

Social Link Inference via Multi-View Matching Network from Spatio-Temporal Trajectories

Wei Zhang , Member, IEEE, Xin Lai, and Jianyong Wang, Fellow, IEEE

Abstract—In this paper, we investigate the problem of social link inference in a target Location-aware Social Network (LSN), which aims at predicting the unobserved links between users within the network. This problem is critical for downstream applications including network completion and friend recommendation. In addition to the network structures commonly used in general link prediction, the studies tailored for social link inference in an LSN leverage user trajectories from the spatial aspect. However, the temporal factor lying in user trajectories is largely overlooked by most of the prior studies, limiting the capabilities of capturing the temporal relevance between users. Moreover, effective user matching by fusing different views, i.e., social, spatial, and temporal factors, remains unresolved, which hinders the potential improvement of link inference. To this end, this paper devises a novel multi-view matching network (MVMN) by regarding each of the three factors as one view of any target user pair. MVMN enjoys the flexibility and completeness of modeling each factor by developing its suitable matching module: i) location matching module, ii) time-series matching module, and iii) relation matching module. Each module learns a view-specific representation for matching, and MVMN fuses them for final link inference. Extensive experiments on two real-world datasets demonstrate the superiority of our approach against several competitive baselines for link prediction and sequence matching, validating the contribution of its key components.

Index Terms—social link inference, spatio-temporal trajectory, multi-view matching, neural point process

I. INTRODUCTION

AS the core part of fast-developing social medias, user linked data has been popularized unprecedentedly around the online world. This makes link prediction [1], aiming to predict the unobserved links between nodes (e.g., users) given some observed links in target networks, attractive to both academic and industrial domains. Link prediction benefits a variety of downstream applications, such as network

Manuscript received xx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. This work was supported in part by the National Key Research and Development Program of China (No. 2019YFB2102600), in part by Shanghai Sailing Program (No. 17YF1404500), in part by the National Natural Science Foundation of China (No. 61702190, No. U1609220, No. 61532010, and 61521002), in part by the foundation of Key Laboratory of Artificial Intelligence, Ministry of Education, P.R. China, and in part by Beijing Academy of Artificial Intelligence (BAAI). (Corresponding author: Wei Zhang.)

W. Zhang is with the School of Computer Science and Technology, East China Normal University, Shanghai 200062, China, and also with the MOE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhangwei.thu2011@gmail.com).

X. Lai is with the School of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: 51184501121@stu.ecnu.edu.cn).

J. Wang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100086, China (e-mail: jianyong@tsinghua.edu.cn).

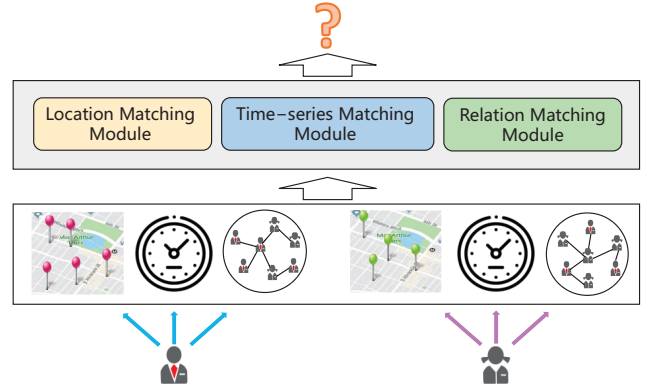


Fig. 1: The sketch of MVMN for social link inference. Three aspects of LSNs, i.e., spatial, temporal, and social factors, are reflected by three subfigures connected with a user.

completion [2], friend recommendation [3], event recommendation [4], to name a few. Most existing studies dive into this problem from the perspective of learning from social network structures, for example, heterogeneous networks [5] and coupled networks [6]. In particular, recent advances in representation learning on graphs [7], [8] significantly promote its development.

In location-aware social networks (LSNs), despite the common social links, users have rich spatio-temporal trajectories recorded by widely prevalent GPS-enabled mobile devices. A typical spatial trajectory involves multiple sequential check-ins from a user, where each check-in is composed of a location and a timestamp. The temporal and spatial properties of check-ins make the mobile trajectories substantially different from other types of sequential data. Since the spatio-temporal trajectories contain informative user mobility patterns, many studies learn to discover knowledge from them for different real applications, including location prediction [9] and recommendation [10], check-in time prediction [11], and trajectory-user linking [12]. To sum up, all these studies are attributed to performing prediction for a single user each time.

In this paper, we investigate the problem of social link inference in a target LSN [13] which estimates whether a given user pair has a social link based on their trajectories. This makes it fundamentally different from the above studies towards LSNs. In comparison with general link prediction, user mobility from spatial trajectories is utilized in this problem. Considerable efforts [13], [14], [15] on this problem adopt feature based classification methods. They require manually crafted mobility features from user trajectory pairs to train user

relation classifiers. The recent approaches [16], [17] benefit from the powerful graph representation learning to encode both users and locations into low-dimensional vectors. They could be further utilized for deriving social relations. However, most of them largely overlook the temporal aspect of user mobility. This might cause information loss. It is intuitive that if two users visit the same physical location around the same time, there is a larger chance that they might have some interactions and establish a connection (see Table I). Moreover, as shown later in Fig. 2, the users with social links have large temporal similarities than unlinked users. As a result, the ignorance of temporal information would limit the accuracy of correlation modeling of any two user trajectories. Although the study [16] takes a step in this direction by discretizing continuous timestamps into time intervals and further treat them as nodes in graphs, the temporal information is not directly leveraged for linking users. Moreover, the continuity and sequentiality of time are not well preserved by the intervals.

It is worth noting that the problem of user identity linkage across different LSNs [18], [19], [20], [21], focusing on linking online IDs (existing in different social networks) of the same user, shares some similar spirits with the studied problem because of the consideration of user mobility traces. Nevertheless, an essential difference is that there are more than one social network involved in user identity linkage. This makes the manners of using network structures for the two problems fundamentally different. For example, network alignment [22], [23], [24] techniques proposed for node linkage across networks are not applicable for the scenarios with only one target social network. As such, although the above methods could be partially adapted to our problem, the ways of utilizing social relations should be carefully reconsidered.

In summary, all the previous methods lack the mechanisms of performing user linking in a target LSN by effectively fusing the multiple views — social, spatial, and temporal views. To this end, we devise a Multi-View Matching Network (MVMN) and make the main contributions as follows:

- i) We address the issue of overlooking the temporal factor in social link inference through detailed data analyses of the temporal influence on this problem. Motivated by this, we develop a time-series module composed by a loosely coupled neural temporal point process. It is benefited from the powerful temporal point process [25], with a simple extension of loosely coupling two Recurrent Neural Networks (RNNs) in Recurrent Marked Temporal Point Process (RMTTP) [26]. It welcomed for the capability of learning to match any pair of user time-series by capturing their continuity and sequentiality. To our best knowledge, this is the first work to adopt point process for learning to match user trajectories, which might benefit the application studies of temporal point process in the literature.
- ii) Our proposed model MVMN learns the user-to-user correlation based on its multiple views, which is welcomed for the adequate utilization of different information types. The sketch of MVMN is shown in Fig. 1, wherein each view corresponds to a tailored module in MVMN to contribute to correlation computation. Despite the above time-series module, a location matching module uses a pairwise

matching matrix to measure the relevance of each location pair. Meanwhile, a relation matching module employs graph neural networks to characterize the proximity between users. The learned multiple view-specific representations for matching are seamlessly fused for final link inference. Such a design provides flexibility and completeness of modeling each view more effectively.

iii) Experimental results on real public datasets demonstrate the superiority of our approach over several strong competitors, including general link prediction methods (e.g., GAT [27]), link inference approaches for LSNs (e.g., HeterGCN [17]), user identity linkage methods (e.g., DPLink [21]), and sequence matching models (e.g., ESIM [28]). Ablation studies show the effectiveness of each module in MVMN. In particular, the incorporation of RMTTP into our model, accompanied by temporal loss functions, has been verified to indeed contribute to inference performance.

II. RELATED WORK

In this section, we review the literature from three aspects: social link inference, spatio-temporal trajectory mining, and the core background techniques.

A. Social Link Inference

The history of link inference in social networks is not very long, dating back to the early study [1] which investigates how the different network based measures of node proximity affect inference performance. Since then, link prediction has been extensively studied. On the methodological aspect, various approaches like supervised random walk [29], factor graph [30], and deep learning [31], have been applied to the problem. On the other hand, different categories of social networks have been explored, covering heterogeneous networks [5] having more than one type of nodes and edges, signed social networks [32] having positive and negative edges, and coupled networks [6] whereby more than one social network are considered.

In contrast to the above studies focusing on network structures, some other studies additionally leverage spatio-temporal trajectories of users to perform link inference. The seminal work [13], as a representative of feature based classification methods [33], [14], [15], designs hand-crafted features such as the co-occurrence of visited locations and overlapping ratios of social friends. The study [34] jointly factorizes the user-location interaction matrix, user-event interaction matrix, and social relation interaction matrix through the Bayesian latent factor model for friend recommendation. The recent studies leverage the power of graph representation learning to obtain low-dimensional representations of users and locations [17], as well as time intervals [16]. However, only user representations are directly leveraged for user matching, while the spatial and temporal views of user matching are not fully exploited. In a nutshell, most of the above studies have overlooked the temporal aspect of user trajectories. Although the study [16] regards timestamps as nodes to learn the representations, the discretization of timestamps misses the continuity and sequentiality of time. In this paper, we regard spatial, temporal,

and social factors as different views of user pairs and perform view-specific user matching to sufficiently utilize the three types of information.

B. User Spatio-temporal Trajectory Mining

User spatio-temporal trajectory mining is a hot research theme for understanding user mobility patterns. Despite the aforementioned research on user link prediction with spatial trajectories, a variety of other applications have been investigated [35], wherein recent advances have abundantly utilized sequential representation learning. Liu et al. [9] developed context-aware RNNs for the next location prediction. Chang et al. [10] conducted successive location recommendation by combining location and content representation learning. Moreover, the check-in time is estimated by incorporating the check-in representations learned by RNNs into survival analysis [36] and point process [37], respectively. Similarly, RNNs are employed to learn trajectory representation [12], used as the input to a softmax function for calculating the probabilities of all candidate users to be linked. In computer vision, some studies propose spatial-temporal image networks for video super-resolution [38], action recognition and localization [39]. However, as introduced in Section I, all the above problems take only one (user) sequence as input each time to estimate its target variable. This makes their problem settings substantially different from the studied user link inference problem.

User identity linkage across different LSNs is a more similar problem setting. It links accounts of the same user from different social networks based on their spatio-temporal trajectories. Specifically, the spatial and temporal spaces are divided into different time intervals [18]. The co-occurrence of two users in the same interval determines their similarities. Chen et al. first considered to measure the similarity of stay regions, local and global time distributions [19], and then further improved accuracy using kernel density estimation [20]. The study [21] is the first one to use deep learning approaches in this problem. RNNs are leveraged to encode a pair of user trajectories and an attention based selector is proposed to capture the correlations between the two trajectories. However, this problem involves multiple social networks and only concentrates on linking the same user across networks. Hence the consideration of social relations is fundamentally different and the existing user relations are commonly not used in these studies.

C. Core Background Techniques

Temporal point process [25] is a principled approach for handling temporal sequences in a continuous space. It has been applied to paper citation count modeling [40], tweet popularity modeling [41], merger and acquisition activity prediction [42], and patient flow prediction [43], to name a few. More recently, point process also finds its application in spatio-temporal domain [44]. However, few relevant studies have considered applying it to sequence matching tasks. To capture the continuity and sequentiality of time, we simply extend RMTTP [26], a powerful recurrent neural point process model, to a loosely coupled version to enable the simultaneous modeling of two time-series.

Graph neural networks have emerged in recent years as effective tools for encoding high-order relations of graphs into low-dimensional node embeddings [45], [7]. In particular, graph convolution network (GCN) [46] and graph attention network (GAT) [27] are two representatives. GraphSAGE [47] leverages node features such as text attributes for achieving inductive graph representation learning. For the reason that GAT can automatically determine the weights of user neighbors, we adopt it to model social relations.

Multi-view learning [48], [49] is a paradigm exploiting the different views of the same input data to acquire abundant information for learning. It has a variety of applications, e.g., learning view-specific embedding for clustering [50] and leveraging multi-view observations as supervisory signals for pose prediction [51]. In particular, the studies [52], [53], [17] investigate the effectiveness of multi-view learning for recommendation, wherein the studies [52], [17] regard each item domain as one view or take different parts of news as their different views. By contrast, each view in our study corresponds to one type of matching between two users. As far as we know, this paper is the first study of leveraging multi-view learning for social link inference.

III. PRELIMINARIES

A. Problem Formulation

Let $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{P})$ represent a target location-aware social network to be investigated. It involves three fundamental elements: (1) \mathcal{U} is a set of user IDs in the network, (2) \mathcal{E} denotes the known social links between users, and (3) \mathcal{P} represents a set of locations IDs corresponding to POIs in the real world. In this type of social network, one of the main user behaviors is check-ins defined as follows:

Definition 1 (Check-in). *A check-in c is a tuple w.r.t. a location $l \in \mathcal{P}$ and a timestamp t , i.e., $c = (l, t)$.*

Given any user ID $u \in \mathcal{U}$, we define its check-in trajectory as follows:

Definition 2 (Check-in Trajectory). *The check-in trajectory of user u is constituted by multiple consecutive check-ins denoted as $\mathcal{S}_u = \{c_1^u, \dots, c_{\ell(u)}^u\}$, where $\ell(u)$ represents the number of check-ins.*

Given user check-in trajectories and existing social links, the aim of social link inference in an LSN is to estimate the existence of social links between users which currently do not belong to \mathcal{E} . We formulate the studied problem as follows:

Problem 1 (Social Link Inference in an LSN). *Given a pair of users u_m and u_n , the goal of this problem is to learn a model $\mathcal{F}(\mathcal{S}_{u_m}, \mathcal{S}_{u_n}, \mathcal{E}) \rightarrow \hat{y} \in \{0, 1\}$ based on existing user trajectories and social links, which can accurately estimate whether they have a social link (1) or not (0).*

The above problem setting illustrates that the key to success is effective model design and training.

B. Motivation of Temporal Influence on Link Inference

In this part, we carry out preliminary data analyses on Gowalla and Foursquare (shown in Section V-A) to validate the

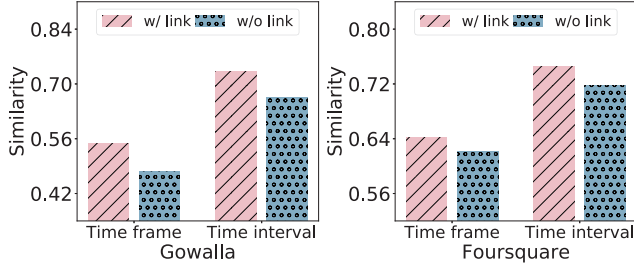


Fig. 2: Comparison of temporal similarities.

rationality of considering the temporal influence on social link inference. The analyses involve two aspects of comparisons, i.e., temporal similarities of linked users versus temporal similarities of unlinked users and ratios of spatial co-occurrence versus ratios of spatio-temporal co-occurrence for linked users:

- 1) We define two types of temporal similarities to realize the first comparison, i.e., time frame based similarities and time interval based similarities. Time frames indicate the time when user check-in behaviors happen, while time intervals show the temporal difference between two consecutive check-in behaviors. Specifically, we have 24 time frames, each of which corresponds to one hour per day, to denote the frame based check-in distributions of users' behaviors. Consequently, the time frame based similarities are calculated by the average cosine similarities of linked users (w/ link) or unlinked users (w/o link). Without loss of generality, we further define 7 time intervals in hours: $[0, 1)$, $[1, 2)$, $[2, 6)$, $[6, 12)$, $[12, 24)$, and $[24, \infty)$, which are utilized to represent interval based check-in distributions. Similar to the time frame based similarities, we also use average cosine similarities to obtain time interval based similarities. Fig. 2 compares the similarities for linked users (w/ link) or unlinked users (w/o link). It is clear that linked users have larger temporal similarities than unlinked users.
- 2) Ratios of spatial co-occurrence versus ratios of spatio-temporal co-occurrence for linked users are calculated by the following manner. Assume the number of user pairs visiting at least one common location is N^L , wherein the count of user pairs also having social links is N_S^L . Spatial ratio computation is given as: $SR = \frac{N_S^L}{N^L}$. We further compute spatio-temporal ratio in a similar fashion, by denoting the number of user pairs visiting at least one common location in the same time interval (the length of time interval is set to 1 hour) is N^{LT} , and the corresponding number of user pairs having social links as well is N_S^{LT} . Based on the two counts, spatio-temporal ratio is calculated by: $STR = \frac{N_S^{LT}}{N^{LT}}$. Table I compares the two types of ratios, from which we observe the additional consideration of temporal information could bring benefits to the social relation refinement.

In summary, the above analyses demonstrate that the temporal aspect of user check-in behaviors has a great effect on social relations, which motivates us to utilize the powerful temporal point process to capture this for boosting the performance of social linkage inference.

TABLE I: Ratio comparison for spatial co-occurrence and spatio-temporal co-occurrence.

Dataset	spatial ratio	spatio-temporal ratio
Gowalla	35.8%	73.4%
Foursquare	26.8%	52.6%

IV. THE COMPUTATIONAL METHODOLOGIES

The architecture of our model MVMN is presented in Fig. 3. The input for MVMN is composed of a user pair with check-in trajectories and a social relation graph containing the existing user links. The output of the model is the estimation of the possibility that the user pair has a social connection. In MVMN, there are three major modules: *location matching module*, *time-series matching module*, and *relation matching module*. The three modules provide different views to characterize the degree of matching between two users. In the end, the matching based representations from multiple views are integrated to calculate the relevance score of the two given users. The decomposition of relevance computation into different views enables the flexibility of designing tailored methods for each view.

A. Input Representation

We encode locations, timestamps, and users into low-dimensional vector spaces for ease of later representation learning. Specifically, the one-hot encoding \mathbf{x}_u of user u can be first acquired by its ID. Afterwards, a commonly adopted lookup operation is performed, i.e., $\mathbf{e}_u^U = \mathbf{E}_U^T \mathbf{x}_u$. Similarly, location l is converted into \mathbf{e}_l^L based on its ID and embedding matrix \mathbf{E}_L as well. It is a little different for dealing with timestamps due to its continuity. Here we first divide each timestamp into different time intervals, and later will use temporal point process for modeling continuous timestamps. We also encode each time interval by the usage of the time embedding matrix. Thus timestamp t is represented as \mathbf{e}_t^T .

Without loss of generality, we take u_m and u_n as an example for specifying the matching details of MVMN. The spatial, temporal, and social aspects are distinguished by different views of the user pair. In particular, user u_m is associated with a spatial sequence $L_m = \{\mathbf{e}_{m,1}^L, \dots, \mathbf{e}_{m,\ell(u_m)}^L\}$, a temporal sequence $T_m = \{\mathbf{e}_{m,1}^T, \dots, \mathbf{e}_{m,\ell(u_m)}^T\}$, and a social representation \mathbf{e}_m^U . Likewise, it is the same for user u_n to have its corresponding representations.

B. Location Matching Module

From the geographical view of the user trajectory pair, we first calculate a pairwise matching matrix to measure the correlation between one location in the spatial sequence of user u_m and another location related to user u_n . Formally, given the i -th location in L_m and the j -th location in L_n , the cosine similarity is computed as follows:

$$S_{i,j} = \cos(\mathbf{e}_{m,i}^L \cdot \mathbf{e}_{n,j}^L). \quad (1)$$

After traversing all pairs of locations in the two sequences, we can obtain the similarity matrix \mathbf{S} where each entry corresponds to one location pair.

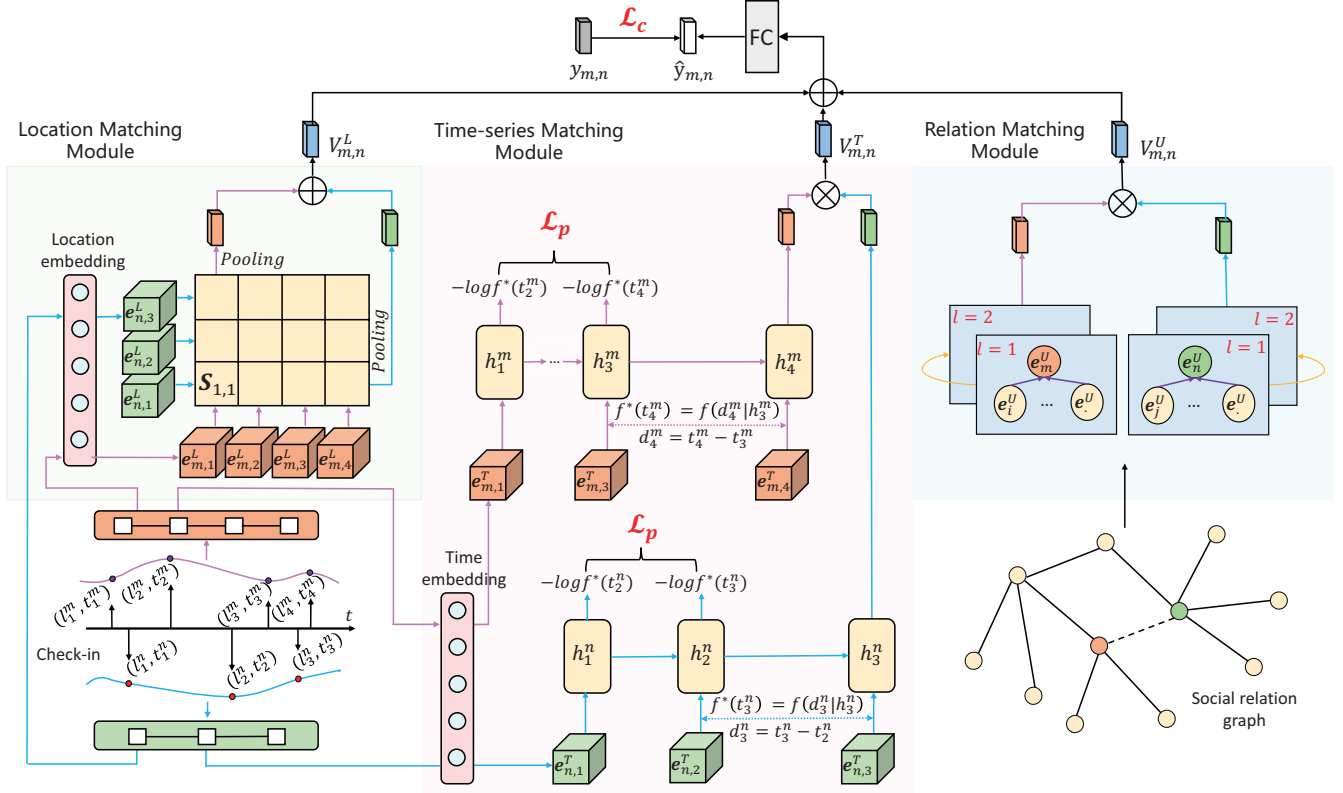


Fig. 3: The architecture of MVMN. We visualize it with two toy trajectory examples. One is for user u_m with length 4 and the other is for user u_n with length 3. The three color regions correspond to different modules. The blue and pink arrows denote the information flow of different users. The loss functions are marked with red font.

It is intuitive that if the two spatial sequences are more similar, there is a bigger chance that the two users will have a social connection. With this intuition, we perform row-wise max-pooling and column-wise max-pooling computations as:

$$\mathbf{s}_i^m = \max_{a \in \{1, \dots, \ell(u_n)\}} \mathbf{S}_{i,a} \quad i \in \{1, \dots, \ell(u_m)\}, \quad (2)$$

$$\mathbf{s}_j^n = \max_{a \in \{1, \dots, \ell(u_m)\}} \mathbf{S}_{a,j} \quad j \in \{1, \dots, \ell(u_n)\}. \quad (3)$$

The above equations return the largest relevance score for each location, which plays a role in summarizing the interaction of the two trajectories.

The two obtained relevance vectors are combined together to form the representation $\mathbf{V}_{m,n}^L = [\mathbf{s}^m \oplus \mathbf{s}^n]$ to capture the correlation of user spatial trajectories, where $\mathbf{s}^m = \{\mathbf{s}_1^m, \dots, \mathbf{s}_{\ell(u_m)}^m\}$ and $\mathbf{s}^n = \{\mathbf{s}_1^n, \dots, \mathbf{s}_{\ell(u_n)}^n\}$. \oplus represents the concatenation operation. In practice, we set a maximal length (K) of a trajectory to ensure the embedding size is fixed, which facilitates the later adoption of fully-connected layers to yield the final relevance score.

C. Time-series Matching Module

The temporal aspect of user trajectories is reflected in the continuity and sequentiality of time. As mentioned before, we propose loosely coupled RMTTP to learn time-series matching. Specifically, we first feed the pair of temporal sequences $T_m = \{\mathbf{e}_{m,1}^T, \dots, \mathbf{e}_{m,\ell(u_m)}^T\}$ and $T_n = \{\mathbf{e}_{n,1}^T, \dots, \mathbf{e}_{n,\ell(u_n)}^T\}$

into recurrent neural networks which are implemented by LSTM [54] for its better performance in our experiments. We omit the computational details of LSTM and define the brief formulas for obtaining sequential hidden representations as follows:

$$\mathbf{h}_i^m = \text{LSTM}(\mathbf{e}_{m,i}^T, \mathbf{h}_{i-1}^m) \quad i \in \{1, \dots, \ell(u_m)\}, \quad (4)$$

$$\mathbf{h}_j^n = \text{LSTM}(\mathbf{e}_{n,j}^T, \mathbf{h}_{j-1}^n) \quad j \in \{1, \dots, \ell(u_n)\}, \quad (5)$$

where $\text{LSTM}(\cdot)$ corresponds to each time step.

We adopt the finally returned embeddings $\mathbf{h}_{\ell(u_m)}^m$ and $\mathbf{h}_{\ell(u_n)}^n$ to represent each user's time-series. Given this, the correlation representation of user time-series is computed as follows:

$$\mathbf{V}_{m,n}^T = \tanh(\mathbf{h}_{\ell(u_m)}^m \otimes \mathbf{h}_{\ell(u_n)}^n), \quad (6)$$

where \otimes is the Hardward product. The usage of the activation function $\tanh(\cdot)$ constrains the value in each dimension of $\mathbf{V}_{m,n}^T$ in the range of $[-1, 1]$. As such, they are consistent with the cosine similarities adopted in the location matching module, which are more interpretable. We have also tried some advanced techniques such as attention mechanisms [55] to fuse multiple hidden representations. In particular, the attention computation is conducted over all the hidden representations and the query representations are obtained by their mean-pooling. Yet the results show no significant improvements.

In this way, the sequentiality of user time-series could be captured. To further model the continuity of time to guide the effective representation learning of $\mathbf{h}_{\ell(u_m)}^m$ and $\mathbf{h}_{\ell(u_n)}^n$, we

introduce a temporal point process based loss for time-series. In temporal point process, the conditional intensity function $\lambda^*(t)$ plays a key role in capturing the temporal dynamics in a continuous space. By convention, $*$ in the function denotes it is history dependent. Within a short time interval $[t, t+dt)$, $\lambda^*(t)$ represents the occurrence rate of a new check-in event given the history \mathcal{H}_t and satisfies: $\lambda^*(t) dt = P\{c \in [t, t+dt) | \mathcal{H}_t\}$. Based on this, the density function is given as:

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{t_j}^t \lambda^*(\epsilon) d\epsilon\right), \quad (7)$$

where t_j could be a happened timestamp of the last check-in event or a starting timestamp.

In RMTTP, $\lambda^*(t)$ has a well-designed parametric form associated with the accumulative influence of past check-ins, the temporal gap since the last event happened, and a global base intensity to denote the context-independent occurrence of a next event. For the given two users, their conditional intensity functions are defined as follows:

$$\lambda_m^*(t) = \exp\left(\mathbf{v}^\top \mathbf{h}_i^m + \omega(t - t_i^m) + b\right), \quad (8)$$

$$\lambda_n^*(t) = \exp\left(\mathbf{v}^\top \mathbf{h}_j^n + \omega(t - t_j^n) + b\right), \quad (9)$$

where \mathbf{v} , ω , and b are trainable parameters. Through Eq. 6, the above two functions of recurrent temporal point process are loosely coupled.

Due to this simple $\lambda^*(t)$, the density function has an analytical form. Here we clarify the density function $f_m^*(t)$ for user u_m :

$$\begin{aligned} f_m^*(t) &= \lambda_m^*(t) \exp\left(-\int_{t_i^m}^t \lambda_m^*(\epsilon) d\epsilon\right) \\ &= \exp\left\{\mathbf{v}^\top \mathbf{h}_i^m + \omega(t - t_i^m) + b + \frac{1}{\omega} \exp(\mathbf{v}^\top \mathbf{h}_i^m + b) \right. \\ &\quad \left. - \frac{1}{\omega} \exp(\mathbf{v}^\top \mathbf{h}_i^m + \omega(t - t_i^m) + b)\right\}. \end{aligned} \quad (10)$$

Similarly, $f_n^*(t)$ could be derived.

In the end, the loss function of loosely coupled RMTTP is denoted as the negative joint log-likelihood of generating the time-series:

$$\mathcal{L}_p = -\sum_{i=1}^{\ell(u_m)-1} \log f_m^*(t_{i+1}^m | \mathbf{h}_i^m) - \sum_{j=1}^{\ell(u_n)-1} \log f_n^*(t_{j+1}^n | \mathbf{h}_j^n). \quad (11)$$

Through the optimization towards \mathcal{L}_p , the hidden representations $\mathbf{h}_{\ell(u_m)}^m$ and $\mathbf{h}_{\ell(u_n)}^n$ could receive useful signals distilled from the continuous timestamps of check-ins.

D. Relation Matching Module

As discussed previously, we adopt GAT to learn user embeddings in the view of their social relations. The core advantage of GAT is the ability to determine the importance weights of neighbor nodes for each target user node. This is appropriate for our study since the original binary values of social edges could not differentiate the relation strength between different users. To be specific, the attention coefficient of the given user pair is computed as follows:

$$c_{m,n} = \text{FC}([\mathbf{W}\mathbf{e}_m^U \oplus \mathbf{W}\mathbf{e}_n^U]), \quad (12)$$

where $\text{FC}(\cdot)$ denotes one or more fully-connected layers. \mathbf{W} is a parameter matrix for linear transformation.

Given the obtained attention coefficients, the importance weights of neighbors are normalized to be probability distributions for ease of updating representations later. Taking user u_m for illustration, we have the following formula to get the corresponding distribution:

$$\alpha_{m,n} = \frac{\exp\left(\text{LeakyReLU}(c_{m,n})\right)}{\sum_{i \in \mathcal{N}_m} \exp\left(\text{LeakyReLU}(c_{m,i})\right)}, \quad (13)$$

where \mathcal{N}_m includes the user itself and its neighbor users. $\alpha_{m,n}$ reflects the contribution of user u_n to updating the representation for user u_m . The updated representation is calculated by an aggregation function:

$$\mathbf{e}_m^U = \sigma\left(\sum_{i \in \mathcal{N}_m} \alpha_{m,i} \mathbf{W}\mathbf{e}_i^U\right), \quad (14)$$

where $\sigma(\cdot)$ is a nonlinear activation function (e.g., $\text{ELU}(\cdot)$). Besides, multi-header attention is commonly adopted as an extension of single-header attention to learn multiple attention coefficients to obtain multiple user node representations. We take the average of these representations as the updated representations. More details can be referred to the study [27].

We summarize the above basic procedures (from Eq. 12 to Eq. 14) for one-time graph propagation as $\text{GAT}(\cdot)$. Then it could be generalized to a multi-layer computational formula, given as:

$$\mathbf{e}_{m,(k+1)}^U = \text{GAT}(\mathbf{W}, \mathbf{e}_{m',(k)}^U) \quad m' \in \mathcal{N}_m, \quad (15)$$

which corresponds to the representation propagation from layer k to layer $k+1$. For starting propagation, $\mathbf{e}_{m,(0)}^U$ is set to \mathbf{e}_m^U .

After K layers of propagation, we obtain the final social representations $\mathbf{e}_{m,(K)}^U$ and $\mathbf{e}_{n,(K)}^U$ for the two users. The Hardward operation is leveraged as well to acquire the representation for social matching, defined as follows:

$$\mathbf{V}_{m,n}^U = \tanh(\mathbf{e}_{m,(K)}^U \otimes \mathbf{e}_{n,(K)}^U). \quad (16)$$

The activation function $\tanh(\cdot)$ plays a role similar to Eq. 6.

E. Multi-view Fusion and Link Estimation

In order to measure user-to-user correlations from a holistic perspective, our model integrates the view-specific representations by $[\mathbf{V}_{m,n}^L \oplus \mathbf{V}_{m,n}^T \oplus \mathbf{V}_{m,n}^U]$. The final social relation prediction is conditioned on the above integrated representation, which is defined as:

$$\hat{y}_{m,n} = \varphi\left(\text{FC}([\mathbf{V}_{m,n}^L \oplus \mathbf{V}_{m,n}^T \oplus \mathbf{V}_{m,n}^U])\right), \quad (17)$$

where $\varphi(\cdot)$ is a sigmoid function which outputs the probability of the user pair to have a social link.

The cross-entropy loss is adopted to quantify the gap between the predicted relation and ground-truth relation:

$$\mathcal{L}_c = -\left(y_{m,n} \log \hat{y}_{m,n} + (1 - y_{m,n}) \log(1 - \hat{y}_{m,n})\right), \quad (18)$$

where $y_{m,n}$ corresponds to the ground-truth relation label. The above loss function is dedicated to the user pair of u_m and u_n , and could be easily generalized to all other user pairs in the training stage. Since for each user, the total number of its linked users in training data is relatively small compared to that of unlinked users, we sample a small number of unlinked users to build pseudo negative pairs.

The final objective function is a combination of the cross-entropy loss and the point process loss, which is written as:

$$\mathcal{L} = \mathcal{L}_c + \beta \mathcal{L}_p, \quad (19)$$

where β controls the relative effect of the point process loss and could be tuned on the validation dataset.

V. EXPERIMENTAL SETUP

To evaluate the performance of our model, we first illustrate basic experimental setups, including the datasets we used, the adopted evaluation metrics, the carefully chosen strong baselines, and some implementation details.

A. Datasets

We conduct experiments on two real-world and publicly available location-aware social networks:

Gowalla¹: This is a widely used dataset for relevant studies, collected by the work [56]. The raw dataset consists of 196,591 users and 950,327 relations. The number of corresponding check-ins is 6,442,890 crawled during the period of Feb. 2009 to Oct. 2010. Although the dataset contains the sampled users over the planet, the relations mainly exist between two users in the same city due to the low possibility of check-in behaviors happened around the world. Therefore it is not very meaningful to conduct relation inference for all these users. For this reason, we choose one of the largest cities, New York City, to select users whose check-ins are usually located in the city. In particular, we first specify the region based on its longitude and latitude. Consequently, we filter out those users with small proportions of her/his total check-ins (less than 0.1) within the specified region. The subset selection by different cities in location-aware social networks is commonly used by the prior studies [57], [16]. To ensure the reliability of comparison, we remove users with less than one friend and ten check-ins.

Foursquare²: This dataset is crawled from the most popular LSN, i.e., Foursquare. Similar to the data selection procedure for Gowalla, we choose another large city, Los Angeles City, to gather training data, which ensures data diversity. We follow the same preprocessing methods applied for Gowalla to filter out some inactive users in the chosen city.

TABLE II: Statistics of the experimental datasets.

Dataset	#Users	#Locations	#Check-ins
Gowalla	1,270	16,615	67,194
Foursquare	1,092	24,133	98,351

Through the above procedures of preprocessing, we have obtained our experimental datasets and their characteristics are shown in Table II. For ease of testing, we randomly divide social relations into training sets, validation sets, and test sets with the ratios of 0.8, 0.1, and 0.1. As is known to all, the number of negative social relation pairs is very large. Therefore it is very time-consuming for evaluations. For this consideration, we follow [58], [59] by randomly selecting 50 users not linked with each target user as her/his candidates. Finally, we have 18,304 and 19,128 relation pairs in the test sets of Gowalla and Foursquare, respectively.

B. Metrics

The performance of user link inference in an LSN is measure by three widely used metrics, i.e. precision at position k (P@k), recall at position k (R@k), and area under the receiver operating characteristic curve (AUC). Among them, P@k measures the faction of accurately linked users contained by returned top k candidates, and R@k denotes the faction of accurately estimated user links contained by all ground-truth links. To be specific, P@k and R@k are given as follows:

$$P@k = \frac{1}{Q_U} \sum_u \frac{N_{true}^u}{k}, \quad R@k = \frac{1}{Q_U} \sum_u \frac{N_{true}^u}{R_u}, \quad (20)$$

where N_{true}^u denotes the number accurately inferred links in top-K ranked candidates. R_u is the number of actual links in the ground truth set of user u . Q_U is the number of test users. In the experiments, we set k in P@k and R@k to 10 for comparison.

AUC relies on the true positive rate and false positive rate to compute its value. It could be explained as the probability of ranking positive links higher than negative links. The expectation value of AUC is equal to 0.5 for a random guess. The computational details of AUC can be referred to [21]. For simplicity, we use Scikit-learn to obtain the metric scores.

C. Baselines

We organize the baselines into four groups, according to the original problem each baseline focuses on.

G1 — Social link inference in an LSN:

Fea_LR and **Fea_RF**. The pioneering study [13] adopts feature based classification methods. Following this, we use logistic regression and random forest as the classification models, and name them as Fea_LR and Fea_RF, respectively. The well-designed features, include social, temporal (the similarities mentioned in Section III-B) and geographical features, are taken as the input of the two models.

LBSN2Vec. LBSN2Vec³ [16] is a hypergraph embedding method which encodes user-user edges and user-time-POI-semantic hyperedges. We adapt this method to our problem by only removing semantic nodes that might be missing in some domains and are not considered by this study.

Heter-GCN. The model [17] benefits from graph convolutional networks for learning social relations and recurrent neural networks to model trajectories. The integration is achieved

¹<https://snap.stanford.edu/data/loc-Gowalla.html>

²<https://sites.google.com/site/dbhonzhi/>

³<https://github.com/eXascaleInfolab/LBSN2Vec>

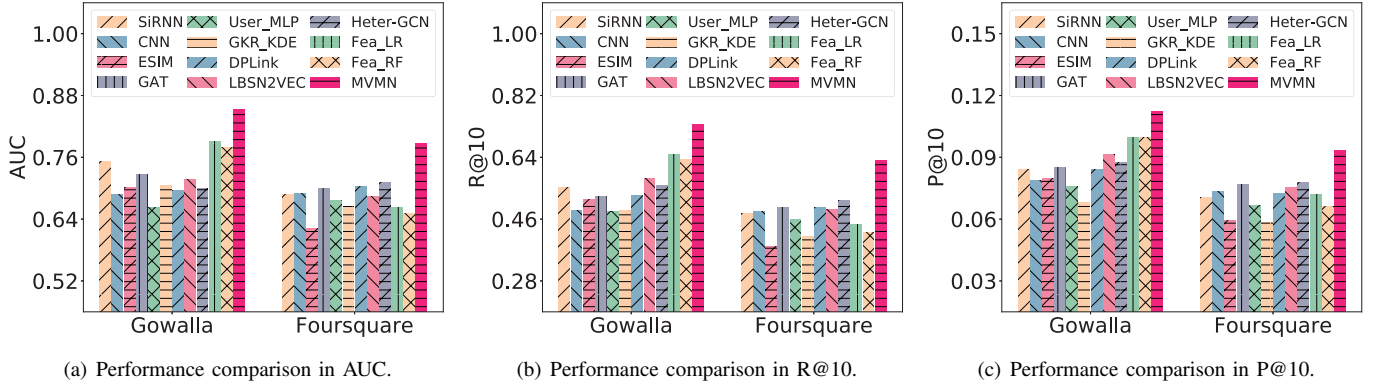


Fig. 4: Evaluation results by our proposed model and baselines on the Gowalla and Foursquare datasets.

by initializing the graph node representations with the hidden representations obtained by RNNs.

G2 — User identity linkage across LSNs:

GKR_GKE. GKR_GKE⁴ [20] is a kernel density based method which improves DG [19] proposed earlier by the same authors.

DPLink. Compared with other methods for user identity linkage, the novel insights of DBLink⁵ [21] are the first appliance of deep learning techniques and the proposal of attention mechanism based selective matching.

G3 — Sequence matching:

SiRNN. This is partially inspired by the study [60] which proposes to use siamese RNNs for sentence matching. The objective function for classification is based on cross-entropy loss, being consistent with ours.

CNN. Convolutional neural networks are utilized to measure the correlation of patient event sequences [61]. We train this model in a supervised manner.

ESIM. ESIM⁶ [28] builds on two recurrent neural networks and seizes the interaction between two sequence embeddings by its proposed interactive attention computation method.

G4 — General user recommendation:

User_MLP. We feed two user representations into multi-layer perceptrons (MLP) which correspond to the deep modeling part of NeuMF [59].

GAT. GAT [27] is directly applied to our problem by only considering social relations as edges and users as nodes. A similar application can be found in the work [62] which models a user-item bipartite graph.

D. Implementations

The hyperparameters of our model and baselines are tuned to achieve their better performance on the validation sets. The maximal trajectory length K is set to 200. If a user's trajectory is longer than the threshold, we keep its recent 200 check-ins. For the negative sampling adopted in model training mentioned in Section IV-E, we set the number of sampled

users to 4. By default, the embedding size is set to 64, the dimension of the hidden unit in RNNs (e.g., LSTM) is set to 128. For the model GAT and our relation matching module, the number of attention header is equal to 3. We use the Adam optimizer to train our model and other representation learning based methods. The learning rate is $1e-4$ and the batch size is 64. We also consider using dropout with a ratio of 0.5 for intermediate layers to overcome the overfitting issue. All the adopted models are run on a small server with two GPU GTX 1080Ti cards. We implement our model by PyTorch 1.1. If the baselines have source codes, we use them for comparison with some necessary minor modifications. Otherwise, we implement them.

VI. EXPERIMENTAL RESULTS

In this section, we conduct comprehensive comparisons to answer the following crucial research questions for a better understanding of our model, especially for the rationality of fusing multiple views for linking users:

Q1: What is the performance of the proposed MVMN for link inference compared to strong competitors?

Q2: Does each view-specific module in MVMN contribute to the final performance?

A. Performance Comparison (Q1)

Fig. 4 shows the performance of our model MVMN and other baselines, from which we observe the performance comparison is not exactly the same across the two datasets.

Among the baselines, we can observe that the feature based classification methods — Fea_LR and Fea_RF achieve the best performance in the Gowalla dataset, and in the Foursquare dataset, Heter-GCN behaves a little better than other baselines. This might be attributed to the reason that these methods within group G1 are tailored for the problem the same as ours. LBSN2VEC is a little worse since its representation learning is not directly guided by optimizing social relations. The other baselines have some issues when applied to our problem: (1) for the methods within group G4, only user relations are well utilized for inference; (2) the commonly adopted sequence matching models in G2 do not utilize temporal factors and social relations; and (3) the approaches belonging to group

⁴<https://www.dropbox.com/s/mrkfao8gr8zktkw/ICDE-18-Chenwei.zip?dl=0>

⁵<https://github.com/vonfeng/DPLink>

⁶<https://github.com/lukec1231/nli>

TABLE III: Ablation study of our model on the two datasets.

Dataset	Method	AUC	R@10	P@10
Gowalla	MVMN	0.852	0.734	0.112
	V1	0.826	0.709	0.105
	V2	0.802	0.676	0.101
	V3	0.780	0.621	0.093
	V4	0.724	0.541	0.081
Foursquare	MVMN	0.787	0.631	0.093
	V1	0.740	0.558	0.082
	V2	0.714	0.529	0.076
	V3	0.681	0.492	0.079
	V4	0.699	0.503	0.075

G3 do not address social relations and there is still room to improve the representation learning for trajectory matching. Thus the observation meets our expectations to some extent.

By further comparing the methods in group G1, we observe that the two feature based methods obtain relatively good results on Gowalla, significantly beating other baselines. It might be the reason that the well-crafted features containing much domain-specific knowledge. On the other hand, the graph representation learning based models obtain better results than the feature based methods on Foursquare. This reflects the prospect of representation learning if it is well treated.

In the end, we can see the proposed MVMN outperforms all the competitors on both datasets. In particular, on the Gowalla dataset, MVMN improves the state-of-the-art Fea_LR method from 0.647 to 0.734 in R@10, and from 0.100 to 0.112 in P@10. Moreover, on the Foursquare dataset, MVMN improves the state-of-the-art Heter-GCN model from 0.515 to 0.631 in R@10, and from 0.078 to 0.093 in P@10. We demonstrate later that the improvements are not only from the modeling of temporal dynamics based on temporal point process, but also thanks to the fusion of the learned matching representations from multiple views.

B. Ablation Studies (Q2)

We conduct ablation experiments to verify the contribution of each module. This is a crucial step to understand how our model works. To achieve this, we define the following variants of our model:

- * **V1: MVMN w/o (RM)**. It removes the relation matching module from MVMN, which is equivalent to not explicitly modeling social relations in the model architecture.
- * **V2: MVMN w/o (RM+PP)**. To show how temporal point process influences our model, this method further erases \mathcal{L}_p by setting β to 0. Noting that the temporal correlation computation of users in the time-series matching module is still maintained.
- * **V3: MVMN w/o (RM+TM)**. This variant removes not only the relation matching module but also the time-series matching module. It is equivalent to only using the location matching module for social link inference in a target LSN.
- * **V4: MVMN w/o (RM+LM)**. It removes both the relation matching module and the location matching module. Hence this variant only involves the time-series matching module.

Table III presents the results of our full model and its variants, from which we have the following key observations:

TABLE IV: Performance of alternative designs.

Dataset	Method	AUC	R@10	P@10
Gowalla	V3	0.780	0.621	0.093
	V3 w/ RNN	0.625	0.442	0.064
	V2	0.802	0.676	0.101
	V3 w/ \oplus Time	0.778	0.619	0.094
	MVMN	0.852	0.734	0.112
	MVMN w/ GCN	0.818	0.694	0.107
Foursquare	MVMN w/ Attention	0.822	0.705	0.106
	V3	0.681	0.492	0.079
	V3 w/ RNN	0.589	0.333	0.046
	V2	0.714	0.529	0.076
	V3 w/ \oplus Time	0.688	0.509	0.079
	MVMN	0.787	0.631	0.093
	MVMN w/ GCN	0.732	0.553	0.081
	MVMN w/ Attention	0.736	0.548	0.080

- 1) The performance comparison between V1 and the full model MVMN demonstrates the positive contribution of the relation matching module. The results are intuitive since social relations could reveal the user-to-user proximity from the aspect of network structures. Moreover, compared to V1, both V3 and V4 incur performance drop. This reflects the location matching module and the time-series matching module contribute to the final performance. As such, the fusion of multiple views for matching is effective.
- 2) Compared to V1, V2 removes the point process loss \mathcal{L}_p , thereby only using twin LSTMs for modeling user temporal behavior sequences. The results of V2 are significantly inferior to V1. This indicates our model indeed benefits from the consideration of temporal point process.

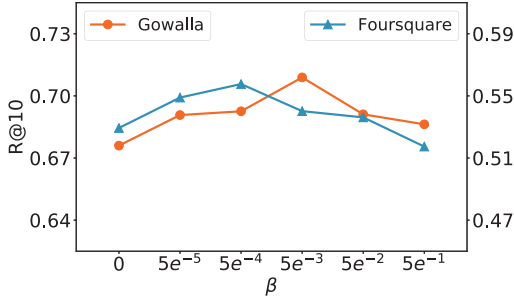
C. Alternative Designs for MVMN

We design some alternatives of MVMN and its variants to better understand the intuition behind the proposed model:

- * **V3 w/ RNN** uses RNNs in V3 to learn location representations for location-to-location correlation calculation.
- * **V3 w/ \oplus Time** concatenates temporal embeddings with location embeddings in V3 for correlation calculation, in contrast to using LSTMs for representing temporal sequences in V2.
- * **MVMN w/ GCN** leverages GCN instead of GAT for learning social relations.
- * **MVMN w/ Attention** performs attention computation to determine the importance weights of each view and further fuse them with weighted combination.

Table IV shows the results of the alternatives, from which we find:

- 1) V3 w/ RNN degrades the performance of link inference across the two datasets when using RNN modeling.
- 2) Compared to V2, V3 w/ \oplus Time performs worse in most cases. It indicates that using LSTMs for modeling temporal sequences is more effective than tightly fusing spatial and temporal information. As such, it might be better to attribute spatial and temporal information to different views.
- 3) MVMN w/ GCN is inferior to MVMN using GAT, showing the benefit of automatically quantifying the weights of social relations. Besides, adopting attention computation to

Fig. 5: Performance variation with different β .

fuse the representations of different views does not improve the performance.

D. More Discussions

1) *Effect of β* : We show how the performance of link inference changes with different values of β in Fig. 5. The results corresponding to $\beta = 0$ are also reported for ease of illustration. Through comparison, we observe that when β takes values larger than 0, the results are much better than the results with $\beta = 0$. This phenomenon indicates the consideration of temporal point process is beneficial for the studied problem.

2) *Effect of propagation layer number*: Fig. 6 depicts how the user linking performance is affected by the number of representation propagation layers. We can see the results are improved with a larger number of propagation layers at the beginning, and consequently, the results remain to be relatively stable. This phenomenon reveals that learning high-order user relations beyond existing direct relation brings additional advantages.

VII. CONCLUSION

In this paper, we have investigated the problem of social link inference in a given location-aware social network by treating spatial, temporal, and social factors of user pairs as different views for effective user linking. We have devised the multi-view matching network which is welcomed for the benefit of learning the view-specific representation by each matching module. Recurrent neural temporal point process is incorporated into our model to guide the effective representation learning of user time-series. We have conducted extensive experiments and demonstrated our model consistently outperforms strong competitors proposed for this problem and related ones. The rationality of our model design has also been validated by ablation studies.

REFERENCES

- [1] D. Liben-Nowell and J. M. Kleinberg, "The link prediction problem for social networks," in *CIKM*, 2003, pp. 556–559.
- [2] M. Kim and J. Leskovec, "The network completion problem: Inferring missing nodes and edges in networks," in *SDM*, 2011, pp. 47–58.
- [3] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom, "Suggesting friends using the implicit social graph," in *SIGKDD*, 2010, pp. 233–242.

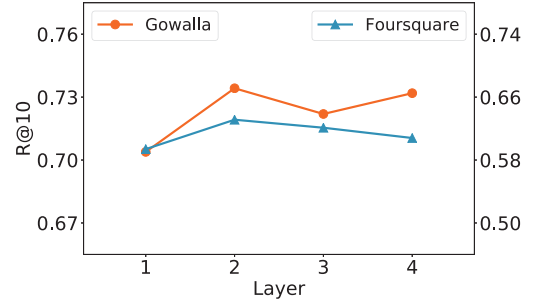


Fig. 6: Performance variation with different layer number in representation propagation in GNN.

- [4] W. Zhang and J. Wang, "A collective bayesian poisson factorization model for cold-start local event recommendation," in *SIGKDD*, 2015, pp. 1455–1464.
- [5] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *ASONAM*, 2011, pp. 121–128.
- [6] Y. Dong, J. Zhang, J. Tang, N. V. Chawla, and B. Wang, "Coupledlp: Link prediction in coupled networks," in *SIGKDD*, 2015, pp. 199–208.
- [7] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017.
- [8] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE TKDE*, 2018.
- [9] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *AAAI*, 2016, pp. 194–200.
- [10] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, "Content-aware hierarchical point-of-interest embedding model for successive POI recommendation," in *IJCAI*, 2018, pp. 3301–3307.
- [11] J. Pan, V. A. Rao, P. K. Agarwal, and A. E. Gelfand, "Markov-modulated marked poisson processes for check-in data," in *ICML*, 2016, pp. 2244–2253.
- [12] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *IJCAI*, 2017, pp. 1689–1695.
- [13] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *SIGKDD*, 2011, pp. 1046–1054.
- [14] H. Hsieh and C. Li, "Inferring social relationships from mobile sensor data," in *WWW (Companion Volume)*, 2014, pp. 293–294.
- [15] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *WSDM*, 2014, pp. 303–312.
- [16] D. Yang, B. Qu, J. Yang, and P. Cudré-Mauroux, "Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach," in *TheWebConf*, 2019, pp. 2147–2157.
- [17] Y. Wu, D. Lian, S. Jin, and E. Chen, "Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference," in *IJCAI*, 2019, pp. 3898–3904.
- [18] C. J. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *WWW*, 2016, pp. 707–719.
- [19] W. Chen, H. Yin, W. Wang, L. Zhao, W. Hua, and X. Zhou, "Exploiting spatio-temporal user behaviors for user linkage," in *CIKM*, 2017, pp. 517–526.
- [20] W. Chen, H. Yin, W. Wang, L. Zhao, and X. Zhou, "Effective and efficient user account linkage across location based social networks," in *ICDE*, 2018, pp. 1085–1096.
- [21] J. Feng, M. Zhang, H. Wang, Z. Yang, C. Zhang, Y. Li, and D. Jin, "Dplink: User identity linkage via deep neural network from heterogeneous mobility data," in *WWW*, 2019, pp. 459–469.
- [22] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *PNAS*, vol. 105, no. 35, pp. 12 763–12 768, 2008.
- [23] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with lbsns and global consistency," in *SIGKDD*, 2015, pp. 1485–1494.
- [24] X. Du, J. Yan, and H. Zha, "Joint link prediction and network alignment via cross-graph embedding," in *IJCAI*, 2019, pp. 2251–2257.

- [25] J. Durbin and G. S. Watson, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.
- [26] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *SIGKDD*, 2016, pp. 1555–1564.
- [27] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
- [28] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *ACL*, 2017, pp. 1657–1668.
- [29] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM*, 2011, pp. 635–644.
- [30] J. E. Hopcroft, T. Lou, and J. Tang, "Who will follow you back?: reciprocal relationship prediction," in *CIKM*, 2011, pp. 1137–1146.
- [31] X. Li, N. Du, H. Li, K. Li, J. Gao, and A. Zhang, "A deep learning approach to link prediction in dynamic networks," in *ICDM*, 2014, pp. 289–297.
- [32] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW*, 2010, pp. 641–650.
- [33] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. Barabási, "Human mobility, social ties, and link prediction," in *SIGKDD*, 2011, pp. 1100–1108.
- [34] Y. Lu, Z. Qiao, C. Zhou, Y. Hu, and L. Guo, "Location-aware friend recommendation in event-based social networks: A bayesian latent factor approach," in *CIKM*, 2016, pp. 1957–1960.
- [35] Y. Zheng, "Trajectory data mining: an overview," *ACM TIST*, vol. 6, no. 3, p. 29, 2015.
- [36] G. Yang, Y. Cai, and C. K. Reddy, "Spatio-temporal check-in time prediction with recurrent neural network based survival analysis," in *IJCAI*, 2018, pp. 2976–2983.
- [37] W. Liang, W. Zhang, and X. Wang, "Deep sequential multi-task modeling for next check-in time and location prediction," in *DASFAA*, 2019, pp. 353–357.
- [38] X. Zhu, Z. Li, X. Zhang, C. Li, Y. Liu, and Z. Xue, "Residual invertible spatio-temporal network for video super-resolution," in *AAAI*, 2019, pp. 5981–5988.
- [39] X. Zhang, H. Shi, C. Li, and P. Li, "Multi-instance multi-label action recognition and localization based on spatio-temporal pre-trimming for untrimmed videos," in *AAAI*, 2020.
- [40] S. Xiao, J. Yan, C. Li, B. Jin, X. Wang, X. Yang, S. M. Chu, and H. Zha, "On modeling and predicting individual paper citation count over time," in *IJCAI*, 2016, pp. 2676–2682.
- [41] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "Seismic: A self-exciting point process model for predicting tweet popularity," in *SIGKDD*, 2015, pp. 1513–1522.
- [42] J. Yan, S. Xiao, C. Li, B. Jin, X. Wang, B. Ke, X. Yang, and H. Zha, "Modeling contagious merger and acquisition via point processes with a profile regression prior," in *IJCAI*, 2016, pp. 2690–2696.
- [43] H. Xu, W. Wu, S. Nemati, and H. Zha, "Patient flow prediction via discriminative learning of mutually-correcting processes," *IEEE TKDE*, vol. 29, no. 1, pp. 157–171, 2016.
- [44] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda, "Deep mixture point processes: Spatio-temporal event prediction with rich contextual information," in *SIGKDD*, 2019, pp. 373–383.
- [45] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE TNNLS*, vol. 20, no. 1, pp. 61–80, 2009.
- [46] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3837–3845.
- [47] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.
- [48] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.
- [49] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.
- [50] Q. Yin, S. Wu, and L. Wang, "Multiview clustering via unified and view-specific embeddings learning," *IEEE TNNLS*, vol. 29, no. 11, pp. 5541–5553, 2018.
- [51] S. Tulsiani, A. A. Efros, and J. Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *CVPR*, 2018, pp. 2897–2905.
- [52] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *WWW*, 2015, pp. 278–288.
- [53] I. Palomares, F. Browne, and P. Davis, "Multi-view fuzzy information fusion in collaborative filtering recommender systems: Application to the urban resilience domain," *Data Knowl. Eng.*, vol. 113, pp. 64–80, 2018.
- [54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [55] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.
- [56] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*. ACM, 2011, pp. 1082–1090.
- [57] W. Zhang, J. Wang, and W. Feng, "Combining latent factor model with location features for event-based group recommendation," in *SIGKDD*, 2013, pp. 910–918.
- [58] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *SIGKDD*, 2008, pp. 426–434.
- [59] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *TheWebConf*, 2017, pp. 173–182.
- [60] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *AAAI*, 2016, pp. 2786–2792.
- [61] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring patient similarities via a deep architecture with medical concept embedding," in *ICDM*, 2016, pp. 749–758.
- [62] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.



Wei Zhang received his PHD degree in computer science and technology from Tsinghua university, Beijing, China, in 2016. He is currently an associate researcher in the School of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests mainly include data mining and machine learning applications, especially for user generated data modeling. He has published more than 30 papers at some leading international conferences and journals. He served as an area chair for ICDM, a PC member for some leading international conferences, such as SIGKDD, SIGIR, WWW, and an invited reviewer for top journals like IEEE TKDE, IEEE TNNLS, ACM TKDD, etc.



Xin Lai received the B.Sc. degree in computer science and technology from Xiangtan University, Xiangtan, China, in 2018. He is currently pursuing the master degree with the Department of Computer Science and Technology, East China Normal University, Shanghai. His main research area is sequential user behavior modeling.



Jianyong Wang received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 1999. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He was an assistant professor with Peking University, and visited Simon Fraser University, the University of Illinois at Urbana-Champaign, and the University of Minnesota at Twin Cities before joining Tsinghua University in December 2004.

His research interests mainly include data mining and Web information management. He has co-authored more than 60 papers in some leading international conferences and some top international journals. He is serving or has served as a PC member for some leading international conferences, such as SIGKDD, VLDB, ICDE, WWW, and an associate editor of the IEEE Transactions on Knowledge and Data Engineering and the ACM Transactions on Knowledge Discovery from Data. He is a fellow of the IEEE.