A Multi-Type Transferable Method for Missing Link Prediction in Heterogeneous Social Networks

Huan Wang, Ziwen Cui, Ruigang Liu, Lei Fang, and Ying Sha

***Abstract*—Heterogeneous social networks, which are characterized by diverse interaction types, have resulted in new challenges for missing link prediction. Most deep learning models tend to capture type-specific features to maximize the prediction performances on specific link types. However, the types of missing links are uncertain in heterogeneous social networks; this restricts the prediction performances of existing deep learning models. To address this issue, we propose a multi-type transferable method ($MTTM$) for missing link prediction in heterogeneous social networks, which exploits adversarial neural networks to remain robust against type differences. It comprises a generative predictor and a discriminative classifier. The generative predictor can extract link representations and predict whether the unobserved link is a missing link. To generalize well for different link types to improve the prediction performance, it attempts to deceive the discriminative classifier by learning transferable feature representations among link types. In order not to be deceived, the discriminative classifier attempts to accurately distinguish link types, which indirectly helps the generative predictor judge whether the learned feature representations are transferable among link types. Finally, the integrated $MTTM$ is constructed on this minimax two-player game between the generative predictor and discriminative classifier to predict missing links based on transferable feature representations among link types. Extensive experiments show that the proposed $MTTM$ can outperform state-of-the-art baselines for missing link prediction in heterogeneous social networks.**

***Index Terms*—Missing link prediction, heterogeneous social network, transferable feature representation.**

Manuscript received 31 December 2021; revised 27 November 2022; accepted 15 December 2022. Date of publication 2 January 2023; date of current version 6 October 2023. This work was supported by the National Natural Science Foundation of China under Grants 62006089, 62272188, 62102265, in part by the Nature Science Foundation of Hubei Province under Grant 2020CFB168, in part by the Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness under Grant HNTS2022032, in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant GML-KF-22-29, in part by the Natural Science Foundation of Guangdong Province of China under Grant 2022A1515011474, and in part by the Independent Science and technology Innovation Fund project of Huazhong Agricultural University under Grant 2662019QD047. Recommended for acceptance by Ziyu Guan. *(Corresponding author: Ying Sha.)*

Huan Wang and Ying Sha are with the College of Informatics, Huazhong Agricultural University, Wuhan, Hubei 430070, China, and also with the Key Laboratory of Smart Farming for Agricultural Animals, Hubei Engineering Technology Research Center of Agricultural Big Data, Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, Zhengzhou Xinda Institute of Advanced Technology, Zhengzhou, Henan 450001, China (e-mail: hwang@mail.hzau.edu.cn; shaying@mail.hzau.edu.cn).

Ziwen Cui is with the College of Informatics, Huazhong Agricultural University, Wuhan, Hubei 430070, China (e-mail: cuiziwen@webmail.hzau.edu.cn).

Ruigang Liu is with the China Mobile (Hangzhou) Information Technology Co., Ltd., Hangzhou, Zhejiang 311000, China (e-mail: liuruigang@cmhi.chinamobile.com).

Lei Fang is with the School of Computer Science, University of St Andrews, Scotland SC013532, U.K. (e-mail: lf28@st-andrews.ac.uk).

Digital Object Identifier 10.1109/TKDE.2022.3233481

I. Introduction

VARIOUS research lines in social network analysis have attracted a lot of attention [1], from information diffusion [2] to anomalous node detection [3], and from ranking users [4] to unobserved path identification [5]. Among them, the research on missing link prediction in heterogeneous social networks has become a unique challenge. Heterogeneous social networks are usually represented as general network graphs wherein the nodes represent individuals belonging to different categories (e.g., people, organizations, or other social entities), and the links represent interactions of different types (e.g., friendship, co-working, or information exchange). Owing to the complexity of real-world social networks [6], [7], it is difficult to construct a complete network graph to represent the whole heterogeneous social network by observing all existent links. In particular, we refer to the existent links that have not been observed in the network structure as missing links. The existence of missing links is a common phenomenon in the collected heterogeneous network graphs in actual applications [8], which severely affects the integrity of the heterogeneous social network and causes misleading conclusions in social network analysis. Therefore, it is important to develop a universal missing link prediction method for heterogeneous social networks to shape complete social intersections among individuals.

Thus far, various methods have been proposed for missing link prediction, mainly including topological calculation methods [9] and deep learning methods [10]. Topological calculation methods exploit the topological structural attributes among nodes to approximate the likelihood of the existence of a link. By contrast, deep learning methods have achieved impressive performance improvements owing to their strength in automatic feature extraction. However, owing to the type uncertainty of missing links, we cannot obtain the prior information of the types of missing links. The formation of missing links on different types in the social network can follow different evolution mechanisms to shape different type-specific features [11]. Verified links on different types are characterized by type-specific features that are not sharable. Existing deep learning methods tend to extract type-specific features to maximize the prediction performances on specific link types, and their trained models cannot generalize well for other different link types, especially for newly emerged link types that are not covered in the training set. This restricts their performances in missing link prediction in heterogeneous social networks wherein there are uncertain types of missing links. Hence, we aim to construct a generalization model to promote the prediction performance of missing

links by learning transferable feature representations among link types.

To learn transferable feature representations among link types, the first challenge is capturing the shared features on different link types by removing the type-specific features. The shared features and type-specific features are coexistent in the feature representations corresponding to different link types. It is difficult to distinguish shared features and type-specific features. The second challenge is predicting missing links based on learned transferable feature representations among link types. The differences among the link representations on different types are changeable during the training stage, and it is difficult to provide accurate characterizations of shared features among different link types. An efficient use of these shared features is the premise of developing a universal missing link prediction method.

To solve the challenges above, we propose a multi-type transferable method ($MTTM$) for missing link prediction in heterogeneous social networks, consisting of a generative predictor and a discriminative classifier. The generative predictor can extract the feature representations of link samples and predict whether the link sample is missing or not. To generalize well for different link types, the generative predictor is required to be trained on transferable feature representations among link types by capturing shared features. Thus, the discriminative classifier is designed to help the generative predictor judge whether the learned feature representations are transferable. Inspired by adversarial networks [12], the generative predictor attempts to learn transferable feature representations among link types to deceive the discriminative classifier. Simultaneously, the discriminative classifier aims to distinguish the types of link samples based on learned feature representations, and not be deceived by the generative predictor. The corresponding loss of the discriminative classifier can be used to evaluate the extent to which the learned feature representations in the generative predictor can fit the criteria of transferable ones. Finally, the integrated $MTTM$ applies the minimax two-player game between the generative predictor and discriminative classifier to predict missing links in heterogeneous social networks.

The remainder of this paper is organized as follows. Section II introduces the background of our study. Section III formally defines the missing link prediction problem in heterogeneous social networks, which is solved by the proposed $MTTM$ as described in Section IV. In Section V, we detail the extensive experiments performed to verify the performance of $MTTM$ on real-world social networks. Section VI concludes this paper and outlines future study directions.

II. Background

In this section, we briefly introduce the preliminary concepts of heterogeneous social networks and adversarial neural networks, and discuss related work.

A. Preliminaries

Heterogeneous Social Network: Most real-world social systems contain multi-typed interacting components that generate numerous social interactions; we model them as heterogeneous social networks with different node categories and link types. The heterogeneous social network extracts relatively complex structural information from real social systems, and strictly distinguishes the heterogeneity of nodes and links in data [13]. Compared with homogeneous networks, the heterogeneous network can effectively fuse more structural information in a unified mechanism and contain rich semantic meaning of node categories and link types. There is an influx of heterogeneous networks in many data mining tasks, especially in recommend systems. Many excellent heterogeneous network representation methods [14], [15], [16] have been developed for recommend systems. Heterogeneous social networks have also brought new challenges for missing link prediction because of the type uncertainty of missing links.

Adversarial Neural Networks: Adversarial neural networks are composed of a generative model and a discriminative model [12]. The task of the generative model is to generate natural-looking and realistic instances that are similar to the original data in real-world applications. Simultaneously, the discriminative model is designed to attempt to determine whether a given generated instance appears genuine. These two models are in constant conflict with each other. The generative model attempts to deceive the discriminative model by generating indistinguishable instances to simulate the original data, and the discriminative model attempts not to be deceived by the generative model. Through adversarial training, the generative model and the discriminative model continue to develop and finally achieve a balance when they become optimal. Adversarial neural networks have been adopted for many tasks in different application domains.

B. Related Work

As an important line of link prediction research [17], the research of missing link prediction has attracted a lot of attentions. Many useful methods can be applied to solve the missing link prediction problem, mainly including topological calculation methods and deep learning methods.

Topological calculation methods have been in existence for a long time, which exploit the topological structural attributes among nodes to approximate the likelihood of the existence of a link, such as similarity-based methods [18], probabilistic models [19], [20], and maximum likelihood methods [21], [22]. Tian and Zafarani [23] considered that the specific assumptions in most link prediction methods may lead to the bad generalization ability for different networks. They proposed general link prediction methods that captured network-specific patterns. Their proposed methods measured the pairwise similarities between nodes more accurately, even when only using common neighbor information. Yu et al. [24] studied fast high-quality link-based similarity search on billion-scale graphs. They devised a "varied-D" method to accurately compute SimRank in linear memory and aggregated duplicate computations. They proposed a novel "cosine-based" SimRank model to circumvent the "connectivity trait" problem.

Concomitant with the development of deep learning technologies [25], many deep learning methods have been developed to

solve the issue of missing link prediction. Because of superior feature extraction [26], deep learning methods have achieved excellent results in missing link prediction. Zhang and Chen [27] proposed a graph neural network framework ($SEAL$) to obtain heuristics from each local subgraph and learn the heuristic learning paradigm. Their method captured general graph structure features from local enclosing subgraphs and defined a function that mapped the subgraph patterns to the link existence. In addition, Cen et al. [28] proposed a general attributed multiplex heterogeneous network embedding ($GATNE$) to address the problem of embedding learning and provided an effective link prediction approach for attributed multiplex heterogeneous networks. Their proposed framework both captured rich attribute information and utilized multiplex topological structures from different node categories for transductive and inductive learning.

Further, many missing link prediction methods have been specially designed to exploit heterogeneous structural information. Hu et al. [29] developed a heterogeneous general adversarial network ($HeGAN$) for heterogeneous networks embedding, which trained both a discriminator and a generator in a minimax game. Their generator learned the node distribution to generate negative samples and cooperated with a discriminator to capture the rich heterogeneous semantics. Negi et al. [30] consider the link prediction in heterogeneous networks as a multitask, metric learning problem. They utilized both network and node features to learn the distance measure in a coupled fashion by employing the multitask structure preserving metric learning setup. In addition, Wang et al. [31] designed a self-supervised learning of contextual embedding ($SLiCE$) model using localized attention driven mechanisms. They pre-trained their model in a self-supervised manner by introducing higher-order semantic associations and masking nodes, and then fine-tuned it for a specific link prediction task. Chen et al. [32] proposed a projected metric embedding model (PME) on heterogeneous networks for link prediction. They captured both first-order and second-order proximities in a unified manner to alleviate the potential geometrical inflexibility of existing metric learning approaches. Further, Zhang et al. [33] proposed a heterogeneous graph neural network model by jointly considering heterogeneous structural information and content information. They leveraged a graph context loss and a mini-batch gradient descent procedure to train the model in an end-to-end manner. Fu et al. [34] proposed a Metapath Aggregated Graph Neural Network by employing node content transformation, intra-metapath, and inter-metapath aggregation. The content information of nodes was fully exploited in their embedding work [33], [34].

However, although the aforementioned studies can achieve good performances of missing link prediction in heterogeneous social networks, their ignoration of the potential uncertainty in the corresponding types of missing links restricts their prediction performances. The missing links may belong to different link types, and each link type has its type-specific features that are not sharable with other types. Existing deep learning models tend to capture the type-specific features from link samples and may provide contradictory prediction results of missing links on different types. Therefore, this study aims to capture shared features among different link types to propose a universal missing link prediction method in heterogeneous social networks.

TABLE I
THE FREQUENTLY-USED SYMBOLS IN THIS STUDY

Symbol	Meaning	Symbol	Meaning
G	Heterogeneous social network.	T_p^h	Set of historical link types.
A	Set of node categories.	φ	Mapping function.
V	Set of observed nodes.	e	A link sample.
E_U	Set of unobserved links.	T_p	Possible link-type sets.
Ms	Matching set.	T_p^n	Set of new link types.
E	Set of observed links.	S	Link sample set.
$r_{(e)}$	Feature Representation of the link sample e.	$f(u)$	Feature representation of node u.
$P(e)$	Existent likelihood of the link sample e.	R_F	Set of feature representations of link samples.
L_{final}	Final loss of this mini-max game.	(a_1, a_2)	Link type of the link sample e.
L_p	Prediction loss.	L_c	Classification loss.
S_T	Training sample set.	η	Learning rate.
Y_T	Set of sample labels.	Z_T	Type label set

III. PROBLEM DEFINITION

To facilitate the presentation in this study, Table I summarizes the frequently used symbols. A heterogeneous social network is expressed as a network graph denoted by $G = (V, E, A, \varphi)$. V and E denote the sets of observed nodes and links, respectively. A denotes the set of node categories and $|A| \geq 2$. φ denotes the mapping function from the node in V to the node category in A. $\varphi(v)$ denotes the node category of a node $v \in V$ and $\varphi(v) \in A$. In addition, although the social interactions among individuals are diverse, social data can be universally represented in the form of undirected and unweighted graphs. To develop a generalized method for missing link prediction in heterogeneous social networks, G is defined as an undirected and unweighted graph.

The set of unobserved links can be denoted by $E_U = \{(i,j)|i \in V \cap j \in V \cap i \neq j \cap (i,j) \notin E\}$. The unobserved links in E_U may contain missing links, the existence of which we aim to predict. The set of possible link types can be denoted as $T_p = \{\langle a_1, a_2\rangle | a_1 \in A \cap a_2 \in A\}$, where a_1, a_2 and a_2, a_1 represent the same link type. The types of links in E are collected to construct the historical link-type set T_p^h. Except for T_p^h, the remaining types in T_p construct the new link-type set T_p^n. The types of the missing links in E_U are unknown, and each missing link may belong to a certain historical type in T_p^h or a certain new type in T_p^n. The uncertainty of the types of missing links creates the prediction challenge in a heterogeneous social network. Formally, we define this problem as follows: given $G = (V, E, A, \varphi)$, we need to design a matching set $Ms = \{(e, \delta)|e \in E_U \cap \delta \in [01]\}$ for the unobserved links in E_U, where each unobserved link e in E_U is assigned with a reasonable value δ to quantify its existent likelihood. The missing link prediction problem can be considered as a binary classification, which classifies the unobserved links in E_U into the missing link set E_m and the nonexistent link set E_n. The

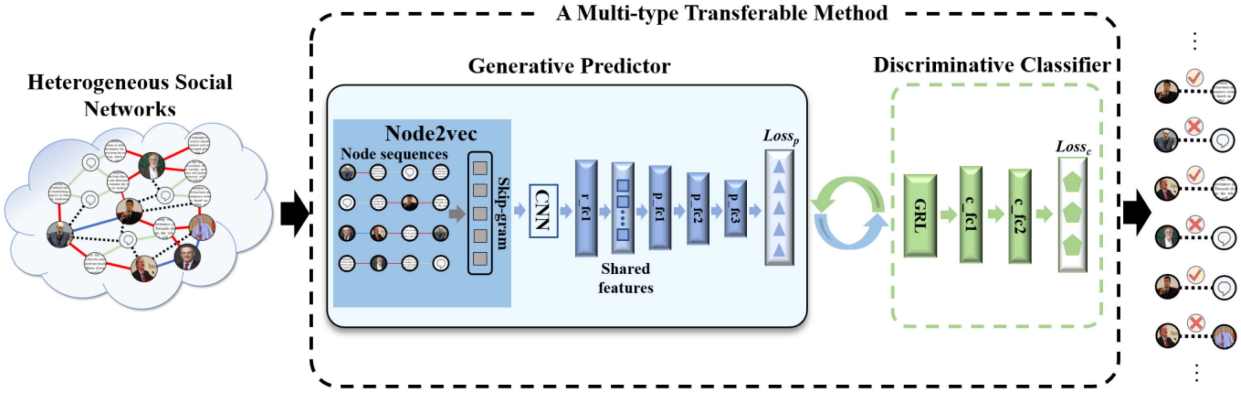


Fig. 1. Framework of $MTTM$.

perfect solution to this problem is that $\delta = 1$ for the link in E_m and $\delta = 0$ for the links in E_n.

IV. Methodology

We detail the proposed $MTTM$ in this section. An overview of $MTTM$ is presented in Section IV-A. Section IV-B explains the generative predictor that predicts whether the link sample is missing or not based on learned feature representations. Section IV-C explains the discriminative classifier used to distinguish the accurate types of different link samples. In Section IV-D, based on the generative predictor and discriminative classifier, the integration of $MTTM$ to predict missing links in a heterogeneous social network is detailed.

A. Framework

In this section, the framework of our proposed $MTTM$ is illustrated in Fig. 1. Differing from traditional homogeneous networks, the inherent heterogeneity of heterogeneous social networks leads to the diversity and uncertainty of missing links. To develop a universal method for missing link prediction on different link types, we propose $MTTM$ to provide a generalized prediction method based on transferable feature representations among link types. To achieve this, as shown in Fig. 1, the generative predictor and discriminative classifier in $MTTM$ cooperate in missing link prediction through a minimax two-player game. Based on the heterogeneous structure of links, the generative predictor attempts to learn transferable feature representations among different link types to deceive the discriminative classifier. Simultaneously, to avoid being deceived by the generative predictor, the discriminative classifier distinguishes the link types and attempts to predict the accurate link types. Finally, the integrated $MTTM$ combines the generative predictor and discriminative classifier to learn transferable feature representations among link types as input for missing link prediction in the heterogeneous social network.

B. Generative Predictor

The generative predictor aims to predict whether link samples are missing links or not based on learned feature representations. Many state-of-the-art representation learning methods have been developed to produce a low-dimensional vector embedding for each node in the heterogeneous social network, such as Metapath2vec [35], ie-HGCN [36], and MAGNN [37]. However, to extract shared features among link types, we use node2vec as core module to reduce the extra semantic information brought by the type heterogeneity of the heterogeneous social network [38].

We first introduce $f(u)$ as the mapping function from a node u to its feature representation by node2vec. For a link $e = (u, v)$, we first obtain the feature representations $f(u)$ and $f(v)$ of the nodes u and v, respectively. Then, the initial feature representation of the link sample e is expressed as follows.

$$r_{(e)} = f(u) * f(v) \tag{1}$$

We denote the set of inputted link samples as S. To train the generative predictor to obtain generalization ability, S is required to contain link samples on more than one type during the training stage. To keep the input flexibility in actual applications, S can contain link samples on one or more types during the prediction stage. To represent the initial feature representations of all link samples in S, we define the initial representation set R_F as follows.

$$R_F = \{r_{(e)} | e \in S\} \tag{2}$$

A convolutional neural network and a full connection layer are added to aggregate the transferable feature representations based on the initial feature representations in R_F. θ_r is used to represent the parameters to be learned in this process. To achieve this, we introduce the aggregated representation set $\bar{R}_F$. Similar to the construction of R_F, we define $\bar{R}_F = \{\bar{r}_{(e)} | e \in S\}$, where $\bar{r}_{(e)}$ represents the aggregated feature representation of the link sample e in S. In the competition process between the generative predictor and discriminative classifier, the aggregated feature representations in $\bar{R}_F$ are consistently adjusted to be transferable ones by capturing the shared features in R_F. The final aggregated feature representations in $\bar{R}_F$ are considered as transferable ones, which are used as the final ground truth for training. We further use $G_p(S; \theta_r, \theta_p)$ to denote the generative predictor. Three fully connected layers with the softmax function are employed to distinguish between missing links and nonexistent

links. θ_p represents their contained parameters. For a given link sample e, the output of the generative predictor can be denoted as follows.

$$P(e) = G_p(\{e\}; \theta_r, \theta_p) \tag{3}$$

Here, $P(e)$ denotes the existent likelihood of e. Among unobserved link samples, the existent likelihoods of missing links are reasonable to have larger values than these of nonexistent links. A large $P(e)$ value implies that the link sample e has a large likelihood to be a missing one. For a given link sample set S, we define the prediction loss of the generative predictor by cross entropy as follows.

$$L_p(\theta_r, \theta_p) = -\sum_{e \in S} [m_e \log(P(e)) + (1 - m_e) \log(1 - P(e))] \tag{4}$$

Here, $m_e \in \{0, 1\}$. $m_e = 1$ denotes that the link sample e is positive; otherwise, $m_e = 0$. To achieve a better prediction performance of missing links, the preliminary task of the generative predictor is to minimize the prediction loss. This process to seek the optimal parameters $\hat{\theta}_r$ and $\hat{\theta}_p$ can be expressed as follows.

$$\left(\hat{\theta}_r, \hat{\theta}_p\right) = arg \min_{\theta_r, \theta_p} L_p(\theta_r, \theta_p) \tag{5}$$

The direct minimization of the prediction loss in (5) promotes the learning performance of discriminable representations by capturing both type-specific features and shared features in R_F. However, the challenge for missing link prediction in heterogeneous social networks is that the types of missing links are uncertain, and each missing link may belong to a historical type or a new type. The deep learning models based on discriminable representations have not enough generalization ability to achieve a general performance promotion on different link types. Especially, without enough generalization ability, these models are difficult to achieve a good prediction performance on the new link types that are not covered in the training set. To improve the prediction performance of missing links, we are inspired to learn transferable feature representations to generalize well among both historical types and new types. Therefore, we need to enable the generative predictor to learn more general feature representations that can be transferable from one link type to other link types. Such feature representations should capture the shared features among link types and remove their type-specific features. To achieve this, we require the following discriminative classifier to help the generative predictor judge whether the learned feature representations in $\bar{R}_F$ fit the criteria of transferable ones. With the help of the discriminative classifier, we can train the generative predictor with a certified $\bar{R}_F$, further obtaining the generalization ability to predict missing links.

C. Discriminative Classifier

The discriminative classifier is a neural network that consists of two fully connected layers with corresponding activation functions. We use $G_c(\bar{R}_F; \theta_r, \theta_c)$ to represent the discriminative classifier, where θ_c represents the parameters to be learned, and $\bar{R}_F$ and θ_r are from the generative predictor. The task of the discriminative classifier is to judge whether the learned feature representations in $\bar{R}_F$ are transferable. To achieve this task, the discriminative classifier is designed to distinguish the link types and attempts to predict the accurate types of link samples based on the learned feature representations in $\bar{R}_F$.

We define the classification loss of the discriminative classifier by cross entropy as follows.

$$L_c(\theta_r, \theta_c) = -\sum_{e \in S} \sum_{a_1, a_2 \in T_p^h} n_e \log \times \left(G_c^{a_1, a_2}\left(\left\{\bar{r}_{(e)}\right\}; \theta_r, \theta_c\right)\right) \tag{6}$$

Here, T_p^h contains the types of the link samples in S, and $G_c^{a_1,a_2}(\{\bar{r}_{(e)}\}; \theta_r, \theta_c)$ represents the probability that classifies the link type of the link sample e as a_1, a_2 in T_p^h. If the correct link type of e is a_1, a_2, $n_e = 1$; otherwise, $n_e = 0$. When the $L_c(\theta_r, \theta_c)$ value is smaller, the discriminative classifier obtains a better performance to classify the link samples in S into correct types. The parameter of the discriminative classifier after minimizing this loss $L_c(\theta_r, \theta_c)$ is expressed as follows.

$$\hat{\theta}_c = arg \min_{\theta_c} L_c(\theta_r, \theta_c) \tag{7}$$

The classification loss $L_c(\theta_r, \hat{\theta}_c)$ can indirectly evaluate the extent to which the learned feature representations in $\bar{R}_F$ can fit the criteria of transferable ones. When the $L_c(\theta_r, \hat{\theta}_c)$ value is larger with a worse classification performance, the learned feature representations in $\bar{R}_F$ can better fit the criteria of transferable ones. Therefore, we need to maximize the $L_c(\theta_r, \hat{\theta}_c)$ by seeking the optimal parameters θ_r during the training stage, which helps the feature representations in $\bar{R}_F$ fit the criteria of transferable ones. Therefore, we are inspired to construct the minimax two-player game between the generative predictor and discriminative classifier. The generative predictor $G_p(S; \theta_r, \theta_p)$ tries to learn the transferable feature representations to deceive the discriminative classifier $G_c(\bar{R}_F; \theta_r, \theta_c)$ by capturing shared features and removing type-specific features for $\bar{R}_F$, and the discriminative classifier $G_c(\bar{R}_F; \theta_r, \theta_c)$ attempts not to be deceived by discovering the type-specific features in $\bar{R}_F$ to recognize the link type. Based on the minimax two-player game, the comprehensive loss function is defined as follows.

$$L_{final}(\theta_r, \theta_p, \theta_c) = L_p(\theta_r, \theta_p) - \lambda L_c(\theta_r, \theta_c) \tag{8}$$

Here, λ is introduced to control the trade-off between the prediction loss and the classification loss. The parameter set we seek is the saddle point of the comprehensive loss function.

$$\left(\hat{\theta}_r, \hat{\theta}_p\right) = arg \min_{\theta_r, \theta_p} L_{final}\left(\theta_r, \theta_p, \hat{\theta}_c\right) \tag{9}$$

$$\hat{\theta}_c = arg \max_{\theta_c} L_{final}\left(\hat{\theta}_r, \theta_c\right) \tag{10}$$

As shown in Fig. 1, the gradient reversal layer (GRL) [39] is added between the generative predictor and discriminative classifier. Based on the GRL, the feature extraction process in the generative predictor has an opposite goal with the discriminative classifier, namely reducing the classification performance and confusing the types of links. The GRL multiplies gradient with

$-\lambda$ and passes the results to the preceding layer during backprop stage. As recommended by [39], we introduce a learning rate η and the update of θ_r can be expressed as follows.

$$\theta_r \leftarrow \theta_r - \eta \left(\frac{\partial L_p}{\partial \theta_p} - \lambda \frac{\partial L_c}{\partial \theta_c} \right) \tag{11}$$

D. Method Integration

Deploying deep learning models requires positive and negative link samples during the training stage. In the problem of missing link prediction, we consider the observed links in E as positive samples, and randomly select $|E|$ unobserved links with the same types from the unobserved link set E_U as negative samples ($|E| \ll |E_U|$). We combine these positive and negative samples to construct the training sample set S_T, and obtain its corresponding sample label set Y_T and type label set Z_T. Based on a minimax two-player game between $G_p(S_T; \theta_r, \theta_p)$ and $G_c(\bar{R}_F; \theta_r, \theta_c)$, the generative predictor $G_p(S_T; \theta_r, \theta_p)$ is trained on $\{S_T, Y_T, Z_T\}$ to learn the generalization prediction model during the training stage, which continuously optimizes the shared features in $\bar{R}_F$ to deceive the discriminative classifier $G_c(\bar{R}_F; \theta_r, \theta_c)$ to confuse the types of link samples.

Without the help of the discriminative classifier, we still can train the generative predictor to predict missing links based on the initial feature representations in R_F. However, to generalize well for different link types, the generative predictor is expected to predict missing links based on the transferable feature representations in $\bar{R}_F$. To archives this, we require the discriminative classifier to help the generative predictor judge whether the learned feature representations in $\bar{R}_F$ fit the criteria of transferable ones. Using this adversarial network, the minimax two-player game between the discriminative classifier and the generative predictor is the premise of the transition process from the initial feature representations in R_F to the certified transferable feature representations in $\bar{R}_F$. Finally, the trained generative predictor in $MTTM$ is carried out to predict missing links in E_U during the prediction stage. Given a heterogeneous social network $G = (V, E, A, \varphi)$, $MTTM$ quantifies the existent likelihoods of the unobserved link samples in E_U to obtain the matching set Ms. The process of $MTTM$ is detailed as follows.

Considering the number of the nodes in V, the total number of node feature representations in the feature extraction process can be estimated as $|V| \times d$ and the corresponding cost will be $O(|V| \times d)$, where d represents the dimension of the feature representation. Subsequently, the dot product is the main step for calculating the feature representations of all link samples, and its cost is quadratic to the number of nodes, denoted as $O(|V|^2 \times d)$. Further, the generative predictor and discriminative classifier are assumed to contain h_1 and h_2 hidden layers, respectively. Since the adversarial training that involves the parameter calculation needs to provide us with a good approximation of the total edge number $|E|$, we can approximate this cost as $O(|E| \times (h_1 + h_2) \times d)$. Finally, the overall time complexity of $MTTM$ is $O((|V|^2 + |E| \times (h_1 + h_2)) \times d)$, which is comparable to most of existing link prediction methods.

Method: $MTTM$

INPUT: G – Heterogeneous social network.
S_T – Training set.
Y_T – Sample label set corresponding to S_T.
Z_T – Type label set corresponding to S_T.
η – Learning rate.
OUTPUT: Ms– Matching set

1: $Ms = \emptyset$
2: For each train iteration **do**
3: Update the parameters θ_r:
4: $\theta_r \leftarrow \theta_r - \eta(\frac{\partial L_p}{\partial \theta_p} - \lambda \frac{\partial L_c}{\partial \theta_c})$
5: Update the parameters θ_p:
6: $\theta_p \leftarrow \theta_p - \eta \frac{\partial L_p}{\partial \theta_p}$
7: Update the parameters θ_c:
8: $\theta_c \leftarrow \theta_c - \eta \frac{\partial L_c}{\partial \theta_c}$
9: End for
10: For each unobserved link sample e in E_U **do**
11: Calculate $P(e)$ by (3).
12: $Ms = Ms + \{(e, P(e))\}$.
13: End for
14: Output Ms.

V. Experimental Analysis

This section describes the extensive experimental evaluation conducted. In Section V-A, we detail the experimental setup, including datasets, comparison methods, and parameter settings. In Section V-B, we verify the performance of $MTTM$ by comparing it with state-of-the-art methods. In addition, $MTTM$ is constructed on the assumption learning the transferable feature representations among link types is beneficial to improve the prediction performance of missing links in heterogeneous social networks. Therefore, we analyze the importance of learning transferable feature representations among link types in $MTTM$ in Section V-C. As an important parameter for $MTTM$, the sample ratio r decides the division of the test set and the training set. Thus, Section V-D discusses the effect of the setting of r on the prediction performance of $MTTM$. Further, the heterogeneity of heterogeneous social networks leads to the type uncertainty of missing links. To insight into $MTTM$, we design a case study to illustrate the influence of the type uncertainty of missing links in Section V-E.

A. Experimental Setup

Datasets: We consider the following two real-world datasets drawn from disparate fields. (1) Facebook [40]. Facebook dataset is a page–page graph of verified Facebook sites collected through the Facebook Graph API. The nodes represent official Facebook pages whereas the links are mutual likes between sites. The collected pages are divided into four categories: politicians, governmental organizations, television shows, and companies. (2) DBLP [41]. DBLP is a bibliographic network dataset in computer science collected from four research areas: database, data mining, machine learning, and information retrieval. DBLP

TABLE II
BASIC STATISTICS OF FOUR DATASETS

Dataset	Facebook	DBLP	IMDB	Yelp
#Node	22470	14475	45,496	28,759
#Node category	4	4	4	4
#Link	171002	170794	136093	247698
#Effective Link type	10	3	3	3

TABLE III
NODE CATEGORIES AND LINK TYPES OF FOUR DATASETS

	#Node categories	#Effective Link types
Face-book	politician company government tvshow	<politician, company> <government,tvshow> <government,political> <government,company> <tvshow,politician> <tvshow,company> <tvshow,tvshow> <politician,politician> <company,company> <government,government>
DBLP	paper venue term author	<paper,venue> <paper, term> <paper, author>
IMDB	user movie actor director	<user, movie> <actor, movie> <director, movie>
Yelp	user business location category	<business, location> <business, user> <business, category>

demonstrates how extensive literature references can lead to the emergence of various structural properties. (3) IMDB [42]. IMDB is a link data set collected from the Internet Movie Data. The network used in the experiment contains four types of objects. In the dataset, 1357 movies are labeled with at least one of the 23 labels. (4) Yelp [42]. The data set was extracted from a user review website in America, Yelp, containing four types of objects. Note that when the original heterogeneous social network is weighted or directed, we treat it as a simple network by ignoring its weights and directions. The basic statistics of these four datasets are summarized in Table II, and their node categories and link types are presented in Table III.

Comparison Methods: As mentioned in Section II-B, the five state-of-the-art prediction methods, $SEAL$[27], $GATNE$ [28], $HeGAN$[29], $SLiCE$ [31], and PME [32] are considered as comparison methods. In addition, the missing link prediction problem can be considered as a binary classification that distinguishes between missing links and nonexistent links. Thus, we also develop three typical missing link prediction methods based on support vector machine (SVM) [43], Logistic regression (LR) [44], and random forest (RF) [45].

Parameter Settings: The settings of $SEAL$, $GATNE$, $HeGAN$, $SLiCE$, and PME are consistent with their original settings. Based on the preliminary feature representations of links in the generative predictor, SVM, LR, and RF are developed to distinguish positive samples from negative samples according to their recommended settings [35], [36], [37], [38], [39], [40], [41], [42], [43]. SVM adopts the Linear kernel as its kernel function and sets the penalty coefficient as 50. The parameter of $solver$ in LR selects the optimization algorithm lbf. We set $n\_estimators = 50$ in RF and the learning rate $\eta = 0.001$ in the stochastic gradient descent. In addition, for each iteration of the generative predictor and discriminative classifier training, we use a batch size of 32. We run 29 iterations to train the generative predictor and discriminative classifier in each epoch. The number of embedding dimensions for all methods are set as 64. For random walk-based methods, we set the walk number of each node to $w = 10$, the walk length to $l = 5$, and the window size to $\tau = 10$. We set λ as 1 without tuning the trade-off parameter. The corresponding code link of the GitHub page is https://github.com/xihairanfeng/A_Multi_type_Transferable_Method.

B. Performance Comparison

To verify the prediction performance of our proposed $MTTM$, we compare it with eight comparison methods. First, for each dataset, we use r to denote the sample ratio. We randomly select $r \times |E|$ links from the observed link set E as positive samples in the training set, and the remaining observed links are considered as the positive samples in the test set. Secondly, we randomly select $r \times |E|$ unobserved links from E_U as negative samples in the training set, and randomly select $(1 - r) \times |E|$ unobserved links from the remaining unobserved links as negative samples in the test set. We set $r = 0.5$ as an example in this section. The positive samples and negative samples are considered as missing links and nonexistent links, respectively. Based on the training set, the problem of missing link prediction requires $MTTM$ and eight comparison methods to predict the positive link samples in the test set.

To avoid the randomness influence in the selection process of positive and negative link samples, each prediction method is repeated 40 independent times for each dataset. Three common evaluation indices are introduced to verify the prediction performances of missing links: AUC, $Accuracy$, and $Precision$[46], [47], [48]. AUC focuses on the whole prediction performance, while $Precision$ focuses on the prediction performance of top-ranked link samples. $Accuracy$ is commonly defined as the ratio $\frac{TP+TN}{N}$, where N denotes the total number of link samples, and TP and TN denote the correct prediction times of positive samples and negative samples, respectively. The values of AUC, $Precision$, and $Accuracy$ obtained by $MTTM$ and the eight comparison methods are shown in Table IV. Their highest values are highlighted in bold.

Among nine missing link prediction methods, $MTTM$ always exhibits the best prediction performances of missing links in heterogeneous social networks. As shown in Table IV, compared with $SEAL$, $GATNE$, $HeGAN$, $SLiCE$, PME, SVM, LR, and RF, $MTTM$ obtains the higher values of AUC, $Accuracy$, and $Precision$ on four datasets. The missing links on the datasets of Facebook, DBLP, IMDB, and Yelp can contain multiple link types. Ignoring the type uncertainty

TABLE IV
AUC, *Precision*, AND *Accuracy* VALUES OBTAINED BY $MTTM$ AND EIGHT COMPARISON METHODS

Dataset	Method	*AUC*	*Precision*	*Accuracy*
Facebook	*MTTM*	***0.9204***	***0.7191***	***0.8206***
	SEAL	0.8261	0.7782	0.7546
	GATNE	0.4503	0.1731	0.7425
	HeGAN	0.5118	0.1698	0.4146
	SLiCE	0.5144	0.1456	0.7956
	PME	0.4981	0.1545	0.7335
	MLAN	0.8693	0.6780	0.8012
	SVM	0.7599	0.2931	0.6316
	RF	0.8197	0.4342	0.8020
	LR	0.5715	0.1795	0.2864
DBLP	*MTTM*	***0.8439***	***0.8242***	***0.8285***
	SEAL	0.80291	0.4667	0.4995
	GATNE	0.6152	0.6914	0.5651
	HeGAN	0.4157	0.1940	0.3305
	SLiCE	0.5437	0.5766	0.5670
	PME	0.4376	0.5217	0.4502
	MLAN	0.8238	0.6123	0.7973
	SVM	0.5572	0.6905	0.5172
	RF	0.5692	0.7914	0.5163
	LR	0.6396	0.6702	0.6686
IMDB	*MTTM*	***0.9892***	***0.8524***	***0.8046***
	SEAL	0.7341	0.6486	0.4615
	GATNE	0.5826	0.5369	0.5734
	HeGAN	0.5771	0.5439	0.5730
	SLiCE	0.5027	0.4894	0.5142
	PME	0.4820	0.4642	0.4826
	MLAN	0.9123	0.6523	0.7823
	SVM	0.5024	0.5112	0.5181
	RF	0.8275	0.8449	0.7300
	LR	0.7058	0.6155	0.6961
Yelp	*MTTM*	***0.9136***	***0.7887***	***0.7453***
	SEAL	0.7702	0.6928	0.5567
	GATNE	0.5376	0.5275	0.5301
	HeGAN	0.5594	0.6224	0.5436
	SLiCE	0.4902	0.4887	0.4852
	PME	0.4970	0.5002	0.5059
	MLAN	0.9002	0.6023	0.6234
	SVM	0.5897	0.7125	0.5875
	RF	0.6630	0.7034	0.6753
	LR	0.7937	0.7061	0.6928

of missing links, the comparison methods focus on the direct minimization of the prediction loss of missing links on known link types in the training set. They tend to capture type-specific features that cannot generalize well among different link types. However, the types of missing links in the test set are uncertain and diversity, and each missing link may belong to a known link type in the training set or a new effective link type. The lack of the prior type information of missing links in the test set restricts the prediction performances of the comparison methods in heterogeneous social networks. In contrast, $MTTM$ resists the disturbance of the type uncertainty of missing links in the test set, which exploits adversarial networks to learn transferable feature representations among link types. Therefore, $MTTM$ displays a substantial prediction improvement in heterogeneous social networks.

Furthermore, all methods are implemented in Python, and the experiments are executed using four threads of a 2.60 GHz Intel(R) Xeon(R) Gold 6132 processor. We do not measure the common time to load datasets into memory. The average computing time of an independent run of $MTTM$, $SEAL$, $GATNE$, $HeGAN$, $SLiCE$, PME, SVM, RF and LR is 1.02, 1.24, 1.36, 1.82, 1.75, 1.23, 0.42, 0.45 and 0.56 minutes, respectively. They can be speed up by parallel algorithms. Except for the machine learning based methods SVM, RF and LR, the computing time of $MTTM$ is less than that of $SEAL$, $GATNE$, $HeGAN$, $SLiCE$, and PME.

C. Importance of Learning Transferable Feature Representations

To demonstrate the importance of learning transferable feature representations during the training stage, we design a variant of the proposed $MTTM$ for comparison. To enable our proposed $MTTM$ to learn the transferable feature representations, we require the discriminative classifier to decide whether the extracted features fit the criteria of transferable feature representations through the maximization process of (10). However, even without this maximization process, we still can predict missing links in heterogeneous social networks based on their preliminary feature representations. Thus, we design a variant of the proposed model, named $MTTM^{1-}$. The only difference between $MTTM$ and $MTTM^{1-}$ is that $MTTM^{1-}$ does not consider the maximization process of (10) in the discriminative classifier. We then conduct $MTTM$ and $MTTM^{1-}$ to predict missing links on Facebook and DBLP. Averaged over 40 independent runs, the histograms of the performance comparisons between $MTTM$ and $MTTM^{1-}$ are shown in Fig. 2.

Compared with $MTTM^{1-}$, $MTTM$ exhibits an obvious performance improvement. This is a strong suggestion that learning transferable feature representations is important and beneficial to improve the prediction performance of missing links in heterogeneous social networks. As shown in Fig. 2, on the datasets of Facebook, DBLP, IMDB, and Yelp, $MTTM$ always obtains larger evaluation values than $MTTM^{1-}$ in terms of AUC, $precision$, and $accuracy$. Based on a minimax two-player game between the generative predictor and discriminative classifier, $MTTM$ is capable of learning general link representations that can be transferred from one link type to other link types. The generative predictor attempts to capture the shared features among link types to deceive the discriminative classifier, while the discriminative classifier tries to distinguish link types to not be deceived. In contrast, without the maximization process of (10), $MTTM^{1-}$ focuses on learning nontransferable type-specific features, which makes it lose the generalization to resist the type uncertainty of missing links in the prediction process. Therefore, benefiting from the learning of transferable feature representations, $MTTM$ is reasonable to achieve a better prediction performance than $MTTM^{1-}$.

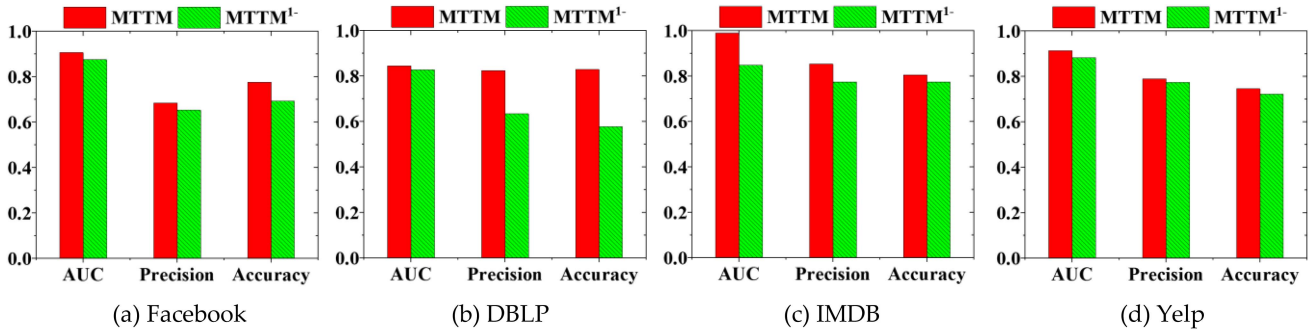


Fig. 2. AUC , $Precision$, and $Accuracy$ values using $MTTM$ and $MTTM^{1-}$ on Facebook, DBLP, IMDB and Yelp.

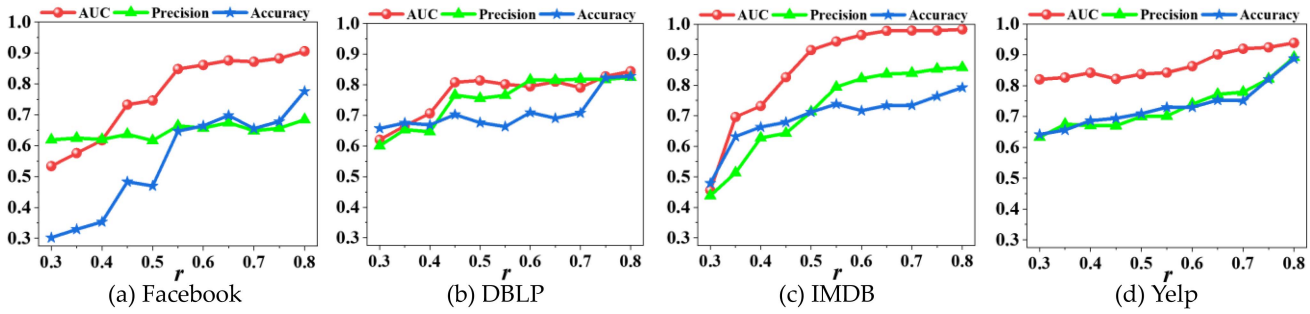


Fig. 3. Changes with r in AUC, $Precision$, and $Accuracy$ values by $MTTM$ on Facebook, DBLP, IMDB and Yelp.

D. Parameter Analysis

The division of the training set and the test set plays a crucial role in the performance evaluation of $MTTM$. This section discusses the effect of the setting of r on the prediction performance of $MTTM$. To achieve a controllable division, we adjust the sample ratio r to analyze the performance change of $MTTM$. Along with the change in the r value, the changes in the AUC, $Precision$, and $Accuracy$ values obtained by $MTTM$ on Facebook and DBLP are shown in Fig. 3.

As shown in Fig. 3, the division of the training set and the test set is important and affects the prediction performance of $MTTM$. The two aspects underlying the observations can be summarized as follows. (1) As the r value increases, more positive samples and negative samples are added to enlarge the size of the training set, which effectively promotes the prediction performance of $MTTM$ in the test set. Thus, the AUC, $Precision$, and $Accuracy$ values are always increasing on four datasets. (2) In most cases, the growth velocities of the AUC, $Precision$, and $Accuracy$ values decrease with the increase in r. When the r value increases to a certain degree, $MTTM$ can obtain a sufficient number of labeled link samples during the training stage, and the addition of new labeled link samples cannot significantly improve its performance.

E. Case Study

In this section, to analyze the influence of the type uncertainty of missing links, we design a case study to conduct comparison experiments as follows. To ensure the setting flexibility of link types, we use the dataset of Facebook with 10 effective link types as a case. The experiment is designed as follows. First, we use T_v to represent the set of effective link types. For each dataset, we randomly select $\lfloor \frac{2}{3} \times |T_v| \rfloor$ link types in T_v to construct the historical link-type set T_p^h, and the other link types are used to construct the new link-type set T_p^n. Next, we use the link samples on the link types in T_p^h to construct the training set and the test set. Furthermore, to achieve a controllable type uncertainty in the test set, we control the addition of the link samples on new link types in T_p^n. We construct the new sample set S', where the observed links on the link types in T_p^n are considered as its positive samples and the equivalent unobserved links within same link types are randomly selected as its negative samples.

Based on the steps above, the types of the link samples in S' have not been existed in the training set and the test set. We set a ratio ξ and randomly select $\xi|S'|$ link samples from S' to add into the test set. Along with the increase in the ξ value, the test set has more link samples on new types, and the type uncertainty of missing links in the test set increases. We adjust the ratio ξ and use AUC to evaluate the overall performance of $MTTM$ and eight comparison methods. The changes in AUC values by $MTTM$ and eight comparison methods are shown in Fig. 4.

As shown in Fig. 4, with different ξ values, the overall prediction performance of our proposed $MTTM$ is always better than these of the eight comparison methods. This confirms that our proposed $MTTM$ is robust to resist the disturbance of the type uncertainty of missing links in the prediction process. Based on the limited number of verified link samples in the training set, the increase in the ξ value results in the number increase of link samples on the test set, which degrades the overall prediction performance of all methods. Though the AUC

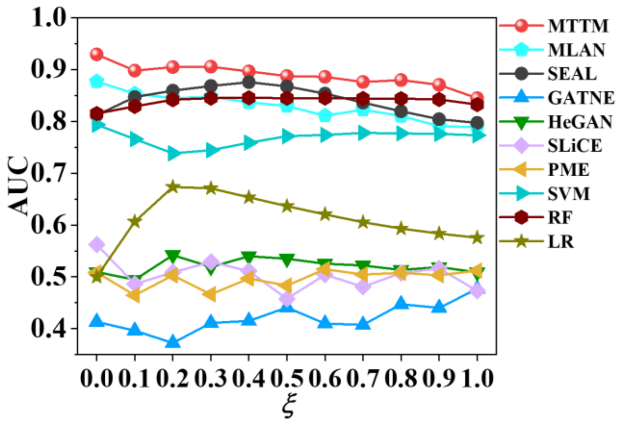


Fig. 4. Changes in AUC values by different methods.

values of $MTTM$ and eight comparison methods gradually decrease with the gradual increase in the ξ value, the AUC values of $MTTM$ are consistently higher than these of the eight comparison methods. Different from comparison methods, $MTTM$ tries to learn on transferable feature representations by capturing shared features and removing type-specific features, which enables its generalization to predict missing links on different types in a unified way.

VI. Conclusion and Future Work

In this study, we formally recognize the challenge of the type uncertainty of missing links, and propose a multi-type transferable method to address it. Our proposed method is constructed on a minimax two-player game between a proposed generative predictor and a proposed discriminative classifier. The generative predictor aims to predict whether the unobserved link is a missing link or not based on learned link representations. It attempts to capture the shared features among link types to deceive the discriminative classifier, while the discriminative classifier attempts to distinguish different link types to not be deceived. As a result, $MTTM$ effectively learns transferable feature representations to improve the prediction performance of missing links in heterogeneous social networks. Extensive experimental investigation shows that our proposed method outperforms state-of-the-art comparison methods in missing link prediction.

Based on our existing study, many additional methods can be explored to improve the prediction performance of missing links in heterogeneous social networks. One possibility is to exploit the attribute information of nodes and links, such as additional description of texts and images. Our proposed method is a general framework for missing link prediction. The extraction of shared features among link types can be easily designed for multi-modal situations. Another possibility is to explore the use of the type-specific information in our method. The simple removal of type-specific information may restrict the performance of our method in some cases. We can further discuss the balance between shared features and type-specific features.

References

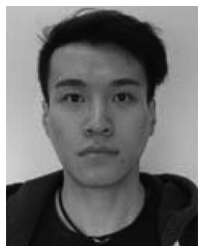
[1] A. Al-Baghdadi, G. Sharma, and L. Xiang, "Efficient processing of group planning queries over spatial-social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2135–2147, May 2022.
[2] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Non-linear dynamics of information diffusion in social networks," *ACM Trans. Web*, vol. 11, no. 2, pp. 1–40, 2017.
[3] H. Wang, C. Qiao, X. Guo, L. Fang, Y. Sha, and Z. Gong, "Identifying and evaluating anomalous structural change-based nodes in generalized dynamic social networks," *ACM Trans. Web*, vol. 15, no. 4, pp. 1–22, 2021.
[4] H. Zhao, X. Xu, Y. Song, D. L. Lee, Z. Chen, and H. Gao, "Ranking users in social networks with motif-based PageRank," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2179–2192, May 2021.
[5] H. Wang et al., "Existence identifications of unobserved paths in graph-based social networks," *World Wide Web J.*, vol. 24, no. 1, pp. 157–173, 2021.
[6] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, "Toward link predictability of complex networks," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 8, pp. 2325–2330, 2015.
[7] M. Coscia, "Noise corrected sampling of online social networks," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, pp. 1–21, 2021.
[8] J. Xu et al., "Robust network enhancement from flawed networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3507–3520, Jul. 2022.
[9] H. Wang, W. Hu, Z. Qiu, and B. Du, "Nodes' evolution diversity and link prediction in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2263–2274, Oct. 2017.
[10] L. Wang, J. Ren, B. Xu, J. Li, and F. Xia, "Model: Motif-based deep feature learning for link prediction," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 503–503–516, Apr. 2020.
[11] H. Wang and C. M. Qiao, "A nodes' evolution diversity inspired method to detect anomalies in dynamic social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 1868–1880, Oct. 2020.
[12] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
[13] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Jan. 2017.
[14] J. Jin et al., "An efficient neighborhood-based interaction model for recommendation on heterogeneous graph," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 75–84.
[15] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[16] J. Zhao et al., "IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2347–2357.
[17] T. Zhou, "Progresses and challenges in link prediction," *iScience*, vol. 24, no. 11, 2021, Art. no. 103217.
[18] H. Liu, Z. Hu, H. Haddadi, and H. Tian, "Hidden link prediction based on node centrality and weak ties," *Europhysics Lett.*, vol. 101, no. 1, 2013, Art. no. 18004.
[19] L. Getoor and B. Taskar, *Probabilistic Entity-Relationship Models, PRMs, and Plate Models*, Cambridge, MA, USA: MIT Press, 2011, pp. 55–60.
[20] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *Abstraction, Reformulation, and Approximation.* Berlin, Germany: Springer, 2000.
[21] M. Sales-Pardo and L. A. N. Amaral, "An extracting the hierarchical organization of complex systems," *Proc. Nation Acad. Sci. USA*, vol. 104, no. 39, 2007, Art. no. 15224.
[22] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, pp. 7821–7826, 2002.
[23] H. Tian and R. Zafarani., "Exploiting common neighbor graph for link prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 3333–3336.
[24] W. Yu, J. McCann, C. Zhang, and H. Ferhatosmanoglu, "Scaling high-quality pairwise link-based similarity retrieval on billion-edge graphs," *ACM Trans. Inf. Syst.*, vol. 40, 2022, Art. no. 78.
[25] S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2018.
[26] M. Sun, I. Konstantelos, and G. Strbac, "A deep learning-based feature extraction framework for system security assessment," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5007–5020, Sep. 2019.

[27] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 5165–5175, 2018.
[28] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1358–1368.
[29] B. Hu, Y. Fang, and C. Shi, "Adversarial learning on heterogeneous information networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 120–129.
[30] S. Negi and C. Santanu, "Link prediction in heterogeneous social networks," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 609–617.
[31] P. Wang, K. Agarwal, C. Ham, S. Choudhury, and C. K. Reddy, "Self-supervised learning of contextual embeddings for link prediction in heterogeneous networks," in *Proc. Web Conf.*, 2021, pp. 2946–2957.
[32] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1177–1186.
[33] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 793–803.
[34] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, 2020, pp. 2331–2341.
[35] Y. Dong, N. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 135–144.
[36] Y. Yang, Z. Guan, J. Li, W. Zhao, J. Cui, and Q. Wang, "Interpretable and efficient heterogeneous graph convolutional network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1637–1650, Feb. 2023.
[37] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, 2020, pp. 2704–2710.
[38] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.
[39] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 2014, *arXiv:1409.7495*.
[40] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," 2019, *arXiv:1909.13021*.
[41] C. Moreira, P. Calado, and B. Martins, "Learning to rank academic experts in the DBLP dataset," 2015, *arXiv:1501.05132*.
[42] G. Fu, B. Yuan, Q. Duan, and X. Yao, "Representation learning for heterogeneous information networks via embedding events," in *Proc. Int. Conf. Neural Inf. Process.*, 2019, pp. 327–339.
[43] W. S. Noble, "What is a support vector machine?," *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.
[44] Y. Dong et al., "Link prediction and recommendation across heterogeneous social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 181–190.
[45] R. Guns and R. Rousseau, "Recommending research collaborations using link prediction and random forest classifiers," *Scientometrics*, vol. 101, no. 2, pp. 1461–1473, 2014.
[46] J. A. Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
[47] W. Hu, H. Wang, Z. Qiu, C. Nie, L. Yan, and B. Du, "An event detection method for social networks based on hybrid link prediction and quantum swarm intelligent," *World Wide Web*, vol. 20, no. 4, pp. 775–795, 2017.
[48] S. Soundarajan and J. Hopcroft, "Using community information to improve the precision of link prediction methods," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 607–608.

Huan Wang is currently an Associate Professor with the College of Informatics, Huazhong Agricultural University, Wuhan, China. His current research interests include focuses on social network analysis and deep learning.

Ziwen Cui is currently working toward the master's degree with College of Informatics, Huazhong Agricultural University. His research interests include link prediction in heterogeneous information networks with applications to social network analysis.

Ruigang Liu is an algorithm engineer of China Mobile (Hangzhou) Information Technology Co., Ltd. His research interests include social network analysis, heterogeneous information networks and their applications in link prediction.



Lei Fang is a lecturer with the School of Computer Science, University of St Andrews since 2020. His research interests include centre around Bayesian inference, statistical learning, Bayesian non-parametrics, sensor-based human activity recognition, and uncertainty reasoning.

Ying Sha is a professor with College of Informatics, Huazhong Agricultural University. His research interests include natural language processing, machine learning, and artificial intelligence.