Link Prediction in Multilayer Networks via Cross-Network Embedding

Guojing Ren[1], Xiao Ding[2], Xiao-Ke Xu[3], Hai-Feng Zhang[2*]

[1]Institutes of Physical Science and Information Technology, Anhui University
[2]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Mathematical Science, Anhui University
[3] Computational Communication Research Center and School of Journalism and Communication, Beijing Normal University
{rengj, xiaoding2021}@stu.ahu.edu.cn, xuxiaoke@foxmail.com, haifengzhang1978@gmail.com

Abstract

Link prediction is a fundamental task in network analysis, with the objective of predicting missing or potential links. While existing studies have mainly concentrated on single networks, it is worth noting that numerous real-world networks exhibit interconnectedness. For example, individuals often register on various social media platforms to access diverse services, such as chatting, tweeting, blogging, and rating movies. These platforms share a subset of users and are termed multilayer networks. The interlayer links in such networks hold valuable information that provides more comprehensive insights into the network structure. To effectively exploit this complementary information and enhance link prediction in the target network, we propose a novel cross-network embedding method. This method aims to represent different networks in a shared latent space, preserving proximity within single networks as well as consistency across multilayer networks. Specifically, nodes can aggregate messages from aligned nodes in other layers. Extensive experiments conducted on real-world datasets demonstrate the superior performance of our proposed method for link prediction in multilayer networks.

Introduction

Link prediction (Kumar et al. 2020; Daud et al. 2020) is a fundamental task in network analysis that aims to predict missing or potential links in a network. It plays a crucial role in various fields, including (i) social network analysis (Kossinets and Watts 2006): suggesting new friendships; (ii) recommender systems (Vahidi Farashah et al. 2021): recommending relevant items or products to users; (iii) biological networks (Coşkun and Koyutürk 2021): predicting protein-protein interactions; and (iv) pandemic forecasting (Ma et al. 2022): predicting the spread of infectious diseases. The objective of link prediction is to infer the likelihood of a link between two nodes in a network based on the observed network structural features and, if available, node attribute features.

Numerous existing approaches have been developed for link prediction, employing various techniques such as similarity indices (Newman 2001; Zhou, Lü, and Zhang 2009), maximum likelihood models (Clauset, Moore, and Newman 2008; Guimerà and Sales-Pardo 2009), matrix factorization methods (Ding, Li, and Jordan 2010; Ma, Sun, and Qin 2017), Skip-gram embedding (Grover and Leskovec 2016; Tang et al. 2015), deep learning models (Wang, Cui, and Zhu 2016), and graph neural networks (GNNs) (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018). These methods have mainly focused on single networks. However, there might be missing links or noise in the single network, due to limitations in observation or sampling. This data insufficiency problem hinders the performance of link prediction methods, which are sensitive to network topology. Moreover, mining information from a single network provides one-sided insights, as users exhibit distinct characteristics and behavior patterns across different networks. For example, the Facebook network captures social friendships, the LinkedIn network focuses on employment relationships, the Douban network contains a common interest in movies, and the DBLP network reveals coauthorship among scholars. We cannot tell if someone is genuinely interested in the movie "Fast X" or just influenced by their friends, using only the limited information revealed in the movie rating network without knowledge of their social friendships. To address these challenges, some researchers have turned their attention to *multilayer networks* (Dickison, Magnani, and Rossi 2016).

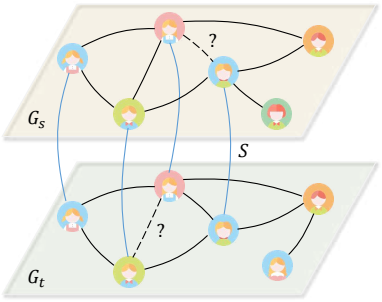

Figure 1: A example of multilayer networks. black lines represent intralayer links, blue lines represent interlayer links, and black dashes represent non-observed intralayer links that need to be predicted.

Interconnectedness is pervasive among real-world net-

*Hai-Feng Zhang is the corresponding author of this paper.

works. For example, individuals often register on multiple social media platforms to access various services, such as chatting, tweeting, blogging, and rating movies. Some users display accounts from other platforms on their profiles, indicating cross-platform connections. Similarly, collaborative networks in different fields share a subset of researchers, implying connections between these fields. Additionally, knowledge graphs share named entities across domains. These interconnected networks are modeled as multilayer networks. The relationships between different networks hold valuable information that can enhance our understanding of the structure, patterns, and evolution of networks. By considering multilayer networks as complementary information, we can improve link prediction in the target network. While previous research has explored link prediction in multilayer networks (Liu et al. 2017; Cao et al. 2018; Najari et al. 2019; Luo et al. 2022), these studies assume complete knowledge of the correspondence between nodes in all layers, e.g., multiplex networks. However, in practice, obtaining complete interlayer relationships is challenging and expensive due to user privacy concerns and platform policies. Consequently, researchers typically only have access to a limited subset of interlayer links. Therefore, this paper focuses on link prediction in multilayer networks, where nodes in different layers are allowed to be partially overlapped.

We propose a novel cross-network embedding model for link prediction in multilayer networks, which represents different networks in a shared latent space based on GNNs. Specifically, a random-walk-based objective function is employed to preserve proximity within each single network. To leverage interlayer relationships as complementary information, nodes aggregate messages not only from their neighbors within the same layer but also from aligned nodes in the other layer. Furthermore, consistency across multilayer networks is maintained by minimizing the distance between aligned nodes, which allows networks to be close in the embedding space. The contributions of our work are summarized as follows:

- We introduce a GNN-based model that enables nodes to simultaneously aggregate messages from both their neighbors in the same layer and aligned nodes in the other layer. Each layer learns complementary information from its counterpart layer.
- We develop a joint objective function to train the model, which effectively preserves both the proximity within single networks and consistency across multilayer networks.
- We extend some state-of-the-art methods designed for single networks and compare them with our proposed model in multilayer networks. Extensive experiments conducted on real-world datasets demonstrate that our proposed method outperforms the baselines for link prediction in multilayer networks, especially with few interlayer links.

Related Works

Link Prediction

Various link prediction methods have been developed which can be categorized into several types.

Similarity-based metrics assign a similarity score for each non-observed link. Common Neighbors (CN) assumes that two individuals with more common friends are more likely to establish a friendship, and it calculates the number of common neighbors for a given pair of nodes (Newman 2001). Resource Allocation Index (RA) considers the resource allocation process in networks and calculates the amount of resource transported through the common neighbors of two nodes (Zhou, Lü, and Zhang 2009).

Maximum likelihood models evaluate the likelihood of each non-observed link, which may not be suitable for large-scale networks due to their complexity and time-consuming nature. Hierarchical structure model (HSM) suggests that many real-world networks are hierarchically structured, and it infers the likelihood of a hierarchical random graph to predict missing links (Clauset, Moore, and Newman 2008). Stochastic block model (SBM) distributes nodes into blocks or communities and computes the link reliability (Guimerà and Sales-Pardo 2009).

Matrix factorization methods extract the latent features of each node and are considered dimensionality reduction techniques. Some authors apply the Singular Value Decomposition (SVD), which maintains important information based on the eigenvalues (Ding, Li, and Jordan 2010). Nonnegative matrix factorization has also been used to learn latent structural features incorporating additional node/link attribute information (Ding, Li, and Jordan 2010; Ma, Sun, and Qin 2017).

Embedding-based methods have gained significant attention. Node2vec is a Skip-gram model that preserves neighborhoods of nodes through biased random walks, taking into account both exploration and exploitation (Grover and Leskovec 2016). LINE preserves both first- and second-order proximity (Tang et al. 2015). SDNE is a semi-supervised deep autoencoder model that jointly preserves local and global structure features (Wang, Cui, and Zhu 2016). *GNN-based methods* have achieved great success in recent years. GCN aggregates information from a node's local neighborhood (Kipf and Welling 2017, 2016). GraphSAGE generates embeddings in an inductive manner and is capable of handling large-scale graphs (Hamilton, Ying, and Leskovec 2017). GAT introduces self-attention to assign different importance to each pair of nodes (Veličković et al. 2018). Chen et al. generalized GCN to simplicial complexes by integrating interactions among multiple higher-order graph structures (Chen, Gel, and Poor 2022).

Multilayer Link Prediction

Above works are modeled for link prediction in single-layer networks having homogeneous links. However, many real-world networks might be heterogeneous, developing different types of links in multiple layers. Compared to single-layer networks, multilayer networks can express richer information, thus drawing more attention recently.

Similarity-based metrics: Hristova et al. extended Jaccard Coefficient and Adamic/Adar Index to a multilayer scenario (Hristova et al. 2016). Yao et al. proposed NSILR which aggregates inter-layer and intra-layer similarity scores based on layer-wise correlations (Yao et al. 2017). Najari et al. developed LPIS synthesizing probability that combines intra-layer and inter-layer information (Najari et al. 2019). Aleta et al. generalized the Adamic-Adar Index to multiplex networks via triadic closure (Aleta et al. 2020). Luo et al. introduced EMLP algorithm that integrates similarity scores from all layers using evidence theory (Luo et al. 2022).

Embedding based methods: Liu et al. proposed layer co-analysis which modifies node2vec to multilayer networks, traversing between layers by leveraging interactions among layers (Liu et al. 2017). Cao et al. trained multiple neural networks (MNN) targeting heterogeneous feature channels and assigned the same embedding for aligned nodes (Cao et al. 2018). Zhan et al. proposed collective link fusion (CLF) to predict link probability using collective random walk with restart (Zhan, Zhang, and Yu 2019). Jiang proposed partially aligned GCNs that jointly learn embeddings incorporating interlayer information (Jiang 2021). Du et al. trained a Skip-gram embedding model CELP via a biased random walk based on intra-network and cross-network distributions (Du et al. 2022). Alnaimy et al. employed matrix factorization to obtain embeddings on the expanded graph (EG) (Alnaimy and Desouki 2022).

Preliminaries

In this section, we will define some terminology and notations used in this paper, and provide the problem formulation of link prediction in multilayer networks.

Definition 1. *Multilayer networks*: For simplicity, we consider two undirected and unweighted networks $G_s = (V_s, E_s)$ and $G_t = (V_t, E_t)$, where V_s, V_t are the sets of nodes, and E_s, E_t are the sets of edges (intralayer links), respectively. G_s, G_t can be referred to as layers, and they share some nodes belonging to the same entities, i.e., $V_s \cap V_t \neq \emptyset$. $S = \{(v_i, v_j) | v_i \in V_s, v_j \in V_t\}$ is the set of interlayer links between G_s and G_t. For $\forall v_i \in V_s \cup V_t$, at most one interlayer link exists, i.e., $|S| \leq \min(|V_s|, |V_t|)$. The multilayer networks can be defined by a triplet (G_s, G_t, S), as Figure 1 shows. Note that multiplex networks are a special case of general multilayer networks, where $V_s = V_t = V$, and $|S| = |V|$.

Definition 2. *Link prediction*: For the target network $G = (V, E)$, $\frac{|V| \cdot (|V|-1)}{2}$ is the number of all possible links, denoted as the universal set U. The set of non-existing links is $U - E$, and there may be some missing or potential links in the set $U - E$. The aim of link prediction is to find such missing or potential links. In the above multilayer network, $G \in \{G_s, G_t\}$.

To evaluate the effectiveness of link prediction methods, E is randomly divided into two parts E^T and E^P, named the training set and the probe set (i.e. test set) respectively. In general, a link prediction algorithm provides a similarity score or linkage probability for each non-observed link $(x, y) \in U - E^T$.

Proposed Method

Cross-Network Embedding

GNNs can learn node representations by aggregating information from neighbors, thereby capturing the underlying connectivity patterns, which is beneficial for the link prediction task. Multilayer networks provide valuable information to enhance link prediction. In light of this, our approach involves utilizing GNNs in multilayer networks, which learn a cross-network embedding simultaneously integrating intralayer and interlayer structural features.

General GNN Layer Various GNNs have been developed for node representation. In general, a typical GNN layer follows the form:

$$\boldsymbol{h}_u^k \leftarrow \theta(\boldsymbol{W}_1^k \cdot \boldsymbol{h}_u^{k-1} + \boldsymbol{W}_2^k \cdot \sum_{v \in \mathcal{N}_u} a_{uv} \boldsymbol{h}_v^{k-1}), \quad (1)$$

where $\boldsymbol{h}_u^k$ represents the output vector of node u in the k-th layer, $\forall k \in \{1, \ldots, K\}$. θ denotes an activation function (e.g., ReLU). $\boldsymbol{W}_1^k$ and $\boldsymbol{W}_2^k$ are matrices that weight the contributions of the node itself and its neighbors, respectively. a_{uv} indicates the importance of link (u, v). In the case of GCN (Kipf and Welling 2017), a_{uv} corresponds to an element of the symmetrically normalized adjacency matrix, and $a_{uv} = \frac{1}{\sqrt{|\mathcal{N}_u| \cdot |\mathcal{N}_v|}}$; for GAT (Veličković et al. 2018), a_{uv} represents attention coefficients; for GraphSAGE (Hamilton, Ying, and Leskovec 2017), $a_{uv} = \frac{1}{|\mathcal{N}_u|}$ when using a mean aggregator. Tang et al. argued that intralayer links connected with small degree nodes have the most significant impact on capturing interlayer features (Tang et al. 2022). This observation might be explained from a resource allocation perspective (Zhou, Lü, and Zhang 2009). For instance, an individual who is popular in school may have many friends, but due to time constraints (e.g., having only 2.5 hours for daily social activities on average), they have less opportunity to interact with each specific friend. Conversely, an individual with few friends may share more attention with each friend. In other words, intralayer links connected with small degree nodes are more important for capturing both intralayer and interlayer features. Therefore, we adopt the GCN form for a_{uv} that suppresses the contributions of neighbors with large degrees. It indicates that nodes with larger degrees transmit fewer messages to each of their neighbors.

Cross-GNN Layer We assume that different layers of multilayer networks have inherent structural consistency to some extent, which is a prerequisite for link prediction in multilayer networks. Aggregating messages from other layers will help complement node information, which enhances the understanding of connectivity patterns. A cross-GNN layer integrating interlayer information is formulated as follows:

$$\begin{aligned} \boldsymbol{h}_u^k \leftarrow & \theta(\boldsymbol{W}_1^k \cdot \boldsymbol{h}_u^{k-1} + \boldsymbol{W}_2^k \cdot \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u| \cdot |\mathcal{N}_v|}} \boldsymbol{h}_v^{k-1} \\ & + \boldsymbol{W}_3^k \cdot b_{uu'} \boldsymbol{h}_{u'}^{k-1}), \end{aligned} \quad (2)$$

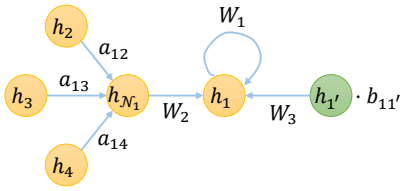

Figure 2: An illustration of the cross-GNN layer. Nodes 2, 3, and 4 are neighbors of 1, denoted as $\mathcal{N}_1$. $1'$ is the aligned node of 1. Information from node 1 itself, neighbors, and aligned node is aggregated with different weights.

where u' is aligned with u, $b_{uu'}$ denotes the importance of u' to u, and $b_{uu'} = \sigma(\boldsymbol{h}_u^{k-1^T} \cdot \boldsymbol{h}_{u'}^{k-1})$, $\sigma(x) = 1/(1+\exp^{-x})$ is the sigmoid function. $\boldsymbol{W}_1^k, \boldsymbol{W}_2^k, \boldsymbol{W}_3^k$ are matrices weighting the contribution of the node itself, its neighbors, and the aligned nodes, respectively. An example can be seen in Figure 2. Note that if u is not aligned with any nodes in the other network (i.e. $u' = \text{None}$), then $\boldsymbol{h}_{u'}^{k-1} = \boldsymbol{0}$.

As the aggregator functions are defined, we employ them to the multilayer network (G_s, G_t, S). It is essential that $\boldsymbol{h}_u^{k-1}$ and $\boldsymbol{h}_{u'}^{k-1}$ have the same dimensions. To achieve this, we use the GCN-form aggregator to capture first-order proximity and ensure consistent output dimensions for both layers. Subsequently, the cross-network aggregator captures higher-order proximity and information from the aligned nodes. The input node features are denoted as $\boldsymbol{X}$, which can be the adjacency matrix in attribute-free networks. The representation vectors are formulated as follows:

$$\begin{aligned}\boldsymbol{H}^1 &= \theta(\boldsymbol{X}\boldsymbol{W}_1^1 + \tilde{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_2^1)\\ \boldsymbol{H}^k &= \theta(\boldsymbol{H}^{k-1}\boldsymbol{W}_1^k + \tilde{\boldsymbol{A}}\boldsymbol{H}^{k-1}\boldsymbol{W}_2^k + \boldsymbol{B}^k\boldsymbol{H}_*^{k-1}\boldsymbol{W}_3^k).\end{aligned} \tag{3}$$

Here, $\boldsymbol{H}^k \in \mathbb{R}^{n\times c_k}$ represents the output representation vectors of the k-th layer, $\forall k \in \{2,\ldots,K\}$. The symmetrically normalized adjacency matrix is denoted as $\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{A}$ is the adjacency matrix, and $\boldsymbol{D}$ is the degree matrix. Additionally, $\boldsymbol{B}^k$ is the importance matrix of $\boldsymbol{H}_*^{k-1}$ to $\boldsymbol{H}^{k-1}$, defined as $\boldsymbol{B}^k = \sigma(\boldsymbol{I}(\boldsymbol{H}^{k-1} \odot \boldsymbol{H}_*^{k-1})^T)$, where $\odot$ represents the Hadamard product. The matrix $\boldsymbol{I} \in \mathbb{R}^{n\times c_k}$ is filled with 1 to summarize the Hadamard product of each aligned pair $(\boldsymbol{h}^{k-1}, \boldsymbol{h}_*^{k-1})$ and broadcast. The aligned representation matrix $\boldsymbol{H}_*^{k-1}$ has the same shape as $\boldsymbol{H}^{k-1}$, and it can be rewritten as:

$$\boldsymbol{H}_*^{k-1}[u] = \begin{cases}\boldsymbol{h}_{u'}^{k-1}, & \text{if } (u,u') \in S\\ \boldsymbol{0}, & \text{otherwise.}\end{cases} \tag{4}$$

The model defined as Eq. (3) is a K-layer GNN, where the first layer is a GCN-form layer, and the following are cross-GNN layers. The model is trained for both G_s and G_t, and parameters in cross-GNN layers are shared. Notably, there is no activation function in the last layer of the GNN model. The final output vectors $\boldsymbol{H}^K$ are denoted as $\boldsymbol{Z}$.

Objective Functions

The objective function includes intralayer and interlayer loss, in which intralayer loss mainly retains intralayer structural features, and interlayer loss retains interlayer structural features. Then the total loss is jointly optimized to unify the two networks to the same latent space better.

Intralayer Loss Random walks are often employed to link prediction tasks. If two nodes co-occur on fixed-length random walks frequently, it indicates a higher probability of a link between them. Random walks thus capture higher-order proximity and provide insights into the connectivity patterns and potential links within a network. Therefore, to learn embedding $z_i, \forall i \in V$, we apply a random-walk-based objective function in an unsupervised setting:

$$\begin{aligned}L_r = &-\log(\sigma(\boldsymbol{z}_i^T \cdot \boldsymbol{z}_j))\\ &- Q \cdot \mathbb{E}_{k\sim P_n(v)} \log(1-\sigma(\boldsymbol{z}_i^T \cdot \boldsymbol{z}_k)),\end{aligned} \tag{5}$$

where j is a node that co-occurs with i in a window from sequences of random walks, σ is the sigmoid function, $P_n(v)$ is the negative sampling probability distribution, and Q defines the number of negative samples. Proximity nodes are encouraged to have similar embeddings, while discrete nodes are distinct in the embedding space. The intralayer loss is the sum of the random-walk-based loss of G^s and G^t:

$$L_{intra} = L_r^s + L_r^t. \tag{6}$$

By minimizing the intralayer loss, the intralayer structural features of both networks can be preserved.

Interlayer Loss Besides intralayer structural features, interlayer structural features are crucial to multilayer networks. We assume that the multilayer networks are consistent to some extent, and aligned nodes should be close in the latent space, i.e., they share similar embeddings. Therefore, we can build an anchor-aware loss to minimize the distance of aligned nodes.

Obviously, the number of non-interlayer links is far more than interlayer links (i.e., $|V_s|\cdot|V_t| - |S| \gg |S|$). To overcome the sample imbalance problem, we adopt undersampling, which selects the several nearest non-interlayer links for each interlayer link. These non-interlayer links termed hard negatives are more informative and helpful for maintaining consistency between multilayer networks. In detail, for each interlayer link (v_i^s, v_j^t), we randomly sample η non-interlayer links (v_i^s, v_k^t) for v_i^s, where $v_k^t \in \mathcal{N}(v_j^t)$, indicating that v_k^t is chosen from the neighbors of v_j^t. The same selection is performed for v_j^t. The set of interlayer links is called S^+ with the label 1, while the set of these sampled non-interlayer links is called S^- (i.e., $|S^-| = 2\eta|S^+|$) with the label -1. The interlayer loss is defined as follows:

$$\begin{aligned}L_{inter} = &\frac{1}{|S^+|}\sum_{(v_i^s,v_j^t)\in S^+}(1-\cos(\boldsymbol{z}_i^s, \boldsymbol{z}_j^t))+\\ &\frac{1}{|S^-|}\sum_{(v_k^s,v_l^t)\in S^-} max(\cos(\boldsymbol{z}_k^s, \boldsymbol{z}_l^t)) - \epsilon, 0),\end{aligned} \tag{7}$$

Algorithm 1: CGNN

Input: Training multilayer networks (G_s, G_t, S^+)
Parameter: Batch size of random walks b_r, batch size of interlayer links b_s, margin ϵ, weight parameter α, numbers of negative samples Q, η
Output: Embedding vectors $\boldsymbol{Z}_s$ for G_s, $\boldsymbol{Z}_t$ for G_t

Obtain node attribute vectors $\boldsymbol{X}_s$ for G_s, $\boldsymbol{X}_t$ for G_t
Initialize set of model parameters $\boldsymbol{W}$
while not converged **do**
    Sample a batch of positive and negative pairs from random walks
    Sample a batch of interlayer links from S^+ and non-interlayer links from S^-
    Generate embedding vectors $\boldsymbol{Z}^s$, $\boldsymbol{Z}^t$ by Eq. (3)
    Calculate the total loss by Eq. (8)
    Update $\boldsymbol{W}$ with Adam optimizer
end while
return $\boldsymbol{Z}_s$, $\boldsymbol{Z}_t$

where ϵ is a margin parameter: if $\cos(\boldsymbol{z}_k^s, \boldsymbol{z}_l^t) > \epsilon$, non-interlayer link $(\boldsymbol{z}_k^s, \boldsymbol{z}_l^t)$ is hard to be distinguished from interlayer link $(\boldsymbol{z}_i^s, \boldsymbol{z}_j^t)$, therefore $\cos(\boldsymbol{z}_k^s, \boldsymbol{z}_l^t)$ is encouraged to decrease; otherwise, it is not taken into account. By minimizing interlayer loss, we can preserve interlayer structural information.

Total Loss The total objective function is defined as a linear combination of both intralayer and interlayer loss, so that we can preserve intralayer and interlayer structural information by jointly optimizing this:

$$L = \alpha \cdot L_{intra} + (1 - \alpha) \cdot L_{inter}. \tag{8}$$

Here, α is the weight parameter to tradeoff the two components of the objective function. We employ Adam optimizer to update the parameters of the GNN layers.

For convenience, we denote our proposed model as **CGNN** (**C**ross-network **G**raph **N**eural **N**etworks). The pseudocode is shown as Algorithm 1.

Link Prediction

To calculate the probability of the existence of each non-observed link $(x, y) \in U - E^T$, we can simply use cosine similarity. To leverage the latent information contained in the node embeddings, we employ the Logistic Regression classifier to predict the linkage probability. The input feature is edge embedding, which is a concatenation of two node embeddings and the Hadarmard product of them (Qu et al. 2016):

$$\mathcal{I}(x, y) = \text{concat}(\boldsymbol{z}_x, \boldsymbol{z}_x \odot \boldsymbol{z}_y, \boldsymbol{z}_y). \tag{9}$$

Complexity Analysis

Notations The depth of GNN layers K is set to 2. f, c, d are the dimensions of the input feature vectors, hidden vectors and output vectors, respectively. $|E_s|, |E_t|$ are the numbers of edges of G_s, G_t. For random walks, the walk length is l, window size is w, walks per node are m.

Datasets	Nodes#	Edges#	Anchors#
Facebook	1043	4734	1043
Twitter	1043	4860	
Twitter	2562	6967	2177
YouTube	2409	7862	

Table 1: Structural statistics of the datasets.

GNN Layers For a single GNN layer, the convolutional operation has complexity $\mathcal{O}(|E|fc)$. For the 2-layer GNN defined above, the complexity is $\mathcal{O}(|E|c(f+d))$, where $E = \max(|E_s|, |E_t|)$.

Intralayer Loss The number of fixed-length random walks from batch sampling is $b_r m(l - w)$. Then the complexity of intralayer loss is $\mathcal{O}(b_r m(l - w)wd)$.

Interlayer Loss The batch size of interlayer links b_s is $|S|$, i.e. full batch. The complexity of interlayer loss is $\mathcal{O}(|S|d)$, where $|S| = \max(|S^+|, |S^-|)$.

Experiments

Datasets

We select two real-world datasets: (i) Facebook/Twitter (Du et al. 2022); (ii) Twitter-YouTube (Dickison, Magnani, and Rossi 2016). The detailed statistics of these datasets are shown in Table 1. More real-world datasets are shown in the Appendix.

Baselines

We compare our proposed model with the following state-of-the-art baseline methods.

Single-Layer Methods

- **Common Neighbors (CN):** Similarity-based metric calculating the size of common neighbors for a given pair of nodes x and y (Newman 2001).
- **Resource Allocation Index (RA):** Similarity-based metric calculating the resources sent from node x to y through the common neighbors (Zhou, Lü, and Zhang 2009).
- **SVD:** Matrix factorization-based method applying the singular value decomposition technique to generate low-dimensional vectors (Ding, Li, and Jordan 2010).
- **node2vec (n2v):** Skip-gram embedding method using a biased random walk to preserve higher-order proximity (Grover and Leskovec 2016).
- **GAE:** GNN-based method using a GCN encoder and a simple inner product decoder (Kipf and Welling 2016).
- **GAT:** GNN-based method using a GAT encoder and a simple inner product decoder (Veličković et al. 2018).

Multilayer Methods

- **MAA:** Similarity-based metric generalizing the Adamic-Adar method (Adamic and Adar 2003) to multiplex networks via triadic relationships within a single layer and across different layers (Aleta et al. 2020).

Dataset	Method	Ratio of Interlayer Links			
		0	30%	60%	90%
Facebook ↕ Twitter	CN	0.7642 / 0.7880	0.7699 / 0.7913	0.7722 / 0.7984	0.7863 / 0.8057
	RA	0.7649 / 0.7885	0.7708 / 0.7917	0.7725 / 0.7992	0.7878 / 0.8077
	SVD	0.8100 / 0.8330	0.8197 / 0.8384	0.8331 / 0.8581	0.8528 / 0.8827
	n2v	0.8120 / 0.8319	0.8304 / 0.8514	0.8642 / 0.8800	0.9313 / 0.9470
	GAE	0.8352 / 0.8488	0.8492 / 0.8608	0.8660 / 0.8851	0.9123 / 0.9326
	GAT	0.8328 / 0.8545	0.8364 / 0.8529	0.8640 / 0.8878	0.9213 / 0.9344
	MAA	0.7649 / 0.7889	0.7692 / 0.7920	0.7753 / 0.7986	0.7814 / 0.8064
	LPIS	0.7721 / 0.7929	0.7444 / 0.7759	0.7449 / 0.7638	0.8287 / 0.8306
	CLF	0.8637 / 0.8787	0.8875 / 0.8996	0.9122 / 0.9169	0.9365 / 0.9341
	n2v-e	0.8120 / 0.8319	0.8481 / 0.8543	0.8792 / 0.8795	0.9013 / 0.9073
	EG-mini	0.7977 / 0.8559	0.8164 / 0.8729	0.8499 / 0.8908	0.8920 / 0.9202
	CELP	0.8150 / 0.8371	0.8697 / 0.8464	0.8983 / 0.9132	0.9588 / 0.9684
	CGNN	**0.8989 / 0.9125**	**0.9105 / 0.9164**	**0.9444 / 0.9550**	**0.9653 / 0.9740**
Twitter ↕ YouTube	CN	0.7381 / 0.7512	0.7490 / 0.7568	0.7882 / 0.7715	0.8048 / 0.7828
	RA	0.7559 / 0.7529	0.7667 / 0.7597	0.8073 / 0.7740	0.8227 / 0.7914
	SVD	0.6458 / 0.7103	0.6537 / 0.7086	0.6641 / 0.7220	0.6725 / 0.7044
	n2v	0.6276 / 0.6594	0.6332 / 0.6619	0.6687 / 0.7057	0.7130 / 0.7266
	GAE	0.6887 / 0.7225	0.6996 / 0.7403	0.7000 / 0.7567	0.7159 / 0.7782
	GAT	0.7065 / 0.7715	0.7260 / 0.7967	0.7302 / 0.7858	0.7582 / 0.8174
	MAA	0.7551 / 0.7531	0.7602 / 0.7551	0.7740 / 0.7646	0.7741 / 0.7782
	LPIS	0.7093 / 0.8975	0.6824 / 0.8811	0.6199 / 0.8508	0.5304 / 0.8069
	CLF	0.8194 / 0.8920	0.8244 / 0.8850	0.8235 / 0.8800	0.8318 / 0.8752
	n2v-e	0.6276 / 0.6594	0.7332 / 0.8001	0.7418 / 0.8033	0.7656 / 0.8138
	EG-mini	0.7859 / 0.7345	0.7942 / 0.7505	0.8000 / 0.7764	0.8104 / 0.7758
	CELP	0.8942 / 0.8737	0.8908 / 0.8635	0.8965 / 0.8735	0.9042 / 0.8569
	CGNN	**0.9300 / 0.9095**	**0.9327 / 0.9094**	**0.9353 / 0.9125**	**0.9353 / 0.9108**

Table 2: AUC with different ratios of interlayer links.

- **LPIS:** This model predicts intralayer link probability using Logistic regression classifier with intralayer features and then calculates interlayer link probability with interlayer similarity. The total link probability is a combination of them (Najari et al. 2019).
- **CLF:** A collective link fusion model predicting both intralayer and interlayer links (Zhan, Zhang, and Yu 2019). It can be seen as random walks with restart (Tong, Faloutsos, and Pan 2006) across multilayer networks.
- **EG-mini:** For multilayer networks (G_s, G_t, S), we construct an expanded graph $G_e = (V_e, E_e)$, where $V_e = V_s \cup V_t, E_e = E_s \cup E_t \cup S$ (Alnaimy and Desouki 2022). Then we use matrix factorization on the adjacency matrix of the expanded graph to obtain node embeddings.
- **CELP:** Cross-network Skip-gram embedding method employing a biased random walk strategy, which is a balance of intra-network and cross-network empirical distributions (Du et al. 2022).
- **n2v-e:** We employ node2vec on the above expanded graph, called n2v-e. It can be seen as a simplified version of CELP.

For a fair comparison, we extend the single-layer methods to multilayer methods. We apply two strategies respectively, then report the best performances of them.

- **Network Extension:** The assumption for this strategy is that different layers in a multilayer network share similar connection patterns. Based on this, if a pair of nodes are not linked in one layer, but their aligned pairs are linked in the other layer, we can add an edge between them to complement the present network structure (Liu et al. 2017). Formally, given a multilayer network (G_s, G_t, S), the extension of $G_t = (V_t, E_t)$ can be formulated as:

$$E_t \leftarrow E_t \cup \{(u,v) | (u,v) \notin E_t, (u',v') \in E_s, \\ (u,u'), (v,v') \in S\}.$$

- **Score Extension:** Link prediction on each layer of a multilayer network can generate different similarity scores or link probabilities, which contain valuable information from different networks. It can be seen as a simplified version of LPIS (Najari et al. 2019). Formally, given a multilayer network (G_s, G_t, S), the link probability of $(x, y) \in E_t$ can be formulated as:

$$p_t(x,y) \leftarrow c \cdot p_t(x,y) + (1-c) \cdot p_s(x',y'), \\ s.t. \quad (x,x'), (y,y') \in S.$$

Experiment Settings

Parameter Setup For SVD, the embedding dimension is 32. For node2vec and n2v-e, $p = 1, q = 1$, window size is 10, the number of walks per node is 20, walk length is 80, and the embedding dimension is 128. For GAE, the hidden dimension is 32, the embedding dimension is 16, and learning rate is 0.01. For GAT, the first layer consists of 4 attention heads computing 8 features each (for a total of 32 features), the embedding dimension is 16, and learning rate

Method	Facebook	Twitter
CGNN	**0.9653**	**0.9740**
W/o cross-GNN layer	0.8967	0.9106
W/o degree-discount	0.9281	0.9392
W/o intralayer loss	0.5639	0.5793
W/o interlayer loss	0.8859	0.8965
W/o weight-sharing	0.9629	0.9723
W/o negative sampling	0.9648	0.9732

Table 3: Ablation study on Facebook/Twitter dataset.

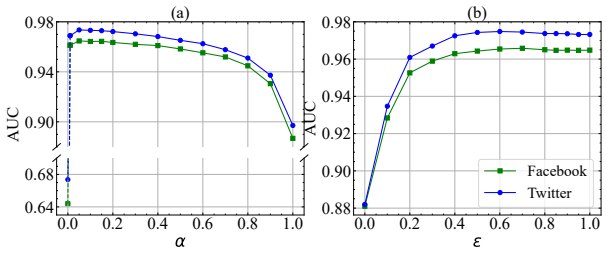

Figure 3: Performance with different hyperparameters. (a) AUC *vs.* α; (b) AUC *vs.* ϵ.

is 0.01. For CGNN, the hidden dimension is 256, the embedding dimension $d = 128$, $\alpha = 0.05, \epsilon = 0.7$, window size is 10, walks per node is 10, walk length is 20, batch size $b_r = 512$, and learning rate is 0.001.

Evaluation Metric We use AUC (Area Under the Curve) as the evaluation metric. Each layer from the multilayer network is divided into the training set and the test set. We predict the linkage probability on each layer respectively.

Hardware SVD, GAE, GAT, and CGNN were run on a Linux server with an NVIDIA A100-40G GPU. These codes are implemented in Python with PyTorch and PyTorch Geometric libraries. All experiments in this paper were carried out 10 times and averaged.

Experimental Results

For each dataset, we randomly select 90% edges as the training set, and the rest 10% edges as the test set. Then we randomly sample negative test edges of the same size as the test set for evaluation. The proposed model is compared with the baselines, and the performance in Facebook/Twitter and Twitter/YouTube is shown in Table 2. One can see that almost all methods perform better with a higher ratio of interlayer links. It proves that interlayer information can indeed enhance link prediction. The improvement of embedding-based methods is greater than that of similarity-based metrics, indicating that embedding-based methods have a better capability of leveraging interlayer information. CGNN is the best, showing the effectiveness of our proposed method. Compared to CELP, CGNN performs better with fewer interlayer links. Results on more real-world datasets are shown in the Appendix.

Ablation Study To investigate the importance of the different components in our proposed model, we conduct an ablation study on the Facebook-Twitter dataset, with 90% training edges and 90% interlayer links. We compare our model CGNN with 6 ablated variants: (i) CGNN without the cross-GNN layer, which replaces the cross-GNN layer with a GCN-form layer (W/o cross-GNN layer); (ii) CGNN without degree-discount, where $a_{uv} = 1$ (W/o degree-discount); (iii) CGNN without the intralayer loss, where $\alpha = 0$ (W/o intralayer loss); (iv) CGNN without the interlayer loss, where $\alpha = 1$ (W/o interlayer loss); (v) CGNN without weight-sharing in the 2-nd layer (W/o weight-sharing); (vi) CGNN without negative sampling in interlayer loss, where $\epsilon = 1$ (W/o negative sampling). The results are shown in Table 3, where black means the optimal results. We can observe that CGNN outperforms most variants, indicating the importance of these components.

Parameter Sensitivity Study To analyze the sensitivity of important parameters in our proposed method, we conduct experiments on the Facebook-Twitter dataset with 90% training edges and 90% interlayer links. Figure 3(a) shows the impact of the weight parameter α that tradeoffs the intralayer loss and interlayer loss. $\alpha = 0$ means only consider interlayer loss, and $\alpha = 1$ means only consider intralayer loss. One can observe that the performance of the Twitter network arrives highest when $\alpha = 0.05$, where interlayer loss is more important. Performance at $\alpha = 0$ is extremely low, indicating that intralayer loss is necessary for link prediction. Figure 3(b) depicts the impact of the margin parameter ϵ that selects indistinguishable interlayer links from non-interlayer links in the embedding space. $\epsilon = 1$ means no negative samples are used. One can observe that performance is better when ϵ becomes larger before arriving 1 and achieves best when $\epsilon = 0.7$. It illustrates the effectiveness of negative sampling. More experiments can be seen in the Appendix.

Conclusion

In this paper, we focus on link prediction in multilayer networks. Multilayer networks provide a more comprehensive view of network analysis. To take advantage of the valuable information in multilayer networks, we present a cross-network GNN model for link prediction in multilayer networks. More specifically, nodes are capable of aggregating messages not only from their immediate neighbors within the same layer but also from corresponding nodes in the other layer. Evidently, each layer learns complementary information from its counterpart layer. For joint model training, we utilize both the intralayer loss based on random walks, which maintains proximity within single layers, and interlayer loss which ensures consistency across the multilayer network. Therefore, different layers of the multilayer network are embedded into the same latent space. Some single-layer state-of-the-art methods are extended to multilayer networks for comparison. Experiments on real-world datasets indicate that our proposed model outperforms baselines for link prediction in multilayer networks, particularly under conditions of limited interlayer links.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61973001) and the University Synergy Innovation Program of Anhui Province (GXXT-2021-032).

References

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

Aleta, A.; Tuninetti, M.; Paolotti, D.; Moreno, Y.; and Starnini, M. 2020. Link prediction in multiplex networks via triadic closure. *Physical Review Research*, 2(4): 042029.

Alnaimy, M.; and Desouki, M. S. 2022. Expanded graph embedding for joint network alignment and link prediction. *Journal of Big Data*, 9(1): 1–15.

Cao, X.; Chen, H.; Wang, X.; Zhang, W.; and Yu, Y. 2018. Neural link prediction over aligned networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-18)*, volume 32.

Chen, Y.; Gel, Y. R.; and Poor, H. V. 2022. BScNets: Block simplicial complex neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-22)*, volume 36, 6333–6341.

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

Coşkun, M.; and Koyutürk, M. 2021. Node similarity-based graph convolution for link prediction in biological networks. *Bioinformatics*, 37(23): 4501–4508.

Daud, N. N.; Ab Hamid, S. H.; Saadoon, M.; Sahran, F.; and Anuar, N. B. 2020. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, 166: 102716.

Dickison, M. E.; Magnani, M.; and Rossi, L. 2016. *Multilayer social networks*. Cambridge University Press.

Ding, C.; Li, T.; and Jordan, M. I. 2010. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(01): 45–55.

Du, X.; Yan, J.; Zhang, R.; and Zha, H. 2022. Cross-Network Skip-Gram Embedding for Joint Network Alignment and Link Prediction. *IEEE Transactions on Knowledge & Data Engineering*, 34(03): 1080–1095.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD-16)*, 855–864.

Guimerà, R.; and Sales-Pardo, M. 2009. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52): 22073–22078.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS-17)*, 1025–1035.

Hristova, D.; Noulas, A.; Brown, C.; Musolesi, M.; and Mascolo, C. 2016. A multilayer approach to multiplexity and link prediction in online geo-social networks. *EPJ Data Science*, 5: 1–17.

Jiang, M. 2021. Cross-Network Learning with Partially Aligned Graph Convolutional Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (SIGKDD-21)*, 746–755.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR-17)*.

Kossinets, G.; and Watts, D. J. 2006. Empirical analysis of an evolving social network. *Science*, 311(5757): 88–90.

Kumar, A.; Singh, S. S.; Singh, K.; and Biswas, B. 2020. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553: 124289.

Liu, W.; Chen, P.-Y.; Yeung, S.; Suzumura, T.; and Chen, L. 2017. Principled multilayer network embedding. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW-17)*, 134–141.

Luo, H.; Li, L.; Dong, H.; and Chen, X. 2022. Link prediction in multiplex networks: An evidence theory method. *Knowledge-Based Systems*, 257: 109932.

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

Ma, Y.; Gerard, P.; Tian, Y.; Guo, Z.; and Chawla, N. V. 2022. Hierarchical spatio-temporal graph neural networks for pandemic forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM-22)*, 1481–1490.

Najari, S.; Salehi, M.; Ranjbar, V.; and Jalili, M. 2019. Link prediction in multiplex networks based on interlayer similarity. *Physica A: Statistical Mechanics and its Applications*, 536: 120978.

Newman, M. E. 2001. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2): 025102.

Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; and Wang, J. 2016. Product-based neural networks for user response prediction. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM-16)*, 1149–1154.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW-15)*, 1067–1077.

Tang, R.; Jiang, S.; Chen, X.; Wang, W.; and Wang, W. 2022. Network structural perturbation against interlayer link prediction. *Knowledge-Based Systems*, 250: 109095.

Tong, H.; Faloutsos, C.; and Pan, J.-Y. 2006. Fast random walk with restart and its applications. In *Proceedings of the 6th International Conference on Data Mining (ICDM-06)*, 613–622.

Vahidi Farashah, M.; Etebarian, A.; Azmi, R.; and Ebrahimzadeh Dastjerdi, R. 2021. A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data*, 8: 1–24.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR-18)*.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD-16)*, 1225–1234.

Yao, Y.; Zhang, R.; Yang, F.; Yuan, Y.; Sun, Q.; Qiu, Y.; and Hu, R. 2017. Link prediction via layer relevance of multiplex networks. *International Journal of Modern Physics C*, 28(08): 1750101.

Zhan, Q.; Zhang, J.; and Yu, P. S. 2019. Integrated anchor and social link predictions across multiple social networks. *Knowledge and Information Systems*, 60: 303–326.

Zhou, T.; Lü, L.; and Zhang, Y.-C. 2009. Predicting missing links via local information. *The European Physical Journal B*, 71: 623–630.