A Diffusion-based Triple Embedding Model for User Identity Linkage across Social Networks

Jingya Zhou(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
Key Laboratory of Data Intelligence and Advanced Computing in Provincial Universities, Soochow University, Suzhou, China
State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, China
jy_zhou@suda.edu.cn

Abstract. User identity linkage identifies anchor users with multiple accounts across different social networks. Current approaches primarily use embedding techniques to match users' content and structural features but struggle with challenges such as similarity dilemmas, network structure variations, and directional linkages. To address these issues, we propose a diffusion-based triple embedding framework (DTE) that leverages information dissemination behaviors during cross-network information diffusion. Each information diffusion is modeled as a triplet within a diffusion network, transforming the linkage problem into a cross-network triplet prediction task. We extract two types of triplet contexts and incorporate them into the learning process, supervised by a small set of anchor user pairs. Our designed triple translation in complex space ensures linkage by maintaining the equivalence between anchor accounts. Experimental results demonstrate that DTE outperforms state-of-the-art methods.

Keywords: User identity linkage · Anchor user · Diffusion network · Triple embedding

1 Introduction

User identity linkage (UIL) [22, 19] aims to identify these anchor users. Current studies can be roughly categorized into four groups: *1) Static content-based methods* [15] are the most intuitive way to identify anchor users by measuring similarities of their profile information, such as username and avatars. *2) Dynamic content-based methods* [2, 19] assume that the dynamic content generated by users contains unique features such as interests, spatiotemporal trajectories, and behavior patterns can be used for identification. *3) Social relationship-based methods* [12] insist that user's social relationships across different networks are constantly stable, e.g., the number of common friends. Recent works [3, 13, 20] have extended the neighborhood relationship to higher-order network structure (e.g., local and global networks). The recent advances in network embedding (NE) have spawned many efforts in designing network embedding-driven methods. *4) hybrid methods* [16, 6, 21] extract features from both content and network structural information to enhance performance. Current studies still have several challenges: **Similarity dilemma.** Network embedding enables structurally similar vertices having similar vectors. Meanwhile, each vertex should ideally have a unique vector to distinguish anchor users from others who have similar structural features. **Network structure variation.** Each user is represented by a vector to capture its structural feature. Once the network structure has been changed, no matter how small, all vector representations must be updated accordingly. Additionally, intentional alterations to network structure can inject noise into vector representations [14], degrading linkage precision. **Directionality of linkage.** UIL is typically formalized to find anchor links from a source network to a destination network, implying directionality in the identified anchor links. But there is no direction for anchor links. A potential solution is to conduct dual linkage by switching the source and destination networks. However, results from opposite directions may be contradictory.

Anchor users often engage in cross-network information diffusion by posting the same content on varied social networks. An analysis of 3 billion tweets shows that one-third originate from sources outside the Twitter network [8]. We collected a dataset containing 12 million posts and 15 million tweets from Facebook and Twitter, respectively. Content analysis revealed that 3.1 million Facebook posts have corresponding tweets with identical content or similar semantics, while 3.5 million tweets have corresponding posts on Facebook. In addition, 94.6% of anchor users participate in the cross-network diffusion. In this paper, we propose a novel framework DTE by leveraging information diffusion across social networks. Considering that anchor users facilitate across-network information diffusion, diffusion provides a crucial clue to overcoming current challenges in UIL. For a specific cross-network information diffusion, each involved user occupies a unique position in the diffusion network, reflecting its contribution to the diffusion. If an involved person has different social accounts u and v on two networks, accounts u and v would have adjacent positions in the diffusion network, i.e., an anchor link connects them, and their contributions are also connected via information flow on the anchor link. Therefore, features extracted from the information diffusion process are key to anchor user identification. Our main contributions are summarized as follows:

- Inspired by the translation-based embedding idea, we use a triplet (u, r, v) to represent the information transmission between users u and v over link r. A carefully designed translation operation significantly alleviates the contradictory dilemma in representation proximity.

- In DTE, each user have multiple vector representations, each corresponding to the user's feature correlated with a specific information diffusion. It is almost impossible for all vector representations of a user to be affected by network structure variation. Moreover, each diffusion network is in charge of diffusing specific information, so the information content is naturally integrated into DTE.

- We do not require to fix source and destination for UIL. All potential anchor users and links are embedded into cross-network triplets with symmetry preserved, ensuring equivalence between accounts at both ends of anchor link. Experimental results on real-world datasets demonstrate DTE's effectiveness.

2 Related Work

1) Static content-based methods often assume that anchor users' accounts exhibit similar profiles [15] and identify them based on similarity scores. However, they strongly assume that every user preserves complete attribute information. In addition, Lim et al. [7] indicated that only 7% of anchor users have similar profiles, while most describe themselves quite differently across networks. *2) Dynamic content-based methods* insist that anchor user identification can be strengthened by the dynamic content generated by users themselves. Nie et al. [9] proposed to leverage interests extracted from users' temporal and post information. Interest is a type of coarse-grained criterion suitable for classifying different user groups rather than distinguishing different users precisely. Zhang et al [19] proposed to capture every user's multi-scale behavior pattern by analyzing their behaviors. Similar work [2] have also been proposed to capture users' trajectory features. *3) Social relationship-based methods* prefer to detect anchor users by capturing users' social relationships using NE techniques. As direct relationships, neighborhood (e.g., follower/followee-ship and friendship) and ego-network are extracted to measure the matching degree of two accounts [12]. Derr et al. [4] proposed a deep adversarial network model to learn a mapping from one network's vector space to another, ensuring consistent data distribution across networks. Xiong et al. [18] investigated multiple structural views of multiplex network embedding and leveraged an attention mechanism to assess the agreement level of each anchor link. TransLink [22] encodes both users and their interactions into a unified vector space, allowing for distinguishing and converting user representations However, challenges such as network variation and link directionality remain inadequately addressed. *4) Hybrid methods* combine content and network information to achieve further improvements in UIL. LHNE [16] learns comprehensive representations of users by capturing the friend-based and interest-based user co-occurrence. Li et al. [6] proposed a type-aware anchor link prediction (TALP) based on multi-type information fusion from both local and global perspectives. Zhou et al. [21] proposed a novel embedding model to learn multiple embeddings for each user from both structural and content aspects. Most works typically embed users by learning features from independent latent spaces and concatenating them, which may not be optimal for identification.

Different from the above work, our framework uses information content to identify cross-network diffusion rather than directly identifying anchor users. Compared to existing hybrid methods, we avoid the issue of poor interpretability associated with vector concatenation. Additionally, unlike user embeddings in a single network, DTE performs triple embeddings that involve both single and cross-network diffusion contexts, effectively improving linkage performance and robustness against network variations. Moreover, DTE addresses the directionality issue by extending triplet entries from real space to complex space.

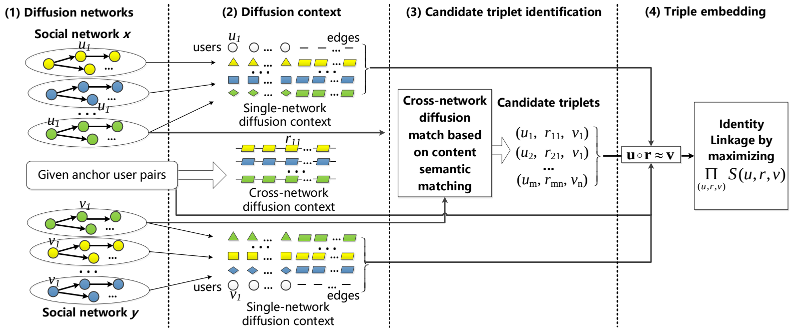

(1) Diffusion networks
(2) Diffusion context
(3) Candidate triplet identification
(4) Triple embedding
Social network *x*
u_1
users
edges
Single-network diffusion context
r_{11}
Given anchor user pairs
Cross-network diffusion context
Cross-network diffusion match based on content semantic matching
Candidate triplets
(u_1, r_{11}, v_1)
(u_2, r_{21}, v_1)
…
(u_m, r_{mn}, v_n)
$\mathbf{u} \circ \mathbf{r} \approx \mathbf{v}$
Identity Linkage by maximizing $\prod_{(u,r,v)} S(u,r,v)$
v_1
Social network *y*

Fig. 1. The workflow illustration of DTE framework. Different colors indicate varied information contents of multiple diffusion networks, and different shapes indicate multiple single network diffusion contexts of a user, while rhombi are used to indicate cross-network diffusion contexts.

3 DTE Framework

Our framework is proposed based on information diffusion in social networks, and its overview is illustrated in Fig. 1. We utilize a diffusion network to model the observed information diffusion, in which the concept of a triplet is borrowed to describe the fundamental process of information transfer from one user to another. Importantly, we correlate independent triplets by capturing both single-network diffusion context and cross-network diffusion context, and use these two types of context as the key issue for further cross-network triplet prediction. Notice that a user might be involved in multiple diffusion networks, hence he/she has different single-network diffusion contexts. Candidate cross-network triplets can be obtained by comparing the content similarity between diffusions on two networks. Finally, triple embeddings are performed with the objective of maximizing score functions for every cross-network triplet. Each of these triplets corresponds to a potential social user identity linkage.

3.1 Diffusion Network

We model a social network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the user set, and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ denotes the relation set. Each edge has a direction indicating the follower/followee relationship observed in many popular social networks. When a user posts, his followers and followers' followers may forward the post in a cascade manner. As a result, the information can be spread to many people. As illustrated in Fig. 2, the post sent by user u_1 is diffused to a group of users $u_2, ..., u_7$, depicted by yellow circles. These involved users, including the initiator itself, form a sub-network, which we define as a diffusion network:

Definition 1. Diffusion Network: *For a specific information diffusion initiated by user u at time t, its* ***diffusion network*** *is denoted by $\mathcal{D}_{u,t} = (\mathcal{V}_{u,t}, \mathcal{E}_{u,t})$, where $\mathcal{V}_{u,t}$ is the set of observed users involved in the diffusion, and $\mathcal{E}_{u,t} \subset \mathcal{V}_{u,t} \times \mathcal{V}_{u,t}$ is the set of edges observed to pass information.*

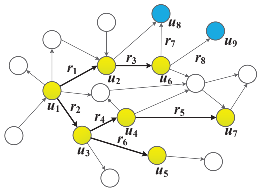


Fig. 2. A diffusion network from u_1.

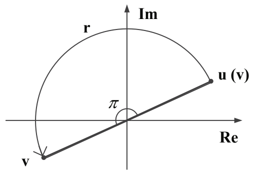


Fig. 3. An example of symmetric translations.

In a specific diffusion process, a user may receive the same information from multiple users, we use the user's forwarding behavior (e.g., retweet on Twitter) to identify the incoming edge. This ensures that each user receives information from only one incoming edge, and the diffusion network is consequently regarded as a directed acyclic graph (DAG). Each user in a DAG has a unique position that can be extracted as his feature by leveraging network embedding techniques such as GraphSAGE++ [5]. However, we do not adopt these popular techniques for two reasons. First, they only embed users into vectors, while as channels for information diffusion, edge embedding has been ignored. Second, there are potential sequences among users in a diffusion network, and these sequences are included in multiple information diffusion paths. Yet, the sequences among users have not been considered in current network embedding studies.

Translation-based embedding techniques [1, 11] embed both vertices and edges at the same time. In a typical translation-based framework, triplet (u, r, v) acts as the basic unit to represent the fact that resides in the three components. In our framework, triplet (u, r, v) is used to represent that user u passes the information to user v via edge r. Note that the vectors in this paper are denoted in bold. In the vector space, triplet $(\mathbf{u},\ \mathbf{r},\ \mathbf{v})$ is interpreted as vector $\mathbf{u}$ plus vector $\mathbf{r}$ approximates vector $\mathbf{v}$, i.e., $\mathbf{u} + \mathbf{r} \approx \mathbf{v}$. The addition operation in vector space satisfies asymmetry, i.e., $(\mathbf{u},\ \mathbf{r},\ \mathbf{v}) \Rightarrow \neg(\mathbf{v},\ \mathbf{r},\ \mathbf{u})$, which means triplet $(\mathbf{u},\ \mathbf{r},\ \mathbf{v})$ and triplet $(\mathbf{v},\ \mathbf{r},\ \mathbf{u})$ do not hold at the same time unless $\mathbf{r} = \mathbf{0}$. If translation $\mathbf{r}$ is a zero vector, the role of translation-based embedding becomes equivalent to network embedding. The asymmetry property is indeed compatible with the fact that information is transferred along one direction on the edge. Nevertheless, there is an exception when we take into account the situation of information diffusion across networks because the anchor link must ensure that information is reachable in both directions.

In terms of Euler's identity, i.e., $e^{i\varphi} = \cos\varphi + i\sin\varphi$, a complex number is equivalent to a rotation in the complex plane. If we put all vectors into the complex space, the translation between vectors can be implemented by the rotation in a complex vector space. Then we define the complex triplet as follows:

Definition 2. Complex Triplet: *Given three vectors* $\mathbf{u}, \mathbf{r}, \mathbf{v} \in \mathbb{C}^d$ representing the complex vectors of three users (u, r, v), *where d is the vector dimension, triplet* $(\mathbf{u},\ \mathbf{r},\ \mathbf{v})$ *is a complex one if it satisfies:* $\mathbf{u} \circ \mathbf{r} = \mathbf{v}$, *where $\circ$ is the Hadamard product of two vectors. For each dimension* k *of a vector in complex space, we have* $\mathrm{v_k} = \mathrm{u_k r_k},\ \mathrm{u_k}, \mathrm{r_k}, \mathrm{v_k} \in \mathbb{C}$ and $|\mathrm{r_k}| = 1$.

In a complex triplet $(\mathbf{u}, \mathbf{r}, \mathbf{v})$, translation vector $\mathbf{r}$ corresponds to an element-wise rotation from vector $\mathbf{u}$ to vector $\mathbf{v}$. Specifically, let vector element $\mathrm{r_k} = e^{i\varphi_\mathrm{k}}$, and element $\mathrm{v_k}$ is regarded as a counterclockwise rotation by φ_k radians from element $\mathrm{u_k}$ in the complex representation space. Thus, we have two properties.
Property 1. *In a complex triplet* $(\mathbf{u}, \mathbf{r}, \mathbf{v})$, $\mathbf{r}$ *contains symmetric relationship.*

Proof. If translation vector $\mathbf{r}$ is symmetric, we have two triplets $(\mathbf{u}, \mathbf{r}, \mathbf{v})$ and $(\mathbf{v}, \mathbf{r}, \mathbf{u})$ hold at the same time, i.e., $\mathbf{u} \circ \mathbf{r} = \mathbf{v} \wedge \mathbf{v} \circ \mathbf{r} = \mathbf{u}$. Combine them together, we have $\mathbf{r} \circ \mathbf{r} = 1$. It implies the square of each $\mathbf{r}$'s element is 1, i.e., $\mathrm{r_k^2} = 1$. As we know, $\mathrm{r_k} = e^0 = 1$ and $\mathrm{r_k} = e^{i\pi} = -1$. As a result, a symmetric translation corresponds to a rotation by 0 or 180 degrees. Fig. 3 shows an example of symmetric translations in a one-dimensional complex space.

Property 2. *In a complex triplet* $(\mathbf{u}, \mathbf{r}, \mathbf{v})$, $\mathbf{r}$ *contains asymmetric relationship.*

Proof. From the derivation of property 1, we conclude that the translation vector $\mathbf{r}$ is asymmetric as long as $\mathrm{r_k} \neq 1$ and $\mathrm{r_k} \neq -1$.

3.2 User Identity Linkage Problem

In our framework, UIL is equivalent to finding out cross-network triplets. Let us use a score function $S(u, r, v)$ to measure the probability that triplet (u, r, v) holds. The objective of the problem is to maximize the joint score of all potential cross-network triplets. Without loss of generality, we formalize the problem as:

Definition 3. User Identity Linkage: *Given two social networks* $\mathcal{G}_x$ *and* $\mathcal{G}_y$*, two sets* $\mathcal{D}_T^x = \{\mathcal{D}_{w1,t1}^x, |w1 \in \mathcal{V}_x, t1 \in [0, T]\}$ *and* $\mathcal{D}_T^y = \{\mathcal{D}_{w2,t2}^y, |w2 \in \mathcal{V}_y, t2 \in [0, T]\}$ *of observed diffusion networks on both social networks during a time window* T*, and a small fraction* Z *of anchor user pairs, the objective is*

$$\max \mathop{\mathit{\Pi}}_{(u,r,v)} S(u, r, v),\ u \in \mathcal{V}_x, v \in \mathcal{V}_y. \tag{1}$$

3.3 Diffusion Context

In the translation-based embedding framework, a social network is decomposed into a set of independent triplets. However, this process loses the information hidden in network structures as well as the connections between triplets. Motivated by users' transmission behaviors during information diffusion, we propose performing triple embedding based on the diffusion network. Specifically, different users in a diffusion network are located in varied positions, and the position reflects how much contribution a user makes to information diffusion. For example, user u_3 contributes more than user u_2 in Fig. 2, since user u_3 helps pass information on to more people. A user's position or contribution is reflected by other users and edges in the corresponding diffusion network. We regard those users and edges as the diffusion context and utilize them in triple embedding. According to the types of observed diffusion networks, we divide context into two categories: single-network diffusion context and cross-network diffusion context.

Single-network Diffusion Context. Given a diffusion network $\mathcal{D}^x_{w,t}$ inside a social network $\mathcal{G}_x$, if user $u \in \mathcal{D}^x_{w,t}$, its single network diffusion context $\theta_{w,t}(u)$ is defined as a set of tuples consisting of translation r and user v, i.e.,

$$\theta_{w,t}(u) = \begin{cases} \left\{(r, v)\,|v \in Suc(u)\right\}, \; if\, u\, is\, not\, a\, leaf\, in\, \mathcal{D}^x_{w,t}, \\ \left\{(r, v)\,|v \in F_T(u)\right\}, \; otherwise, \end{cases} \tag{2}$$

where $Suc(u)$ denotes all successors of u in $\mathcal{D}^x_{w,t}$, and (r, v) are coming from the following paths inside of $\mathcal{D}^x_{w,t}$ originated from user u. $F_T(u)$ denotes the set of u's outgoing neighbors who are observed to forward similar information from u in time window T. If u has no successor, to avoid null context, $\theta_{w,t}(u)$ is selected from u's outgoing neighbors and corresponding edges. For example, as shown in Fig. 2, the single network diffusion context of u_3 is $\{(r_4, u_4), (r_5, u_7), (r_6, u_5)\}$ and u_6's context is $\{(r_7, u_8), (r_8, u_9)\}$. For an arbitrary context $\theta_{w,t}(u)$, the pattern of (translation, tail) pair is designed to facilitate the prediction of user u's appearance as a head in triplets based on the context.

Cross-network Diffusion Context. The single network diffusion context alone is insufficient for supporting anchor user identification across different networks. Based on the given anchor users, we can observe information diffusion across networks and merge multiple diffusion networks into a larger one. We then collect contexts from the merged network and use them for further cross-network triplet prediction. Let $\mathcal{D}^{xy}_{w,t}$ denote the merged diffusion network and let r_a be the given anchor link, the cross-network diffusion context for a pair of users $u \in \mathcal{V}_x \cap \mathcal{V}^{xy}_{w,t}$ and $v \in \mathcal{V}_y \cap \mathcal{V}^{xy}_{w,t}$ is defined as follows:

$$\phi_{w,t}(u, v) = (r_{b1}, ..., r_a, ..., r_{bj}), \tag{3}$$

where $(r_{b1}, ..., r_a, ..., r_{bj})$ is a diffusion path observed from u to v. This path consists of a sequence of edges, including the anchor link r_a, and the path length is $j + 1$. In our framework, the cross-network diffusion context is designed to facilitate the prediction of the occurrence of the tail user v in a triplet, given a head user u in another network.

3.4 Candidate Triplets

Each cross-network triplet corresponds to a potential anchor user linkage. Before performing triple embedding to maximize the likelihood of cross-network triplets, we propose a method to identify candidate triplets for embedding. Assume that $\mathcal{D}^x_{w1,t1}$ and $\mathcal{D}^y_{w2,t2}$ are two diffusion networks residing in two different social networks, respectively, if the information diffused on two networks has high similarity, and the diffusion time interval between $\mathcal{D}^x_{w1,t1}$ and $\mathcal{D}^y_{w2,t2}$ is less than a reasonable value, e.g., $t2 - t1 \leq 1$ hour, we can confidently infer that the information is diffused from $\mathcal{D}^x_{w1,t1}$ to $\mathcal{D}^y_{w2,t2}$ via an anchor link. One end of the anchor link is user v, and the other end is someone hidden in $\mathcal{V}^x_{w1,t1}$. All possible combinations are considered as candidate triplets. Besides one-to-one diffusion network matching, there are probably more than two diffusion networks diffusing similar information. In such cases, these diffusion networks are organized into multiple pairwise tuples, and candidate triplets can be obtained in the same way.

The content similarity of information is measured by the semantic distance in vector space, such as SBERT [10]. Moreover, many existing works based on static content, dynamic content, and social relationships can also be integrated to narrow down the candidate anchor user pairs by eliminating impossible pairs.

3.5 Triple Embedding

Given two types of context $\theta_{w,t}(u)$ and $\phi_{w,t}(u,v)$, the score function for triplet (u,r,v) is defined as a conditional probability that the triplet holds, i.e.,

$$\begin{aligned} S(u,r,v) =& \Pr\left((u,r,v)\left|\theta_{w,t}(u),\phi_{w,t}(u,v)\right.\right) = \Pr\left(u\left|\theta_{w,t}(u),\phi_{w,t}(u,v)\right.\right) \\ & \Pr\left(v\left|\theta_{w,t}(u),\phi_{w,t}(u,v),u\right.\right)\Pr\left(r\left|\theta_{w,t}(u),\phi_{w,t}(u,v),u,v\right.\right), \end{aligned} \tag{4}$$

where the evaluation of triplet (u,r,v) is further decomposed into three components. The first component $\Pr\left(u\left|\theta_{w,t}(u),\phi_{w,t}(u,v)\right.\right)$ measures the conditional probability that u is the head of a triplet given the triplet context. As discussed in Sec. 3.3, u's appearance as a head is mainly decided by its single network context. The component can be approximated by $\Pr\left(u\left|\theta_{w,t}(u)\right.\right)$, which is regarded as the compatibility between u and its single network diffusion context. Following the computation method validated in [17], we formalize it as:

$$\Pr\left(u\left|\theta_{w,t}(u)\right.\right) = \frac{\exp(\delta(u,\theta_{w,t}(u)))}{\sum_{u'\in\mathcal{V}_x}\exp(\delta(u',\theta_{w,t}(u)))}, \tag{5}$$

where function $\delta(\cdot)$ is used to represent the correlation between an arbitrary user u' and u's single network diffusion context, and its definition is given by

$$\delta(u',\theta_{w,t}(u)) = -\frac{1}{|\theta_{w,t}(u)|}\sum\nolimits_{(r,v)\in\theta_{w,t}(u)}\|\mathbf{u'}\circ\mathbf{r}-\mathbf{v}\|, \tag{6}$$

where $|\theta_{w,t}(u)|$ is the size of context, and $\|\cdot\|$ is the L1-norm for complex vector distance. The second component $\Pr\left(v\left|\theta_{w,t}(u),\phi_{w,t}(u,v),u\right.\right)$ measures the conditional probability that v is the tail of triplet given the head u and triplet context. For a cross-network triplet, v's appearance as a tail is mainly decided by its cross-network context. Hence, this component can be approximated by $\Pr\left(v\left|\phi_{w,t}(u,v),u\right.\right)$, and it is computed by

$$\Pr\left(v\left|\phi_{w,t}(u,v),u\right.\right) = \frac{\exp(\eta(v,\phi_{w,t}(u,v)))}{\sum_{v'\in\mathcal{V}_y}\exp(\eta(v',\phi_{w,t}(u)))}, \tag{7}$$

where function $\eta(\cdot)$ reflects the correlation between an arbitrary v' and tuple (u,v)'s cross-network context, and is defined as: $\eta(v',\phi_{w,t}(u,v)) = -\|\mathbf{u}\circ\mathbf{p}-\mathbf{v'}\|$, where $\mathbf{p}$ refers to a path vector composed of multiple translation vectors from u to v. If $p=(r_{b1},...,r_a,...,r_{bj})$, its embedded vector is calculated by $\mathbf{p} = \frac{1}{j+1}\left(\mathbf{r}_{b1}+...+\mathbf{r}_{bj}\right)$.

Algorithm 1: *DTE_ training*

```
Input: training set $\Omega$, the set $Z$ of anchor user triplets, dimension $d$,
       maximum iterations $I_{max}$;
Output: the set of triple embeddings;
if two diffusion networks in $\Omega$ contain anchor users belong to $Z$ then
    Merge two diffusion networks as one;
Initialize u, v and r uniformly for each triplet $(u, r, v) \in \Omega$;
repeat
    for each batch $b_i \subset \Omega$ do
        for each triplet $(u, r, v) \in b_i$ do
            Sample a set $(u', r, v')$ of corrupted triplets;
            Construct two types of diffusion context based on Eqs. (2)(3);
        Update triple embeddings w.r.t. $\sum_{(u,r,v) \in \Omega} \nabla \log S(u, r, v)$;
until convergence or $I_{max}$;
return triple embeddings;
```

The third component $\Pr(r \mid \theta_{w,t}(u), \phi_{w,t}(u, v), u, v)$ measures the conditional probability that translation r holds given the triplet context, head u and tail v. Considering that triplet context has been incorporated in the previous two components, where head and tail have also been determined, respectively, the component is simplified to $\Pr(r \mid u, v)$, i.e.,

$$\Pr(r \mid u, v) = \frac{\exp(\gamma(u, r, v))}{\sum_{r' \in \mathcal{E}_{xy}} \exp(\gamma(u, r', v))}, \tag{8}$$

where $\mathcal{E}_{xy}$ is the given anchor link set, function $\gamma(\cdot)$ is used to represent the connection between translation r' and tuple (u, v), and is defined as follows:

$$\gamma(u, r', v) = -\left\| \mathbf{u} \circ \mathbf{r}' - \mathbf{v} \right\|. \tag{9}$$

As a result, the score function defined in Eq. (4) is approximated by

$$S(u, r, v) = \Pr(u \mid \theta_{w,t}(u)) \Pr(v \mid \phi_{w,t}(u, v), u) \Pr(r \mid u, v). \tag{10}$$

In order to avoid the high computational overhead involved in the score function, we approximate the softmax function based on negative sampling. Besides, we convert the score function into the negative logarithm form, and then the objective is re-formalized as follows:

$$\sum_{(u,r,v)} \log S(u, r, v) = \sum_{u \in \mathcal{V}_x \cup \mathcal{V}_y} (A(u) + B(v) + \Lambda(r)), \tag{11}$$

where $\mathrm{A}(u) = \log \sigma(\delta(u, \theta_{w,t}(u))) + \sum_{u' \neq u}^{k} \log \sigma(-\delta(u', \theta_{w,t}(u)))$; $B(v) = \log \sigma(\eta(v, \phi_{w,t}(u, v))) + \sum_{v' \neq v}^{k} \log \sigma(-\eta(v', \phi_{w,t}(u, v)))$; $\Lambda(r) = \log \sigma(\gamma(u, r, v)) +$

Algorithm 2: *DTE_ linking*

```
Input: two sets $\mathcal{D}_T^x$, $\mathcal{D}_T^y$ of diffusion networks, the set $Z$ of anchor user
       triplets, score threshold $\chi$;
Output: the set $\mathcal{A}$ of identified anchor user triplets;
for each $\mathcal{D}_{w1,t1}^x \in \mathcal{D}_T^x$ do
    for each $\mathcal{D}_{w2,t2}^y \in \mathcal{D}_T^y$ do
        if $|t2 - t1| \leq t^*$ and the information content diffused in $\mathcal{D}_T^x$, $\mathcal{D}_T^y$ are
          similar then
            Construct the set $\Omega$ of candidate triplets;
            Learn triple embeddings by Algorithm 1;
            Find triplet $(u, r, v)$ such that $S(u, r, v) \geq \chi$;
            Append triplet $(u, r, v)$ to $\mathcal{A}$;
return $\mathcal{A}$;
```

$\sum\limits_{r' \neq r}^{k} \log \sigma(-\gamma(u, r', v))$, $\sigma(\cdot)$ is a logistic function, and k is the number of negative samples. From the above equation, we may notice that the single network diffusion context of v has not been used for triple embedding. However, as we have stated, there is no fixed direction during identity linkage, and v may be also the head of a potential triplet. Thereby, v's single network diffusion context might be incorporated already.

3.6 Algorithms

The training process of DTE is illustrated by Algorithm 1, which learns embeddings of users and translations inside triplets. The training set Ω is composed of multiple triplets inside varied diffusion networks. Given a set Z of anchor user triplets, some diffusion networks may be merged as a cross-network diffusion network, and the anchor user triplets are also included in Ω. The embeddings take value in $\mathbb{C}^d$. Both the real and imaginary parts of the triple embeddings are uniformly initialized, and the phases of the translation embeddings are uniformly initialized between 0 and 2π. At each iteration, a small set of triplets is sampled from Ω to serve as the training triplets of the minibatch. For each triple, in order to generate positive and negative triplets, we have to corrupt the head or tail of the triple. For each minibatch in the generated training set of triplets, both single network diffusion context and cross-network diffusion context are constructed. Afterward, we calculate the loss which favors a higher probability for positive triplets than negative triplets, and update triple embeddings iteratively until convergence or the preset iterations I_{max}. Social user identity linkage is illustrated by Algorithm 2. It first identifies all possible pairs of diffusion networks that diffuse similar content within a short time interval, i.e., $|t2 - t1| \leq t^*$. Then it constructs the set Ω of candidate triplets across $\mathcal{G}_x$ and $\mathcal{G}_y$, and learns triple embeddings by the training algorithm. Based on the embeddings, it selects the triplet with a score higher than χ and puts it in the anchor triplet set $\mathcal{A}$.

Complexity. Merging diffusion networks requires traversing set Ω, which takes $O(|\Omega|)$ time. Negative sampling for each triplet takes $O(k)$ time. Let D_{size} denote the average size of a diffusion network. The context construction takes $O(D_{size})$ time. The triple embedding update takes $O(kd)$ time. Therefore, the time complexity of Algorithm 1 is $O(I_{max}|\Omega|k(D_{size}+d))$. In Algorithm 2, similarity determination takes $O(|D_T^x||D_T^y|)$ time. Assume that the fraction of similar diffusion networks is α. Therefore, the overall time complexity of Algorithm 2 is $O\left(\frac{|D_T^x|+|D_T^y|}{2}\alpha I_{max}|\Omega|k(D_{size}+d)+|D_T^x||D_T^y|\right)$.

3.7 Discussions

Selection of Information Flows. When we are building diffusion networks, there are many factors influencing the selection of information flows, such as information carrier, authenticity of information, etc. Here we mainly focus on the information popularity based on the following considerations: 1) Hot information flows are very popular among users, which implies many non-anchor users will post similar information on different social networks and thereby form multiple independent diffusion networks on different social networks. Then more non-anchor candidate triplets need to be filtered out, which is bound to make the problem more difficult. 2) Compared to non-hot information flow, hot one will attract more users included in the diffusion network (i.e., larger D_{size}), and the fraction of similar diffusion networks α and the set size of training triplets $|\Omega|$ will also increase, which increases the complexity of our solution. Therefore, DTE favors non-hot information flows for better identity linkage.

Progressive Identity Linkage. There is a contradictory situation: on the one hand, if the given set Z of anchor users is large enough, more cross-network context could be obtained and DTE is capable of finding out more latent anchor users; on the other hand, it is practically impossible to acquire a large Z due to the extreme hardness of identity linkage. If the newly identified anchor users could be appended into the set Z for the next identification, the tricky situation could be alleviated. However, this progressive identity linkage suffers from a fatal problem – error explosion, once non-anchor users are wrongly appended, errors will accumulate and cause more errors. To this end, we incorporate a self-correcting mechanism into the progressive linkage. Specifically, we define a margin-based loss function for each triplet, i.e.,

$$L(u,r,v)=\sum_{(u',r',v')\in Neg}\max\{0,S(u,r,v)-S(u',r',v')\},$$

where Neg denotes the negative sample set. Besides, we define a credibility score for each triplet, i.e., $Cr(u,r,v)=\pi\left(\kappa\cdot(S(u,r,v)-\chi)\right)$, where $\pi(\cdot)$ is a sigmoid function and κ is a hyper-parameter. Let $\mathcal{Z}$ denote the set of found anchor triplets. Every time before appending $\mathcal{Z}$ to Z, we calculate the credible loss of current $\mathcal{Z}$, i.e., $Cr=\sum_{(u,r,z)\in\mathcal{Z}}Cr(u,r,v)L(u,r,v)$. If $Cr>\vartheta$, the accumulated errors increase rapidly. Then $\mathcal{Z}$ should not be appended and the identity linking must rollback to the previous iteration and update $\mathcal{Z}$ until $Cr\leq\vartheta$.

Table 1. Statistics of Social Network Datasets.

TF	#user	#edge	#post	#anchor	**TM**	#user	#edge	#post	#anchor
Twitter	517,386	16,693,605	15,640,672	72,024	**Tumblr**	112,736	5,232,078	1,354,074	11,608
Facebook	531,764	11,962,350	12,574,252		**Myspace**	105,884	3,028,476	1,086,718	

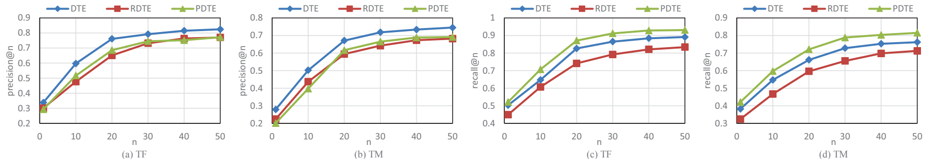
4 Experiments

4.1 Experimental Settings

Datasets. To the best of our knowledge, no qualified public dataset is currently available. Most existing public datasets only provide network structures with ground-truth on anchor users, without including any post content. Fortunately, the website About.me allows people to showcase their profiles along with prominent links to their various social accounts. Following the methodology outlined in [21], we collected data from About.me and built two groups of datasets, referred to as **TF** and **TM**. As Table 1 shows, users in the two networks of **TF** demonstrate greater activity and exhibit more frequent posting behaviors compared to those in **TM**. Thus, these datasets can be viewed as representing distinct benchmarks for evaluation: rich-information networks and poor-information networks.

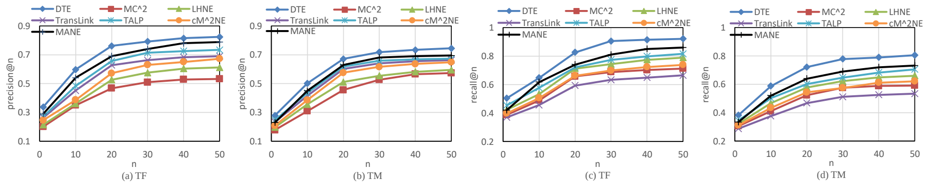
Experiment Setup. In our experiments, diffusion networks are obtained by tracing post-forwarding paths. Not every post corresponds to a diffusion network, since a considerable fraction of posts are rarely forwarded (e.g., retweet on Twitter, reblog on Tumblr, share on Facebook and Myspace). Instead, posts that are forwarded more than 10 times or are two hops away from the original initiator are selected to build diffusion networks. Here, we mainly focus on the text content similarity measured by the sentence semantic distance The hyperparameters for embeddings are set by default as follows, $d = 128$, $t^* = 3\,hours$ for **TF** and $8\,hours$ for **TM**. The learning rate is set to 0.001, following the optimal settings in [1]. The fraction of Z to the ground truth is set to 0.3.

We evaluate DTE by comparing it to several state-of-the-art methods described as follows: 1) *RDTE*. It is a variant of DTE that performs triple embeddings in a real space. Its performance is reported by averaging the linkage results from both directions. 2) *PDTE*. It is another variant of DTE that implements progressive linkage. 3) MC^2 [12]. It identifies anchor users by using matrix factorization to infer the common subspace from the network matrices. 4) *LHNE* [16]. It infers anchor users by embedding them into vector space to capture both friend-based structure and topic of interest. 5) *TransLink* [22]. It jointly embeds users and their interactions into a unified representation space based on translating embeddings. 6) *TALP* [6]. It makes linkage predictions based on the pairwise-similarity between type-fusion vectors of users. 7) cM^2NE [18]. It selects the fitted views for different layers and maximizes the inter-layer mutual information by emphasizing the most informative anchor links. 8) *MANE* [21]. It learns multiple representations for each user and then trains an unbiased mapping function to identify anchor accounts. Here, we use two widely adopted metrics: precision@n and recall@n.



(a) TF (b) TM (c) TF (d) TM

Fig. 4. Performance comparison among DTE and its variants.



(a) TF (b) TM (c) TF (d) TM

Fig. 5. Performance comparison among DTE and state-of-the-art methods.

4.2 Results

Comparison with variants. We first compare DTE with its two variants. Fig. 4 shows the precision@n results for different values of n. Compared to **TM**, **TF** provides more diffusion networks for triple embeddings (more than twice as many) and achieves 12.5% and 13.6% higher precision and recall, respectively, than **TM**. This indicates that rich-information significantly enhances the performance of our framework, meanwhile, our framework still works well even on the poor-information dataset. Besides, we notice that DTE achieves the highest precision@n on both datasets, while its recall@n is slightly lower than that of PDTE. When PDTE appends newly identified anchor triplets to a given anchor triplet set, the appended false positive triplets may mislead PDTE, resulting in incorrect anchor triple embeddings and reduced overall linkage accuracy. However, the appended true positive triplets simultaneously improve the recall rate. Triple embeddings in RDTE only preserve the asymmetry property, but users within an arbitrary anchor triplet are mutually equivalent, resulting in RDTE's performance being lower than that of DTE.

Comparison with SOTA methods. In this group of experiments, we compare DTE with state-of-the-art methods. Fig. 5 (a)(b) show precision@n, where DTE consistently outperforms the others. Compared to MC^2 and LHNE, DTE achieves 61% and 46% higher precision on **TF** (44% and 32% higher on **TM**), respectively. The structural features used in MC^2 and LHNE are extracted from the neighborhood relationship, which contains substantial noise due to the heterogeneity of neighborhood relationships across different social networks. cM^{2}NE takes advantage of varied connection types by learning on both intra-layer and inter-layer, alleviating the impact of noise to some extent, but its precision is significantly lower than DTE's (33% and 21% lower on both datasets). Even though content features are incorporated in user representation, they do not bring significant accuracy gains due to the loose coupling between content and

Table 2. Comparison results under varying degrees of network structure variations.

		presion@30						recall@30					
	p_v	0.01%	0.1%	1%	10%	20%	30%	0.01%	0.1%	1%	10%	20%	30%
TF	DTE	**0.792**	**0.784**	**0.763**	**0.712**	**0.633**	**0.579**	**0.901**	**0.894**	**0.876**	**0.815**	**0.747**	**0.711**
	MC^2	0.499	0.455	0.367	0.273	0.181	0.121	0.681	0.659	0.591	0.487	0.355	0.289
	LHNE	0.561	0.530	0.437	0.338	0.221	0.182	0.738	0.703	0.642	0.543	0.429	0.376
	TransLink	0.653	0.606	0.521	0.362	0.279	0.220	0.626	0.607	0.519	0.356	0.228	0.157
	TALP	0.708	0.682	0.637	0.435	0.328	0.263	0.770	0.728	0.687	0.559	0.441	0.391
	cM^{2}NE	0.623	0.591	0.512	0.343	0.232	0.201	0.692	0.676	0.610	0.502	0.385	0.317
	MANE	0.731	0.695	0.643	0.458	0.363	0.302	0.801	0.770	0.706	0.586	0.474	0.415
TM	DTE	**0.717**	**0.681**	**0.659**	**0.612**	**0.543**	**0.496**	**0.772**	**0.765**	**0.739**	**0.671**	**0.584**	**0.518**
	MC^2	0.518	0.489	0.378	0.255	0.166	0.105	0.567	0.530	0.461	0.369	0.273	0.208
	LHNE	0.532	0.509	0.403	0.286	0.201	0.142	0.612	0.587	0.520	0.433	0.322	0.281
	TransLink	0.634	0.608	0.527	0.316	0.233	0.151	0.503	0.476	0.394	0.230	0.137	0.101
	TALP	0.653	0.617	0.542	0.335	0.250	0.172	0.643	0.602	0.551	0.467	0.360	0.315
	cM^{2}NE	0.614	0.585	0.496	0.282	0.194	0.138	0.572	0.543	0.484	0.372	0.291	0.226
	MANE	0.672	0.640	0.566	0.431	0.334	0.252	0.692	0.676	0.610	0.502	0.385	0.317

structure, e.g., TALP is 17% and 11% lower than DTE, MANE is 9% and 10% lower than DTE. TransLink improves linkage accuracy by embedding both users and their interactions into a latent space, but still yields 15%∼20% lower accuracy than DTE. DTE is capable of filtering out a large proportion of user pairs that are unlikely to match. Moreover, DTE utilizes complex triple embeddings to alleviate the similarity dilemma.

Fig. 5 (c)(d) report recall@n, revealing that DTE achieves 14%∼40% higher recall rate on **TF** (16%∼49% higher on **TM**) than other methods. In DTE, candidate anchor triplets are built by identifying diffusion networks with similar content information. Many candidate anchor triplets may be involved in multiple diffusion networks, and are matched iteratively throughout the linkage process, potentially increasing the number of returned true positive triplets. We also observe that other methods perform better than TransLink in terms of recall rate. These methods attempt to return as many potential anchor users as possible by leveraging various kinds of content (where cM^{2}NE mainly relies on contrastive learning on the hypersphere), leading to the growth of both true positive and false positive anchor users. For MC^2, LHNE, cM^{2}NE, and MANE, the recall rate is improved by sacrificing link accuracy to some extent.

Comparison under network structure variations. To assess the robustness of methods against network dynamics, we simulate network structure variations by randomly adding/removing a percentage p_v of edges, with the ratio between added and removed edges ranging from 0.8 to 1.2. Table 2 shows the comparison results on two performance metrics as p_v varies from 0.01% to 30%. DTE's precision@30 decreases by an average of 28%, and its recall@30 decreases by an average of 26%, whereas other methods experience a degradation exceeding 80% and 63%, respectively, on both metrics. As p_v grows, the original network structure becomes more severely compromised. The neighborhood relationship being the fundamental structural unit of an arbitrary social network, is more susceptible to variations in network structure. Although features extracted from user-generated content assist LHNE and MANE in mitigating the impact of structural changes, their effectiveness is notably limited in this context. In contrast, DTE relies on the diffusion network to explore anchor user identifi-

Table 3. Comparison results (recall@30) under intentional changes.

Dataset	**TF**					**TM**				
$p_a\,(\%)$	(1, 10]	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(1, 10]	(10, 20]	(20, 30]	(30, 40]	(40, 50]
DTE	**0.869**	**0.833**	**0.820**	**0.768**	**0.711**	**0.758**	**0.722**	**0.699**	**0.656**	**0.581**
MC^2	0.621	0.569	0.477	0.403	0.275	0.547	0.511	0.435	0.338	0.252
LHNE	0.742	0.670	0.611	0.483	0.380	0.603	0.576	0.522	0.460	0.367
TransLink	0.631	0.586	0.538	0.463	0.401	0.518	0.490	0.462	0.407	0.341
TALP	0.753	0.708	0.625	0.528	0.422	0.620	0.593	0.525	0.467	0.403
cM^{2}NE	0.672	0.600	0.518	0.414	0.301	0.568	0.530	0.508	0.403	0.278
MANE	0.770	0.737	0.671	0.594	0.493	0.661	0.628	0.575	0.506	0.454

cation. Almost every anchor user in two datasets is involved in more than one diffusion network. When the fraction of broken edges is not high (e.g., less than 10% of the original edge set in our experiments, the actual rate of edge change is usually below 6%), the probability that an anchor user's all diffusion networks are completely broken is extremely low. Consequently, the negative impact of network variations is potentially alleviated.

To assess the resilience to intentional modifications of the network structure, we select half of the anchor users deliberately altering their structural properties. Specifically, for each pair of selected anchor user accounts u_i and v_j, we increase u_i's PageRank value by p_a percent by adding/removing edges within two hops away from u_i, and reduce v_j's PageRank value by p_a percent in the same way. As shown in Table 3, all methods exhibit varying degrees of decline as p_a increases, where DTE shows the smallest decrease, e.g., 18%∼24% reduction when $40\% < p_a \leq 50\%$. As p_a increases, an increasing number of edges associated with anchor users have been changed, and the connections between anchor users and non-anchor users change accordingly. In our framework, as long as enough information is observed to diffuse over the unchanged edges, the anchor user's diffusion context with little noise can still be effectively extracted and utilized for anchor triplet identification.

5 Conclusion

In this paper, we investigated the UIL problem by leveraging diffusion-based triple embeddings and proposed a novel framework DTE. We modeled information diffusion as a diffusion network, where the triplet serves as the fundamental unit to depict the information flow between users. To identify cross-network triplets (i.e., anchor triplets), we introduced two types of triplet contexts and generalized the triplet element vectors into the complex space. Moreover, the translation operation designed in the complex space ensures the directionality of information diffusion and the equivalence of anchor links. Extensive experiments demonstrated that DTE outperforms state-of-the-art methods.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 61972272), Qinlan Project of Jiangsu Province of China, and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS (2013)
2. Chen, W., Wang, W., Yin, H., Zhao, L., Zhou, X.: HFUL: a hybrid framework for user account linkage across location-aware social networks. VLDB J. **32**(1) (2023)
3. Chen, X., Heimann, M., Vahedian, F., Koutra, D.: Cone-align: Consistent network alignment with proximity-preserving node embedding. In: CIKM (2020)
4. Derr, T., Karimi, H., Liu, X., Xu, J., Tang, J.: Deep adversarial network alignment. In: CIKM (2021)
5. Jiawei, E., Zhang, Y., Yang, S., Wang, H., Xia, X., Xu, X.: Graphsage++: Weighted multi-scale GNN for graph representation learning. NPL **56**(1), 24 (2024)
6. Li, X., Shang, Y., Cao, Y., Li, Y., Tan, J., Liu, Y.: Type-aware anchor link prediction across heterogeneous networks based on graph attention network. In: AAAI (2020)
7. Lim, B.H., Lu, D., Chen, T., Kan, M.: #mytweet via instagram: Exploring user behaviour across multiple social networks. In: ASONAM. pp. 113–120 (2015)
8. Myers, S.A., Zhu, C., Leskovec, J.: Information diffusion and external influence in networks. In: KDD. pp. 33–41 (2012)
9. Nie, Y., Jia, Y., Li, S., Zhu, X., Li, A., Zhou, B.: Identifying users across social networks based on dynamic core interests. Neurocomputing **210**, 107–115 (2016)
10. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP. pp. 3980–3990 (2019)
11. Song, D., Zhang, F., Lu, M., Yang, S., Huang, H.: Dtranse: Distributed translating embedding for knowledge graph. TPDS **32**(10), 2509–2523 (2021)
12. Sun, L., Zhang, Z., Li, G., Ji, P., Su, S., Yu, P.S.: Mc^2: Unsupervised multiple social network alignment. TIST **14**(4), 70:1–70:22 (2023)
13. Sun, L., Zhang, Z., Wang, F., Ji, P., Wen, J., Su, S., Yu, P.S.: Aligning dynamic social networks: An optimization over dynamic graph autoencoder. TKDE **35**(6), 5597–5611 (2023)
14. Sun, L., Dou, Y., Yang, C., Zhang, K., Wang, J., Yu, P.S., He, L., Li, B.: Adversarial attack and defense on graph data: A survey. TKDE **35**(8), 7693–7711 (2023)
15. Wang, M., Wang, W., Chen, W., Zhao, L.: EEUPL: towards effective and efficient user profile linkage across multiple social platforms. WWWJ **24**(5), 1731–1748 (2021)
16. Wang, Y., Feng, C., Chen, L., Yin, H., Guo, C., Chu, Y.: User identity linkage across social networks via linked heterogeneous network embedding. WWWJ **22**(6), 2611–2632 (2019)
17. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: EMNLP (2014)
18. Xiong, H., Yan, J., Pan, L.: Contrastive multi-view multiplex network embedding with applications to robust network alignment. In: KDD (2021)
19. Zhang, Z., Ren, F., Zhang, J., Su, S., Yan, Y., Wei, Q., Sun, L., Zhu, G., Guo, C.: When behavior analysis meets social network alignment. TKDE **35**(7) (2023)
20. Zhou, F., Cao, C., Trajcevski, G., Zhang, K., Zhong, T., Geng, J.: Fast network alignment via graph meta-learning. In: INFOCOM. pp. 686–695 (2020)
21. Zhou, J.: Mane: A multi-cascade adversarial network embedding model for anchor link prediction. In: DASFAA. pp. 168–184 (2024)
22. Zhou, J., Fan, J.: Translink: User identity linkage across heterogeneous social networks via translating embeddings. In: INFOCOM (2019)