



TULRN: Trajectory user linking on road networks

Yu Sang¹ · Zhenping Xie¹ · Wei Chen² · Lei Zhao²

Received: 27 September 2022 / Revised: 30 October 2022 / Accepted: 15 November 2022 /
Published online: 1 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Linking trajectories to users who generate them with deep learning techniques has been a popular research topic in recent years, due to the large-scale trajectory data obtained by ubiquitous GPS-enabled devices and the widespread applications served by the study, such as route planning, next location prediction, and destination prediction. To address the TUL (Trajectory User Linking) problem more effectively, we propose a novel semi-supervised model TULRN (Trajectory User Linking on Road Networks) based on GNN (Graph Neural Network) and BiLSTM (Bi-directional Long Short-Term Memory). The main difference between our study and existing ones is that the TUL problem is extended onto road networks in this work, where both the structure of road networks and the sequential characteristics of trajectories will be fully utilized in a unified manner. The reason behind the extension is that many trajectories are usually generated on road networks in real life, and based on which we can model the relationships between trajectories and users more precisely. Specifically, our proposed model TULRN contains four main components: (1) transforming each trajectory into a sequence of road segments and constructing a road network-aware trajectory sequence graph RTSG; (2) learning the representation of a node in RTSG with a weight-aware GNN module; (3) learning the representation of a trajectory with a BiLSTM-based module; (4) linking trajectories to users based on the embedding of each trajectory. The extensive experiments conducted on a real-world dataset demonstrate that the proposed model TULRN performs better than the state-of-the-art methods.

This article belongs to the Topical Collection: *Special Issue on Spatiotemporal Data Management and Analytics for Recommend*

Guest Editors: Shuo Shang, Xiangliang Zhang and Panos Kalnis

✉ Zhenping Xie
xiezp@jiangnan.edu.cn

Yu Sang
7213107001@stu.jiangnan.edu.cn

Wei Chen
robertchen@suda.edu.cn

Lei Zhao
zhaol@suda.edu.cn

¹ School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China

² School of Computer Science and Technology, Soochow University, Suzhou, China

Keywords Trajectory user linking · Representation learning · Road networks

1 Introduction

Benefiting from the ubiquitous GPS-enabled devices (e.g., mobile phones and on-board navigation system-equipped cars), people have witnessed the unprecedented increase of trajectory data in the past decade. The mobility patterns of moving individuals and groups can be found from the trajectory data, so as to provide methods and decision-making guidance for the research of location-based services, crowd management, tourism monitoring, and other related application fields. It has become a research hot point in the field of data mining and has attracted extensive attention from academia. Based on the large-scale trajectory data, there has been a lot of work, including trajectory similarity computation [1–6], activity trajectory search [7–10], road network-based path analysis [11–13], etc. Recently, some researchers pay attention to trajectory user linking (TUL), due to the significant benefits brought by it, such as making next-visit-location prediction [14–18] and friends recommendation on location-based social networks [19].

The above-mentioned study TUL aims to find a mapping between trajectories and their owners, and the work was firstly proposed in [20], where a Recurrent Neural Network (RNN) based semi-supervised learning model TULER was proposed. Following this, a novel approach called TULVAE was introduced in [21], where authors investigated trajectory user linking with variational autoencoder. Then a method called TGAN was proposed in [22], where the TUL problem was solved in an adversarial network. In recent advances, a novel model called GNNTUL composed of a graph neural network module and a classifier was applied to the TUL task [23]. Commonly, all of existing studies tackle the TUL problem with deep learning techniques, due to the excellent performance of these techniques in capturing human sequential and semantic information from a large number of trajectories ordered by timestamps. These studies have made great contributions in addressing the TUL problem, but none of them investigate the problem on road networks. Unfortunately, in real life, most of trajectories are generated on road networks by many different vehicles, such as buses, cars, and taxis.

As the skeleton of a city, the road network is the “lifeline” of the entire city and the carrier of urban social and economic activities and transportation. Comprehensive and accurate road network information is an important foundation for building a smart city. People walk through the urban road network every day, generating massive spatiotemporal trajectory data. The analysis and modeling of road network-based trajectory data not only bring an opportunity for understanding people’s movement patterns, but also provide a new perspective for urban planning, traffic management and forecasting, and public travel. Consequently, we extend the problem TUL onto road networks. Specifically, the reason for investigating the TUL problem on road networks is threefold. (1) Although many trajectories are collected from a road network in real life, there has been no work that links trajectories to users on it, and our study can fill this gap. (2) Existing studies need to develop methods to transform trajectories into segments before learning representations and the performance of segmentation is deeply affected by their designed approaches. Different from them, we can obtain the trajectory sequence (TS) introduced in Section 3 automatically, based on the given map information. (3) Apart from the sample points of a trajectory, given a road network we can employ more information, such as road segments and intersections to enhance the performance of trajectory representation learning.

Despite the significance of exploring trajectory user linking on road networks, the task turns out to be challenging due to the following problem, i.e., how to fully incorporate and utilize the topology information of a road network and the sequential characteristics of trajectories in an effective manner for better trajectory representation learning. To address the problem, we propose a novel semi-supervised model TULRN, which contains four main components. (1) Construction of a road network-aware trajectory sequence graph RTSG: we transform trajectories into a sequence of road segments after map matching based on the location information of trajectories, then construct this graph by merging the trajectory sequence with road network graph. (2) Node embedding: we propose a weight-aware GNN module to learn the representation of a node in RTSG. (3) Trajectory embedding: we develop a BiLSTM-based module to learn the representation of each trajectory. (4) Linking module: following the embedding of trajectory, we feed the output of part (3) into a fully-connected neural network to conduct multi-class classification with the softmax function and cross-entropy loss function. To sum up, we make the following contributions in this work.

- To the best of our knowledge, we are the first to extend the TUL problem onto road networks.
- We develop a novel model TULRN to address the problem TUL effectively, by fully utilizing the structure of a road network and sequential information of trajectories.
- We conduct extensive experiments on a real-world dataset, and the results demonstrate that our proposed model outperforms the state-of-art approaches.

The rest of the paper is organized as follows. We introduce the related work in Section 2, and formulate the problem in Section 3. The overview of TULRN is presented in Section 4, and the details of TULRN are introduced in Section 5. The experiments are conducted in Section 6 and the paper is concluded in Section 7.

2 Related work

We review related studies to our work in this section, which contain trajectory representation learning and trajectory user linking.

2.1 Trajectory representation learning

Embedding trajectory to low-dimension representation has been a popular research topic in spatio-temporal database, due to the wide range of applications of this study, such as trajectory clustering [24, 25], movement behavior analysis [26], travel time and destination prediction [27–30].

To detect trajectory clusters where within-cluster similarity occurs in different regions and periods, Yao et al. [24] used a sliding window to extract moving behaviors and employed a sequence-to-sequence auto-encoder to learn fixed-length deep representations. To compute trajectory similarity with low sampling rates and noisy points, a method called t2vec was proposed in [31]. A hierarchical reinforcement learning algorithm namely SeCTAR, which can be used for navigation and object manipulation, was introduced in [32]. The algorithm used a bottom-up approach to learn continuous representations for trajectories without explicit need for hand-specification or subgoal information. To define the similarity between two attribute-aware trajectories, an approach called MAEAT was developed by Boonchoo et al. [33]. MAEAT was built upon a sentence embedding algorithm and directly

learned the whole trajectory embedding via predicting the context aspect tokens. In recent work, Fu et al. [27] explored road networks for trajectory representation learning. Their proposed framework contains three main components: road network matching, road segment representation learning, and trajectory representation learning.

2.2 Trajectory user linking

The study of trajectory user linking (TUL) aims to find a mapping between given trajectories and users, and has been explored by some existing work.

The TUL problem was firstly introduced in [20], where authors proposed a RNN-based semi-supervised learning model TULER starting with trajectory segmentation and check-in embedding. Then, a softmax-based method was designed to link trajectories to their owners. Following this, Zhou et al. [21] proposed a semi-supervised learning model TULVAE, which learned the human mobility in a neural generative architecture with stochastic latent variables that span hidden states in RNN. Considering the sparsity of human trajectories, Miao et al. [34] proposed a novel model DeepTUL, which is composed of a feature representation layer and a recurrent network with attention mechanism, to solve the TUL task. DeepTUL not only combines multiple features that govern user mobility to model high-order and complex mobility patterns, but also learns from labeled historical trajectories to capture the multi-periodic nature of user mobility. Due to the insufficient data, Zhou et al. [22] introduced a Trajectory Distribution Approximation(TDA) problem and proposed the TGAN - a generative adversarial samples-based individual trajectory generation algorithm. As an approach to enable learning users' motion patterns and location distribution, TGAN aims to improve the performance of identifying human mobility. Due to the discovery that RNN-based models could not distinguish trajectories correctly when the trajectories are out of length [35]. A conception of trajectory semantic vector had been proposed in the TULAR, which focuses on selected parts of the source trajectories when linking. TULAR introduces the Trajectory Semantic Vector (TSV) via unsupervised location representation learning and RNN, by which to reckon the weight of parts of the source trajectory. In recent work, both users' personalized moving preferences and the prior knowledge behind human mobility were considered for more precisely linking in. Fan et al. [23] made it in an efficient model GNNTUL. It also addresses the human mobility discrimination problem by utilizing graph neural network to capture higher-order spatio-temporal information, as well as the implicit transition patterns between check-ins from the constructed graph.

Existing studies have made great contributions in terms of trajectory representation learning and trajectory user linking. However, linking trajectories to users on road networks has not been investigated. Consequently, we propose the model TULRN to address the issue in this work.

3 Preliminaries

In this section, we first present notations used throughout the paper in Table 1, then give definitions and formulate the problem TUL.

3.1 Problem definition

Definition 1 Road Network. A road network is represented as a directed graph $G = (V, E)$, where V is a set of nodes (i.e., intersections), and E is a set of edges (i.e., road segments).

Table 1 Definitions of notations

Notation	Definition
u	A user
U	A set of users
τ	A trajectory
p	A sample point in a trajectory
$TS(\tau)$	Trajectory sequence of τ
S	A set of trajectories
v	An intersection
\mathbf{v}	Representation of the intersection v
e	A road segment
$H(e)$	Renyi entropy of e
$w(e)$	Weight of e
\mathbf{e}_τ	Representation of the trajectory τ

Definition 2 Trajectory. Let $p = (lat, lng, t)$ be a sample point on a road network, where lat and lng represent the latitude and longitude respectively, t denotes the time-stamp, a trajectory is a sequence of sample points, denoted as $\tau = (p_1, p_2, \dots, p_n)$.

Definition 3 Trajectory Sequence. Given a trajectory $\tau = (p_1, p_2, \dots, p_n)$ collected from a road network, the trajectory sequence of τ is defined as $TS(\tau) = (v_k, e_i, \dots, e_j, v_l)$, where e_i is a road segment (i.e., an edge of G) and v_k is an intersection (i.e., a node of G).

Consider the example in Figure 1, there are four trajectories $\tau_1, \tau_2, \tau_3,$ and τ_4 on road networks, and the trajectory sequences of them are $TS(\tau_1) = (v_2, e_2, v_5, e_7, v_9, e_{10}, v_{10})$, $TS(\tau_2) = (v_6, e_5, v_5, e_4, v_4, e_3, v_3)$, $TS(\tau_3) = (v_7, e_8, v_8, e_9, v_9, e_{10}, v_{10})$, and $TS(\tau_4) = (v_{11}, e_{11}, v_8, e_6, v_4, e_3, v_3)$, respectively.

Problem formulation Given a road network G , a set of users $U = \{u_1, u_2, \dots, u_n\}$ and a set of trajectories $S = \{\tau_1, \tau_2, \dots, \tau_m\} (m \gg n)$ collected from G , our proposed semi-supervised model TULRN provides a mapping that will link each trajectory τ_i to a user u_j : $S \mapsto U$.

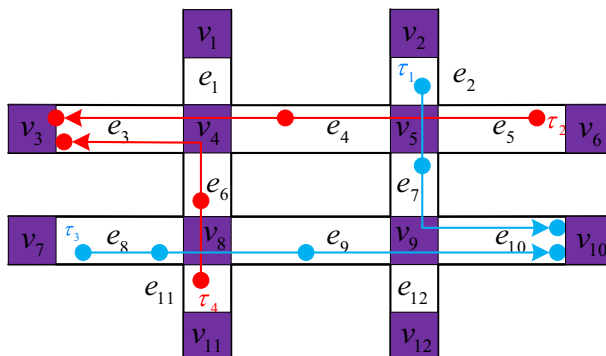


Figure 1 An example of trajectory sequence

4 Overview of TULRN

The overview of our proposed model TULRN is presented in Figure 2. Observed from which, the model contains the following four main components.

Component 1: We divide each trajectory τ into a sequence of road segments based on the given road network and location information of trajectories, and the corresponding sequence is denoted as $TS(\tau)$. Next, we construct a road network-aware trajectory sequence graph RTSG by merging the trajectory sequence and road network graph.

Component 2: To fully utilize the topology information of a road network, we develop a weight-aware GNN module to learn the presentation of a node in RTSG. Different from the naive GNN [36], we calculate the weight for each edge based on Renyi entropy before node sampling with the goal of enhancing the performance of node embedding.

Component 3: We develop a BiLSTM-based module to take full advantage of the sequential features involved in trajectories. The final output trajectory representation e_τ will be fed into the next module.

Component 4: A multi-class classification module, which is designed based on a fully-connect neural network and a softmax function, is trained to link each trajectory to its owner.

5 Proposed model TULRN

As presented in Figure 2, we will transform each trajectory into a sequence of road segments firstly. Before the transformation, we need to conduct road network matching, i.e., align sample points of a trajectory onto the road network [27]. To achieve the matching effectively, in this work, we adopt a state-of-the-art model [37], which is developed based on the

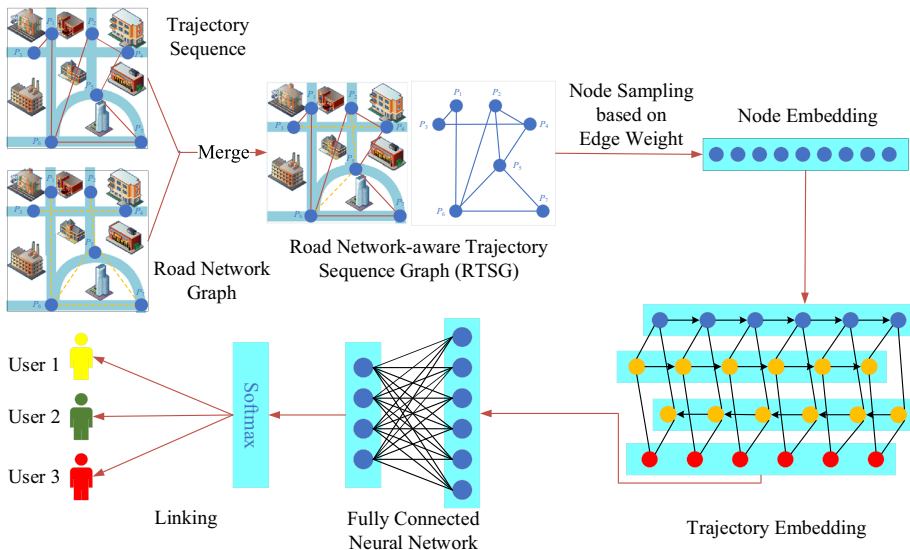


Figure 2 Overview of the proposed model TULRN

Hidden Markov Model and code available¹. Notably, the core part of this study is linking trajectories to their owners with well-designed deep learning modules, thus the details of preprocessing (i.e., map-matching) of the input data are omitted here. After map-matching, we obtain the trajectory sequence $TS(\tau)$ for each trajectory τ in S .

5.1 Construction of the graph RTSG

Following the map-matching, we construct a road network-aware trajectory sequence graph called RTSG. Specifically, we connect each road segment of a trajectory sequence $TS(\tau)$ to an edge in graph G , and connect each intersection of $TS(\tau)$ to a node of G . Finally, we obtain the graph RTSG by merging all trajectory sequences with the given road network graph.

5.2 Node embedding

For most GNN models, they only consider the node features of the graph yet ignore the edge features. To obtain a fixed-length node sequence for learning, these methods usually use the adjacency matrix for sampling according to the neighbors of a node, and ignore the edge weight during the sampling. When the number of nodes in the entire graph is large, these methods tend to sample unremarkable nodes and omit unique nodes, which will cause a serious loss of feature information of the graph. Having observed the shortcoming of existing work, we design a novel sampling strategy based on edge weight. Specifically, the calculation of edge weight is presented as follows.

We claim that the edges that have been visited by users in graph RTSG are not equally important. The edges corresponding to hot road segments are usually useless for distinguishing a user from others, while the edges visited by fewer users are more discriminative. Based on this idea, we propose to compute the edge weight with entropy, which is the expected value of the information contained in each message in relation to the importance of the message². Shannon entropy is a typical entropy, which is a common tool and has been widely used in many applications [38, 39], Renyi entropy is a generalized version of it and has proven more powerful [40]. Given a set of users $U = \{u_1, u_2, \dots, u_n\}$, the Renyi entropy of an edge e is defined as follows:

$$H(e) = \frac{1}{1-q} \log \sum_{i=1}^n \left(\frac{N_{u_i}(e)}{M_{u_i}} \right)^q \quad (1)$$

where $N_{u_i}(e)$ is the number of times that u_i passes through the road segment corresponding to e , and M_{u_i} denotes the total number of road segments that u_i has visited. Following the work [38, 40], the parameter q makes Renyi entropy much more expressive and flexible than Shannon entropy, and it indicates entropy's sensitivity to the number $N_{u_i}(e)$, and more details are discussed in [40]. Next, we give the definition of the edge weight $w(e)$ as follows

¹<https://github.com/bmwcarit/barefoot/>

²<https://en.wikipedia.org/wiki/Entropy>

based on $H(e)$,

$$\begin{aligned}
 w(e) &= \exp(-H(e)) \\
 &= \exp\left(-\frac{1}{1-q} \log \sum_{i=1}^n \left(\frac{N_{u_i}(e)}{M_{u_i}}\right)^q\right) \\
 &= \left(\sum_{i=1}^n \left(\frac{N_{u_i}(e)}{M_{u_i}}\right)^q\right)^{\frac{1}{q-1}}
 \end{aligned}
 \tag{2}$$

Apart from the edge weight calculation, we construct a feature vector f_i for each node e_i in RTSG based on word embedding before sampling. Next, to obtain a complex representation for each node with interactive information, we sort the edges in RTSG according to the weights of them from large to small. During the aggregation process, we sample nodes with different probabilities, i.e., the node with a larger weight is more likely to be sampled and vice versa. The feature matrix $F = \{f_1, f_2, \dots, f_n\}$ provides the initial input for each sampled point. Suppose that the representation of the sampling point p_i (i.e., a node in RTSG) in the k -th layer is expressed as h_i^k , and $\{h_N^k\}$ denotes the representation of the sequence of nodes to be sampled by p_i in k -th layer, the representation of the sampling point p_i in the $(k + 1)$ -th layer is:

$$h_i^{k+1} = \sigma(W \cdot \text{Mean}(h_i^k \oplus \{h_N^k\}))
 \tag{3}$$

Following the node embedding, we can obtain the trajectory representation based on the trajectory sequence. By way of illustration, the red and blue dotted lines in Figure 3 denote the trajectory sequences of τ_1 and τ_2 in RTSG respectively. Then, we can obtain trajectory representation e_{τ_1} by concatenating the embeddings of p_3, p_4, p_5 , and p_7 . Similarly, the representation e_{τ_2} is obtained by concatenating the embeddings of p_7, p_6, p_1 , and p_3 .

5.3 BiLSTM-based trajectory embedding

To further utilize the sequential information contained by trajectories for better representation learning, we construct a BiLSTM-based module, the input of which is the trajectory embedding, which contains abundant topology information of a road network, learned by weight-aware GNN module.

As an extension of the traditional RNN, the LSTM model introduces memory cells with different structures. The LSTM cell at each moment contains an input gate i_t , a forget gate

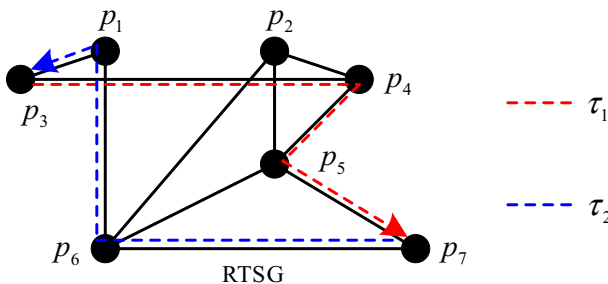


Figure 3 An example of RTSG and trajectory sequence-based embedding

f_t , and an output gate o_t . The input of the cell at the current moment will include the current input x_t , the output of the cell at the previous moment h_{t-1} and its state C_{t-1} ,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

where all W and b are parameters, \odot is the symbol for multiplying a vector by a matrix. σ and \tanh are the activation functions. The input gate i_t is used to retain the information of the previous input, the forget gate f_t is used to control the forgetting degree of the previous input, and the output gate o_t is used to control the output of the next time. BiLSTM is a combination of the forward LSTM and backward LSTM. At time t , the output trajectory representation of BiLSTM is the concatenation of h_t of the forward LSTM and \hat{h}_t of the backward LSTM, i.e.,

$$e_\tau = \text{Concat}(h_t, \hat{h}_t) \tag{8}$$

Taking full advantage of the GNN-based module in capturing topology information and the BiLSTM-based module in capturing sequence features, we obtain the final and effective trajectory representation, which is the input of the next linking module.

5.4 Trajectory user linking

Following the learning of trajectory representation on road networks, we link trajectories to users who generate them, and the architecture of the linking module is presented in Figure 4.

Observed from Figure 4, two parts are involved in linking, i.e., a fully-connected Multi-Layer Perception (MLP) and softmax. The input of MLP is the representation e_τ of a trajectory, and we assume the output of MLP is $Y = (y_1, y_2, \dots, y_n)$ after computing $We_\tau + b$, then each probability $p(y_i)$ in softmax is defined as:

$$p(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)} \tag{9}$$

where W and b denote the weight matrix and bias vector in MLP, respectively. The predicted label is generated with $\hat{y} = \text{argmax } p(y)$. Generally, trajectory user linking can be regarded

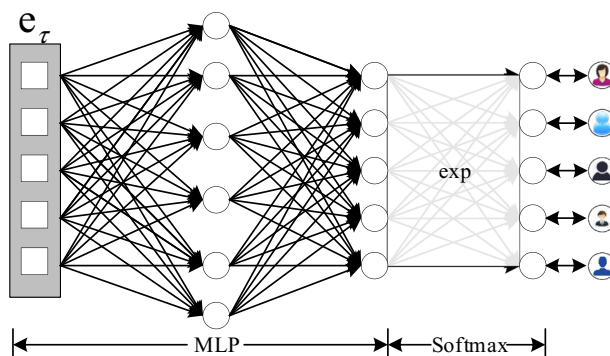


Figure 4 Architecture of the linking module

as a multi-class classification problem, we apply the cross-entropy as a loss function and use backward propagation and Adam [41] to train our model. The cost function is given as:

$$\mathcal{J} = -\frac{1}{L} \sum_{i=1}^L g_i \log(p(y_i)) \quad (10)$$

where L is the number of trajectories used in training and g_i is the one-hot represented ground truth of the trajectory.

Continue the example in Figure 1, assume τ_1 and τ_3 are generated by u_1 , τ_2 and τ_4 belong to u_2 , after embedding τ_1 , τ_2 , τ_3 , and τ_4 into vectors, we use \mathbf{e}_{τ_1} and \mathbf{e}_{τ_2} to train the linking module in Figure 4. Then, we have $p(y) = (0.73, 0.27)$ for τ_3 and $p(y) = (0.32, 0.68)$ for τ_4 in test, which means our proposed model TULRN can make correct classification in this case.

6 Experiment study

We conduct extensive experiments on a real-world dataset in this section. First, we present the experimental setup, statistics of the dataset, compared methods, and evaluation metrics. Then, we compare the performance of our proposed model TULRN with those of baselines and discuss the impact of parameters. Notably, all algorithms are implemented with Python 3.8 and run on a Linux Server with 256GB memory.

6.1 Experimental setup

The parameters used in our experiments are presented in Table 2. The dimension of a node in graph RTSG and the number of iterations are set to 250 and 100, respectively. Max length of a sequence is set to 50, the batch size is 64, and the learning rate of TULRN is 0.0001.

6.2 Dataset

The road network, which contains 56201 intersections and 75268 road segments including 306954 sample points on them, is collected from the city of Beijing, and the map is presented in Figure 5. The trajectory dataset used to conduct trajectory user linking contains 10567 trajectories with 415216 samples points, generated by 493 taxis (i.e., users) from 2012-10-1 to 2012-10-7. We divide the trajectory dataset of each user into training set and test set according to the ratio of 7:3.

Table 2 Experimental Setup

Parameter	Value
embedding size	250
number of iterations	100
max length	50
batch size	64
learning rate	0.0001



Figure 5 Road network of the city Beijing

6.3 Compared methods

We compare the performance of TULRN with the following state-of-the-art methods, which are focusing on the TUL problem with trajectory data.

- TULER. The model, which has two variants (i.e., TULER-LSTM and TULER-GRU), is proposed in [20] and has better performance in more evaluation metrics by embedding trajectories with LSTM, thus we reproduce the variant TULER-LSTM in this work.
- TULVAE. The approach is introduced in [21], where the TUL problem is tackled with a semi-supervised learning framework that learns human mobility in neural generative architecture with stochastic latent variables that span hidden states in RNN.
- TLUTE. The method leverages a graph-based location embedding method to learn the semantics of locations and has better performance while embedding trajectories with LSTM compared with GRU. Consequently, we implement the method with TULTE-LSTM.
- AdattTUL. To make adversarial mobility learning for the TUL problem, the model AdattTUL is proposed in [42]. To train the model, multiple human preferences are considered and an attention mechanism is used to dynamically capture the complex relationships of user check-ins from trajectory data.
- DeepTUL. The model, which not only combines multiple features that govern user mobility but also learns from labeled historical trajectory, is composed of a feature representation layer and an attentive recurrent network with attention mechanism and proposed by [34].
- GNNTUL. It is the first GNN-based human mobility learning model [23] exploiting implicit transition patterns behind sparse user traces on social networks while extracting users' unique motion features and discriminating the motion traces.

Notably, we report the best performance of all compared methods in this section, by optimizing parameters for each of them in the given dataset.

6.4 Evaluation metrics

The TUL task can be regarded as a multi-class classification problem. We use ACC@K and Macro-F1 to evaluate the performance, which are common metrics in multi-class classification area [20, 21, 23]. Specifically, ACC@K is to evaluate the accuracy of prediction and can be represented as:

$$\text{ACC@K} = \frac{\# \text{correctly identified trajectories@K}}{\# \text{trajectories}} \quad (11)$$

and Macro-F1 is the harmonic mean of the precision (Macro-P) and recall (Macro-R) that are averaged across all classes:

$$\text{Macro-F1} = \frac{2 \times \text{Macro-P} \times \text{Macro-R}}{\text{Macro-P} + \text{Macro-R}} \quad (12)$$

6.5 Experimental results

We present the ACC@1, ACC@5, and Macro-F1 of all methods in Table 3. Observed from which, TULVAE performs better than TULER since it is able to capture the semantics of mobility patterns and has incorporated unlabeled data into the training. TULER suffers from the shallow generation in modeling sequence information although the variant LSTM is adopted. TLUTE leverages both spatial and temporal information of trajectories to mine the underlying movement pattern of users and utilizes the pattern to improve the performance of TUL, thus TLUTE performs better than TULER and TULVAE. The model AdattTUL leverages an attention mechanism in trajectory encoding to address the importance of each road segment, which leads to the higher performance compared with that of TLUTE. DeepTUL performs better than all baselines except GNNTUL and the reason is that the attention module of the model learns the multi-periodic nature of user mobility on each road segment and generates the most related context of the current trajectory to improve the accuracy. The recent study GNNTUL performs best in all compared methods, and the reason is that the GNN module in GNNTUL captures spatial and high-order correlations among locations and introduces non-exist by reasonable transition patterns into the encoded location embedding vectors. Without surprise, our proposed model TULRN outperforms other methods and the reason is twofold. (1) Compared with segmenting trajectories based on location information in some developed methods, the trajectory sequence obtained based on road networks is more likely to characterize the sequential behaviors of a user. (2) The topology and sequential information involved in trajectories have been fully utilized in the weight-aware GNN module and BiLSM-based module, respectively.

Table 3 Results of all methods

Algorithm	ACC@1	ACC@5	Macro-F1
TULER	24.14%	59.58%	21.61%
TULVAE	24.97%	61.45%	23.98%
TLUTE	25.16%	63.07%	25.01%
AdattTUL	27.12%	65.38%	27.85%
DeepTUL	28.86%	67.91%	28.41%
GNNTUL	29.97%	69.98%	30.21%
TULRN	32.11%	73.71%	32.67%

Table 4 Property of compared approaches

	Weight-aware GNN	BiLSTM
TUL-GNN	✓	×
TUL-BiLSTM	×	✓
TULRN	✓	✓

6.6 Ablation study

Specifically, we use the weight-aware GNN module in TULRN to utilize the topology information of a road network, and use the BiLSTM-based module to utilize the sequential features of trajectories, during the embedding of trajectories. To investigate the benefits brought by these two modules separately, we develop the following baselines (i.e., TUL-GNN and TUL-BiLSTM), and the property of them are presented in Table 4.

Observed from Table 5, the model TULRN performs much better than TUL-GNN and TUL-BiLSTM, since in which both the structure of a road network and sequential characteristics of trajectories are taken into account. In contrast, only one of these two features is considered in TUL-GNN and TUL-BiLSTM. The results in Table 5 demonstrate the effectiveness and indispensability of the weight-aware GNN module and Bi-LSTM-based module for high-performance trajectory embedding.

6.7 Analysis of parameters

Varying embedding size The dimension of node embedding is of critical importance for TULRN, the performance of it while varying the embedding size from 50 to 400 is presented in Figure 6(a). Observed from which, our proposed model TULRN achieves higher ACC@1, ACC@5, and Macro-F1 with the increase of the embedding size. This is because more latent information will be contained by the learned vectors in this progress. Additionally, we observe that the ACC@1 and Macro-F1 have little change from the dimension 250 to 400, especially the ACC@5 has remained virtually unchanged. Consequently, we set the node embedding size to 250 with the goal of saving running time and memory costs.

Varying number of iterations Another important factor to be investigated is the number of iterations. Seen from Figure 6(b), TULRN obtains higher ACC@1, ACC@5, and Macro-F1 with the increase of this number, since the parameters in TULRN are optimized in this progress. Additionally, we observe that it is enough to set the number of iterations to 100, a too large iteration number needs more running time and memory cost, and may lead to the overfitting problem. Consequently, we set this number to 100 in this work.

Varying learning rate The performance of TULRN, while varying the learning rate from 0.001 to 0.01, is presented in Figure 7(a). Observed from which, the increase of learning

Table 5 Performance of compared approaches

Algorithm	ACC@1	ACC@5	Macro-F1
TUL-GNN	29.85%	66.35%	27.25%
TUL-BiLSTM	30.12%	65.42%	28.68%
TULRN	32.11%	73.71%	32.67%

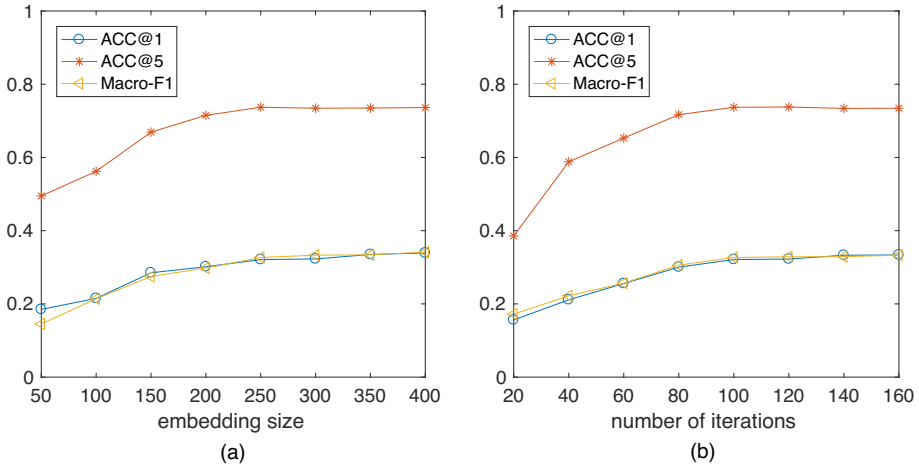


Figure 6 Performance of TULRN w.r.t. (a) embedding size and (b) the number of iterations

rate firstly enhances the performance of TULRN, as the local optimum can be avoided in this progress. However, the model may miss the global optimum with a too large learning rate. Finally, this parameter is set to 0.005 in this work.

Varying dropout The results for varying dropout are presented in Figure 7(b). As expected, the increase of dropout initially leads to the increase of the performance of TULRN, since the problem of overfitting can be further alleviated in this progress. However, a too large dropout may also lead to the loss of information. To obtain the best performance for TULRN, the dropout is set to 0.025 in our experiments.

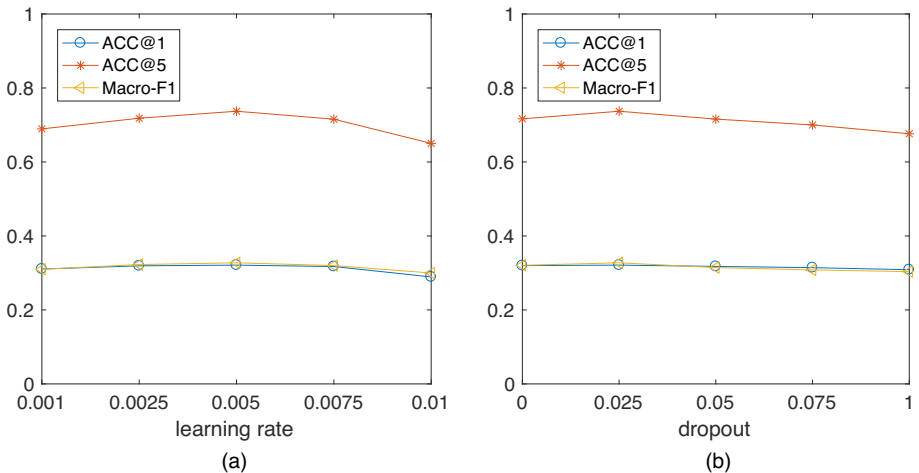


Figure 7 Performance of TULRN w.r.t. (a) learning rate and (b) dropout

7 Conclusion and future work

The problem of trajectory user linking has received increasing attention in recent years, due to the wide range of applications underlay by the study. Despite the great contributions made by traditional studies, there has been no work focusing on tackling the problem on road networks. To fill the gap, we propose a novel model namely TULRN, which differs from existing work with two characteristics: (1) a road network-aware trajectory sequence graph called RTSG is developed to reorganize the input trajectories; (2) the structure of the given road network and sequential information of trajectories are fully utilized during the embedding. The experimental results conducted on a real-world dataset demonstrate that our proposed model outperforms state-of-the-art methods. In future work, we can take the time information into account for better trajectory representation learning. Additionally, the TUL problem can be extended to multiple platforms, where a novel pruning strategy is necessary to prune the search space.

Acknowledgements This work is supported by National Natural Science Foundation of China (No. 61872166), Six Talent Peaks Project of Jiangsu Province (2019 XYDXX-161).

Author Contributions Yu Sang wrote the manuscript, Zhenping Xie modified the proposed model, Wei Chen and Lei Zhao polished the paper.

Funding The funding concludes the National Natural Science Foundation of China (No.61872166), Six Talent Peaks Project of Jiangsu Province (2019 XYDXX-161).

Availability of Supporting Data The road network and trajectory data in experiments are non-public datasets.

Declarations

Competing interests We declare that we have no conflict of interest.

References

1. Chen, L., Ozsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: SIGMOD, pp. 491–502 (2005)
2. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering Similar Multidimensional Trajectories. In: ICDE, pp. 673–684 (2002)
3. Ranu, S.P.D., Telang, A.D., Deshpande, P., Raghavan, S.: Indexing and Matching Trajectories under Inconsistent Sampling Rates. In: ICDE, pp. 999–1010 (2015)
4. de Sousa, R.S., Boukerche, A., Loureiro, A.A.F.: Vehicle trajectory similarity: models, methods, and applications. *ACM Comput. Surv.* **53**(5), 94–19432 (2020)
5. Han, P., Wang, J., Yao, D., Shang, S., Zhang, X.: A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks. In: KDD, pp. 556–564 (2021)
6. Chen, L., Shang, S., Jensen, C.S., Yao, B., Kalnis, P.: Parallel Semantic Trajectory Similarity Join. In: ICDE, pp. 997–1008 (2020)
7. Zheng, K., Shang, S., Yuan, N.J., Yang, Y.: Towards efficient search for activity trajectories. In: ICDE, pp. 230–241 (2013)
8. Chen, W., Zhao, L., Xu, J., Liu, G., Zheng, K., Zhou, X.: Trip oriented search on activity trajectory. *JCST* **30**(4), 745–761 (2015)
9. Zheng, B., Zheng, K., Scheuermann, P., Zhou, X., Nguyen, Q.V.H., Li, C.: Searching activity trajectory with keywords. *World Wide Web* **22**(3), 967–1000 (2019)
10. Chen, L., Shang, S., Feng, S., Kalnis, P.: Parallel subtrajectory alignment over massive-scale trajectory data. In: IJCAI, pp. 3613–3619 (2021)

11. Delling, D., Schieferdecker, D., Sommer, C.: Traffic-Aware Routing in Road Networks. In: ICDE, pp. 1543–1548 (2018)
12. Ticha, H.B., Absi, N., Feillet, D., Quilliot, A., Woensel, T.V.: A branch-and-price algorithm for the vehicle routing problem with time windows on a road network. *Networks* **73**(4), 401–417 (2019)
13. Al-Baghdadi, A., Lian, X., Cheng, E.: Efficient path routing over road networks in the presence of ad-hoc obstacles. *Inf. Syst.*, 88 (2020)
14. Lin, Y., Wan, H., Guo, S., Lin, Y.: Pre-Training Context and Time Aware Location Embeddings from Spatial-Temporal Trajectories for User Next Location Prediction. In: AAAI, pp. 4241–4248 (2021)
15. Han, P., Li, Z., Liu, Y., Zhao, P., Li, J., Wang, H., Shang, S.: Contextualized Point-Of-Interest Recommendation. In: IJCAI, pp. 2484–2490 (2020)
16. Zhao, P., Luo, A., Liu, Y., Xu, J., Li, Z., Zhuang, F., Sheng, V.S., Zhou, X.: Where to go next: a spatio-temporal gated network for next POI recommendation. *IEEE Trans. Knowl. Data Eng.* **34**(5), 2512–2524 (2022)
17. Hu, X., Xu, J., Wang, W., Li, Z., Liu, A.: A graph embedding based model for fine-grained POI recommendation. *Neurocomputing* **428**, 376–384 (2021)
18. Sun, H., Xu, J., Zhou, R., Chen, W., Zhao, L., Liu, C.: HOPE: A hybrid deep neural model for out-of-town next POI recommendation. *World Wide Web* **24**(5), 1749–1768 (2021)
19. Wang, G., Liao, D., Li, J.: Complete user mobility via user and trajectory embeddings. *IEEE Access* **6**, 72125–72136 (2018)
20. Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., Zhang, F.: Identifying Human Mobility via Trajectory Embeddings. In: IJCAI, pp. 1689–1695 (2017)
21. Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., Zhang, F.: Trajectory-User Linking via Variational Autoencoder. In: IJCAI, pp. 3212–3218 (2018)
22. Zhou, F., Yin, R., Trajcevski, G., Zhang, K., Wu, J., Khokhar, A.A.: Improving human mobility identification with trajectory augmentation. *GeoInformatica* **25**(3), 453–483 (2021)
23. Zhou, F., Chen, S., Wu, J., Cao, C., Zhang, S.: Trajectory-User Linking via Graph Neural Network. In: ICC, pp. 1–6 (2021)
24. Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J.: Trajectory Clustering via Deep Representation Learning. In: IJCNN, pp. 3880–3887 (2017)
25. Yao, D., Zhang, C., Zhu, Z., Hu, Q., Wang, Z., Huang, J., Bi, J.: Learning deep representation for trajectory clustering. *Expert Syst. J. Knowl. Eng.* **35**(2) (2018)
26. Yang, W., Zhao, Y., Zheng, B., Liu, G., Zheng, K.: Modeling Travel Behavior Similarity with Trajectory Embedding. In: DASFAA, vol. 10827, pp. 630–646 (2018)
27. Fu, T., Lee, W.: Trembr: Exploring road networks for trajectory representation learning. *ACM Trans. Intell. Syst. Technol.* **11**(1), 10–11025 (2020)
28. Liu, D., Wang, J., Shang, S., Han, P.: MSDR: Multi-Step Dependency Relation Networks for Spatial Temporal Forecasting. In: KDD, pp. 1042–1050 (2022)
29. Xu, S., Zhang, R., Cheng, W., Xu, J.: MTLM: A multi-task learning model for travel time estimation. *GeoInformatica* **26**(2), 379–395 (2022)
30. Xu, J., Zhao, J., Zhou, R., Liu, C., Zhao, P., Zhao, L.: Predicting destinations by a deep learning based approach. *IEEE Trans. Knowl. Data Eng.* **33**(2), 651–666 (2021)
31. Li, X., Zhao, K., Cong, G., Jensen, C.S., Wei, W.: Deep Representation Learning for Trajectory Similarity Computation. In: ICDE, pp. 617–628 (2018)
32. Co-Reyes, J.D., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., Levine, S.: Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In: ICML, pp. 1008–1017 (2018)
33. Boonchoo, T., Ao, X., He, Q.: Multi-aspect embedding for attribute-aware trajectories. *Symmetry* **11**(9), 1149 (2019)
34. Miao, C., Wang, J., Yu, H., Zhang, W., Qi, Y.: Trajectory-User Linking with Attentive Recurrent Network. In: AAMAS, pp. 878–886 (2020)
35. Sun, T., Xu, Y., Wang, F., Wu, L., Qian, T., Shao, Z.: Trajectory-User Link with Attention Recurrent Networks. In: ICPR, pp. 4589–4596 (2020)
36. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2009)
37. Newson, P., Krumm, J.: Hidden Markov Map Matching through Noise and Sparseness. In: GIS, pp. 336–343 (2009)
38. Pham, H., Shahabi, C., Liu, Y.: EBM: an Entropy-Based Model to Infer Social Strength from Spatiotemporal Data. In: SIGMOD, pp. 265–276 (2013)
39. Bulusu, K.V., Plesniak, M.W.: Shannon entropy-based wavelet transform method for autonomous coherent structure identification in fluid flow field data. *Entropy* **17**(10), 6617–6642 (2015)

40. Chen, W., Yin, H., Wang, W., Zhao, L., Zhou, X.: Effective and Efficient User Account Linkage across Location Based Social Networks. In: ICDE, pp. 1085–1096 (2018)
41. Kingma, D.P., Ba, J.: Adam: a Method for Stochastic Optimization. In: ICLR (2015)
42. Gao, Q., Zhang, F., Yao, F., Li, A., Mei, L., Zhou, F.: Adversarial mobility learning for human trajectory classification. *IEEE Access* **8**, 20563–20576 (2020)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.