldsymbol{A}_S$ according to (9);
▷ **Pre-training:**
$\boldsymbol{W}_1, \boldsymbol{H}_1 \leftarrow$ ShallowNMF$(\boldsymbol{A}_S, r_1)$;
for $i = 2$ **to** l **do**
  $\boldsymbol{W}_i, \boldsymbol{H}_i \leftarrow$ ShallowNMF$(\boldsymbol{H}_{i-1}, r_i)$;
end for
▷ **Fine-tuning:**
while Condition (29) not met **do**
  Update $\boldsymbol{R}$ by (19)
  while Condition (28) not met **do**
    Update $\boldsymbol{D}$ by (17)
    for $i = 1$ **to** l **do**
      $\boldsymbol{\Psi}_{i-1} \leftarrow \prod_{\tau=1}^{i-1} \boldsymbol{W}_\tau (\boldsymbol{\Psi}_0 \leftarrow \boldsymbol{I})$;
      $\boldsymbol{\Phi}_{i+1} \leftarrow \prod_{\tau=i+1}^{l} \boldsymbol{W}_\tau (\boldsymbol{\Phi}_{l+1} \leftarrow \boldsymbol{I})$;
      Update $\boldsymbol{W}_i$ by (23);
      $\boldsymbol{\Psi}_i \leftarrow \boldsymbol{\Psi}_{i-1} \boldsymbol{W}_i$;
      Update $\boldsymbol{H}_l$ by (27);
    end for
  end while
end while
return the reconstructed network $\bar{\boldsymbol{A}}_S = \boldsymbol{W}_1 \ldots \boldsymbol{W}_l \boldsymbol{H}_l$.

4.5. Complexity analysis

The proposed model consists of two stages: a pre-training phase and a fine-tuning phase, as described in Algorithm 1. Thus, we examine the computational complexity for each phase separately. To facilitate this analysis, we use certain notations: let k represent the maximum size of any layer, n represent the number of nodes, and l represent the total number of layers in the model. The pre-training phase, which is less computationally intensive, utilizes the *ShallowNMF* algorithm. For this phase, the computational complexity for each layer is $\mathcal{O}(n^2 k)$. Consequently, for all l layers, the computational complexity is $\mathcal{O}(ln^2 k)$. If the shallow model achieves convergence after t_p iterations, the overall computational complexity of the pre-training phase is given by: $T_{\text{pretrain}} = \mathcal{O}(t_p l n^2 k)$. Similarly, in the fine-tuning phase, the computational complexity for the learner is: $\mathcal{O}(t_{\text{in}} l n^2 k)$ and the computational complexity for the attacker is: $\mathcal{O}(t_{\text{out}} n^2 k)$ where t_{in} and t_{out} represent the number of iterations required for the learner and the attacker, respectively. Therefore, the total computational complexity of the fine-tuning phase is given by: $T_{\text{finetune}} = O(t_{\text{out}}(t_{\text{in}} l n^2 k + n^2))$ To sum up, the overall computational complexity for LPADNMF is: $T_{\text{total}} = T_{\text{pretrain}} + T_{\text{finetune}} = \mathcal{O}(t_p l n^2 k) + \mathcal{O}(t_{\text{out}}(t_{\text{in}} l n^2 k + n^2))$. This complexity

is comparable to related deep models, particularly when considering the input data size and the number of layers, making LPADNMF a competitive alternative in terms of computational efficiency.

5. Experimental results

This section evaluates the effectiveness of LPADNMF model against 13 similarity based and embedded based methods. We evaluate the overall effectiveness of the proposed model in comparison to all other approaches, assess the robustness of LPADNMF across all datasets, and verify the model convergence.

5.1. Evaluation criteria

We utilize two evaluation metrics, specifically AUC (Area Under the Curve) (Hanley and McNeil, 1982) and Precision (Herlocker et al., 2004), to evaluate the effectiveness of the investigated link prediction methods. The definitions of these metrics are as follows:

- AUC (Area Under the Curve) signifies the likelihood that the scores of a randomly chosen missing link (i.e., a link in E^P) will surpass the scores of a randomly selected non-existent connection. In each instance, we select a random set of missing and non-existent connections to compare their similarity scores. If n independent comparisons are conducted, where n_1 is the number of missing links with a score higher than the non-existent connections and n_2 is the number of missing links with a score equal to the non-existent links, the general definition of the AUC metric is as follows:

$$\text{AUC} = \frac{n_1 + 0.5 \times n_2}{n}. \tag{30}$$

- Precision is defined as the ratio of correctly selected relevant links to the total number of chosen links, given a score ranking for unobserved links obtained through an algorithm. If we consider the top-L links as the predicted links and there are only l_c correctly predicted links in the probe set E^P, then the Precision value equals:

$$\text{Precision} = \frac{l_c}{L}. \tag{31}$$

- Recall is defined as the ratio of m existing links in all links M, namely:

$$\text{Recall} = \frac{m}{M}. \tag{32}$$

- The F-Measure, also known as the F1-score, is a metric that combines both Precision and Recall to provide a single value representing a balance between the two. Precision and Recall are often in tension: increasing precision might decrease recall and vice versa. The F-Measure helps to find a middle ground by calculating the harmonic mean of Precision and Recall, giving a more holistic view of an algorithm's effectiveness and is calculated as:

$$\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{33}$$

5.2. Datasets

This study employs eight real-world networks from various fields, encompassing biological, social, and technical networks (Rossi and Ahmed, 2015), to assess the performance of the proposed method. This section outlines the characteristics of the networks used in our experiments, and Table 2 presents basic statistical properties.

Table 2
Outline of the structural attributes of eight networks. $|V|$ denotes the node count, while $|E|$ indicates the number of edges. M represents the maximum degree, $\bar{d}$ stands for the average degree, CC signifies the clustering coefficient, and S denotes the sparsity coefficient, in that order.

Dataset	$\|V\|$	$\|E\|$	M	$\bar{d}$	CC	S
C.elegans	297	2345	134	15	0.31	5.317
Usair97	332	2126	139	12	0.63	3.858
Metabolic	453	2040	239	9	0.65	1.988
Email-Univ	1133	5451	71	9	0.22	0.849
Bio-Sc-Gt	1715	33 987	549	39	0.35	2.311
Bio-Ce-Gn	2219	53 683	242	48	0.18	2.18
SciMet	3084	10 413	164	6	0.15	0.219
Kohonen	4469	12 731	740	5	0.21	0.127

- C.elegans (Duch and Arenas, 2005): This network models the neural connections within the C.elegans worm's brain. Each neuron is represented as a node, and synaptic connections are depicted as links. The dataset comprises 297 nodes and 2,345 links.
- USAir97 (Colizza et al., 2007): This dataset includes 332 nodes and 2,126 links, depicting the U.S. aviation network, where airports serve as nodes and flight routes as connections.
- Metabolic (Rossi and Ahmed, 2015): This dataset illustrates the metabolic processes of the Escherichia Coli bacteria. Nodes represent metabolites, and directed connections symbolize product reactions. It consists of 453 nodes and 2,040 links.
- Email-Univ (Rossi and Ahmed, 2015): The network is derived from email data within a European research institution. The dataset contains 1,133 nodes, representing researchers, and 5,451 links, indicating the communication through emails among them.
- Bio-Sc-Gt (Cho et al., 2014): This dataset represents biological interactions, where nodes correspond to genes and links indicate genetic interactions. It includes 1,715 nodes and a substantial 33,987 links.
- Bio-Ce-Gn (Cho et al., 2014): This network focuses on biological networks and highlights interactions between bacteria genes and Archaea organisms. Nodes symbolize genes, and there are 2,219 of them, connected by 53,676 links.
- SciMet (Rossi and Ahmed, 2015): This dataset is related to citation analysis. Its graph includes 3,084 nodes representing individual papers and 10,413 links indicating citations among them.
- Kohonen (Rossi and Ahmed, 2015): The Kohonen dataset is an example network generated using the Histcitep program. It comprises 4,469 nodes and 12,731 links, forming a comprehensive network.

5.3. Results

In this experiment, we create the probe set E^P by removing 10% of the links, while the remaining links form the training set E^T, which is used to construct an observable network. Subsequently, for each method applied, we calculate the scores for all pairs of nodes within the observed network that are currently unlinked. These pairs are then arranged in descending order based on their likelihood of forming connections. We employ AUC and Precision as the evaluation metrics for predicting missing links. Tables 3 and 4 present the results for both the proposed and comparative methods. As it can be seen, our method demonstrates superior performance compared to other in most cases. This result demonstrates that generally speaking, the adversarial learning strategy has outperformed other methods in terms of optimizing factorization and predicting missing connections. In Table 3, the proposed model exhibits improvements over the second-best method in term of AUC by 4.61%, 1.83%, 1.41%, 1.50%, 3.31%, 1.47% for the *Celegans*, *Metabolic*, *Email-univ*, *Bio-Sc-Gt*, *Bio-Ce-Gn* and *Kohonen* datasets, respectively. Table 4 illustrates the Precision values of all methods across eight datasets. Notably, the LPADNMF method

Table 3
AUC performance of the similarity-based and NMF-based methods. The results correspond to distinct executions with a random training set of 90% and a probing set partitioning of 10%. The best results are emphasized in **bold**, whereas the second-bests are indicated in underline style.

Methods	C.elegans	Usair97	Metabolic	Email-Un	Bio-Sc-Gt	Bio-Ce-Gn	SciMet	Kohonen
Jaccard	0.8007	0.9081	0.7958	0.8575	0.9407	0.8771	0.8006	0.8040
CN	0.8522	0.9446	0.9304	0.8596	0.9469	0.8861	0.8032	0.8211
RA	0.8779	**0.9677**	0.9719	0.8612	0.9549	0.8914	0.8042	0.8263
AA	0.8698	0.9594	0.9661	0.8618	0.9511	0.8881	0.8042	0.8264
PA	0.7559	0.8810	0.8182	0.7851	0.8719	0.8508	0.8208	0.8595
NMF	0.8667	0.9224	0.8646	0.8751	0.9625	0.9428	0.8815	0.8658
SASNMF	0.8929	0.9385	0.8925	0.8807	0.9568	0.9402	0.8900	0.8780
GWNMF	0.6955	0.8689	0.7399	0.8128	0.8874	0.8057	0.8130	0.8213
NMFKL	0.8533	0.9136	0.9048	0.8882	0.9386	0.8968	0.8949	0.8657
RC-NMF	0.8952	0.9455	0.9143	0.8968	0.9624	0.9461	0.9006	0.8870
MS-RNMF	0.8808	0.9454	0.9169	0.8742	0.9647	0.9418	0.8916	0.8819
FSSDNMF	0.8824	0.9300	0.8820	0.8850	0.9623	0.9365	0.8852	0.8918
LRNP	0.7904	0.9429	0.8806	0.7907	0.9323	0.8202	0.6732	0.7072
LPANMF	0.9197	0.9549	0.9330	0.9110	0.9688	0.9533	**0.9171**	0.8989
DANMFL	0.8215	0.9160	0.7914	0.8781	0.9467	0.9110	0.8851	0.8607
LPADNMF	**0.9621**	0.9607	**0.9897**	**0.9238**	**0.9833**	**0.9849**	0.9091	**0.9121**

Table 4
Precision performance of the similarity-based and NMF-based methods. The results correspond to distinct executions with a random training set of 90% and a probing set partitioning of 10%. The best results are emphasized in **bold**, whereas the second-bests are indicated in underline style.

Methods	C.elegans	Usair97	Metabolic	Email-Un	Bio-Sc-Gt	Bio-Ce-Gn	SciMet	Kohonen
Jaccard	0.0199	0.0939	0.0835	0.0917	0.1377	0.0408	0.0164	0.0012
CN	0.0660	0.3615	0.1597	0.1064	0.2620	0.0923	0.0529	0.0590
RA	0.0647	0.3709	0.1695	0.0771	0.3038	0.0853	0.0385	0.0437
AA	0.0597	0.3709	0.1548	0.1083	0.2724	0.0957	0.0529	0.0528
PA	0.0489	0.3038	0.1124	0.0185	0.0858	0.0356	0.0121	0.0282
NMF	0.1286	0.3756	0.2882	0.1385	0.4853	0.2996	0.0933	0.0813
SASNMF	0.1505	0.4296	0.3251	0.1349	0.4949	0.2916	0.0938	0.0896
GWNMF	0.0777	0.3404	0.0813	0.0633	0.3046	0.1015	0.0144	0.0365
NMFKL	0.0801	0.3451	0.1281	0.1394	0.2449	0.1054	0.0740	0.0758
RC-NMF	0.1650	0.4366	0.3153	0.1440	0.4818	0.2935	0.0942	0.0899
MS-RNMF	0.1529	0.4390	0.2906	0.1440	0.4981	0.2930	0.0928	0.0797
FSSDNMF	0.1508	0.4624	0.3153	0.1450	0.4794	0.2833	0.0766	0.0849
LRNP	0.0950	0.4170	0.1555	0.1456	0.3293	0.1355	0.0963	0.0826
LPANMF	0.1633	**0.4648**	**0.3432**	0.1303	0.5106	0.2692	0.0963	0.0861
DANMFL	0.1515	0.4178	0.2709	0.1156	0.4007	0.1993	0.1011	0.0750
LPADNMF	**0.1667**	0.4635	0.3227	**0.1615**	**0.5172**	**0.3158**	**0.1046**	**0.0906**

Table 5
Recall performance of the similarity-based and NMF-based methods. The results correspond to distinct executions with a random training set of 90% and a probing set partitioning of 10%. The best results are emphasized in **bold**, whereas the second-bests are indicated in underline style.

Methods	C.elegans	Usair97	Metabolic	Email-Un	Bio-Sc-Gt	Bio-Ce-Gn	SciMet	Kohonen
Jaccard	0.1848	0.3911	0.1055	0.2584	0.4833	0.2467	0.0926	0.0046
CN	0.3399	0.567	0.367	0.3222	0.5325	0.281	0.2216	0.2353
RA	0.3515	0.6802	0.5565	0.3418	0.5982	0.289	0.2303	0.268
AA	0.3603	0.6011	0.5082	0.3507	0.5509	0.2853	0.2491	0.2805
PA	0.2793	0.563	0.3518	0.1096	0.3203	0.1839	0.0706	0.1367
NMF	0.8433	0.8819	0.8194	0.6212	0.7715	0.6134	0.4787	0.5514
SASNMF	0.8168	0.8584	0.8103	0.6209	0.7318	0.5949	0.4852	0.5314
GWNMF	0.6634	0.7498	0.692	0.3107	0.6479	0.4191	0.1883	0.1901
NMFKL	0.3801	0.5753	0.387	0.3791	0.5489	0.3001	0.261	0.246
RC-NMF	0.8189	0.8725	0.8079	0.6139	0.7562	0.5957	0.4764	0.5432
MS-RNMF	0.5766	0.6823	0.4165	0.2932	0.5974	0.3697	0.107	0.186
FSSDNMF	0.8708	0.8963	0.8445	0.6628	0.7935	0.6445	0.521	0.5887
LRNP	0.3737	0.5858	0.4042	0.0170	0.0528	0.0372	0.0069	0.0057
LPANMF	0.855	0.8909	0.8391	0.6219	0.7872	0.6193	0.4797	0.5454
DANMFL	0.679	0.6762	0.6927	0.6207	0.6267	0.4067	0.5071	0.4936
LPADNMF	**0.8857**	**0.9198**	**0.8718**	**0.6829**	**0.8094**	**0.6512**	**0.5352**	**0.5963**

demonstrates significant improvements over the second-best method by 1.03%, 10.92%, 1.27%, 5.41%, 8.62%, and 0.78% for the *Celegans*, *Email-univ*, *Bio-Sc-Gt*, *Bio-Ce-Gn*, *SciMet*, and *Kohonen* datasets, respectively. Furthermore, the improvement rate for the Recall criterion, as presented in Table 5, is 1.71%, 2.62%, 3.23%, 3.03%, 2%, 1.04%, 2.73%, and 1.29%, respectively, for the following datasets: *Celegans*, *Usair97*, *Metabolic*, *Email-univ*, *Bio-Sc-Gt*, *Bio-Ce-Gn*, *SciMet*, and *Kohonen*.

The reason for the relatively lower performance of latent-based methods in the *Usair97* and *Metabolic* datasets can be due to the specific structure of those networks. These datasets have a greater clustering coefficient than other datasets, as seen in Table 2. These methods evidently do not leverage the high clustering coefficient of networks when discussing link prediction. As the clustering coefficient increases, their performance declines (Robledo et al., 2022). Nevertheless, the

Table 6
F-Measure performance of the similarity-based and NMF-based methods. The results correspond to distinct executions with a random training set of 90% and a probing set partitioning of 10%. The best results are emphasized in **bold**, whereas the second-bests are indicated in underline style.

Methods	C.elegans	Usair97	Metabolic	Email-Un	Bio-Sc-Gt	Bio-Ce-Gn	SciMet	Kohonen
Jaccard	0.0359	0.1514	0.0932	0.1354	0.2143	0.0700	0.0279	0.0019
CN	0.1105	0.4415	0.2226	0.1600	0.3512	0.1390	0.0854	0.0943
RA	0.1093	0.4800	0.2599	0.1258	0.4030	0.1317	0.0660	0.0751
AA	0.1024	0.4587	0.2373	0.1655	0.3645	0.1433	0.0873	0.0889
PA	0.0832	0.3946	0.1704	0.0317	0.1353	0.0597	0.0207	0.0468
NMF	0.2232	0.5268	0.4264	0.2265	0.5958	0.4026	0.1562	0.1417
SASNMF	0.2542	0.5726	0.4640	0.2216	0.5905	0.3914	0.1572	0.1533
GWNMF	0.1391	0.4682	0.1455	0.1052	0.4144	0.1634	0.0268	0.0612
NMFKL	0.1323	0.4314	0.1925	0.2038	0.3387	0.1560	0.1153	0.1159
RC-NMF	0.2747	0.5820	0.4536	0.2333	0.5886	0.3932	0.1573	0.1543
MS-RNMF	0.2417	0.5343	0.3423	0.1931	0.5432	0.3269	0.0994	0.1116
FSSDNMF	0.2571	0.6101	0.4592	0.2379	0.5977	0.3936	0.1336	0.1484
LRNP	0.1515	0.4872	0.2246	0.0304	0.0910	0.0584	0.0129	0.0107
LPANMF	0.2742	0.6109	**0.4872**	0.2155	0.6194	0.3753	0.1604	0.1487
DANMFL	0.2477	0.5165	0.3895	0.1949	0.4888	0.2675	0.1686	0.1302
LPADNMF	**0.2806**	**0.6164**	0.4710	**0.2612**	**0.6311**	**0.4253**	**0.1750**	**0.1573**

Table 7
Evaluation results for the LPADNMF with different number of layers.

Size of layers		Dataset							
		C.elegans	*Usair97*	*Metabolic*	*Email-Univ*	*Bio-Sc-Gt*	*Bio-Ce-Gn*	*SciMet*	*Kohonen*
[512, 256, 16]	AUC	0.9611	0.9500	0.9524	0.8980	0.9750	0.9755	0.8997	0.8933
	Prec	0.1338	0.4319	0.2826	0.1229	0.4244	0.1829	0.0751	0.0748
	Rec	0.5717	0.7801	0.6293	0.3850	0.6467	0.4129	0.2829	0.3432
	F1	0.2168	0.5560	0.3900	0.1863	0.5125	0.2535	0.1187	0.1228
[512, 256, 32]	AUC	0.9612	0.9548	0.9753	0.9136	0.9791	0.9778	0.9011	0.8955
	Prec	0.1475	0.3803	0.2899	0.1394	0.4872	0.2260	0.0733	0.0713
	Rec	0.6820	0.8288	0.7306	0.4640	0.7094	0.4846	0.3472	0.4309
	F1	0.2425	0.5214	0.4151	0.2144	0.5777	0.3082	0.1210	0.1224
[512, 256, 64]	AUC	**0.9621**	0.9578	**0.9897**	0.9143	0.9821	0.9846	**0.9091**	0.9060
	Prec	**0.1667**	0.4022	**0.3227**	0.1462	0.5094	0.2783	**0.1046**	0.0775
	Rec	**0.8857**	0.8822	**0.8718**	0.5636	0.7592	0.5658	**0.5352**	0.4987
	F1	**0.2806**	0.5525	**0.4710**	0.2322	0.6097	0.3731	**0.1750**	0.1342
[512, 256, 128]	AUC	0.9528	**0.9607**	0.9711	**0.9238**	**0.9833**	**0.9849**	0.9054	**0.9121**
	Prec	0.1565	**0.4635**	0.3048	**0.1615**	**0.5172**	**0.3158**	0.0998	**0.0906**
	Rec	0.7986	**0.9198**	0.8039	**0.6829**	**0.8094**	**0.6512**	0.4317	**0.5963**
	F1	0.2617	**0.6164**	0.4420	**0.2612**	**0.6311**	**0.4253**	0.1621	**0.1573**

results presented in Tables 3 to 6 demonstrate that the proposed deep model surpasses both the latent-based and similarity-based approaches. This advantage arises from its remarkable ability to adeptly capture the complex structures present in these networks. The information depicted in Fig. 4, represented by a radar chart, provides a concise summary of the results presented in Tables 3 to 6. It also highlights a comparison between the proposed method and NMF-based methods based on four metrics, AUC, Precision, Recall, and F-Measure across eight different datasets.

5.4. Parameter analysis

In this subsection, we explore how the dimensions of hidden layers and different hyperparameters affect the performance of the proposed model. To thoroughly study the three hyperparameters, mentioned in Section 4, we assessed the proposed model with different parameters values across all datasets using a grid-search approach. In the case of the loss function (15), LPADNMF involves λ, α, and β hyperparameters. We varied α as $\{0, 1, 1.1, 1.2, 1.3, 1.4, 1.5\}$, β as $\{0, 0.5, 1, 1.5, 2\}$, and λ as $\{1.5, 2, 2.5, 5\}$.

5.4.1. Impact of latent space dimension and layers

This section focuses on detailing the dimension and layers of the latent space. In the context of a low-rank approximation problem, the precision and computational cost of the proposed method are directly influenced by the number of layers and the size of the latent space. Therefore, choosing the optimal size for the latent space is of utmost importance. In order to exclusively investigate the efficacy of the latent space of layers, we excluded the influence of other components by assigning zero values to the parameters α and β. Additionally, we set a high value for the parameter λ to eliminate adversarial perturbations. According to Eq. (15), a high value of λ results in a significantly reduced attacker effect, nearly approaching zero. For our experiment, we configured the model with three layers. We observed that altering the sizes of the first two hidden layers did not significantly impact the model's performance. As a result, we fixed the sizes of the first two layers at [512, 256], while allowing the size of the third layer to vary based on the characteristics of network. Specifically, we experimented with different third layer sizes, choosing from the set $\{16, 32, 64, 128\}$. The results of these experiments are detailed in Table 7. From this results, we can derive that the configuration $[512, 256, 128]$ consistently performs well across most datasets, achieving either the highest or competitive evaluation metrics values. This suggests that $[512, 256, 128]$ could be a robust choice when optimizing the LPADNMF model for a balance between evaluation metrics values, especially if the goal is to perform well across a variety of datasets.

5.4.2. Impact of α, β and λ

In this section, we analyze the impact of the three remaining parameters by applying a grid search while keeping the optimal size of the

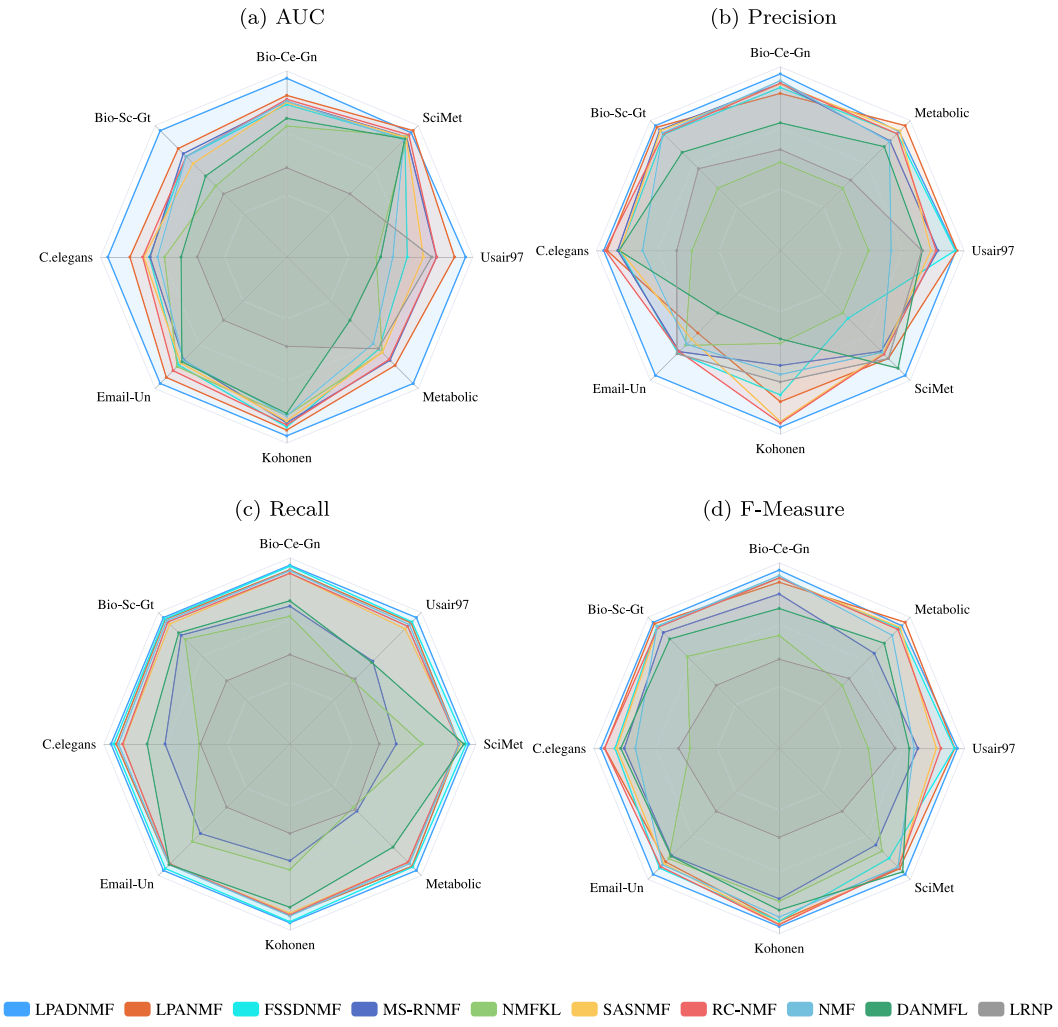


Fig. 4. Comparing the proposed method with NMF-based methods in terms of *AUC*, *Precision*, *Recall*, and *F-Measure* metrics. The chart presents the normalized results of this comparison, with values ranging between 0.5 and 1.

latent layer constant, as determined in Section 5.4.1. According to (15), LPADNMF has three hyperparameters, λ, α, and β, which correspond to the attack intensity, compromise parameter between the local and global structures, and Frobenius regularization terms of basis matrices and coefficients, respectively. Figs. 5 and 6 illustrate the analysis of these parameters on the eight datasets. Each subfigure has 140 evaluation points corresponding to seven distinct αs, five distinct βs and four distinct λs. In these figures, performance is illustrated using a range of colors. Warm hues (such as orange to yellow) indicate high AUC values, while cool hues (like blue to dark blue) indicate lower values for the measure.

As we can observe in Figs. 5 and 6, optimal results were obtained with higher α parameter values for the *C.elegans*, *SciMet*, *Kohonen* datasets, whereas, for the other five datasets, the median α value generally yielded superior results. Additionally, it is worth noting that in most datasets, higher AUC results were achieved when both β and λ values were set to higher levels. In Table 8, the optimal parameter values for assessing AUC, Precision, Recall, and F-Measure for eight datasets are shown.

5.5. Robustness analysis

This section assess the robustness of the proposed LPADNMF model. We aim to vary the input sparsity by removing a portion of the links from the original dataset randomly. Hence, we conducted evaluations on the training sets with observation rates spanning from 20% to 90%. The experimental results are presented in Fig. 7. Some analyses are outlined below:

- The Fig. 7 illustrates that, for the majority of cases, the AUC curve of LPADNMF consistently surpasses those of the compared methods. This indicates that the proposed approach is more robust than other predictors across a range of training set ratios.
- From Fig. 7 we can derive that as the proportion of the training set decreases from 90% to 20%, the number of missing links increases. However, LPADNMF still outperforms other methods even when the number of observed links is limited. Considering the sparsity of datasets in the real world, this advantage becomes crucial.

5.6. Convergence analysis

Based on the derived updating rules outlined earlier, our optimization algorithm iteratively optimizes the objective function and adjusts the matrices. This leads to the ongoing convergence of the objective function values. To better visualize the evolution of the cost function's convergence throughout the iterations, the convergence rate across the four datasets is illustrated in Fig. 8. As you can see, every 80 minimization iterations, a model attack is executed. The loss exhibits a long-term, non-monotonic pattern because the Majorization-Minimization

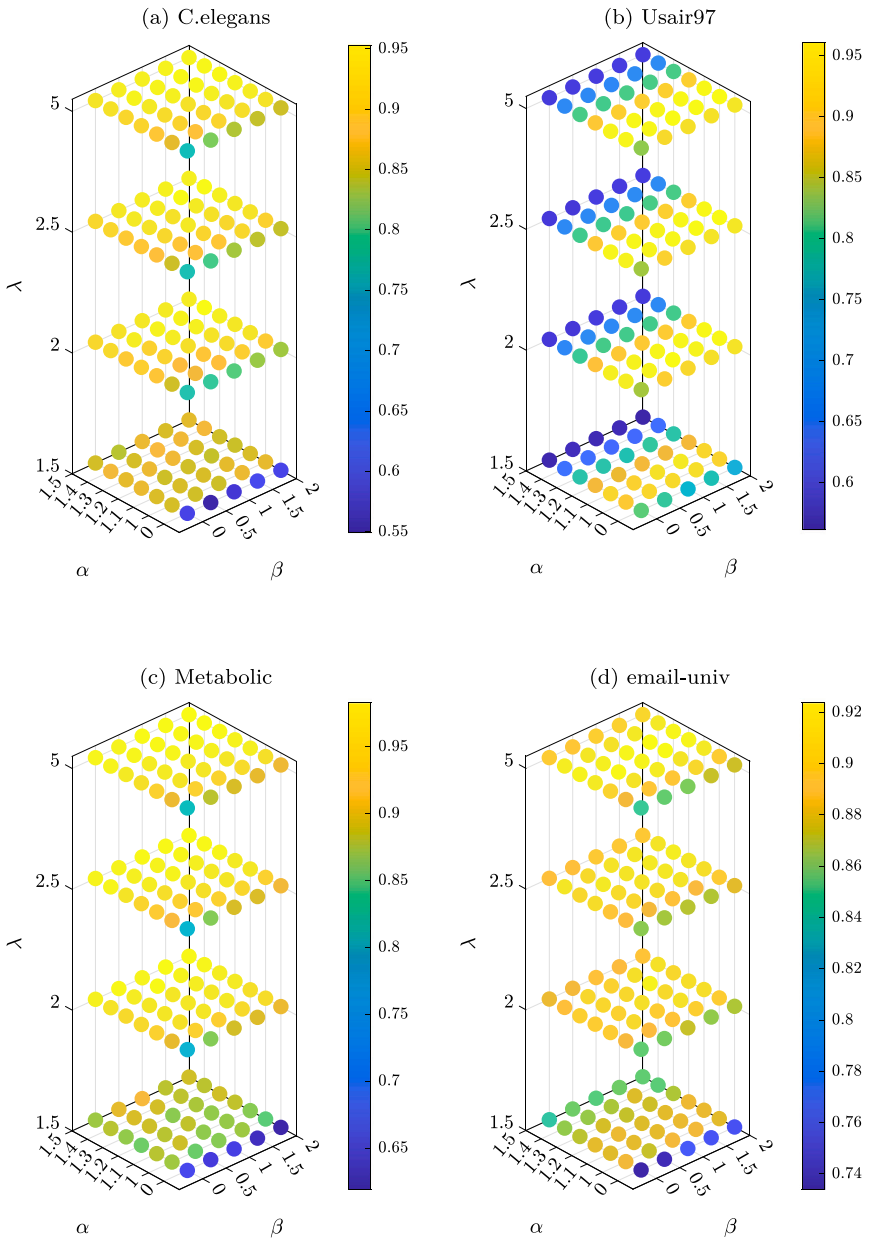


Fig. 5. Parameter analysis of LPADNMF method on four datasets concerning the *AUC* measure.

Table 8
The optimal value of hyperparameters for assessing *AUC*, *Precision*, *Recall*, and *F-Measure* for eight datasets.

Dataset	C.elegans		Usair97		Metabolic		Email-Univ		Bio-Sc-Gt		Bio-Ce-Gn		SciMet		Kohonen	
Metric	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC	Prec
α	1.5	1.5	1	0	1.2	1.5	1.3	1.2	1.2	1.1	1.2	1	1.4	1.1	1.5	1.1
β	2	2	2	2	2	2	1.5	1.5	1.5	2	2	2	1.5	1	1.5	2
λ	5	5	5	5	2	5	5	5	5	5	5	5	5	5	5	5
Metric	Rec	F1	Rec	F1	Rec	F1	Rec	F1	Rec	F1	Rec	F1	Rec	F1	Rec	F1
α	1	1.4	1	1	1.3	1.5	1.3	1.2	0	1	1	0	1.5	1.5	0	1.1
β	1	2	1.5	2	1	2	0	2	2	2	1.5	2	2	2	1.5	0.5
λ	5	5	5	5	2.5	5	5	5	5	5	5	5	5	5	5	5

property holds for both $\boldsymbol{R}$ and the factors in LPADNMF. As depicted in the Figures, there are minor fluctuations in the loss, demonstrating the attack impact on the optimization. This non-monotonic pattern helps avoid getting stuck in local optima by alternating between minimization and maximization steps, allowing the process to explore different regions of the loss landscape. Adversarial training with minimax optimization also improves generalization by having the learner and attacker constantly challenge each other, becoming more robust and less prone to overfitting the training data as they oscillate around the saddle point solution (Xing et al., 2021).

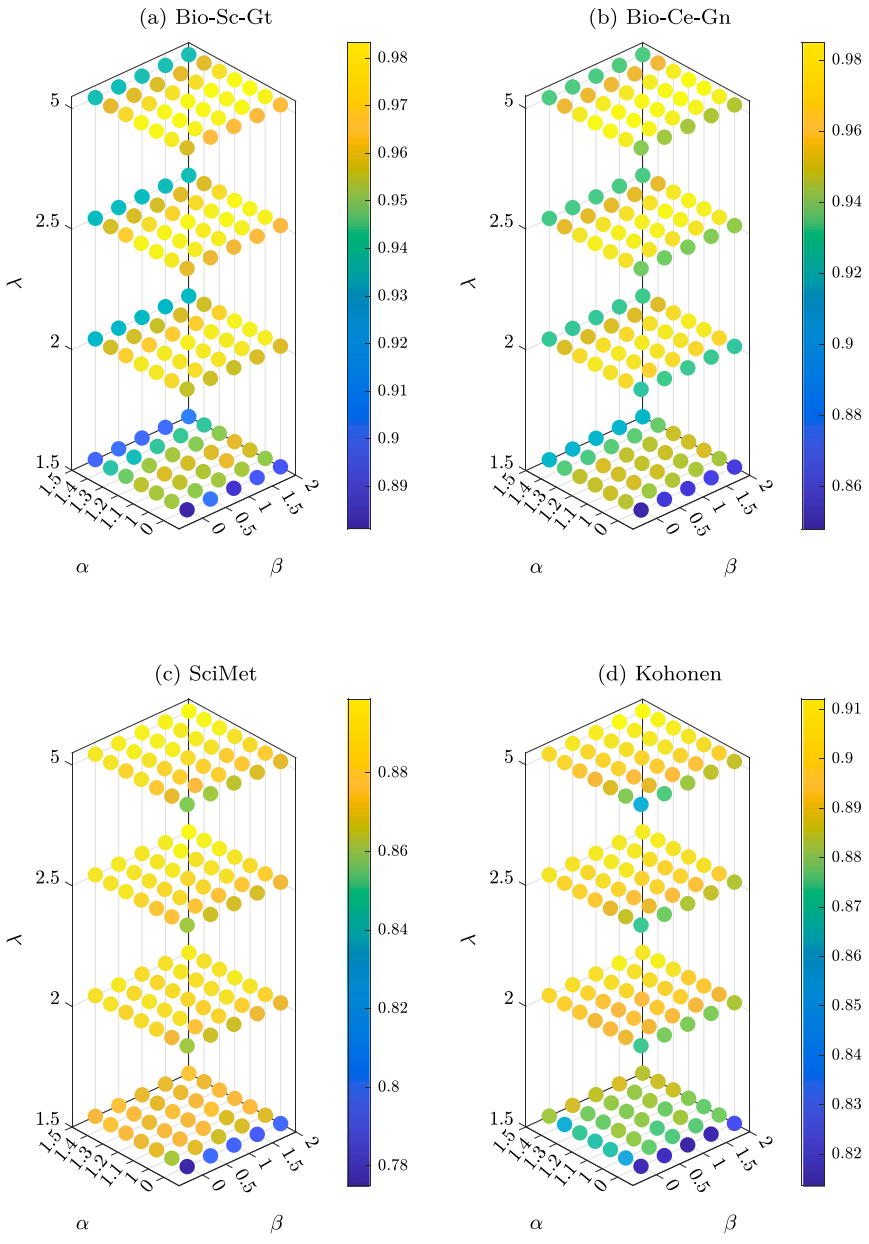


Fig. 6. Parameter analysis of LPADNMF method on four datasets concerning the *AUC* measure.

6. Conclusion

This paper introduced a modified adversarial method in the Deep NMF framework, namely the Link Prediction using Adversarial Deep NMF (LPADNMF), to enhance generalization in sparse graphs for link prediction. By incorporating adversarial training with a bounded attack on the input matrix using the sparse $L_{2,1}$ norm, the model achieves robustness against perturbations and noises in the data. Additionally, the inclusion of second-order similarity and smooth L_2 regularization further improves the performance of the proposed method. The efficient Majorization-Minimization algorithm is utilized to implement this adversarial training, while the Multiplicative Update Rules algorithm is employed to attain the optimal solution for the Deep NMF model. Experiments illustrate the efficacy of the LPADNMF model, showcasing its ability to outperform existing state-of-the-art models in the link prediction task on various datasets. These findings validate the effectiveness of incorporating adversarial training and second-order similarity into the matrix factorization framework.

The limitation of the current work is the handling of signed graphs and the non-negativity constraint in Deep NMF, which restricts its applicability to graphs with negative edges. Future work could explore extending the model to consider negative values by relaxing the non-negativity constraint of Deep NMF (Deep Semi-NMF). Moreover, it is well-established that the MUR optimization algorithm can be computationally intensive, leading to slower convergence times. As an avenue for future exploration, we could investigate the adoption of the Hierarchical Alternating Least Squares (HALS) optimization technique, which has been shown to offer improved computational efficiency. In addition, we recommend exploring the extension of the LPADNMF model to handle dynamic and evolving graphs. Adapting the model to time-dependent networks would open up new possibilities for predicting links in temporal graphs and capturing temporal patterns in network evolution. Furthermore, investigating the LPADNMF method's performance on heterogeneous graphs and exploring its potential applications in other domains such as recommendation systems or social network analysis would be promising directions for further research.

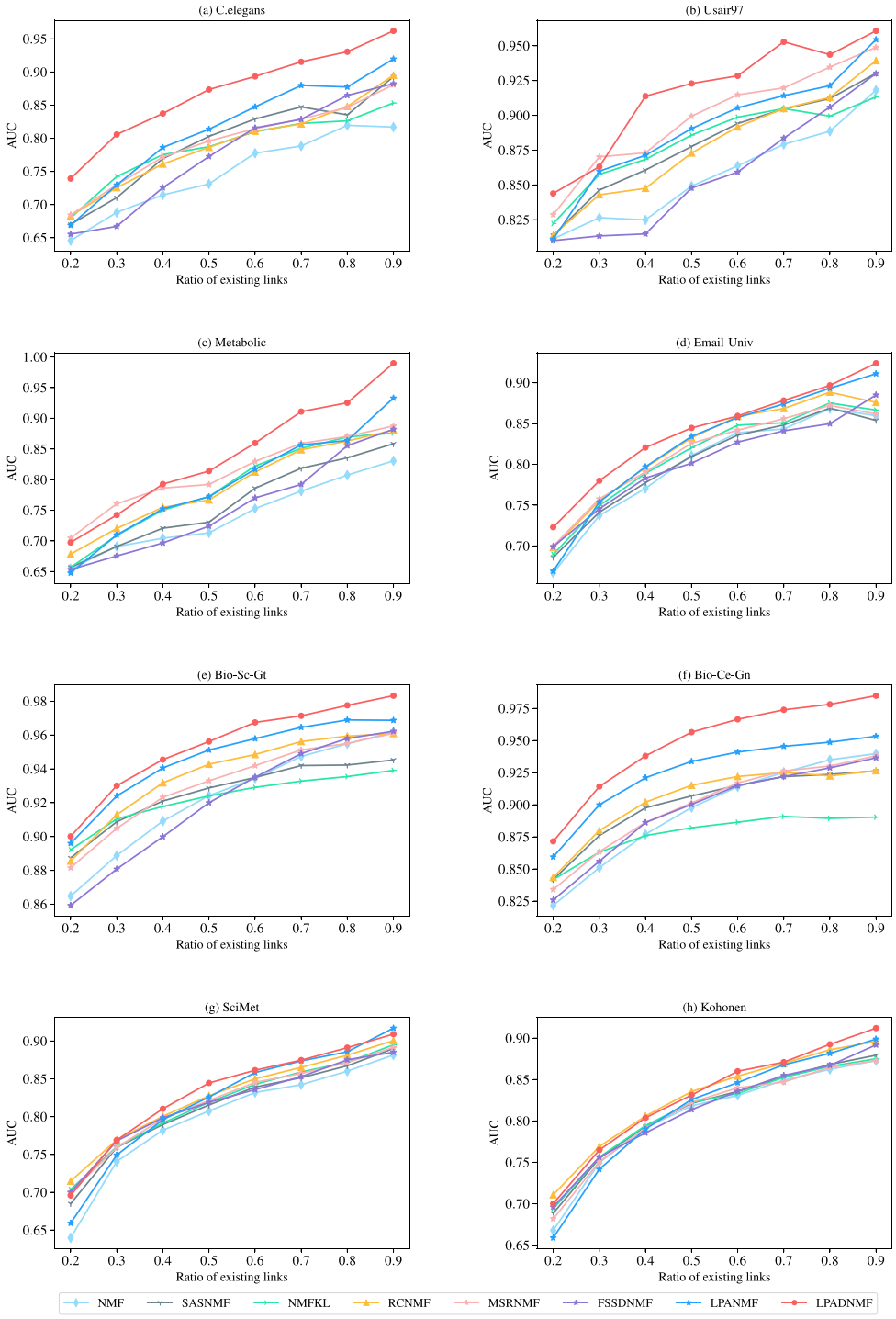


Fig. 7. Examining the *AUC* across eight networks with varying ratios of existing links (ranging from 20% to 90%). Subfigures (a) to (h) depict distinct graphs. In addition to our method (LPADNMF), we extend the comparison to seven other NMF-based methods. Each data point represents the maximum value obtained from multiple runs.

CRediT authorship contribution statement

Reza Mahmoodi: Writing – original draft, Visualization, Software, Methodology, Data curation. **Seyed Amjad Seyedi:** Writing – review & editing, Software, Methodology, Investigation, Conceptualization. **Alireza Abdollahpouri:** Writing – review & editing, Validation, Supervision, Formal analysis. **Fardin Akhlaghian Tab:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

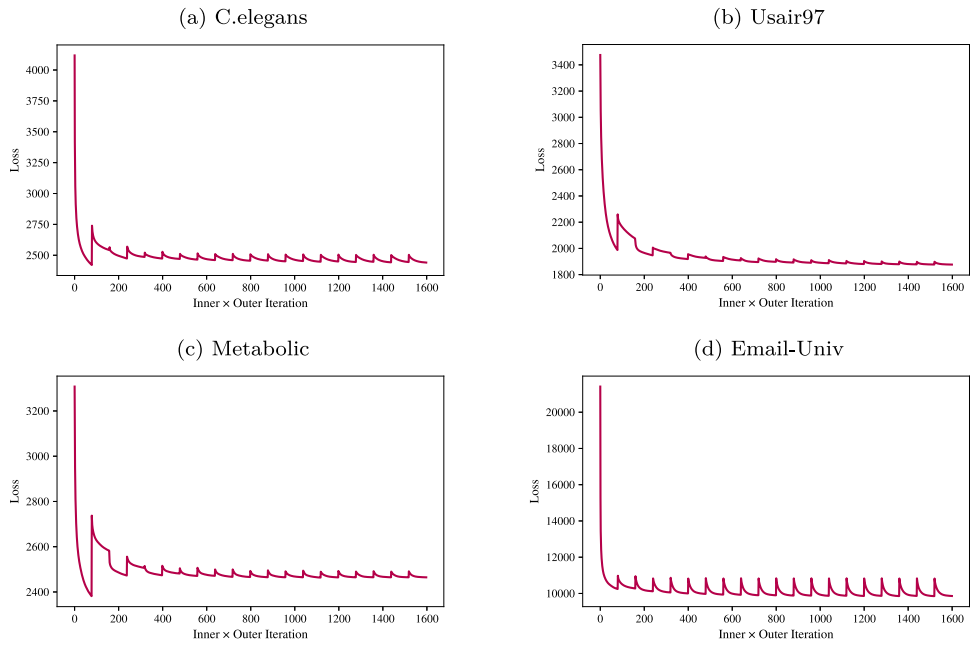


Fig. 8. Convergence analysis of Adversarial Deep NMF (LPADNMF) on the four datasets. An attack is applied to the model after every 80 minimizing iterations.

References

Abdollahi, R., Amjad Seyedi, S., Reza Noorimehr, M., 2020. Asymmetric semi-nonnegative matrix factorization for directed graph clustering. In: 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 323–328.

Adamic, L.A., Adar, E., 2003. Friends and neighbors on the web. Social Networks 25 (3), 211–230.

Ahmed, N.M., 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 Min. Anal. 1 (1), 19–33.

Bai, S., Fang, S., Li, L., Liu, R., Chen, X., 2019. Enhancing link prediction by exploring community membership of nodes. Internat. J. Modern Phys. B 33 (31), 1950382.

Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Cai, T., Tan, V.Y.F., Févotte, C., 2021. Adversarially-trained nonnegative matrix factorization. IEEE Signal Process. Lett. 28, 1415–1419.

Chai, L., Tu, L., Wang, X., Chen, J., 2022. Network-energy-based predictability and link-corrected prediction in complex networks. Expert Syst. Appl. 207, 118005.

Chai, L., Tu, L., Yu, X., Wang, X., Chen, J., 2023. Link prediction and its optimization based on low-rank representation of network structures. Expert Syst. Appl. 219, 119680.

Chen, G., Wang, H., Fang, Y., Jiang, L., 2022. Link prediction by deep non-negative matrix factorization. Expert Syst. Appl. 188, 115991.

Chen, G., Xu, C., Wang, J., Feng, J., Feng, J., 2019. Graph regularization weighted nonnegative matrix factorization for link prediction in weighted complex network. Neurocomputing 369, 50–60.

Chen, G., Xu, C., Wang, J., Feng, J., Feng, J., 2020a. Nonnegative matrix factorization for link prediction in directed complex networks using PageRank and asymmetric link clustering information. Expert Syst. Appl. 148, 113–290.

Chen, G., Xu, C., Wang, J., Feng, J., Feng, J., 2020b. Robust non-negative matrix factorization for link prediction in complex networks using manifold regularization and sparse learning. Phys. A 539, 122882.

Chen, L., Zhang, X., Li, Y., Sun, M., 2024. Noise-robust voice conversion using adversarial training with multi-feature decoupling. Eng. Appl. Artif. Intell. 131, 107807.

Cho, A., Shin, J., Hwang, S., Kim, C., Shim, H., Kim, H., Kim, H., Lee, I., 2014. WormNet v3: A network-assisted hypothesis-generating server for caenorhabditis elegans. Nucleic Acids Res. 42 (W1), W76–W82.

Clauset, A., Moore, C., Newman, M.E., 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453 (7191), 98–101.

Colizza, V., Pastor-Satorras, R., Vespignani, A., 2007. Reaction–diffusion processes and metapopulation models in heterogeneous networks. Nat. Phys. 3 (4), 276–282.

De Handschutter, P., Gillis, N., Siebert, X., 2021. A survey on deep matrix factorizations. Comp. Sci. Rev. 42, 100423.

Duch, J., Arenas, A., 2005. Community identification using extremal optimization. Phys. Rev. E 72, 027104.

Faraji, M., Seyedi, S.A., Tab, F.A., Mahmoodi, R., 2024. Multi-label feature selection with global and local label correlation. Expert Syst. Appl. 246, 123198.

Farnia, F., Zhang, J.M., Tse, D., 2018. Generalizable adversarial training via spectral normalization. arXiv preprint arXiv:1811.07457.

Fortunato, S., 2010. Community detection in graphs. Phys. Rep. 486 (3–5), 75–174.

Getoor, L., Friedman, N., Koller, D., Pfeffer, A., 2001. Learning probabilistic relational models. In: Relational Data Mining. Springer, Berlin, Germany, pp. 307–335.

Getoor, L., Taskar, B., 2007. Probabilistic entity-relationship models, PRMs, and plate models. In: Introduction to Statistical Relational Learning. pp. 200–238.

Ghodsi, S., Seyedi, S.A., Ntoutsi, E., 2024. Towards cohesion-fairness harmony: Contrastive regularization in individual fair graph clustering. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 284–296.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014a. Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014b. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Guimerà, R., Sales-Pardo, M., Amaral, L.A.N., 2007. Classes of complex networks defined by role-to-role connectivity profiles. Nat. Phys. 3, 63–69.

Hajiveiseh, A., Seyedi, S.A., Akhlaghian Tab, F., 2024. Deep asymmetric nonnegative matrix factorization for graph clustering. Pattern Recognit. 148, 110179.

Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143 (1), 29–36.

He, C., Liu, H., Tang, Y., Fei, X., Li, H., Zhang, Q., 2020. Network embedding using deep robust nonnegative matrix factorization. IEEE Access 8, 85441–85453.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) 22 (1), 5–53.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Hu, H., Zhu, C., Ai, H., Zhang, L., Zhao, J., Zhao, Q., Liu, H., 2017. LPI-ETSLP: lncRNA–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction. Mol. Biosyst. 13 (9), 1781–1787.

Huang, J., Zhang, T., Yu, W., Zhu, J., Cai, E., 2021. Community detection based on modularized deep nonnegative matrix factorization. Int. J. Pattern Recognit. Artif. Intell. 35 (02), 2159006.

Hunter, D.R., Lange, K., 2000. Quantile regression via an MM algorithm. J. Comput. Graph. Statist. 9 (1), 60–77.

Kumar, A., Mishra, S., Singh, S.S., Singh, K., Biswas, B., 2020. Link prediction in complex networks based on significance of higher-order path index (SHOPI). Phys. A 545, 123790.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791.

Li, H.-C., Feng, X.-R., Zhai, D.-H., Du, Q., Plaza, A., 2021. Self-supervised robust deep matrix factorization for hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. 60, 1–14.

Li, S., Huang, J., Liu, J., Huang, T., Chen, H., 2020. Relative-path-based algorithm for link prediction on complex networks using a basic similarity factor. Chaos 30 (1), 013104.

Li, L., Wen, Y., Bai, S., Liu, P., 2022. Link prediction in weighted networks via motif predictor. Knowl.-Based Syst. 242, 108402.

Li, T., Zhang, R., Yao, Y., Liu, Y., Ma, J., 2024. Link prediction using deep autoencoder-like non-negative matrix factorization with L21-norm. Appl. Intell. 1–26.

Lin, C.-J., 2007. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Trans. Neural Netw. 18 (6), 1589–1596.

Lü, L., Zhou, T., 2011. Link prediction in complex networks: A survey. Phys. A: Stat. Mech. Appl. 390 (6), 1150–1170.

Ma, X., Sun, P., Qin, G., 2017. Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability. Pattern Recognit. 71, 361–374.

Mahmoodi, R., Seyedi, S.A., Akhlaghian Tab, F., Abdollahpouri, A., 2023. Link prediction by adversarial nonnegative matrix factorization. Knowl.-Based Syst. 280, 110998.

Marcus, G., 2018. Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631.

Martínez, V., Berzal, F., Cubero, J.-C., 2016. A survey of link prediction in complex networks. ACM Comput. Surv. (CSUR) 49 (4), 1–33.

Menon, A.K., Elkan, C., 2011. Link prediction via matrix factorization. In: Machine Learning and Knowledge Discovery in Databases. pp. 437–452.

Mozafari, M., Seyedi, S.A., Mohammadiani, R.P., Tab, F.A., 2024. Unsupervised feature selection using orthogonal encoder-decoder factorization. Inform. Sci. 120277.

Nath, V., Chattopadhyay, C., Desai, K., 2023. On enhancing prediction abilities of vision-based metallic surface defect classification through adversarial training. Eng. Appl. Artif. Intell. 117, 105553.

Newman, M.E.J., 2001. Clustering and preferential attachment in growing networks. Phys. Rev. E 64, 025102.

Ou, Q., Jin, Y.-D., Zhou, T., Wang, B.-H., Yin, B.-Q., 2007. Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. Phys. Rev. E 75, 021102.

Pavlov, M., Ichise, R., 2007. Finding experts by link prediction in co-authorship networks. FEWS 290, 42–55.

Qian, Z., Huang, K., Wang, Q.-F., Zhang, X.-Y., 2022. A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies. Pattern Recognit. 131, 108889.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Robledo, O.F., Zhan, X.-X., Hanjalic, A., Wang, H., 2022. Influence of clustering coefficient on network embedding in link prediction. Appl. Netw. Sci. 7 (1), 1–20.

Rossi, R.A., Ahmed, N.K., 2015. The network data repository with interactive graph analytics and visualization. In: AAAI.

Salahian, N., Tab, F.A., Seyedi, S.A., Chavoshinejad, J., 2023. Deep autoencoder-like NMF with contrastive regularization and feature relationship preservation. Expert Syst. Appl. 214, 119051.

Salton, G., McGill, M., 1983. McGraw Hill, Introduction to modern information retrieval. In: Computer Science Series.

Seyedi, S.A., Ghodsi, S.S., Akhlaghian, F., Jalili, M., Moradi, P., 2019. Self-paced multi-label learning with diversity. In: Proceedings of The Eleventh Asian Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 101, PMLR, pp. 790–805.

Seyedi, S.A., Tab, F.A., Lotfi, A., Salahian, N., Chavoshinejad, J., 2023. Elastic adversarial deep nonnegative matrix factorization for matrix completion. Inform. Sci. 621, 562–579.

Shajarian, Z., Seyedi, S.A., Moradi, P., 2017. A clustering-based matrix factorization method to improve the accuracy of recommendation systems. In: 2017 Iranian Conference on Electrical Engineering. ICEE, IEEE, pp. 2241–2246.

Sinha, A., Namkoong, H., Volpi, R., Duchi, J., 2017. Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:1710.10571.

Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B., 2014. A deep semi-nmf model for learning hidden representations. In: International Conference on Machine Learning. PMLR, pp. 1692–1700.

Wang, W., Feng, Y., Jiao, P., Yu, W., 2017. Kernel framework based on non-negative matrix factorization for networks reconstruction and link prediction. Knowl.-Based Syst. 137, 104–114.

Wang, J., Mu, R., 2021. A regularized convex nonnegative matrix factorization model for signed network analysis. Soc. Netw. Anal. Min. 11 (1), 1–12.

Wang, W., Tang, M., Jiao, P., 2018. A unified framework for link prediction based on non-negative matrix factorization with coupling multivariate information. PLoS One 13 (11), 1–22.

Xiao, Z., Xing, H., Qu, R., Feng, L., Luo, S., Dai, P., Zhao, B., Dai, Y., 2024. Densely knowledge-aware network for multivariate time series classification. IEEE Trans. Syst., Man, Cybern.: Syst..

Xiao, Z., Xing, H., Zhao, B., Qu, R., Luo, S., Dai, P., Li, K., Zhu, Z., 2023. Deep contrastive representation learning with self-distillation. IEEE Trans. Emerg. Top. Comput. Intell..

Xie, F., Chen, Z., Shang, J., Feng, X., Li, J., 2015. A link prediction approach for item recommendation with complex number. Knowl.-Based Syst. 81, 148–158.

Xing, Y., Song, Q., Cheng, G., 2021. On the generalization properties of adversarial training. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, vol. 130, pp. 505–513.

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. CIKM '18, pp. 1393—-1402.

Yu, K., Chu, W., Yu, S., Tresp, V., Xu, Z., 2006. Stochastic relational models for discriminative link prediction. In: Advances in Neural Information Processing Systems, vol. 19.

Zhang, L., Jiang, C., Chai, Z., He, Y., 2024. Adversarial attack and training for deep neural network based power quality disturbance classification. Eng. Appl. Artif. Intell. 127, 107245.

Zhang, M., Zhou, Z., 2020. Structural deep nonnegative matrix factorization for community detection. Appl. Soft Comput. 97, 106846.

Zhao, Y., Li, S., Zhao, C., Jiang, W., 2015. Link prediction via a neighborhood-based nonnegative matrix factorization model. In: Mu, J., Liang, Q., Wang, W., Zhang, B., Pi, Y. (Eds.), The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems. Springer International Publishing, Cham, pp. 603–611.

Zhao, J., Mathieu, M., LeCun, Y., 2016. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126.

Zhou, T., Lü, L., Zhang, Y.-C., 2009. Predicting missing links via local information. Eur. Phys. J. B 71, 623–630.