User Identity Linkage via Co-Attentive Neural Network from Heterogeneous Mobility Data

Jie Feng, Yong Li, *Senior Member, IEEE*, Zeyu Yang, Mingyang Zhang, Huandong Wang, Han Cao, Depeng Jin, *Member, IEEE*

Abstract—Online services are playing critical roles in almost all aspects of users' life. Users usually have multiple online identities (IDs) in different online services. In order to fuse the separated user data in multiple services for better business intelligence, it is critical for service providers to link online IDs belonging to the same user. On the other hand, the popularity of mobile networks and GPS-equipped smart devices have provided a generic way to link IDs, *i.e.*, utilizing the *mobility traces* of IDs. However, linking IDs based on their mobility traces has been a challenging problem due to the highly heterogeneous, incomplete and noisy mobility data across services. In this paper, we propose *DPLink*, an end-to-end deep learning based framework, to complete the user identity linkage task for heterogeneous mobility data collected from different services with different properties. *DPLink* is made up by a *feature extractor* including a location encoder and a trajectory encoder to extract representative features from trajectory and a *comparator* to compare and decide whether to link two trajectories as the same user. Particularly, we propose a pre-training strategy with a simple task to train the *DPLink* model to overcome the training difficulties introduced by the highly heterogeneous nature of different source mobility data. Besides, we introduce a multi-modal embedding network and a co-attention mechanism in *DPLink* to deal with the low-quality problem of mobility data. By conducting extensive experiments on two real-life ground-truth mobility datasets with eight baselines, we demonstrate that *DPLink* outperforms the state-of-the-art solutions by more than 15% in terms of hit-precision. Moreover, it is expandable to add external geographical context data and works stably with heterogeneous noisy mobility traces.

Index Terms—deep learning; mobility trajectory; user identity linkage

1 Introduction

Smartphones and other mobile devices have made it easy for users to access various online services nearly everywhere and at any time. It is very common for a user to have multiple online identifiers (IDs) in different services such as online social networks (OSN), e-commerce services, online games, etc. Service providers have strong motivations to massively mine user data for monetization and optimizing user experience [1]. To capture a more comprehensive understanding of user behavior, it is increasingly intriguing to link user IDs across multiple services to fuse the separated data [2], [3].

To these ends, linking online IDs plays a critical role in data fusion for better business intelligence. Early research has explored different ways to link user IDs by using service-specific data such as user profile attributes [4] and social graphs [5]. However, these approaches depend on whether these services have the same data type. For example, e-commerce services often do not have social graphs to match with an online social network. Moreover, users may fill in fake information (*e.g.*, name, gender) in their profiles, which makes the linkage even harder.

In this paper, we explore a more generic approach to link user IDs by leveraging the spatial-temporal locality of user activities. The key intuition is that no matter what online services a user accesses, we can bind them to the user's *physical presence*, which is characterized by time and location. This becomes possible because most online services today have a mobile version with the location as parts of the service (*e.g.*, Uber, Yelp, Twitter). Besides, with some tolerance on granularity, even network accessing related information can be translated into location [6]. Figure 1 presents an intuitive example of linking IDs with mobility trajectories. Our goal is to link multiple online IDs that belong to the same users across different services. Despite the inspiring prospects and results of user identity linkage, several key challenges remain to be solved:

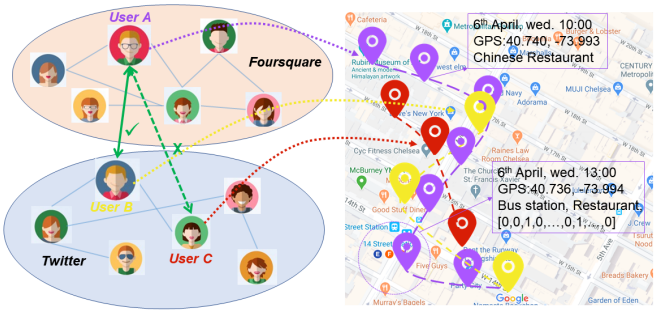


Fig. 1. An example of linking user accounts with mobility trajectories. We want to link user A (purple) from Foursquare with user B (yellow) and C (red) from Twitter. Based on the left trajectory of three accounts, we find that the trajectory of user B (yellow) is much closer to the trajectory of user A (purple). Thus, we can link the user A account with the user B account successfully.

- *J. Feng, Y. Li, Z. Yang, H. Wang, H. Cao and D. Jin are with Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. Email: liyong07@tsinghua.edu.cn. M. Zhang is with the department of Computer Science and Engineering, Hong Kong University of Science and Technology.*

Heterogeneity nature of mobility data: Due to the different usage behavior of users and various collection mechanisms, the properties of mobility data (*e.g.*, sample rate, time periods) are drastically different across services. For example, the mobility traces collected by an Internet Service Provider (ISP) are over 4 times denser than those collected by an online social network. Early works simply assume mobility traces of different services have similar a sample rate [7] or time period [8], which are sensitive to the heterogeneous mobility data and do not perform well in practice. Due to this challenge, the general trajectory similarity [9], [10] algorithm designed for the trajectory from the same data source also fails to model the correlated relationship between the different data sources and can not be effectively applied in the linkage problem.

Poor quality of mobility data: The data quality of collected mobility data is not always so good. *On the one hand*, the collected data only records the time and location information of mobility which is not enough to mine the hidden semantics of it. *On the other hand*, due to the limitation of devices and other artificial reasons [11], [12], the collected data usually contains noisy records to generate significant spatial and temporal mismatches between trajectories from different services. Because of the limitation of algorithms and the lack of proper data, existing approaches [8], [13] ignore the hidden semantics of mobility trajectory. Although, some works [14] propose prior knowledge-based solutions to address the mismatch problem. These solutions require proper manual parameter settings and are difficult to be applied in reality.

In this paper, we propose *DPLink*, an end-to-end deep learning based framework, to achieve linking IDs belonging to the same user for mobility data collected from different services with heterogeneous nature. *DPLink* consists of two main components: a *feature extractor module* and a *comparator module*. The *feature extractor module* is designed to extract vector features from input raw trajectory and model relations between the trajectories from the different data sources. The following *comparator module*, implemented as a multilayer feed-forward network, is aimed to yield the final similarity score of the extracted representative trajectory feature vectors.

As the core component of *DPLink*, *feature extractor* contains two encoders: *location encoder* and *trajectory encoder*. The *location encoder* is designed to integrate multi-dimensional input and extract the low-level feature of isolated locations. The multi-modal embedding based design in location encoder makes *DPLink* expandable to other available geographical features like PoI context. The following *trajectory encoder* is designed to capture the transitional relations of a single trajectory itself and model the correlation between two different trajectories. In the *trajectory encoder*, the transitional relations of a single trajectory are captured by a *sequential encoder*. Then a *selector* is introduced at the end of *trajectory encoder* to force the model to focus on the discriminative parts of trajectory and model the correlations between two different trajectories. With the help of this attention-based selector, *DPLink* not only can observe the similar parts of two trajectories but also works robustly with noisy and missing trajectory.

Besides, we propose a pre-training mechanism to address the training challenges introduced by the heterogeneity nature of mobility data. Following our proposed training mechanism, our model is first trained to complete a warm-up task as linking trajectory from one mobility data but with different time periods. Due to the regularity of human mobility [15], [16] and the consistency and high quality of used mobility data, this single data based linkage task is much easier for *DPLink* to complete than the linkage task on cross-domain datasets with different quality. Intuitively, this single data based linkage task acts as a simple *auxiliary task* to help the model to first learn about the basic knowledge of the trajectory space. Then, the pre-trained network with prior knowledge of the physical world and trajectory data is trained to complete the final *target task*: user identity linkage task on different data collected from different services.

Our contributions can be summarized as follows:

- We are the first to use deep learning techniques in the user identity linkage problem based on the heterogeneous mobility traces collected from different services with different quality. Our model does not assume any property of mobility data and works with heterogeneous mobility data across services.
- We utilize a recurrent network with pooling unit as a trajectory encoder to extract transitional features from each single trajectory and introduce co-attention based selector to capture potential correlations between trajectories from two different data sources. With fusing these two features, our model obtains a comprehensive high-level understanding of trajectory.
- We propose a simple yet effective pre-training mechanism to adapt to the heterogeneity nature of different source mobility data. And the proposed training mechanism greatly improves the performance and robustness of our deep learning model on the user identity linkage task on the different data sources.
- We perform experiments on three real-life mobility datasets and compare the performance of our model with nine baselines. Extensive results demonstrate that *DPLink* outperforms state-of-the-art solutions by more than 15%. Moreover, our model succeeds in utilizing additional geographical context data to further improve the performance and works robustly with noisy data.

Compared with the conference version [17], we extend this work from three aspects. First of all, we propose an enhanced version of the original model by adding a similarity matcher that directly considers the spatial-temporal similarity from the embedding space. The similarity matcher not only helps to improve the effectiveness of our model but also makes it insensitive and robust to the specific network structure including recurrent network type and attention mechanism. We also give a more detailed discussion and illustration of the model design, which will help readers better understand the intuition and design of our model. Secondly, we add a new public data to evaluate the performance of our model and a new state-of-the-art baseline with two new widely used metrics. The extensive results guarantee the effectiveness and reproducibility of our methods in the user identity linkage task. Finally, we extend the expansion capability and stability experiment and also add a hyper-parameter study experiment on three

datasets. The results give more insights on the effectiveness and limitation of the proposed model, which sheds light for the future direction.

2 Problem Formulation

TABLE 1
A list of commonly used notations.

Notat.	Description
$\mathcal{A}$	The set of all online IDs
$\boldsymbol{S}$	The set of types (platform) for online IDs
$\mathcal{A}^s$	The set of online account IDs on platform s
$\mathcal{T}$	The set of all time slots
t	time stamp
$\mathcal{L}$	The set of all regions
l	location indicator
$\mathcal{E}$	The set of PoIs
e	semantic information
$p = (l, t, e)$	A location record
x	The embedded location record
h	The hidden state of recurrent network
u, v	online user ID
$\mathcal{R}(u)$	location records sequence for online ID u
$r(u)$	trajectory slice for online ID u
$I(u, v)$	Binary variable indicating whether ID u and v belong to the same users
N	The number of linking candidates for online ID u

Let $\mathcal{A}$ represents the set of online account IDs, and S represents the set of different online platforms. Then, $\forall s \in S$, $\mathcal{A}^s$ denotes the set of online account IDs on platform s. As shown in Figure 1, for each user, we define its trajectory as a sequence of tuple $p = (l, t, e)$, which represents a location record in location l with semantic label e at time t. For example, in Figure 1, l is $(40.470, -73.993)$, t is $6, April, Wed., 10:00$, and e is the Chinese restaurant. Note that locations and times may be recorded at a different granularity and levels of precision in different source mobility data (*e.g.*, GPS coordinates or nearest base station in location record). Thus, without loss of generality, locations and times are divided into bins corresponding to geographical regions (*e.g.*, street block) and intervals of time (*e.g.*, one hour). Here, the semantic label e is defined as the distribution of points of interest (PoIs). As the bottom dashed circle in Figure 1 shows, beyond the nearest PoI, for each location l we can calculate its nearby PoI distribution in a certain space as the semantic label, *e.g.*, a multi-hot vector $[0, 0, 1, 0, ..., 0, 1, 0...]$, where the first 1 means l is close to a bus station, the second 1 means it is close to a restaurant, and the left 0 means that other types of PoIs are not existed around location l. We further define $\mathcal{T}$ as the set of all time bins, $\mathcal{L}$ as the set of all locations, and $\mathcal{E}$ as the set of PoIs.

Given any online ID $u \in \mathcal{A}$, we define its location records as, $\mathcal{R}(u) = \{p_1, p_2, ..., p_n\}$, where n denotes the number of location records. Taking the continuity of mobility data into consideration, we further partition the location records into meaningful trajectories $\mathcal{R}(u) = \{r_1(u), r_2(u), ...\}$ with maximum time window T_w (*e.g.*, 1 day). Given a pair of online IDs $u \in \mathcal{A}^1$ and $v \in \mathcal{A}^2$, let a binary variable $I(u, v)$ indicates whether these two IDs belong to the same user when the trajectories of them $r(u), r(v)$ are known,

$$I(u, v) = \begin{cases} 1, & u,v \text{ belong to the same user.} \\ 0, & u,v \text{ belong to different user.} \end{cases}$$

Further, given a target ID u, a list of candidates IDs $v_1, v_2, ..., v_N \subseteq \mathcal{A}^2$ and their trajectories $r(u)$ and $r(v_i)$ for $i = 1, 2, ..., N$, we aim to build a function, which is approximate to the identity function I enough, to find the best matching trajectory $r(v_i)$ and ID v_i.

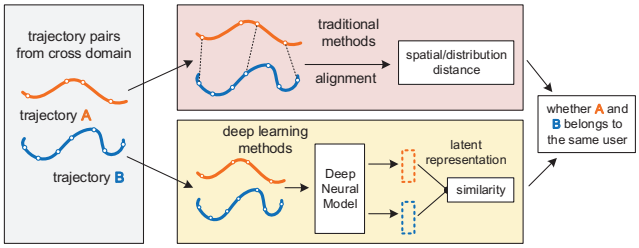


Fig. 2. Different working mechanisms of traditional methods and our proposed model.

Following the aforementioned definitions, many researchers [8], [13], [14] proposed insightful algorithms to work out this problem. However, because of the data quality problem and heterogeneity of mobility trajectory, these methods are still far from application in reality. Inspired by the powerful representation ability of deep learning models, we propose a deep learning based framework to address these challenges and aim to achieve better performance on real-life mobility datasets. The difference in workflows between existing works and our work is shown in Figure 2. In the traditional workflow of existing works, they usually first align the trajectory from the spatial or probabilistic view and then calculate the spatial or distribution distance to obtain a score. However, instead of aligning trajectory directly, we aim to utilize deep learning tools to first obtain the latent representation of mobility trajectory and then calculate the similarity between these two vectors as the similarity of trajectories. The details of our proposed model can be found in the next section.

3 Model and Method

The structure of *DPLink* is presented in Figure 3. *DPLink* contains three major components: *location encoder*, *trajectory encoder*, and *comparator network*. The input trajectory pairs are first processed by *location encoder* and then fed into *trajectory encoder* to extract multi-level representative features. Particularly, *DPLink* uses an co-attention based *selector* at the end of *trajectory encoder* to focus on the similar parts of the trajectory pair and avoid the potentially harmful influence of missing and noisy data. Finally, the generated vector pair, which represents the original trajectory pair, are fed into the *comparator network* to calculate the similarity score of two input trajectories.

3.1 Location Encoder

Location encoder is a multi-modal embedding module, which is designed to reorganize and embed the spatial and temporal features of a trajectory point $p_i = (t_i, l_i, e_i)$ into a single vector x_i. As introduced before, e_i can be defined as the nearest PoI as a one-hot vector or as the multiple distributions as the multi-hot vector. Here, we

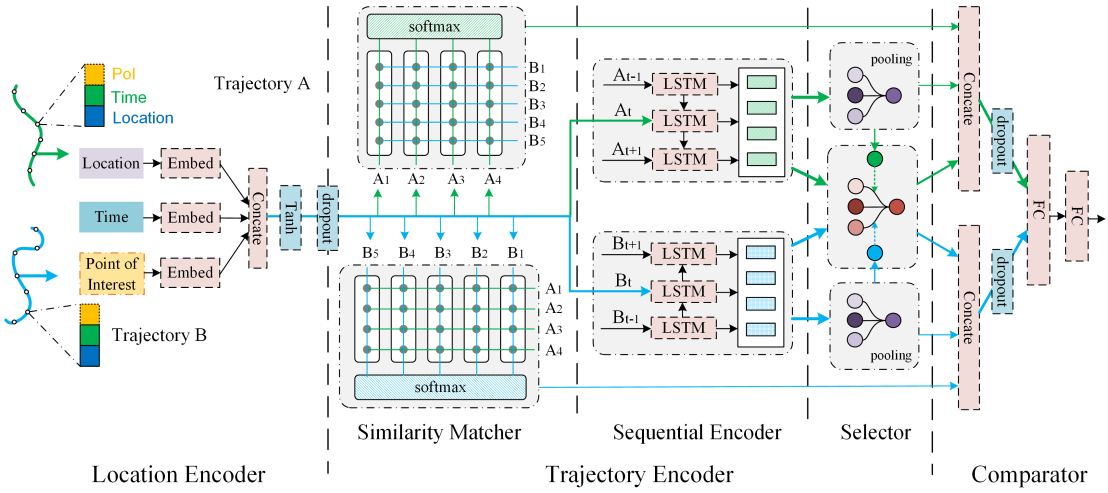


Fig. 3. The main architecture of *DPLink* consisting of three major components: location encoder, trajectory encoder (including recurrent encoder and attention-based selector), and comparator network.

choose the multi-hot vector of PoI distribution as the default settings for the convenience of the experiment and we also compare different kinds of semantic label representations in Section 4.4. As the distribution of nearby PoIs, e_i not only represents the potential semantic motivation of the mobility but also provides us another fuzzy localization method [18] to reduce the harmful effects of noisy and inaccurate GPS records in the raw data. We design three sparse linear embedding layers to encode each type of input (*e.g.*, one-hot) into a dense vector representation. Then, we concatenate them together to obtain an ensemble vector x_i. To strengthen the modeling ability of the embedding module, we add a *tanh* function as the final non-linear activation function. The formulation of the embedding module is as follows,

$$x_i = tanh(W_p p_i + b_p) = tanh([W_t t_i + b_t; W_l l_i + b_l; W_e e_i + b_e]),$$

where W and b denote the learnable parameters of embedding layers, *tanh* denotes the non-linear activation function, $[;;]$ denotes the concatenate function.

As mentioned above, the location fed into our model is in the form of a one-hot vector. Compared with the original geographic coordinates, the one-hot vector loses the information of spatial dependencies, which means that regions close to each other in the physical world can be far away in the one-hot space. However, the spatial dependency is important in measuring trajectory similarity. To enable *DPLink* to learn about the spatial adjacency of location, we introduce the mobility prediction task to help our model learn the meaning of closeness of location. This task is to predict the next location of an object by knowing its trajectory history. Due to the regularity of mobility, the location points of the trajectory which neared in the time dimension is also neared in the spatial dimension. In other words, the location encoder is enforced by the mobility prediction task to embed the adjacent location in the physical world into the adjacent space in the latent high-dimensional space in a neural network.

It is noted that this multi-modal embedding module is shared by the network to simultaneously process two input trajectories. This sharing mechanism guarantees that two types of trajectories from the same geographical space can be projected into another same latent space. Besides, the shared embedding module also greatly reduces the parameters of the whole network. Due to the high-dimensional characteristic of the embedding module, we consider additional regularization techniques to prevent the overfitting of it. Inspired by the successful application in the language modelling, we apply embedding dropout [19] in our location encoder. When p_e is the drop rate of the embedding module, the remaining non-dropped location embeddings are re-scaled by $\frac{1}{1-p_e}$. Besides, we also add a standard dropout layer after the embedding concate layer.

3.2 Trajectory Encoder

Following the *location encoder* is the *trajectory encoder* for multi-level trajectory feature extracting, which is the core component of our model. It consists of a *similarity matcher* to model the similarity between two trajectories in the isolated location level, a *sequential encoder* to extract the transitional level feature of a single trajectory, and a *selector* to extract correlated level features of trajectory pair.

3.2.1 Similarity Matcher

Following the embedded trajectory representation, we first build the similarity matcher module to extract location level features from the trajectory for matching. As shown in Figure 3, the similarity matcher module is made up of a correlation calculator (appears twice in the framework from two views) and two specific soft-max functions. Firstly, we calculate the correlation value between two input trajectories. For example, we have one embedded trajectory $A = \{A_i\}, i = 1, 2, ..., n, A_i \in \mathbb{R}^{k*1}$ and another embedded trajectory $B = \{B_i\}, i = 1, 2, ..., m, B_i \in \mathbb{R}^{k*1}$, where k is the embedding size. After calculating the correlation, we get a correlation matrix $C \in \mathbb{R}^{n*m}$, where each row in it represents the "correlation" between a specific point of trajectory A and the whole trajectory B and each column represents the "correlation" between a specific point of trajectory B and the whole trajectory A. Secondly, we calculate the mean value of each row to construct a n-dimensional vector, which

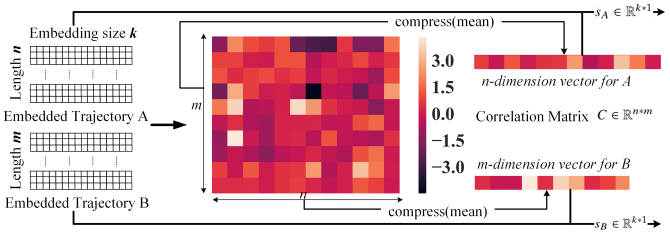


Fig. 4. Illustration of the similarity matcher with an actual correlation matrix.

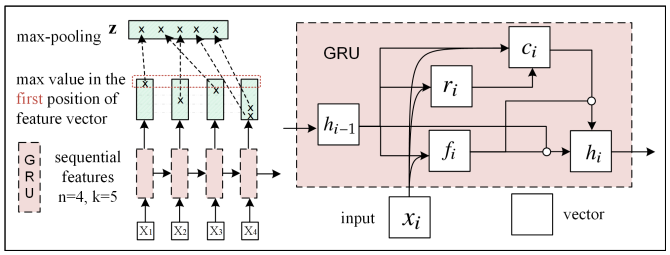


Fig. 5. Details of the sequential encoder in the *DPLink* model.

represents the similarity of each point in the trajectory A to the whole trajectory B. Finally, we apply a soft-max function upon this vector to obtain a normalized distribution and sum up the embedded trajectory A with this normalized vector as weights to obtain the isolated location level representation($s_A \in \mathbb{R}^{k*1}$) of trajectory A. Similarly, we reduce the correlation matrix from the column view by calculating the mean value of each column to obtain a m dimensional vector and normalize it by another soft-max function to obtain the isolated location level representation($s_B \in \mathbb{R}^{k*1}$) of trajectory B. These two low-level trajectory representation will be fused in the final comparator network to construct the full view representation of the trajectory. The detailed design of the similarity matcher is presented in Figure 4.

3.2.2 Sequential Encoder

Paralleled with the former similarity matcher, we build a sequential encoder to extract medium level features of trajectory from the transition view, which is shown in Figure 5. Based on the embedded vector representation $\{x_1, x_2, ..., x_n\}$ of original trajectory $\{p_1, p_2, ..., p_n\}$, sequential encoder captures the sequential transitions and model the mobility pattern of a single trajectory. The output encoded trajectory feature is recorded as vector representations $\{h_1, h_2, ..., h_n\}$. The recurrent encoder is made up of a recurrent neural network with max/mean pooling operation. The recurrent neural network is a standard class of neural networks, which is designed with cycle and internal memory to model the sequential information. We use the widely used long short term memory (LSTM) and its popular variation gated recurrent unit (GRU) as the basic unit of our recurrent encoder. The formulations of GRU are as follows

$$
\begin{aligned}
f_i &= \sigma(W_{fx}x_i + W_{fh}h_{i-1} + b_f),\\
r_i &= \sigma(W_{rx}x_i + W_{rh}h_{i-1} + b_r),\\
c_i &= tanh(W_{cx}x_i + r_t * (W_{ch}h_{i-1}) + b_c),\\
h_i &= (1 - f_i) * c_i + f_i * h_{i-1},
\end{aligned}
$$

where x_i is the input in i time slot, h_{i-1} is the last output of GRU unit, multiple matrix W are different gate parameters, multiple vectors b are the bias vectors for different part, $*$ means element-wise multiplication, f_i is the update weight, r_i is the reset gates, c_i is the update state and h_i is the output state.

Since the number of features extracted from the recurrent network is still identical to the length of trajectory, we introduce pooling operation (*e.g.*, max-pooling and mean-pooling) after recurrent network to obtain fix-length vector representation of trajectory. For n feature vectors with k-dimension, the max-pooling operation is to select the maximum value in each poison of the k-dimension vector from n instances and then generate a k-dimension vector. Mean-pooling means to calculate an average vector of n feature vectors. This operation is beyond only feature reshaping but also achieves a certain range of performance improvement in our experiment. This is because the pooling operation acts as a simple self-feature filter to select important features. As Figure 5 shows, with the help of recurrent network with max-pooling, we model the sequential relations $\{h_1, h_2, ..., h_n\}$ and finally obtain a single feature vector z for each trajectory. And we call this single feature vector z as the *primary representation vector* for trajectory.

3.2.3 Co-Attention based Selector

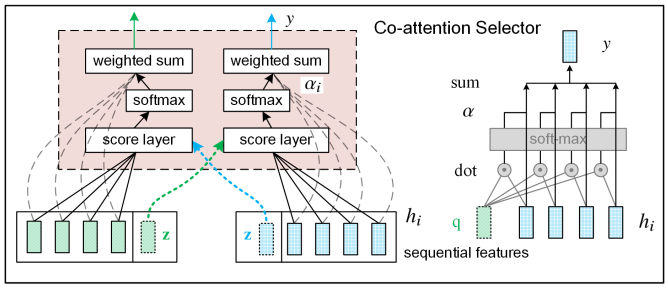


Fig. 6. Co-attention based selector with "dot" based correlation weights.

To handle the data quality problem and enable our model to focus on the critical parts of the trajectory pair for matching and linking, we propose to design a co-attention based selector network, which is presented in Figure 6. The extracted features in this selector are also regarded as the high-level feature of two trajectories. We first briefly introduce the background knowledge of attention mechanisms.

Given a query vector q and a series of candidate vectors $\{h_1, h_2, ..., h_n\}$, attention mechanism can be implemented with two steps: 1) to calculate the "correlation" between the query vector q and all these candidate vectors $\{h_1, h_2, ..., h_n\}$; 2) with these normalized "correlation" as weights $\{\alpha_1, \alpha_2, ..., \alpha_n\}$, to calculate the weighted sum y of candidate vectors as the comprehensive representation of them. This weighted sum y is regarded as the summary of the most related parts of candidates $\{h_1, h_2, ..., h_n\}$ for the query q. There are three widely used attention methods: *dot*, *general*, *mlp*. The main difference between these attention

implementations is the calculation of "correlation". The formulation of above attention methods are as follows,

$$y = \sum \alpha_i h_i, \quad \alpha_i = \sigma(f(q, h_i)),$$
$$f_{dot}(q, h_i) = h_i^T q,$$
$$f_{gen}(q, h_i) = h_i^T W q,$$
$$f_{mlp}(q, h_i) = v^T tanh(Wq + Uh_i),$$

where W, U, v are the learnable parameters, f represents the score function, σ is the soft-max function, h_i represents i_{th} candidate vector, q is the query vector and y is the final output.

Figure 6 presents the core idea of the co-attention based selector. In the common application of attention mechanisms like neural machine translation, there is only one candidate sequence, where the query vector q of attention is naturally constructed from candidates $\{h_1, h_2, ..., h_n\}$, *e.g.*, $q = h_n$. Different from the former, we introduce a co-attention mechanism to adapt the general attention network for pair input to capture and model the correlation relationship. For the candidates $\{h_1^A, h_2^A, ..., h_n^A\}$ from trajectory A, we use the *primary representation vector* z_B of trajectory B as the query vector $q_A = z_B$. Meanwhile, we use the *primary representation vector* z_A of trajectory A as the query vector $q_B = z_A$ for the candidates $\{h_1^B, h_2^B, ..., h_m^B\}$ from trajectory B. In this way, we directly connect two trajectories before the final comparator network and give them opportunities to find the related parts of each other. Further, this co-attention based selector will reduce the harmful effects of potential noisy records. After the processing of the co-attention network, we obtain the final feature vector of trajectory: y_A for original trajectory A $\{p_1^A, p_2^A, ..., p_n^A\}$ and y_B for original trajectory B $\{p_1^B, p_2^B, ..., p_m^B\}$.

3.3 Comparator Network

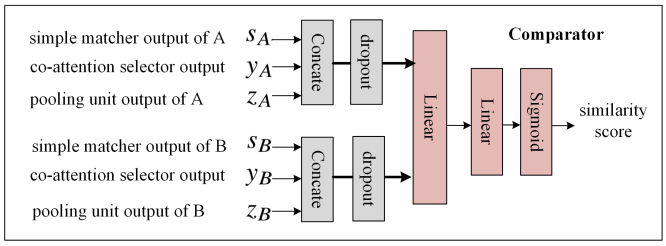


Fig. 7. Illustration of the final comparator network.

The final component of our model is a comparator network, which is shown in Figure 7. It is implemented as a multilayer feed-forward network. The final layer of the comparator network is a neural unit with a *sigmoid* function acting as a logistic regression function to generate the final similarity score. The feature vectors (s_A, y_A, z_A) and (s_B, y_B, z_B) are independently fused to obtain the final vector presentation of each trajectory. To prevent the risk of overfitting, we apply a standard dropout operation after the fusion layer. Then these two vector representations are fed into a multi-layer feed-forward network to yield the similarity *score*. From the view of the problem, the comparator network gives two input trajectories a second chance to exchange information and validate each other. From the view of the network, the comparator network can be regarded as a simple but effective classifier for binary classification to judge whether two input vectors belong to the same class. Due to the *sigmoid* function, the output score is a normalized probability that can be optimized by the binary cross-entropy loss function.

3.4 Training strategy

Our model works in an end-to-end manner without requiring handcrafting features. Since we translate the user identity task into a binary classification problem, we choose the binary cross-entropy loss as the objective loss function.

$$Loss = -\sum_{i=1}^{n} y_i \log x_i + (1 - y_i) \log(1 - x_i),$$

In the training, the Adaptive Moment Estimation (Adam) algorithm is utilized to optimize the model. Several widely used tricks, such as dropout, L2 regularization, and learning rate schedule, are used to avoid the overfitting problem. The detailed settings for these parameters can refer to Section 4.1.4 and Table 4.

However, due to the heterogeneous nature of mobility trajectories from the different services and platforms, our model always fails to converge and stops with really poor performance when directly trained from scratch. The intuition that mobility trajectories from different platforms share identical underlying spatial-temporal patterns is the precondition for completing the user identity linkage task. Nevertheless, due to the heterogeneous nature of mobility trajectories from different data sources, our model struggles to explore the huge spaces and learn the true knowledge of it. On the one hand, different platforms provide users with various services and users generate different mobility behaviors with various intentions. On the other hand, the sample rate and recording mechanism of collecting data on different platforms are also different. All of these lead to the heterogeneous nature of mobility trajectories, which make it difficult for our model to learn valuable knowledge. Consequently, our original model fails to link trajectories across different mobility data.

Here, we propose to pre-train the whole model with a simple task as a warm-up to first obtain the basic knowledge of the physical world and trajectory pattern. Once our model achieves good performance on this simple task, we start to train it with the final user identity linkage task on

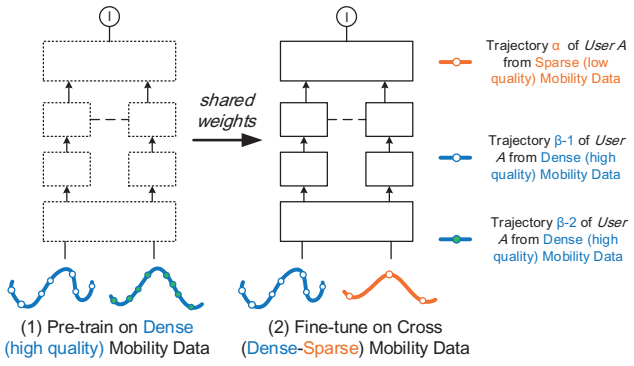


Fig. 8. Two-step training strategy for user identity linkage task.

TABLE 2
Statistics of collected trajectory datasets.

Dataset	Users	Records	Locations	Rec./U.	Loc./U.	Duration
Twitter-F	1228	53337	8975	43	25	21 months
Foursquare	2970	44915	8975	15	12	48 months
ISP	2844	325215	12576	114	19	1 week
Weibo	1761	49651	12576	28	5	1 week
Instagram	2505	428492	59634	171	37	35 months
Twitter-I	1721	447972	46310	260	40	51 months

different mobility data sources. This warm-up task is to identify whether two mobility trajectories from the same platform in different periods (*e.g.*, different days) belong to the same user. This warm-up task is proposed based on the observation from the regularity of human mobility [15], [16], people usually go to the same places (*e.g.*, to and from where he needs to go like home and office) and generate similar even the same trajectories in different workdays. Furthermore, the trajectories used in the warm-up task are both from the identical high-quality mobility dataset, which is denser and without too many missing records. Due to the regularity of mobility and the high-quality mobility dataset, the warm-up task becomes an easier and directional task to help the model to converge and learn the useful prior knowledge for the future difficult task.

The two-step training strategy is presented in Figure 8. Our training strategy can be regarded as a kind of network-based transfer learning method. The prior knowledge learned from the first linkage task on a single dataset is transferred to the final linkage task (*target task*) on different datasets by sharing partial network weights.

4 Performance Evaluation

In this section, we conduct extensive experiments on three real-world datasets with nine baselines to evaluate the performance of our model on user identity linkage task.

4.1 Experimental Settings

4.1.1 Datasets

We carry out experiments with three real-world cross-domain mobility datasets. The first dataset is the mobile network record dataset provided by one of the largest major Internet Service Providers (ISP) in China and the social network location service records obtained from Weibo. The left two datasets are the location-based social network datasets from Foursquare, Twitter and Instagram. Table 2 summarizes the basic statistical information of datasets. Below, we describe these datasets in detail.

ISP-Weibo. The ISP dataset contains 325215 mobility records that cover the metropolitan area of Shanghai from April 19 to April 26 in 2016. The location records are generated in the base station level when users access cellular networks via mobile devices for communication and the Internet. Each trajectory is characterized by an anonymized user ID. It contains a series of spatial-temporal points produced by the user, where each point includes a base station ID and a timestamp. We replace the base station ID with a certain longitude-latitude coordinate according to the base station information provided by ISP. The Weibo dataset is generated from the ISP dataset. Our collaborators from ISP collected the Weibo sessions from the ISP datasets in the same time window with the permission of Weibo. In the Weibo dataset, each mobile trace is characterized by a Weibo ID and contains a series of GPS coordinates that show up in HTTP sessions between the mobile application and the Weibo server. These GPS coordinates are produced when users access the location service in their mobile applications like check-in. Before we access the dataset, the collaborators have mapped the Weibo ID into the same anonymized user ID in the ISP dataset to protect personal privacy.

Foursquare-Twitter [20]. This dataset contains the trajectories data from Foursquare, a popular location-based social network (LBSN), and Twitter, a micro-blogs social network around the world. As its primary function, Foursquare provides users various location-related services like location check-ins and posting online reviews. Twitter also provides users with basic location-based services like location check-ins. Provided by Zhang et al. [20], the data is crawled from the web pages of thousands of users who use both platforms with at least one location records during November 2012 in the east coast of U.S. Based on the crawled raw data, the trajectory is also characterized as a user's account ID and a series of GPS coordinates with timestamp.

Instagram-Twitter [8]. This dataset contains public check-in data from Instagram and Twitter. Instagram is a popular photo-sharing application and service, where users can share pictures and videos with location information through mobile, desktop, laptop, and tablet. Based on the public data on the Instagram, [8] extracts their pubic twitter account and obtains the shared check-ins. After filtering users [8], [21], the final public dataset contains 1717 users with 337934 Instagram records and 447366 Twitter records, where a record is stored as a tuple (user ID, latitude, longitude, time-stamp). The details of how to obtain this data can refer to [8], [21]. As the basic statistics in Table 2 show, the trajectory quality of this data is much better than the former two datasets, which is made up of densely and balanced records from two platforms.

PoI dataset. We crawled 0.75 million points of interest (PoI) of Shanghai from BaiduMap as the additional geographical context of the ISP-Weibo dataset. As Table 3 shows, the crawled PoI dataset contains 20 categories and can be classified into 7 region functions. For every base station in the ISP dataset, we calculate the distribution of PoIs for it in the surrounding 1-kilometer area. This PoIs distribution serves as a soft function label for the region and describes the potential intention of the user's movement. Besides, we also collect PoI data [22] for the Foursquare-Twitter dataset to support the semantic experiments in Section 4.4. When the Instagram-Twitter dataset is distributed around the world, we cannot collect enough PoI data for it and we also think that only considering the spatial effect is enough for it to link different accounts.

It's noted that all the user IDs in our datasets are anonymized to protect user's privacy. Meanwhile, we store the data in a secure local server and only core researchers can access the data with strict non-disclosure agreements.

TABLE 3
PoIs category distribution on the ISP-Weibo dataset.

Related Function	PoI Categories
Residence	residence, life services.
Entertainment	food, hotel, gym, shopping, leisure.
Business	finance, office building, company, trading area.
Industry	factory, industrial estate, economic development zone.
Education	school, campus.
Scenery spot	scenery spot, tourism development.
Suburb	villages, towns.

4.1.2 Baselines

We compare the performance of our model with nine state-of-the-art baselines, including seven classic user identity linkage algorithms and two deep learning based trajectory representation models. The detailed introduction of seven classic methods is as follows.

NFLX: With knowing some external information, Narayanan et al. [23] propose a statistical model based mobility trace similarity score to identify the users in the Netflix dataset. NFLX cannot be directly applied to a cross-domain linkage problem and we follow the method [8] to adapt it to our problem.

MSQ: Ma et al. [7] incorporate general knowledge in forms of global movement constraints and preferences to identify users from the dataset. Specifically, they consider the negative square difference between two mobility traces as their similarity score.

HIST: Naini et al. [24] focus on linking users by matching the location histograms of their mobility traces. Firstly, they compute a user's visiting frequency of each location and then define a similarity score of two histograms based on Kullback-Leibler divergence.

LRCF: Goga et al. [25] take the popularity of different locations into consideration. LRCF applies the term frequency-inverse document frequency(TF-IDF) [26] weighting scheme to location visiting histograms and measures mobility trace similarity using a cosine distance.

WYCI: Rossi et al. [13] propose a time based probabilistic user identifying algorithm. They use the frequency of user login in different locations to approximate the probability of visiting these locations. Then they determine if a mobility trace belongs to a user by computing the user's probability to produce the mobility trace.

POIS: POIS [8] algorithm uses the "encountering" events to measure the similarity of two different-domain mobility traces. It assigns every "encountering" event a weight based on a statistical model and uses the weighted sum as the similarity measure.

GKR-KDE: Chen et al. [21] proposed a kernel density estimation based method, which characterizes the spatial pattern of an individual's check-in activities and then performs user account linkage based on their spatial patterns. Further, GKR-KDE utilized grid-based KDE for computational efficiency and entropy-based weight scheme for negative coincidence.

As we are the first deep learning based model for the user identity linkage task on different mobility datasets, we compare our model with two deep learning based models for trajectory representation.

TULER: Gao et al. [27] utilize the recurrent network to encode the trajectory into a single vector to identify users in single mobility data. TULER cannot be directly applied to our problem, thus we train two independent TULER models for each mobility data and check whether these two models can identify each input trajectory as the same user.

t2vec: Li et al. [9] propose to adapt the seq2seq model with spatial proximity aware loss function to infer and represent the underlying route information of a trajectory for efficient trajectory similarity computation. Following the philosophy of the original paper, we adapt it to our problem by using a sparse (low sampling) trajectory from one dataset as input and using dense (high sampling) trajectory with the same user ID from another dataset as the underlying route to learn the representation of trajectory. Then, we use this representation to link different IDs.

For seven classic methods, six of them are from the pubic implementation [1] and GKR-KDE is from the official implementation [2] from the author of original paper. We use the parameters recommended in the public implementation and adapt them based on the requirement of different datasets. For two deep learning based methods, we use the official implementation of TULER [3] and t2vec [4] from the authors from the original papers. We follow the default parameter settings from the official repository and fine-tune them based on the performance on different datasets.

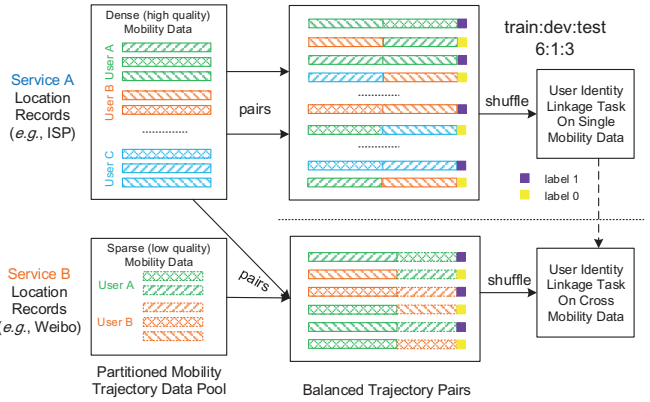


Fig. 9. Build training/validation/testing dataset.

4.1.3 Preprocessing

Figure 9 presents the details of how to build training/validation/testing dataset in our experiment. As Figure 9 shows, the first step is to divide the lengthy location records into continuous mobility trajectories. After this operation, we obtain two independent mobility trajectory pools, where each trajectory is labeled with its user ID, for two different mobility data. As mentioned in the training section, we first pre-train the model with user identity linkage task

1. https://github.com/whd14/De-anonymization-of-Mobility-Trajectories
2. https://www.dropbox.com/s/mrkfao8gr8zktkw/ICDE-18-Chenwei.zip?dl=0
3. https://github.com/gcooq/TUL
4. https://github.com/boathit/t2vec

TABLE 4
Typical parameter settings of our model *DPLink*.

Item	Value	Features	input	output
batch size	32	location	10000	200
learning rate	1e-3	time	24	10
drop out	0.3	PoI	20	10
fine-tune lr	0.0005	hidden state	200	200

on single data as a warm-up and then fine-tune the model with user identity linkage task on cross-domain mobility data. Furthermore, we choose high-quality mobility data with more frequent and longer records, which is called *dense* mobility trajectory data to complete the warm-up linkage task. Another mobility data source with fewer mobility records called *sparse* mobility data is only used in target user linkage task on cross-domain datasets. The mechanism of building data in two training stages are the same, while the main difference is to choose which data to use.

We first introduce the core step of building experiment data for the user identity linkage task on single domain data as follows. We randomly choose two trajectories from the dense mobility data pool with identical owner labels to build positive trajectory pair. The basic intuition behind this operation is the regularity of human mobility [15], [16], which means that the mobility trajectory of one user keeps similar in the different periods with extremely high probability. To meet the sample balance requirement of binary classification problem in the training stage, for every positive trajectory pair we randomly select two trajectories from dense mobility data pool with different user IDs to form a negative trajectory pair. In the training stage and validation stage, only this one negative sample is enough.

However, in reality, we usually have lots of online ID candidates for matching, and we need to compare them together to find the best matching. To simulate this practical case, in the testing stage, we choose N negative candidates for matching. Particularly, for each online ID, we choose those left online IDs whose trajectory has at least one intersection with it as candidates. Based on the statistics of our data, the average number of these candidates for each online ID is around 20. To meet the requirement of batch size for efficient computation, we select 32 as the default setting for N in the experiment. To build experimental data for the user identity linkage task from different data sources (*target task*), we follow the similar steps mentioned before. Different from choosing two input trajectories from the same mobility dataset for the warm-up linkage task, we choose one trajectory from dense mobility data and another trajectory from sparse mobility data for the final user identity linkage task. In our experiment, the proportion of training, validation and testing data is 6:1:3. It's noted that we divide not only the trajectory but also the users by shuffling the trajectory data with the user label. This operation makes sure the generalization of our experiment, where some users may only have few data in the training step and even they only appear in the testing step.

4.1.4 Metrics and Parameter Settings

Our model is implemented on the Pytorch platform and the typical parameter settings of our model are presented in Table 4. Our main code is publicly available [5]. In the experiments, we select GRU as the default recurrent unit and *dot* attention as the default attention mechanism for efficient computation. The results for LSTM and other basic components are similar. Following the aforementioned evaluation dataset, our model is trained and validated in a binary classification manner and finally evaluated in the search and ranking manner.

In the training step, we use the F1 score and AUC to measure the performance of our model, which are both widely used metrics in classification problems. F1 score is calculated as $\frac{2*Precision*Recall}{Precision+Recall}$. AUC, known as the area under the receiver operating characteristic (ROC) curve, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. With P positive instances and Q negative instances, AUC can be calculated by the following formulation,

$$AUC = \frac{\sum_{i \in +} rank_i - P(1+P)/2}{P \times Q}.$$

In the validation step, we also use the F1 score and AUC as the metrics to cooperate with the early stopping mechanism to find the best parameters and avoid overfitting.

In the testing step, we use three widely used metrics in the ranking scene to evaluate the performance of algorithms in the user identity linkage task with more than one (*e.g.*, $N = 32$, as mentioned in the preprocessing section) candidate online IDs. After algorithm deciding the score for each candidate, we rank these candidates by score and calculate the hit-precision@k, MRR@k and nDCG@k. It's noted that the default metric in the paper is hit-precision@5, which can also be aliased as hit@5. Hit-precision of top-k candidates is defined as follows,

$$h(x) = \begin{cases} \frac{k-x}{k}, & \text{if } k > x >= 0 \\ 0, & \text{if } x >= k \end{cases}$$

Mean Reciprocal Rank (MRR) is a statistical measure to evaluate the performance of systems that return a ranked list of answers to queries. For a single query, reciprocal rank is calculated by $\frac{1}{rank}$, where rank is the position of the correct answer in the result. Normalized Discounted Cumulative Gain (nDCG) is often used to measure the effectiveness of ranking in information retrieval. nDCG is the extended version of hit-precision by assigning higher scores to the hits at higher positions in the ranking list. Their formulation is as follows,

$$MRR@k = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\text{rank}_i}, nDCG@k = \frac{\sum_{i=1}^{k}\frac{2^{rel_i}-1}{log_2(i+1)}}{\sum_{i=1}^{N}\frac{2^{rel_i}-1}{log_2(i+1)}}$$

In our experiments, for all the metrics, higher values indicate better performance.

4.2 Overall performance comparison.

We evaluate our model with nine baselines on three datasets to present the performance on user identity linkage task by three widely used ranking metrics. The results are presented in Table 5. The following analysis are based the results on

5. https://github.com/vonfeng/DPLink

TABLE 5
The performance of our model and nine state-of-the-art baselines on three real-life datasets, where higher results are better. **Bold** fronts denotes the best(highest) results and underline denotes the second best results.

Datasets	ISP-Weibo				Foursquare-Twitter				Instagram-Twitter			
Methods	hit@5	hit@10	nDCG@10	MRR@10	hit@5	hit@10	nDCG@10	MRR@10	hit@5	hit@10	nDCG@10	MRR@10
HIST	0.245	0.278	0.251	0.228	0.166	0.182	0.157	0.168	0.321	0.325	0.319	0.316
MSQ	0.268	0.354	0.315	0.257	0.170	0.226	0.157	0.202	0.165	0.245	0.200	0.148
POI	0.317	0.357	0.328	0.301	0.165	0.246	0.211	0.155	0.203	0.307	0.263	0.195
LRCF	0.380	0.450	0.402	0.357	0.157	0.178	0.145	0.158	0.321	0.324	0.318	0.315
NFLX	0.380	0.454	0.404	0.357	0.197	0.222	0.174	0.193	0.353	0.357	0.350	0.346
GKR-KDE	0.289	0.355	0.320	0.276	0.226	0.289	0.276	0.229	0.421	0.438	0.434	0.418
WYCI	0.432	0.506	0.460	0.407	0.307	0.351	0.288	0.322	0.402	0.456	0.403	0.394
t2vec	0.334	0.421	0.376	0.389	0.189	0.270	0.237	0.183	0.433	0.494	0.498	0.491
TULER	0.409	0.476	0.437	0.392	0.324	0.429	0.364	0.297	**0.815**	**0.907**	0.797	0.737
DPLink	**0.499**	**0.591**	**0.535**	**0.535**	**0.449**	**0.483**	**0.451**	**0.426**	0.794	0.857	**0.818**	**0.751**

TABLE 6
Performance comparison on the ISP-Weibo and Instagram-Twitter datasets after tuning the similarity matcher (denoted as 'SM').

Datasets	Methods	hit@5	hit@10	MRR@10	nDCG@10
ISP-Weibo	DPLink	0.499	0.591	0.535	0.535
	DPLink+SM	**0.508**	**0.594**	**0.541**	**0.543**
Ins-Twitter	DPLink	0.794	0.857	**0.818**	0.751
	DPLink+SM	**0.812**	0.857	0.786	**0.816**

hit@5, and the similar results can also be concluded from the other two metrics including MRR and nDCG.

We first analyze the results of the ISP-Weibo dataset. Among seven classic linkage baselines, the WYCI algorithm with the tolerance of different sample rates performs best with 0.432 hit@5. Besides, the performance of NFLX and LRCF with the tolerance of mismatch from the spatial or temporal view are also similar well. Particularly, the hit-precision of our model is 0.499, which is 15% higher than the best baseline WYCI. Besides, we compare our model *DPLink* with other two state-of-the-art trajectory representation algorithms: TULER and t2vec. The results demonstrate that our model outperforms both of them more than 17%. TULER is designed for directly identifying the user from a single mobility dataset. It fails to obtain better performance due to two potential reasons: 1) TULER cannot identify any new users because the user set is decided in the training step. However, in our experiments, the users in the testing step are not identical to the training step; 2) each TULER only focus on its single mobility dataset and ignores the potential correlation between two different mobility dataset. Based on the seq2seq model, t2vec is proposed to infer and represent the underlying route information of a trajectory for efficient trajectory similarity computation. In practice, t2vec needs an extremely high-quality dataset (*e.g.*, taxi trajectory dataset with less than 1 minute sampling period) to learn good enough representative function. It fails to handle the poor quality (low sampling and missing records) and the heterogeneity nature of our cross-domain mobility dataset. *DPLink* is designed for the user identity linkage task and succeeds in dealing with these challenges with special network design and training strategies. On the one hand, the location encoder and trajectory encoder help *DPLink* extract representative features from a single trajectory. On the other hand, the attention-based selector enables *DPLink* to capture the correlation between two trajectories with different properties from different mobility data. Finally, the MLP based comparator network acts as a powerful classifier to obtain the final result. Based on *DPLink*, we design a similarity matcher as an additional component to help our model extract the capture the trajectory similarity from the isolated location view, whose performance on the ISP-Weibo dataset is presented in Table 6. Here, we choose to fine-tune the similarity matcher component on the best *DPLink*. As Table 6 shows, the performance of our whole model has improved again on two datasets. Besides, while the fine-tuning of our model in the small Foursquare-Twitter dataset is easy to be over-fitted, we do not report their results here.

Similar results can also be found in the Foursquare-Twitter dataset. Because of the smaller volume and sparse nature of the Foursquare-Twitter dataset, the performance of all the methods is lower. However, due to the powerful representation ability of neural network and transfer learning based training strategy, our model still achieves about 0.4 hit-precision. Compared with the state-of-the-art methods, the performance gain of our model can be up to 25%. The result of the Instagram-Twitter dataset is a little bit different. First, we can observe that the performance of all methods on it is much better than on the other two datasets. This is due to that the visited locations in this data are spread in multiple cities with a huge spatial range. Thus, the visited locations of different users are quite different, which leads to higher performance in the table. While TULER can achieve competitive results even a little bit better performance with hit-precision, our model still outperforms all the baselines in MRR@10 and nDCG@10 metrics.

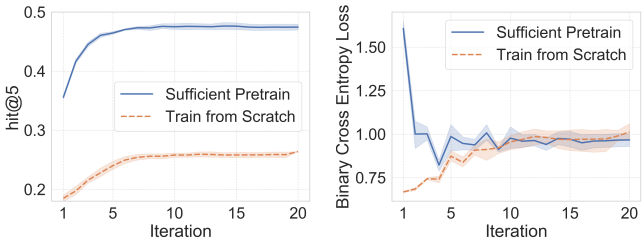
In summary, some baselines can achieve competitive results with our model in easy settings (*e.g.*, Instagram-Twitter dataset with dispersed location distribution in multiple cities), but our model can perform much better in difficult settings (*e.g.*, ISP-Weibo dataset in a single city). The extensive evaluation results of three real-life datasets demonstrate the superiority of our model than other the-state-of-the-art methods on user identity linkage task.

4.3 The effects of the pre-training strategy.

In this section, we conduct experiments to demonstrate the effect of the pre-training strategy for *DPLink*on heterogeneous data. As introduced before, based on the statistic

TABLE 7
Performance of different training mechanisms on the ISP-Weibo dataset. For example, "Full−Location Encoder" denotes that the embedding weights of location encoder are not shared in the pre-train step, which means the location encoder is trained from scratch in user identity linkage task.

Training Strategy	hit@5 (mean+std)	Δ
Train from Scratch	0.257 ± 0.025	-48.3%
Full Pretrain	0.497 ± 0.013	0
Full-Location Encoder	0.404 ± 0.018	-18.7%
Full-Recurrent Encoder	0.438 ± 0.005	-11.9%
Full-Selector	0.395 ± 0.014	-20.5%
Full-Comparator	0.499 ± 0.009	+0.4%



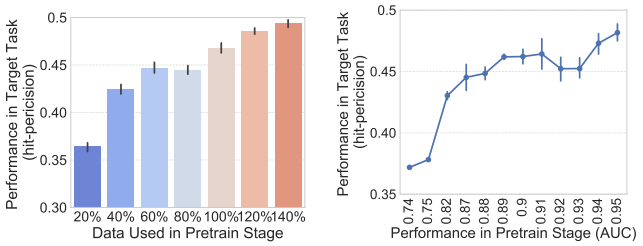
(a) Testing performance curves. (b) Testing loss curves.

Fig. 10. The comparison results between two types of training mechanisms on the ISP-Weibo dataset.

information in Table 2, we find that the ISP-Weibo dataset is more heterogeneous when the records from ISP are over 4 times denser than the records from Weibo. Meanwhile, the data in the other two datasets are not so heterogeneous. Thus, the pre-training testing experiment is only conducted on the ISP-Weibo dataset, whose results are presented in Table 7, and Figure 10-11.

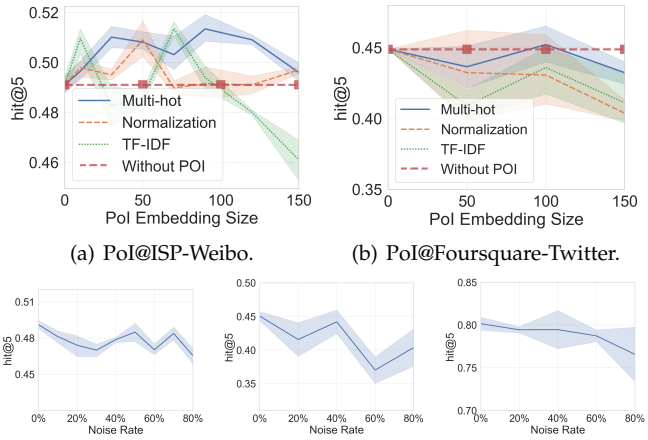
Table 7 shows the performance of our model with different training strategies on the ISP-Weibo dataset. The pre-train step is crucial for better performance. Furthermore, we choose to remove different components of *DPLink* from the sharing weights step to locate the most important component for the performance. As Table 7 shows, removing location embedding and selector weights from the sharing step leads to the worst performance, which denotes the crucial role of location encoder and attention-based selector in modeling. Besides, the effect of removing the recurrent encoder is smaller and removing the comparator network weights even does not harm the performance at all. These results tell us that transitional relations and classification knowledge are not hard to learn from beginning in this task.

We dive into the training process to understand the importance of the pre-train step. As Figure 10(a) shows, after pre-training, the initial performance of our model before fine-tuning on target task can be higher than 0.35, which is better than the best performance of model trained from the scratch. With fine-tuning on the target task, the hit-precision of our model can be further improved to 0.49. Figure 10(b) presents the variation trend of testing loss of models with different training mechanism. Due to the heterogeneity of mobility data, the model trained from scratch becomes overfitted at the very start. Meanwhile, the pre-trained model



(a) The effect of data volume used in pre-train stage. (b) The effect of the performance in pre-train stage.

Fig. 11. The effects of different pre-train parameters on ISP-Weibo dataset.



(a) PoI@ISP-Weibo. (b) PoI@Foursquare-Twitter.

(c) Noise@ISP-Weibo. (d) Noise@Twitter-Foursquare. (e) Noise@Instagram-Twitter.

Fig. 12. The effects of data quality (geographical context and noisy records) on our model. The first two figures present the effects of semantic context (PoI) with different embedding sizes on the ISP-Weibo and the Foursquare-Twitter dataset. The last three figures present the performance of our model on three datasets with different noise input.

learns the knowledge about the task and performs better and better.

Figure 11 presents the effects of data volume and model performance of the pre-train step on the performance of our model on target task. As Figure 11(a) shows, with more and more data utilized on the single domain linkage task in the pre-train step, the final performance of our model on the target task also becomes better and better. This observation tells us that enough unlabeled data on a single domain can help to improve the performance of our model on the target task and reduce the requirement of labeled cross-domain data. Figure 11(b) shows us that better pre-trained model on a single domain linkage task produces better a fine-tune model on the target task.

4.4 Expansion capability and stability study.

In this section, we evaluate whether *DPLink* can utilize external information with the help of a multi-modal embedding based location encoder to improve the performance on user identity linkage task and whether it works robustly with noisy records. The results are presented in Figure 12-14.

To evaluate the expansion capability of multi-modal embedding network, we build three kinds of PoI distribution

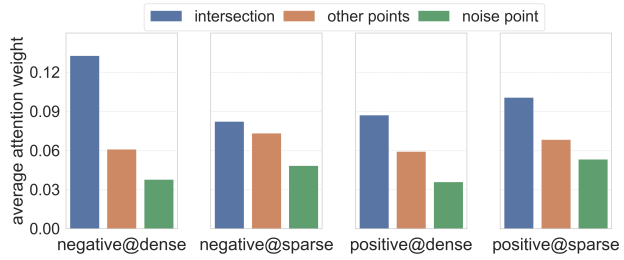
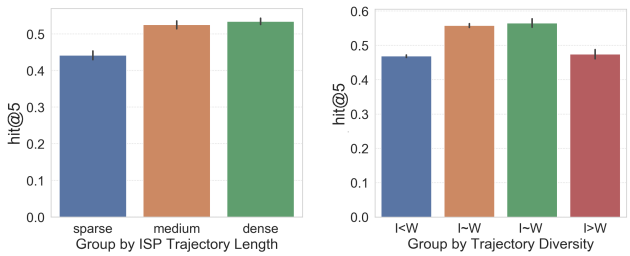


Fig. 13. Average attention wights of three types of points on the testing set of the ISP-Weibo dataset. For example, "negative@dense" means that the attention weights are from the dense(high quality) mobility trajectory with a negative label.



(a) The effects of sample rate(trajectory length) of ISP trajectory.

(b) The effects of the diversity of the sample rate(trajectory length) between trajectories.

Fig. 14. The effects of different trajectory characteristics on the ISP-Weibo dataset. "sparse" denotes short trajectory length and low sample rate in the ISP trajectory. "I<W" means the length of the ISP trajectory is shorter than the length of the Weibo trajectory, while "I⌣W" denotes the length of the ISP trajectory is close to the Weibo trajectory.

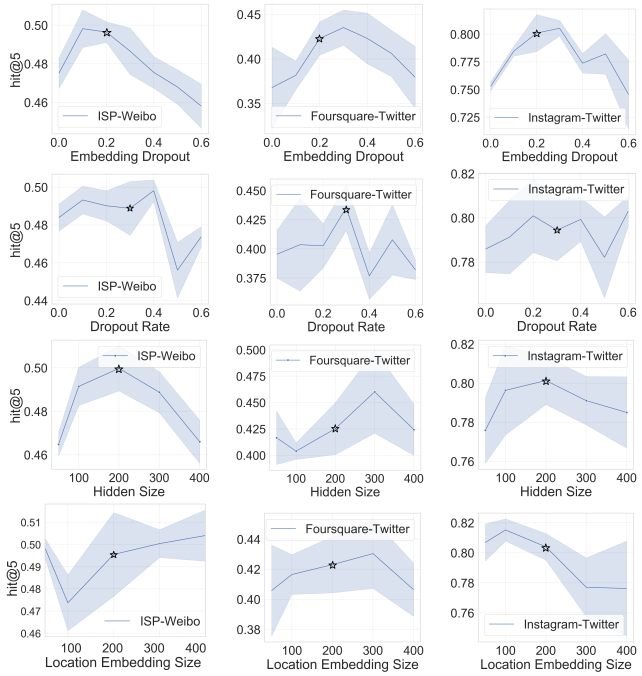


Fig. 15. Hyper-parameter study of *DPLink* model on three datasets, where ⋆ denotes the default parameter value.

features: 1) multi-hot distribution, which only describes whether specific type PoI exists in the region; 2) normalized probability distribution of different PoIs in the region; 3) tf-idf distribution, the enhanced version of normalized probability distribution. The evaluation results are presented in Figure 12(a) and Figure 12(b). The results show that *DPLink* is able to apply external PoI information to further improve the performance from 0.49 to more than 0.51 on the ISP-Weibo dataset with enough PoI data. While the results on the Foursquare-Twitter dataset is a little bit worse, we think this is due to the low quality of the collected PoI data for it whose volume and coverage are limited. In general, we find that the multi-hot feature of PoI distribution performs much better and stable.

Figure 12(c)-12(e) present the performance variations of *DPLink* with noisy trajectory input on three datasets. In these figures, the noise rate is 20% means that 20% trajectories in the data are randomly selected to randomly replace one real data point in it with a noisy point. As Figure 12(c)-12(e) shows, the performance of our model keeps stable with the increase of noise rate on three datasets, which demonstrates the robustness of *DPLink*. Furthermore, we visualize the attention weights of three kinds of points in the trajectory to verify whether our co-attention based selector plays an important role in filtering out the harmful effects of noisy records. We only present the results on the ISP-Weibo dataset, the results on the other two datasets are similar. Three types of points are 1) intersection location between two trajectories; 2) noisy point inserted by the program; 3) other location points. In Figure 13, we analyze the average attention weights of three types points in 4 scenarios which include the dense/sparse (high/low quality) mobility data with positive/negative labels. As Figure 13 shows, the average attention weight for intersection point is the highest and the weight for the noisy point is the lowest, which distinctly demonstrates the selecting value of our selector.

We investigate the effects of data quality (sample rate) for the performance of our model on the ISP-Weibo dataset in Figure 14. As Figure 14(a) shows, the performance of our model in the difficult cases (sparse trajectory with low sample rate) is similar to the performance(0.42) of the best baseline on the whole dataset. With the augment of trajectory sample rate, our model can achieve better and better linkage performance. Figure 14(b) presents the similar results of the performance gain of our model from another view: the diversity of sample rate (trajectory length) between trajectories from two platforms. As Figure 14(b) shows, when the sample rate of trajectories from two platforms is similar(I⌣W), the performance of our model will be better. For those unbalanced samples, the performance of our model will reduce but is still better than the baselines.

In summary, our model with specific designs achieves promising performance over baselines on various scenarios, including considering semantic information, noisy input and different data quality. The extensive results demonstrate the effectiveness of our model: not only the whole framework but also each specifically designed component.

4.5 Hyper-parameters study of *DPLink* model.

In this section, we test the influence of several critical hyper-parameters in the model on the user identity linkage task. Besides, we evaluate the stability of the proposed models. The results are presented in Figure 15 and Figure 16.

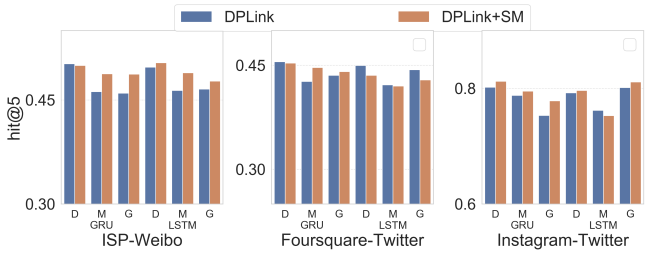


Fig. 16. The effects of different combinations of recurrent network (LSTM and GRU) and attention mechanisms (D(dot), M(mlp), G(general)) on original model *DPLink* and enhanced model *DPLink*+SM, where 'SM' denotes the similarity matcher.

Following the successful practice in language modeling [19], we apply embedding dropout in the location encoder, which is similar to perform variational dropout on the connection between the one-hot input and the embedding lookup. As the figures in the first row of Figure 15 show, suitable embedding dropout provides our model about 4% performance gain and the embedding dropout rate is expected to be small to 0.1 based on the practice. The results of the dropout in the comparator are presented in the second row of Figure 15, compared with the embedding dropout, the influence of dropout in the comparator is smaller, 0.2 is a good initialized value for it. As the figures in the third row of Figure 15 show, the size of hidden vector in the recurrent layer has an important impact for the final performance of the model. According to our experiment, 200 $\sim$ 300 is the sufficient range for the hidden size. In the experiment, we also find that one layer GRU achieves the best performance while more layers with dropout can obtain similar results. The figures in the final row of Figure 15 present the results of different location embedding size. Compared with the hidden size, the effects of location embedding size is smaller.

In Figure 16, we test whether the different combinations of recurrent networks and attention mechanisms significantly influence the final performance on three datasets. All the results are obtained by at least 5 independent experiments with a different random seed. According to Figure 16, we find that both GRU and LSTM can achieve promising performance and dot-based attention performs better among three kinds of attention mechanisms. From the results, we find that the performance of the original version of *DPLink* fluctuate a lot for different attention mechanisms, *e.g.*, the performance of GRU-mlp is about 0.43 which is much lower than the best performance on the ISP-Weibo dataset. To reduce this performance variation introduced by recurrent units and attention mechanism, we design a similarity matcher in the trajectory encoder as supplement. Similarity matcher provides another independent view for modelling the correlation between trajectories, which are parallelized the original relation modelling path including the recurrent encoder and co-attention selector. In this way, the similarity matcher can be regarded as a powerful supplement component for the correlation modelling between trajectories and will play an important role when the performance of correlation modelling is influenced by the selection of recurrent units and attention mechanisms. As Figure 16 shows, the enhanced model *DPLink+SM* achieves more stable performance on different combinations of recurrent network and attention mechanism, *e.g.*, the performance of GRU-mlp is improved from about 0.43 to 0.47, which is close to the reported best performance on the ISP-Weibo dataset. We can find similar results on the other two datasets. In summary, with introducing the independent correlation modelling component similarity matcher, the stability of our model with different combinations of recurrent units and attention mechanisms is successfully improved.

5 RELATED WORK

Applications of Identity Linkage: A number of applications can benefit from linking IDs across services. For example, Kumar et al. [1] investigated the user migration patterns across social media to provide guidance for online social network design. Zafarani et al. [3] and Yan et al. [2] leveraged linked IDs across social networks for better friend recommendations. Yang et al. [28] leveraged linked IDs across sites for better video recommendations. All these works indicate the strong motivations for the service provider to link IDs belonging to the same user. Besides, some researchers [29], [30] tried to protect users from identification attack by matching mobility trajectory.

Identity Linkage using Trajectory Data: A few recent works examine the possibility of linking IDs based on location data [7], [8], [13], [21], [23], [24], [31]–[37]. Mudhakar et al. [31] and Ji et al. [32], [38] focused on linking IDs based on users' graph/network structures. They adapted their algorithms to location trajectories by constructing a "contact graph" to model users encountering with each other. However, these algorithms still require using social network graphs, which are not available in our scenario. Besides, some algorithms are designed to tolerate data noise such as temporal mismatching [23] and spatial mismatching [7]. Wang et al. [14] proposed algorithms with tolerating spatial or temporal mismatches (or both) and modeling user behavior for better linkage performance. Other algorithms implemented de-anonymization attacks based on *individual user's* mobility patterns [13], [24], [33]. Finally, researchers also developed identity linkage algorithms based on "encountering" events [8], [21], [34], [35]. By considering the location context (*e.g.*, user population density), it achieved a better performance [8], [21]. However, none of them is able to capture the trajectory dynamics and integrate semantic information simultaneously.

Representation Learning for Trajectories: Recently, deep learning has been used for spatial-temporal data mining [39]–[41] such as next location prediction [40], [42] and trajectory embedding [9], [27], [39]. Yao et al. [39] used a recurrent network with manually features to cluster trajectory into several clusters. Gao et al. [27] proposed to identify users in one mobility dataset via a recurrent network encoder. However, the uniqueness of an individual in one dataset does not imply that this user can be easily recognized in another dataset. Besides, this method can not be applied to new users whose data has not been trained with the model. Based on one dense taxi trajectory dataset, Li et al. [9] used a seq2seq model to measure the similarity of sub-trajectories extracted from one dense trajectory, which requires high data quality and fails to model the relationship of trajectories from different datasets.

Due to the limitation of model design, all these existing methods are not suitable for the user identity linkage problem for different mobility data. Compared with these methods, our model is not only designed to extract comprehensive trajectory features but also to model the correlated relationship between trajectories. This interaction ability is based on the attention mechanism introduced by Bahdanau et al. [43] in the neural machine translation task. Feng et al. [40] is the first work to introduce the attention model to predict human mobility. However, instead of achieving mobility prediction, our focus is to measure the trajectory similarity from the different data sources to link online IDs.

6 Conclusion

In this paper, we investigated the task of user identity linkage by leveraging the power of deep learning. We proposed an end-to-end deep learning framework to link different accounts from heterogeneous mobility data. The proposed model employs location encoder and trajectory encoder to model the complicated single trajectory feature and apply co-attention based selector to focus on discriminative parts when matching two mobility trajectories. Extensive experiments on two real-life mobility datasets show that *DPLink* significantly outperforms nine baselines on the user identity linkage task. Compared with the existing solutions, the proposed model achieves a general similarity measurement for heterogeneous mobility data. Besides, it is robust to the noise of trajectory and is easy to extend to external information like PoI distribution. There are several future directions for our work. First, because of the limitation of the datasets, we only consider simple external geographical context data like the PoIs category. In the future, we plan to expand the multi-modal embedding module to process the raw textual information in the check-in data. Second, although the pre-training mechanism works well, it is still not easy to directly train a good model for sparse mobility data. Thus, design more stable network structure and better training mechanism is also an important direction.

7 Acknowledgments

This work was supported in part by The National Key Research and Development Program of China under grant SQ2018YFB180012, the National Nature Science Foundation of China under 61971267, 61972223, 61861136003, and 61621091, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] S. Kumar, R. Zafarani, and H. Liu, "Understanding user migration patterns in social media," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

[2] M. Yan, J. Sang, T. Mei, and C. Xu, "Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge," in *Proceedings of the International Conference on Multimedia and Expo (ICME)*, 2013.

[3] R. Zafarani and H. Liu, "Finding friends on a new site using minimum information," in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2014.

[4] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi, "On the reliability of profile matching across large online social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1799–1808.

[5] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377–388, 2014.

[6] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards ip geolocation using delay and topology measurements," in *Proceedings of the ACM SIGCOMM conference on Internet Measurement (IMC)*, 2006.

[7] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," *IEEE/ACM Transactions on Networking (TON)*, 2013.

[8] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *Proceedings of the 25th International Conference on World Wide Web (WWW)*, 2016, pp. 707–719.

[9] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei, "Deep representation learning for trajectory similarity computation," 2018.

[10] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *IEEE Seventh International Conference on Intelligent Computing and Information Systems*, 2016.

[11] G. Wang, S. Y. Schoenebeck, H. Zheng, and B. Y. Zhao, ""will check-in for badges": Understanding bias and misbehavior on location-based social networks," in *Proceedings of the International Conference on Web and Social Media (ICWSM)*, 2016.

[12] F. Xu, G. Zhang, Z. Chen, J. Huang, Y. Li, D. Yang, B. Y. Zhao, and F. Meng, "Understanding motivations behind inaccurate check-ins," *Proceedings of the ACM on Human-Computer Interaction (CSCW)*, 2018.

[13] L. Rossi and M. Musolesi, "It's the way you check-in: identifying users in location-based social networks," in *Proceedings of the second ACM Conference on Online Social Networks (COSN)*, 2014.

[14] H. Wang, C. Gao, Y. Li, G. Wang, D. Jin, and J. Sun, "De-anonymization of mobility trajectories: Dissecting the gaps between theory and practice," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[15] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[16] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, 2010.

[17] 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 *Proceedings of the 2019 World Wide Web Conference*, 2019.

[18] H. Cao, J. Feng, Y. Li, and V. Kostakos, "Uniqueness in the city: Urban morphology and location privacy," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, pp. 1–20, 2018.

[19] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *arXiv preprint arXiv:1708.02182*, 2017.

[20] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *WSDM*, 2014.

[21] W. Chen, H. Yin, W. Wang, L. Zhao, and X. Zhou, "Effective and efficient user account linkage across location based social networks," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 2018, pp. 1085–1096.

[22] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "Gmove: Group-level mobility modeling using geo-tagged social media," in *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[23] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2008.

[24] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli, "Where you are is who you are: User identification by matching statistics," *IEEE Transactions on Information Forensics and Security (TIFS)*, 2016.

[25] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *Proceedings of the 22nd international conference on World Wide Web (WWW)*, 2013.

[26] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[27] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.
[28] C. Yang, H. Yan, D. Yu, Y. Li, and D. M. Chiu, "Multi-site user behavior modeling and its application in video recommendation," in *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2017.
[29] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data," in *Proceedings of the 26th International Conference on World Wide Web (WWW)*, 2017, pp. 1241–1250.
[30] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity and t-closeness," *IEEE Transactions on Network and Service Management*, 2018.
[31] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012.
[32] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1040–1053.
[33] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2011.
[34] A. Cecaj, M. Mamei, and F. Zambonelli, "Re-identification and information fusion between anonymized cdr and social network data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 1, pp. 83–96, 2016.
[35] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu, "Automatic user identification method across heterogeneous mobility data sources," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 978–989.
[36] H. Wang, C. Gao, Y. Li, Z.-L. Zhang, and D. Jin, "From fingerprint to footprint: Revealing physical world privacy leakage by cyberspace cookie logs," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, 2017.
[37] H. Wang, Y. Li, G. Wang, and D. Jin, "You are how you move: Linking multiple user identities from massive mobility traces," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 189–197.
[38] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. A. Beyah, "On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge." in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2015.
[39] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.
[40] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web (WWW)*, 2018, pp. 1459–1468.
[41] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *AAAI*, 2019.
[42] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2016.
[43] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

8 BIOGRAPHY

Jie Feng is a Ph.D. candidate at Department of Electronics Engineering of Tsinghua University, advised by Prof. Yong Li. He reccived B.E. degree in Electrical Engineering from Tsinghua University in 2016. His research interest falls in the area of spatial-temporal data mining. He currently works on applying deep learning methods into spatial-temporal data mining filed to improve the performance of practical model in many challenging practical tasks like mobility prediction and flow forecasting.

Yong Li (M'09-SM'16) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member of the Department of Electronic Engineering, Tsinghua University.

Dr. Li has served as General Chair, TPC Chair, SPC/TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations more than 6900. Among them, ten are ESI Highly Cited Papers in Computer Science, and four receive conference Best Paper (run-up) Awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers, Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.

Mingyang Zhang received the B.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree with the department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include mobile big data, urban computing.

Zeyu Yang is an undergraduate student majoring in electronic engineering from Tsinghua University, Beijing, China. His research interests include mobile data mining and deep learning.

Huandong Wang is a Ph.D. candidate in the Department of Electronic Engineering of Tsinghua University, co-advised by Prof. Depeng Jin, Prof. Yong Li at Tsinghua University, and Prof. Gang Wang at Virginia Tech. He received my bachelor degree of Electronic Engineering from Tsinghua University in 2014. His researches mainly focus on mobile big data mining, social media analysis, and software-defined networks. He has participated in multiple joint research projects with Tencent, China Telecom, and Hitachi.

Han Cao is a junior in electronic engineering from Tsinghua University, Beijing, China. His research interests include data mining, machine learning and their applications.

Depeng Jin (M'2009) received his B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Now he is an associate professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.