Predicting Human Mobility with Semantic Motivation via Multi-task Attentional Recurrent Networks

Jie Feng, Yong Li, *Senior Member, IEEE*, Zeyu Yang, Qiang Qiu, Depeng Jin, *Member, IEEE*

Abstract—Human mobility prediction is of great importance for a wide spectrum of location-based applications. However, predicting mobility is not trivial because of four challenges: 1) the complex sequential transition regularities exhibited with time-dependent and high-order nature; 2) the multi-level periodicity of human mobility; 3) the heterogeneity and sparsity of the collected trajectory data; and 4) the complicated semantic motivation behind the mobility. In this paper, we propose DeepMove, an attentional recurrent network for mobility prediction from lengthy and sparse trajectories. In DeepMove, we first design a multi-modal embedding recurrent neural network to capture the complicated sequential transitions by jointly embedding the multiple factors that govern human mobility. Then, we propose a historical attention model with two mechanisms to capture the multi-level periodicity in a principle way, which effectively utilizes the periodicity nature to augment the recurrent neural network for mobility prediction. Furthermore, we design a context adaptor to capture the semantic effects of Point-Of-Interest (POI)-based activity and temporal factor (e.g., dwell time). Finally, we use the multi-task framework to encourage the model to learn comprehensive motivations with mobility by introducing the task of the next activity type prediction and the next check-in time prediction. We perform experiments on four representative real-life mobility datasets, and extensive evaluation results demonstrate that our model outperforms the state-of-the-art models by more than 10%. Moreover, compared with the state-of-the-art neural network models, DeepMove provides intuitive explanations into the prediction and sheds light on interpretable mobility prediction.

Index Terms—neural network; attention; human mobility

1 Introduction

Human mobility prediction is of great importance in a wide spectrum of applications, ranging from smart transportation and urban planning, to resource management in mobile communications, personalized recommender systems, and mobile healthcare services. By predicting the future locations people tend to visit, governments can design better transportation planning and scheduling strategies to alleviate traffic jams and handle crowd aggregations. Ride-sharing platforms like Uber and Didi also heavily rely on accurate mobility prediction techniques, for better estimating the travel demands of their customers and scheduling resources to meet such demands accordingly. With the proliferation of such mobile applications, it has become a pressing need to understand the mobility patterns of people from their historical traces and foresee their future whereabouts. With sensing user's future movement in advance, the telecom operators can do better resource arrangement and guarantee the high quality of service with minimal costing.

By measuring the entropy of an individual's trajectory, Song et al. [1] find remarkable stability in the predictability of human mobility — 93% human movements are predictable according to their study on a million-scale user base. So far, lots of research efforts [1, 2, 3, 4, 5, 6, 7] have been taken to turn this identified predictability into actual mobility prediction models. Early methods for mobility prediction are mostly pattern-based [8, 9, 10, 11, 12]. They first discover pre-defined mobility patterns (e.g., sequential patterns, periodic patterns) from the trajectory traces, and then predict future locations based on these extracted patterns. Such methods, however, not only suffer from the one-sided nature of the pre-defined patterns but also ignore personal preferences that are critical for mobility prediction. More recent developments turn to model-based methods [2, 3, 13] for mobility prediction. They leverage sequential statistical models, e.g., Markov chain or recurrent neural networks (RNN), to capture the transition regularities of human movements and learn the model parameters from the given training corpus.

Despite the inspiring results of model-based mobility prediction, several key challenges that remain to be solved to realize the high potential predictability of human movements: (1) First, human mobility exhibits *complex sequential transition regularities*. In practice, the transitions between two arbitrary locations can be time-dependent and high-order. For instance, the probability of moving from home to office for a commuter is higher in workday mornings but often low in weekend mornings. Meanwhile, the transition may not follow the simple and exact Markov chain assumption, as people can go to different places (e.g., breakfast places) in their commute routes, which lead to high-order and irregular transition patterns. (2) Second, there is often *multi-level periodicity* that governs human mobility. Periodicity has been

• *J. Feng, Z. Yang, Y. Li, 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. Q. Qiu is with Institute of Computing Technology, Chinese Academy of Science, Beijing, China.*

demonstrated as an important factor that governs human movements [14, 15]. However, existing mobility prediction models are mostly sequential models that only capture the transitional regularities. Further, mobility periodicity is often complex and multi-level, involving daily routines, weekend leisure, yearly festivals, and even other personal periodic activities. All these periodic activities interweave with each other in complex ways and are difficult to be captured. (3) The third challenge is *heterogeneity and sparsity* in the data recording human mobility. Unlike intentionally collected tracking data like taxi trajectories, most data recording human mobility is low-sampling in nature, and the location information is recorded only when the user accesses the location service. Such sparsity makes it difficult for training a mobility model for each individual. Aggregating the data of all users, on the other hand, may face the problem of mixing personalized mobility patterns and suffer from low prediction accuracy. (4) The fourth challenge is how to capture the complicated *semantic motivation* behind mobility. Human mobility does not only follow the simple physical law like continuity but also is influenced by the underlying motivation of related human activities. Without considering the motivation of human mobility and the semantic meaning of locations, the performance of mobility modelling will be limited.

In this paper, we propose DeepMove, an attentional recurrent neural network model for predicting human mobility from lengthy and sparse trajectories. In DeepMove, we utilize a multi-modal recurrent neural network to capture the multiple factors that govern the transition regularities of human movements. Specifically, we design a multi-modal embedding module that converts discrete features (e.g., time of day, location, user ID) of a raw location record into dense representations, which are then fed into a recurrent neural network to model long-range and complex dependencies in a trajectory sequence. DeepMove is capable of discovering the transitional regularities that are shared by all the users, while is also be able to capture personalized movement preferences by flexibly leveraging user embeddings or user identification task. Another key component in DeepMove is a historical attention module, which captures the multi-level periodicity of human mobility in a principled way. The attention component is jointly trained to select historical records that are highly correlated with the current prediction timestamp. It thus flexibly utilizes periodic movement regularities to augment the recurrent neural network and improve prediction accuracy. Meanwhile, the learned attention weights offer an easy-to-interpret way to understand which historical activities are emphasized in the prediction process. Besides, we design a context adaptor to model the semantic effects of POI label, user text, and dwell time of the current location. Furthermore, we introduce two other additional tasks: the task of the next activity type prediction and the next check-in time prediction to cooperate with the original next location prediction task. With the help of context adaptor and multi-task learning, the new enhanced model is encouraged to better model the mobility pattern by observing the underlying semantic motivation of human movement.

Our contributions can be summarized as follows:

- We propose an attentional recurrent model, DeepMove, to predict human mobility from long-range and sparse trajectories. Our model combines two regularities in a principled way: heterogeneous transition regularity and multi-level periodicity. To the best of our knowledge, DeepMove is the first model that simultaneously combines these two important regularities for accurate mobility prediction.
- We design two attention mechanisms that are tailored to cooperate with the recurrent module. The first is to directly embed historical records into independent latent vectors and use the current status to selectively focus on relevant historical steps; while the second preserves the sequential information among historical records. Both unveil the periodicity of human mobility by matching historical records with the current status and rationalize the prediction making process.
- We design a context adaptor with the multi-task prediction framework to achieve better prediction performance by capturing the semantic motivation of human mobility. One the one hand, the context adaptor is utilized to model the semantic effects of PoI label, dwell time, and user text of the current location. On the other hand, our model is designed to predict the next activity type and the next visiting time while predicting the next location to better capture the semantic motivation of human mobility.
- We perform extensive experiments on four representative real-life mobility datasets. Our results demonstrate that DeepMove outperforms state-of-the-art mobility prediction models by more than 10%. DeepMove shows outstanding generalization ability and is robust across trajectory datasets that have different natures. Furthermore, compared with existing RNN models, DeepMove provides intuitive explanations into the prediction and sheds light on interpretable mobility prediction.

Compared with the original conference version [16], we extend this work from two aspects. *First of all*, we introduce two new designs to improve the prediction performance by modelling the semantic motivation of human mobility. On the one hand, we design a new context adaptor to capture the semantic effects of activity, user text, and dwell time of the current location. On the other hand, we introduce the task of next activity type prediction and next visiting time prediction to cooperate with the original next location prediction task to learn the comprehensive pattern of human mobility. Compared with the original DeepMove, the enhanced DeepMove improves the Top1 prediction accuracy by more than 11% on two check-in datasets. *Secondly*, we also extend the experimental results from two dimensions to help readers to better understand our models. We add a new public available data and two new baselines to evaluate the performance of the proposed model. Besides, we study the effects of different hyper-parameters (including the weights of multi-task) and also analyze the necessity of location embedding in location prediction task. The results give more insights on the effectiveness and limitation of proposed models, which shed light on the future direction.

The rest of this paper is organized as follows. We first formulate the problem and discuss the motivation of our work in Section 2. Following the motivation, we introduce details of the architecture of DeepMove in Section 3. After that, we apply our model on three real-world mobility datasets and conduct an extensive analysis of the performance in Section 4. After systematically reviewing the related works in Section 5, we finally conclude our paper in Section 6.

2 PRELIMINARIES

In this section, we first formally formulate the mobility prediction problem, and then briefly introduce the recurrent neural networks. Finally, we discuss the motivation and the overview of our solution. In general, the time of collected mobility trajectory data for each person can be more than one month even year, the lengthy mobility records are not trivial for any machine learning model to handle. Besides, due to the regularity of human mobility [1, 6], the lengthy mobility data can be naturally divided into segments [3], where each segment represents a meaningful and complete mobility behavior sequence. Based on these intuitions, we introduce the definition of trajectory.

2.1 Problem Formulation

Definition 1 (Trajectory Sequence). Spatiotemporal point q is a tuple of timestamp t and location identification l, i.e., $q = (t, l)$. Given a user identification u, trajectory sequence S is a spatiotemporal point sequence, i.e., $S^u = q_1 q_2 ... q_n$.

Furthermore, we enrich the spatiotemporal point q with the semantic information e to enable the understanding of the context of human mobility. Following the practice in previous work [3, 17], we utilize the widely-used Point-Of-Interest (POI) data as basic semantic label e in our trajectory points $q = (t, l, e)$. POI (e.g., shops, restaurants, and museums) of a location describes the common interests of people when visiting the location and it can be regarded as the most straightforward and available proxy of the context of human mobility. It is also noted that the proposed framework for mobility prediction with semantic motivation modelling can be easily extended into any semantic labels obtained from other complicated semantics mining methods [17].

Definition 2 (Trajectory). Given a trajectory sequence S^u and a time window t_w, a trajectory is a subsequence $S^u_{t_w} = q_i q_{i+1} .. q_{i+k}$ of S^u in the time window t_w, if $\forall$ $1 < \text{j} \leqslant \text{k}$, t_{q_j} belongs to t_w.

At the m-th time window t_{w_m}, the current trajectory of user u can be defined as $S^u_{t_{w_m}}$ and the trajectory history can be denoted as $S^u_{t_{w_1}} S^u_{t_{w_2}} ... S^u_{t_{w_{m-1}}}$, where t_w can be one specific day, one week or even one month in the year.

Problem 1 (Next Location Prediction). Given the current trajectory $S^u_{t_{w_m}} = q_1 q_2 .. q_n$ and the corresponding trajectory history $S^u_{t_{w_1}} S^u_{t_{w_2}} ... S^u_{t_{w_{m-1}}}$, next location prediction task can be defined to predict the next location point l_{n+1} in the trajectory.

Coupling with the *next location prediction task*, we define two other tasks as follows. With the same input and condition with next location prediction task, the task of *next activity type prediction* is to predict the semantic label e_{n+1} of next location l_{n+1} and the task of *next visiting time prediction* is to predict the visiting time t_{n+1} of the next location l_{n+1}. Finally, the problem of multi-task based mobility prediction is defined as the combination of three mentioned tasks to predict the next-hop mobility event $q_{n+1} = (l_{n+1}, t_{n+1}, e_{n+1})$ by knowing the mobility history.

2.2 Recurrent Neural Network

Recurrent Neural Networks (RNN) [18] is a class of neural networks with cycle and internal memory units to capture sequential information. Long short-term memory (LSTM) [19] and gated recurrent unit (GRU) [20] are widely used recurrent units. LSTM consists of one cell state and three controlled *gates* to keep and update the cell state. Based on the input and last cell state, LSTM first updates the cell state with parts to keep and parts to drop. Then, LSTM generates the output from the cell state with a learnable weight. The updating formulation of LSTM is as follows:

$$i_t = \sigma(W_{ix} x_t + W_{ih} h_{t-1} + b_i), \tag{1}$$
$$f_t = \sigma(W_{fx} x_t + W_{fh} h_{t-1} + b_f), \tag{2}$$
$$o_t = \sigma(W_{ox} x_t + W_{oh} h_{t-1} + b_o), \tag{3}$$
$$g_t = tanh(W_{gx} x_t + W_{gh} h_{t-1} + b_g), \tag{4}$$
$$c_t = f_t * c_{t-1} + i_t * g_t, \tag{5}$$
$$h_t = o_t * tanh(c_t), \tag{6}$$

where $*$ denotes the element-wise multiplication, x_t, h_t, c_t denote the input, hidden state, and cell state, respectively, i_t, f_t, o_t denote three types gates, and g_t denotes the useful information from the input.

GRU is a popular variant of LSTM which replaces the forget gate and the input gate with only one update gate. The updating formulations of GRU are as follows,

$$f_t = \sigma(W_{fx} x_t + W_{fh} h_{t-1} + b_f), \tag{7}$$
$$r_t = \sigma(W_{rx} x_t + W_{rh} h_{t-1} + b_r), \tag{8}$$
$$c_t = tanh(W_{cx} x_t + r_t * (W_{ch} h_{t-1}) + b_c), \tag{9}$$
$$h_t = (1 - f_t) * c_t + f_t * h_{t-1}, \tag{10}$$

where x_t is the input in time t, h_{t-1} is the last output of GRU unit, multiple matrix W are different gate parameters, multiple vectors b are the bias vectors for different parts, f_t is the update weight, r_t is the reset gates, c_t is the candidate and h_t is the output result. According to Chung et al. [20], GRU achieves similar performance in multiple tasks with less computation, which is used as the basic recurrent unit in our proposed model.

2.3 Overview

As a powerful sequence modeling tool, recurrent neural networks can capture long-range dependencies of sequential information. However, when the sequence is too long, i.e., a long sentence with more than 20 words, its performance will degrade rapidly [21]. According to the typical mobility datasets, the average length of one day's trajectory for mobile application data varies from 20 to 100, which obviously exceeds the processing ability of recurrent neural

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

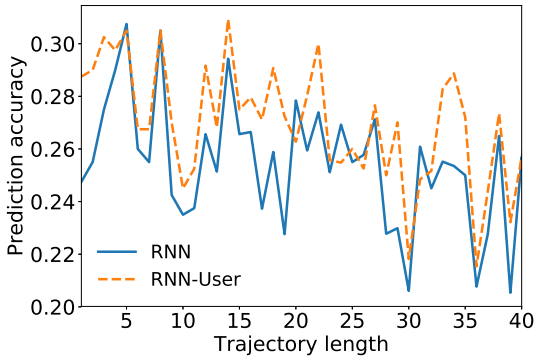


Fig. 1: Performance varies with trajectory length.

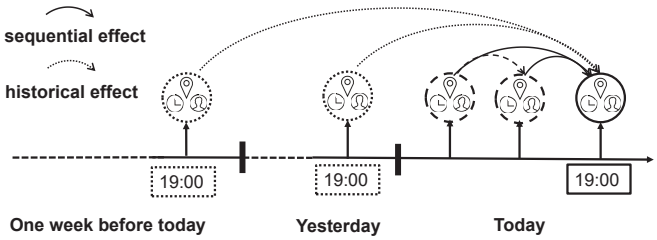


Fig. 2: Motivation and intuition of our solution.

networks. Figure 1 plots the prediction accuracy obtained by a simple recurrent neural network. It shows that the prediction accuracy varies significantly with the testing trajectory. The longer the time extends, the worse performance the prediction achieves. Thus, with the recurrent neural networks, we can only process a limited length trajectory with a short duration of one day or even shorter.

Directly applying the recurrent neural networks to solve the mobility prediction problem is intuitive but inefficient. Except for the long-term nature mentioned above, some other challenges also make it failed. The first is the multi-level periodicity of human mobility. Generally, there exist several periodicities in human activities: day, week, month, and even other personal periodicities, and thus the multi-level periodicity becomes a universe pattern that governs human mobility. However, the general recurrent neural networks can do little to handle this because of the long-term effect and the complicated influencing factors from the trajectory. Besides, because users will not report their activities in every location (unlike uniform sampling traces of taxi [22]), most collected mobility data is sparse and incomplete, which cannot record the periodical nature of human mobility. In general, the data quality problem degrades performance in two ways. The first one is that the missing data will puzzle the recurrent neural network, and induce it to learn the wrong transition information. The other one is that the sparse data makes it difficult to train models for every individual. Even for capturing the transition relations, recurrent neural networks face the problem of the time-dependent and high-order nature of human mobility. In conclusion, the recurrent neural network faces the problem of periodicity, data sparsity, and complicated transitions, which prevent it to achieve high prediction accuracy for human mobility.

Based on the above observations, we propose DeepMove, an attentional recurrent neural network for predicting human mobility from the lengthy, periodical, and incomplete trajectories. Figure 2 presents the intuition behind our solution: not only the sequential information from the current activities decides the next mobility status but also the periodical information from the trajectory history takes effects. In DeepMove, we first involve a multi-modal recurrent neural network to capture the complicated transition relationship. In the multi-modal recurrent neural network, we design a multi-modal embedding module to convert the discrete features (e.g., user ID, location, time of day) into dense representations, which are more expressing and computable. Then, they are jointly fed into a recurrent neural network to capture the complicated transition relationship. With the help of user embedding, DeepMove can distinguish every user and learn the personal preference while training a single model for all users to learn and share similar mobility regularities. Besides, the time representation involved in the embedding gives recurrent neural network the ability to model the time-dependent nature.

Another key component of DeepMove is the historical attention module, which is designed to capture the multi-level periodical nature of human mobility by jointly selecting the most related historical trajectories under the current mobility status. The historical attention module first extracts spatiotemporal features from the historical trajectories by an extractor. Then, these features are selected by the current mobility status based on the spatiotemporal relations to generate the most related context. By combining this context with the current mobility status, we could predict the mobility based on not only the sequential relation but also the multi-level periodical regularity.

To model the semantic motivation behind the mobility, we enhance the original DeepMove by a context adaptor and the multi-task learning techniques. On the one hand, we design a context adaptor to model the semantic effects of POI label, user text, and dwell time of the current location. In the context adaptor, parts of the mentioned discrete semantic features are fed into the multi-modal embedding layer to obtain the comprehensive representation of location record while the temporal feature is utilized to filter and fine-tune the hidden state vector before the output layer. On the other hand, we introduce multi-task learning techniques by introducing the task of the next activity type prediction and the next visiting time prediction to cooperate with the original next location prediction task. With the help of two additional tasks and context adaptor, our model can achieve better prediction performance by understanding the semantic motivation behind the mobility.

3 The Design of DeepMove

Figure 3 presents the architecture of DeepMove, which consists of three major components: 1) feature extracting and embedding module; 2) recurrent module and historical attention modules; and 3) context adaptor and multi-task prediction output layer. Details of the three components are introduced as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

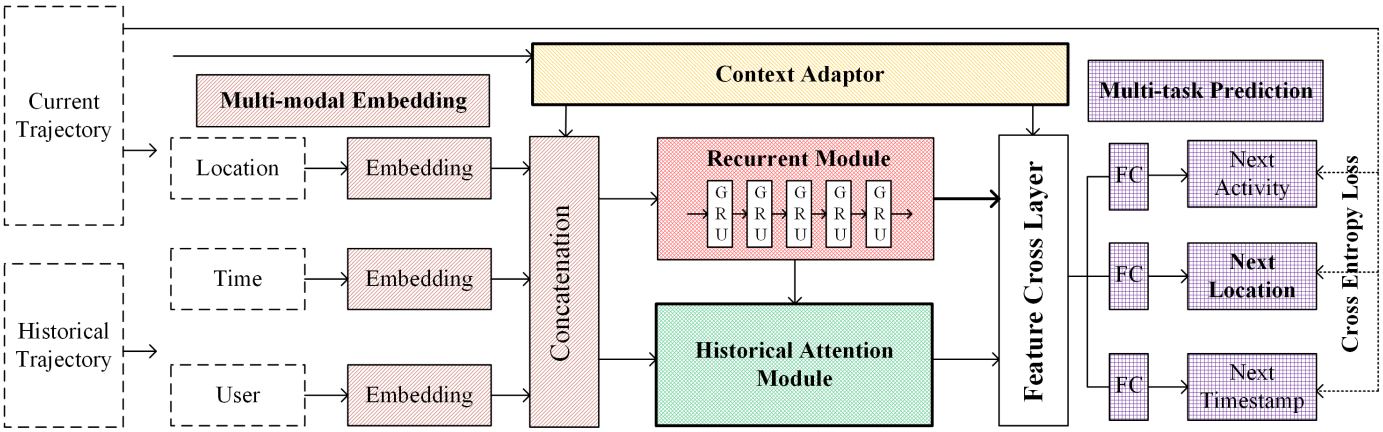


Fig. 3: The main architecture of DeepMove, which contains three major parts: 1) multi-modal embedding module for raw feature extracting; 2) recurrent module and historical attention module for sequential modelling; 3) context adaptor and multi-task prediction output for semantic modelling.

3.1 Feature Extracting and Embedding Module

The trajectories are partitioned into two parts: current trajectory and historical trajectory. The current trajectory is processed by the recurrent layer to model the complicated sequential information. The trajectory history is handled by the historical attention module to extract the regularity of mobility. Before that, all the trajectories are first embedded by the multi-modal embedding module. Simple models like Markov chains can only describe the transitions between independent states like locations. However, mobility transitions are governed by multiple factors like time of day and user preference. Thus, we design a multi-modal embedding module to jointly embed the spatiotemporal features and personal features into dense representations to help model the complicated transitions. In practice, all the available features of one trajectory point including time t, location l, user ID u can be numbered. Then, the numbered features are translated into one-hot vectors and input to the multi-modal embedding module. The ID number of any new user, which does not appeared in the training set, is fixed as 0. We formulate the multi-modal embedding layer as follows,

$$x_i = tanh([W_t t_i + b_t; W_l l_i + b_l; W_e e_i + b_e]), \qquad (11)$$

where W and b denote the learnable parameters of different embedding layers, $tanh$ denotes the non-linear activation function, $[;;]$ denotes the concatenate function, i denotes the i_{th} record in the trajectory sequence.

It is noted that we represent the location in the input by a one-hot vector, which is general for the discrete locations in many real-world applications (e.g, cellular network localization and the obfuscation of location for privacy concern). Furthermore, we argue that the learnable embedding vector after one-hot representation is also able to learn the most important adjacent information stored in the original coordinate format. For location representation, we introduce an independent location embedding module to map one-hot location into a dense vector in a latent space, where the projected dense location representation is close to each other if their original physical location is close. Due to the regularity and consistency of mobility, two points close in the time dimension may also be close in the spatial dimension. Further, with the help of recurrent networks in the following step, the location embedding module is enforced to embed the adjacent location in the physical world into the adjacent space in the latent high-dimensional space in the neural network. In this way, our location embedding module succeeds in modeling the spatial correlation. Besides, if coordinates information of location is available, we can also consider it as the additional features of locations in the model with a linear network.

3.2 Recurrent Module and Historical Attention

3.2.1 *Overview of Sequential Modelling*

The recurrent module aims to capture the complicated sequential information or long-range dependencies contained in the current trajectory. Following the detailed design introduced in Section 2.2, the recurrent layer takes the spatiotemporal vector sequence embedded by the multi-modal embedding layer as input and outputs the hidden state step by step. The formulation of the recurrent layer is as follows,

$$[h_1, h_2, ..., h_n] = LSTM([x_1, x_2, ..., x_n]), \qquad (12)$$

where h represents the output state of the recurrent neural networks (*e.g.*, LSTM), x denotes the output of the multi-modal embedding layer, n denotes the length of the current trajectory. These output states $[h_1, h_2, ..., h_n]$ are called as the current status of the mobility. Paralleled with the recurrent module is the historical attention module, which is designed to capture mobility regularity from the lengthy historical records. It takes the historical trajectory as the input and outputs the most related context vector when queried by a query vector from the recurrent module. The abstract formulation of historical attention is as follows,

$$h_i^s = HisAttn(h_i, [s_1, s_2, ..., s_m]), \qquad (13)$$

where $HisAttn$ denotes the *historical attention module*, h_i is the output hidden state of the recurrent neural network, s_i denotes the embedding vector of i_{th} records in the historical trajectory, m denotes the length of the historical trajectory.

To capture the multi-level periodicity of human mobility, we need an auto-selector to choose the most related historical records of current mobility status from the trajectory

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

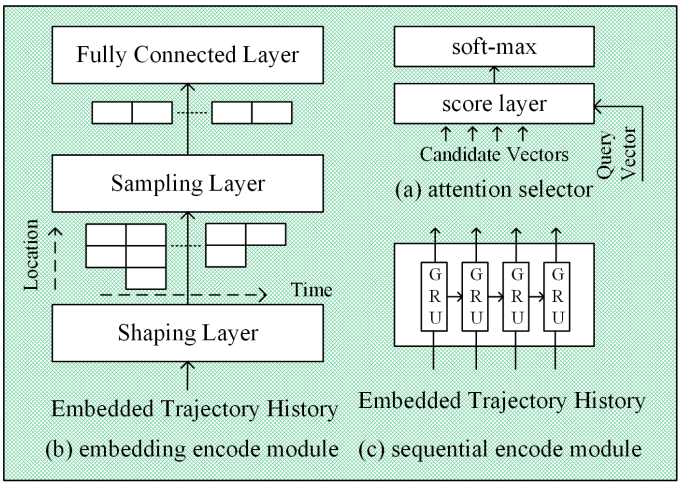


Fig. 4: Architecture of the historical attention module.

history as the periodicity representation. Inspired by the human visual attention nature and the attention mechanism widely used in natural language translation [21], we design a *historical attention module* to implement the auto-selector. As Figure 3 presents, it is comprised of two components: 1) an attention candidate generator to generate the candidates, which are exactly the regularities of the mobility; 2) an attention selector to match the candidate vectors with the query vector, i.e., the current mobility status. We first introduce the basic formulation of the attention module and then discuss two specific candidate generation mechanisms.

3.2.2 Attention Selector

The goal of the attention module is to calculate the *similarity* between the query vector (i.e., the current mobility status) and the candidate vectors to generate the context vector. The attention module is parameterized as a feed-forward neural network that can be trained with the whole neural network. 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 typical attention methods are as follows,

$$c_t = \sum \alpha_i s_i, \tag{14}$$

$$\alpha_i = \sigma(f(h_t, s_i)), \tag{15}$$

$$f_{dot}(h_t, s) = h_t^T s, \tag{16}$$

$$f_{gen}(h_t, s) = h_t^T W s, \tag{17}$$

$$f_{mlp}(h_t, s) = v^T tanh(Ws + Uh_t), \tag{18}$$

where s represents the historical features, W is the learnable parameters, h_t is the query vector which denotes current mobility status from the recurrent layer, f represents the score function, σ is the soft-max function and c_t is the context output representing the periodicity related to the current mobility status. While there are many other variations of attention models [23], we choose the original one for its simplicity and general expressions.

3.2.3 Attention Candidate Generator

To provide the candidate vectors s for the attention selector, we discuss two specific generation mechanisms.

Embedding Encode Module The first is the embedding encode mechanism, whose implementation structure is presented in Figure 4(b). The embedding encodes module directly embeds the historical records into independent latent vectors as the candidate vectors. It is composed of three components: 1) a shaping layer for disorganizing the ordered trajectory sequence into a history matrix with fixed-length temporal dimension and variable-length spatial dimension; 2) a sampling layer for the location and POI sampling; 3) fully connected layers. The shaping layer is a fixed layer, whose structure and parameters are manually assigned. In this layer, we reorganize the trajectory vectors into a two-dimension matrix (for the convenience of discussion, we omit the embedding dimension logically for the time being). In the temporal dimension, we align all the time of trajectory into one week or two days, which is designed to simulate the periodical nature of human mobility. In the spatial dimension, we collect all the locations that appeared in the same time to keep a visited location set for every time slot.

Following the shaping layer is a sampling layer that is designed to sample location from the visited location set in every time slot. We design three kinds of sampling strategies: 1) average sampling; 2) maximum sampling; 3) none sampling. The average sampling strategy adds up all the location embedding vectors in the set at every time slot and calculates the mean value as their representation. In this way, all the historical information can be reserved. The maximum sampling strategy is based on the periodical assumption of human mobility: the most frequently visited location means a lot to the user. It works by selecting the most frequent location embedding vector as the representation for every time slot. The none sampling strategy is to reserve all the location and flatten them along the temporal dimension without any processing. In the last of the paper, the sampling layer with the average sampling strategy is the default settings for the embedding encode mechanism. The final fully connected layers further process the historical spatiotemporal vectors into the appropriate shape.

Sequential Encode Module The second mechanism is the sequential encode mechanism, whose implementation structure is presented in Figure 4(c). The sequential encode module takes the historical records as input and keeps the intermediate outputs of every step as the candidate vectors $\boldsymbol{s}$. Working like the recurrent module introduced before, the formulation of sequential encode module is as follows,

$$[s_1, s_2, ..., s_m] = GRU([x_1^s, x_2^s, ..., x_m^s]), \tag{19}$$

where x_i^s denotes the embedding vector of i_{th} record in the historical trajectory, m is the length of the historical trajectory, s_i denotes the i_{th} historical candidates for attention selection. Different from the embedding encode module, it does not directly simulate the periodicity and reserves all the spatiotemporal information. Based on the multimodal embedding method mentioned above, the recurrent neural network can extract complex sequential information from the historical records. Compared with the embedding encode module, the sequential encode module relies on the follow-up attention selector to capture the periodical information. Besides, the sequential encode module projects

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

the historical records into a latent space which is similar to the current mobility status in. This similar projection result also benefits the follow-up attentional selection.

3.3 Context Adaptor and Multi-task Prediction

To better understand and predict the human mobility by capturing the semantic motivation of human mobility, we first introduce a context adaptor to model the semantic effects of the current location and then utilize the multi-task design to learn the semantic motivation behind the future mobility.

3.3.1 Context Adaptor

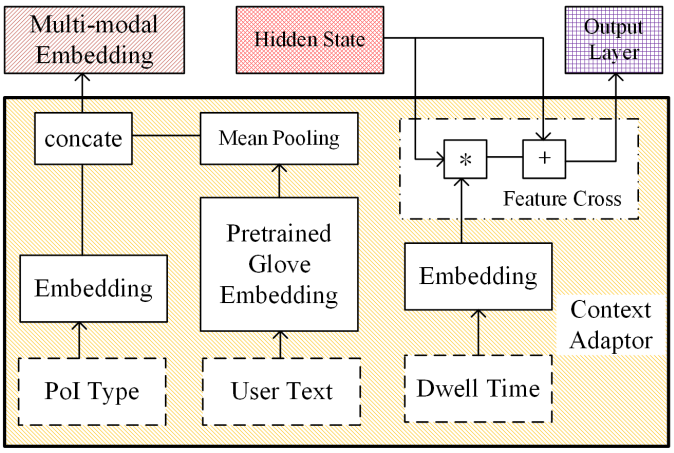


Fig. 5: Architecture of context adaptor.

While aforementioned multi-modal embedding has considered the basic discrete features of the spatiotemporal point including location l, time t, and user u, it fails to capture the semantic effects of the current location. Thus, as Figure 5 shows, we design a context adaptor to model the semantic effects of location like the effects of its POI label, user text, and dwell time. For the discrete POI label, we use a discrete one-hot vector as its representation and design a trainable embedding layer to obtain its dense feature. For user text information, we use the pretrained Glove embeddings [24] to convert the sentence into dense vectors and apply the mean-pooling operation to obtain the final fixed-length representation of user text. After concating the dense features from POI label and user text into one vector, we feed it into the multi-modal embedding layer to enrich the representation of the location.

Furthermore, we consider the effects of dwell time of current location. While dwell time is not directly available in the data, we simply estimate it by two steps: 1) calculating the time interval between the current location and the last location; 2) reshaping the time interval by the reasonable assumptions of human activity like it should be smaller than 12 hours in most cases. Based on the estimating dwell time t^{dw}, we design an embedding layer to convert it into dense vector t^{dwe}. As Figure 5 shows, we design a *feature cross layer* to fuse the dwell time vector with the original hidden state vector h_i, which works like the residual connection manner. We first use dwell time vector t^{dwe} to filter the original hidden state h_i by the element-wise multiplication operation. Then, we add this new vector with the original hidden state h_i as the new representation h_i^c of the hidden state. The formulation of mentioned operation is as follows,

$$h_i^c = h_i + h_i * tanh(W_t t_i^{dw} + b_t), \tag{20}$$

where h_i denotes the i_{th} hidden state output of the recurrent module, t_i^{dw} denotes the estimated dwell time of i_{th} location, W_t and b_t denote the learnable embedding parameters of dwell time embedding layer. Compared with the trainable location embedding table (e.g., 10000x128), the POI type embedding (e.g., 10x128) and dwell time (e.g., 48x128) embedding only introduce limited parameters. The text embedding is introduced based on the pre-trained Glove embedding table, which is fixed during the training.

3.3.2 Multi-task Prediction Output

As mentioned in Section 2.1, we formulate the mobility prediction task into three sub-tasks to conduct a multi-task learning process for better predicting human mobility with semantic motivation. Based on the aforementioned components, the comprehensive representation of current mobility state h_i^c before the output layer is formulated as follows,

$$h_i^c = h_i + h_i * h_i^s + h_i * t_i^{dwe}, \tag{21}$$

where h_i denotes the i_{th} output of the recurrent module for the current trajectory, h_i^s denotes the output of historical attention module, t_i^{dwe} denotes the dwell time embedding from context adaptor. This formulation is an extension of the feature cross layer in formula(20) by combining three kinds of outputs from the recurrent module, historical attention module, and context adaptor. With a comprehensive representation of current mobility state h_i^c as input, we design three independent fully connected layers with softmax function to project it into different targets to complete different tasks simultaneously. The formulation of multi-task prediction output is as follows,

$$l_{i+1} = argmax(softmax(W_l h_i^c + b_l)), \tag{22}$$
$$t_{i+1} = argmax(softmax(W_t h_i^c + b_t)), \tag{23}$$
$$e_{i+1} = argmax(softmax(W_e h_i^c + b_e)), \tag{24}$$

where W and b denote the learnable parameters of the linear layer. It is noted that due to the actual activity type is not available in the data, we regard the POI label of next location as the proxy label for activity type.

3.4 Training Algorithm

Algorithm 1 outlines the training process of DeepMove. As a multi-task learning problem, the final loss function is as follows,

$$L = L_{location} + \alpha L_{time} + \beta L_{activity}, \tag{25}$$

where α and β are the hyper-parameters for balancing the contributions of different tasks during joint training, the loss function of each task is the cross-entropy loss. We use Adam optimizer [25] to train the whole network. The historical attention module is parameterized to a feed-forward neural network that can be jointly trained with the whole model.

Algorithm 1: Training algorithm for DeepMove

Input: Time window: t_w;
Trajectory: $\{S^{u_1}, S^{u_2}, ..., S^{u_n}\}$.
Output: Trained Model $\mathcal{M}$.
//construct training instances
$\mathcal{D} \longleftarrow \emptyset$
for $u \in \{u_1, ..., u_n\}$ **do**
  for $m \in \{1, 2, ..., M\}$ **do**
    $S_c^u = S_{t_{w_m}}^u$; $S_h^u = S_{t_{w_1}}^u S_{t_{w_2}}^u ... S_{t_{w_{m-1}}}^u$
    put a training instance (u, m, S_c^u, S_h^u) into $\mathcal{D}$
initialize the parameters θ
for $i \in \{1, 2, ..., EPOCH\}$ **do**
  select instance $\mathcal{D}_m^u$ with m in order for user u from $\mathcal{D}$
  update θ by minimizing the objective with $\mathcal{D}_m^u$
  stop training when criteria is met
output trained model $\mathcal{M}$

4 Performance Evaluation

4.1 Datasets

We collect three representative real-life mobility datasets to evaluate the performance of our proposed model. The first one is the public Foursquare check-in data, the second one is a mobile application location data from a popular social network vendor, and the last one is call detail records (CDR) data from a major cellular network operator. The generation mechanism of trajectory records of three data is different, which represent three main location generation mechanisms in reality.

- Call detail records data with location records generates in the base station of the cellular network when users access it for communication and data accessing.
- Mobile application data with location records generates in the application servers when users request location service in the application like search, check-in, and so on.
- In Foursquare, users always intentionally publish their location information to share with other friends and the public, which is the check-in location.

Besides, three datasets are collected among three different cities during a different time. All of these features ensure the representativeness of our data. The basic information of three mobility datasets is presented in Table 4.1. Figure 6 shows the spatiotemporal features of the three mobility data. The details about the datasets and basic preprocessing procedure are discussed as follows.

Foursquare-NYC: This data is collected from Foursquare API from Feb. 2010 to Jan. 2011 in New York. Every record in the data consists of user ID, timestamp, GPS location, POI ID, and related text. Further, we follow the official venue category from Foursquare [1] to category different POI into different groups.

Foursquare-TKY [26]: This dataset includes long-term (about 10 months) check-in data in Tokyo collected from

1. https://developer.foursquare.com/docs/resources/categories

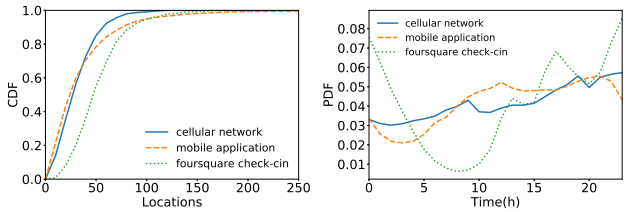


Fig. 6: Spatiotemporal features of mobility datasets.

Foursquare from 12 April 2012 to 16 February 2013. It contains an anonymized user ID, venue ID, venue category, GPS location, and timestamp.

Mobile Application Data: This data is collected from the most popular social network vendor in China. It records the location of users whenever they request the localization service in the applications. The data is collected from 17 Nov. 2016 to 31 Oct. 2016. The localization of the records is mainly achieved by GPS modules on the mobile phone and enhanced by other possible sensors. For the convenience of representation and computation, the GPS location is projected into third level street blocks (31522 blocks in Beijing), which can be represented as a street block ID.

Call Detail Records Data: This data is collected from one major cellular network operator in China. It records the spatiotemporal information of users when they access the cellular networks (i.e., making phone calls, sending short messages, or consuming data plan). The data is collected from 1 Jan. 2016 to 31 Jan. 2016. It contains more than 17000 base stations around the city.

Preprocessing: While Foursquare check-in data is sparse, we filter out the users with less than 10 records and then cut the left trajectories into several sessions based on the interval between two neighbor records. Further, we filter out the sessions with less than 5 records and the users with less than 5 sessions. Here, we choose 72 hours as the default interval threshold based on the empirical experience. Different from the sparse Foursquare check-in data, mobile application data and call detail records data are both dense daily mobility data [27, 28]. To obtain a meaningful trajectory from them, we first split the whole trajectory into different sessions by the date. Further, we split one day into 48 pieces and aggregate the records in the same time slot into one record. In practice, because of the duplication of the raw mobility data, we filter out these records during the same period of time.

To protect the privacy of the users, the base station ID, the street block ID and the user ID are all anonymous. Meanwhile, we want to point out that only the core researchers can access the data with strict non-disclosure agreements. Besides, all the data are stored in a secure local server. After processing data without leaking user privacy, we will open and publish these datasets and codes for the community.

4.2 Experimental Setup

To evaluate the accuracy of our predictive model, we compared the proposed model with several most updated methods:

Markov is widely used to predict human prediction [2, 13] for a long time. They regard all the visited locations as

Dataset	City	Duration	Users	Records	Locations	Rec./User	Loc./User
Foursquare-NYC	New York	1 year	886	82571	10497	93	33
Foursquare-TKY	Tokyo	10 months	2108	537703	21390	255	40
Mobile Application	Beijing	1 month	2407	15007511	31522	3000	48
Cellular Network	Shanghai	1 month	1075	491077	17785	456	40

TABLE 1: Basic statistics of mobility datasets.

states and build a transition matrix to capture the first-order transition probabilities between them.

PMM [14], which is recently proposed, assumes that mobility location follows a spatiotemporal mixture model and predicts the next locations with periodicity into consideration.

FPMC [29] Factorizing personalized Markov chains (FPMC) is the combination of matrix factorization and Markov chains. Further, S-BPR (Sequential Bayesian Personalized Ranking) is used in the training process of FPMC.

Geo-teaser [7] is a combination model of temporal POI embedding model for capturing the dynamic contextual information of location and geographically hierarchical pairwise preference ranking model for capturing the geographical influence.

SimpleRNN [5, 18] can be regarded as a simplification version of our model without the historical attention module, which uses LSTM as the default recurrent network unit. In the experiments of cellular network data and mobile application data, **RNN-Short** means that the trajectory is fed into the model day by day, while **RNN-Long** means that the whole trajectory lasting one month is directly fed into the model.

Parameter settings. The experiments are conducted in terms of test-train mode, where the first 80% of each users' trajectory is selected as training data, the remaining 20% as testing data. For our method, the parameters are divided into two groups: parameters for optimizer (*e.g.*, learning rate, weight decay, gradient clipping, and L2 penalty) and parameters for the model (*e.g.*, embedding size for location, time, user, POI and dwell time, the size of the hidden state). We define search space for each parameter and use the grid search method to find the best values of them. We use the similar approach to find the best parameter groups for neural network based methods (*e.g.*, RNN and Geo-teaser). For other baselines, we also use the grid search method to find the best settings for their crucial parameters.

4.3 Overall Performance

We evaluate our model with the baseline methods on four mobility datasets to present the performance of our model. We rank the candidate locations by the probabilities generated from the model and check whether the ground-truth location appears in the top-k candidate locations. While the semantic information like POI label and user text is not available in the cellular network and mobile application data, we only evaluate the enhanced DeepMove with semantic modelling in two Foursquare datasets.

We first analyze the result of Foursquare check-in data in Table 4.3. In baseline methods, SimpleRNN works better than conventional sequential models Markov and FPMC

Methods	Foursquare-NYC		Foursquare-TKY	
	Top-1	Improv.	Top-1	Improv.
Markov	0.081	-34.68%	0.124	-18.42%
FPMC	0.102	-17.74%	0.132	-13.16%
Geo-teaser	0.107	-13.71%	0.129	-15.13%
SimpleRNN	0.124	0	0.152	0
DeepMove	0.148	+19.35%	0.184	+21.05%
+C&MT	0.167	+34.68%	0.206	+35.53%

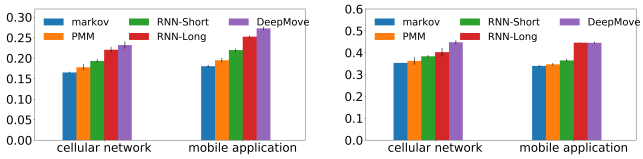
TABLE 2: Overall performance on Foursquare datasets. "+C&MT" denotes the enhanced DeepMove with semantic modelling, which contains a context adaptor and multi-task prediction output.

significantly, which due to the high-order transition modelling capacity of well designed recurrent neural networks. Furthermore, with strength in complex sequential modelling and multi-model embedding, SimpleRNN also outperforms than embedding based baseline Geo-teaser which constructs temporal POI embedding. Compared with the performance of the general RNN, we find that the prediction accuracy of DeepMove is about 20% better on average. This suggests that there indeed exist periodical regularity in human mobility, which helps to improve prediction accuracy. As the general RNN captures the complex sequential transition from the current trajectory, recurrent part of our model can also do like this. Nevertheless, our model utilizes the historical attention module to capture the periodical regularity from the lengthy trajectory history. Such an attention mechanism on the trajectory helps our model understand human mobility and achieve much better prediction accuracy. Furthermore, the enhanced DeepMove with semantic modelling via context adaptor and multi-task prediction output outperforms the original DeepMove again, which further improves the prediction performance by more than 11% respectively. The significant improvement provided by enhanced DeepMove not only shows us the necessity of understanding the semantic motivation of mobility but also testifies the effectiveness of proposed semantic modelling approaches.

Evaluation results of the other two mobility datasets also demonstrate the superiority and generalization of our model. Compared with the Foursquare check-in data, cellular neural network data, and mobile application data completely record human's daily life. As Figure 7 presents, the performance of our historical attention model outperforms the general RNN model over 8.04% on average in mobile application data. The performance gain is 5.16% on average in cellular network data, which demonstrates the generalization of our model. Compared with the general RNN, the advantage of our model is that it can capture

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

the periodical regularity of human mobility from trajectory history. In general, our model significantly outperforms all the baseline methods on three different mobility data in terms of prediction accuracy.



(a) top-1 prediction accuracy (b) top-5 prediction accuracy

Fig. 7: Performance on Cellular Network data and Mobile Application data.

Besides, we compare our models with the baseline methods on the raw schema, i.e., without deleting the duplication, of two daily mobility data. In Table 3, we can observe that our model outperforms the general RNN model by only about 1%, which achieves the prediction accuracy of 69.4%. Meanwhile, even the prediction accuracy of Markov model comes up to 50% in two mobility data. According to a further analysis of the data, we find that many users always stay in a location for several hours during the day. For this kind of trajectory, we can achieve pretty well results in location prediction only by simply copying the current input, where complex methods like attention mechanism take minor effect because of the principal influence of the current input. Apparently, the performance gain of our DeepMove will be limited. However, it will work well on the mobility data where users move around different places.

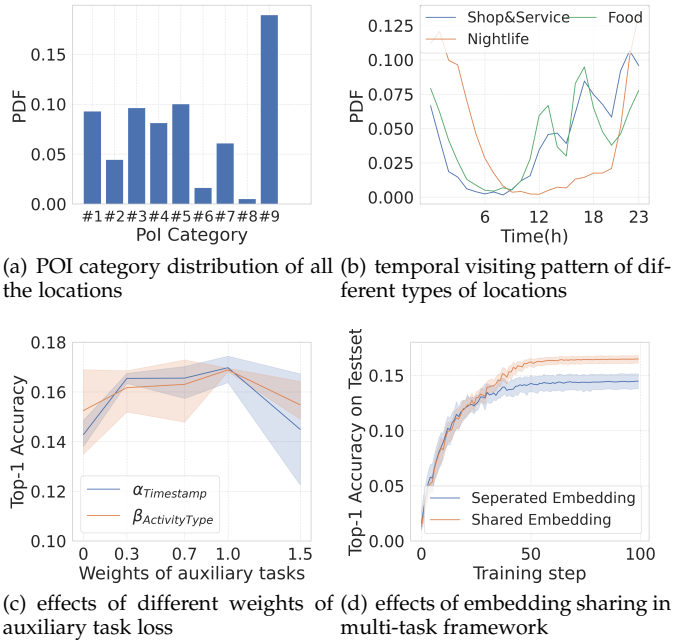
	cellular network	mobile application
Markov	0.459	0.595
RNN-Long	0.595	0.690
Our model	0.593	0.694

TABLE 3: Prediction performance on dense mobility data.

4.4 The Effectiveness of Multi-Task Design

In this section, we use Foursquare-NYC data with the richest semantic information (time, POI, and user text) among our four datasets as an example to evaluate the effectiveness of multi-task design in the mobility prediction problem.

We first directly analyze the relationship between location, POI, and time in the mobility trajectory to present the motivation of considering the next POI prediction and next visiting time as the auxiliary tasks. In our data, all the PoIs are categorized into 9 high-level classes: Shop & Service, Arts & Entertainment, Outdoors & Recreation, Nightlife Spot, Travel & Transport, College & University, Professional & Other Places, Residence, and Food. Based on the above classification, Figure 8(a) presents the visiting distribution of locations with different POI categories. Furthermore, we present the normalized temporal pattern of three kinds of POI: shop & service, nightlife, and food in Figure 8(b). Based on the results in Figure 8(a) and Figure 8(b), we find that the visiting probability of each location is highly related to its POI type and visiting time. In other words, if we can



(a) POI category distribution of all the locations (b) temporal visiting pattern of different types of locations

(c) effects of different weights of auxiliary task loss (d) effects of embedding sharing in multi-task framework

Fig. 8: The motivation and effectiveness of multi-task design on Foursquare-NYC data.

successfully predict the POI type and visiting time of the next location, the prediction accuracy of the next location should also be improved.

As shown in Section 3.3.2 and 3.4, we add two branches for the auxiliary tasks and optimize the whole model with the weighted loss function of three tasks. With fixing the weights of the loss of next location prediction task and other parameters, we try different values for α_{time} and $\beta_{activity}$ in our model and present the results in Figure 8(c). As Figure 8(c) shows, different values of α_{time} and $\beta_{activity}$ influence the location prediction performance significantly. Compared with the single task framework, proper weights for auxiliary tasks will improve the prediction accuracy from around 0.15 to higher than 0.16. Besides, too larger weights of auxiliary tasks can hurt the performance of the main task, e.g., when weights are set as 1.5 the performance of the whole model is as low as the performance without the multi-task design.

Following the common practice of multi-task, we use the shared encoder structure in the model and only design different branches at the end of the network. To evaluate the necessity of shared embedding design, we design separate embedding modules for each task and only share the recurrent module and historical attention module between tasks. As shown in Figure 8(d), the performance of the model with separate embedding design is significantly inferior to the model with shared embedding design by decreasing from 0.167 to 0.15. While three tasks share the same data structure and semantic space, sharing embedding design is parameter efficient and also provides more chances to enable the feature fusion between different tasks.

Finally, we also test the effectiveness of context adaptor and the result is presented in Figure 9. As Figure 9 shows, after removing the context adaptor (blue curves in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

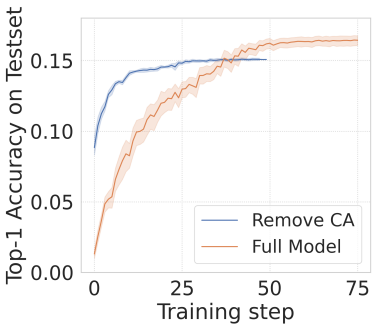


Fig. 9: The effectiveness of context adaptor (denoted as CA in the figure).

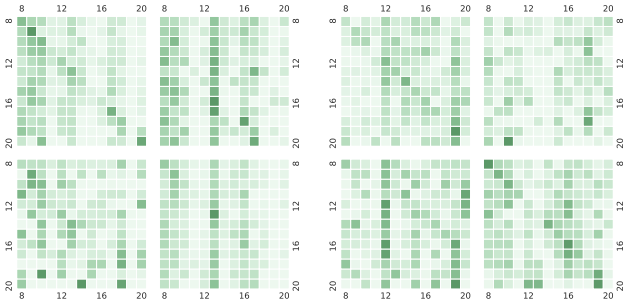
figure), the performance of our model decreases from 0.167 to around 0.15, which demonstrates the effectiveness of the context adaptor for feature fusion. Besides, we also find that the convergence rate of our model is slow down after adding context adaptor, which is caused by the additional parameters and computing for the high-order feature cross.

In summary, we observe the tight relation between three tasks from the mobility data and design an effective multi-task framework accordingly. With fully leveraging the signal from the auxiliary tasks by controlling the weights of their loss and cooperating with the context adaptor, the multi-task design can improve the next location prediction accuracy of our model by more than 11%.

4.5 Reason Interpretation: Visualization of Historical Attention Weights

Because of the importance of the periodicity of human mobility, our model, especially the historical attention module, is designed to capture the periodicity of human mobility. Thus, in this section, we discuss whether our periodicity assumption appears and whether our model really captures it.

In Figure 10, we visualize the output of historical attention module to demonstrate this. To obtain the visualization, we first collect the normalized weight of historical attention module for a little seed users, and align them together on the time dimension. Then, we re-normalize the weights and draw them in Figure 10 in terms of the heatmap. The horizontal axis and vertical axis of every square matrix in Figure 10 are both time period, the shade of the grids describe the weight, where the deeper green means the larger weight. For example, the top-left square matrix in Figure 10(a) shows us the distribution of the historical attention weight from 8.am. to 8. pm. during the weekday via the weekday's historical trajectories in mobile application data. The diagonal entries of it are remarkably larger than other entries, which shows the day-level regularity of human mobility in the different workday. The top-left square matrix in Figure 10(b) shows a similar result, while it is based on another cellular network data. The bottom-right square matrix in Figure 10(b) shows the attention distribution in the weekend in cellular network data, which also reveals a remarkably day-level regularity. In general, the results of Figure 10 show that our model indeed captures the regularity and periodicity from the historical trajectory. Meanwhile, our historical attention module not only improves the prediction accuracy but also offers an easy-to-interpret way to understand which historical activities are emphasized in the future mobility.



(a) mobile application data (b) cellular network data

Fig. 10: Visualization of the output of the historical attention module. Every matrix presents the correlations between the current trajectory and historical trajectory. The diagonal entries of the matrix present the correlations of trajectories in the same time period of different days. The shade of the grids describes the weight where the deeper green means the larger weight. For example, the top-left matrix in (a) shows the correlations of the current trajectory and the historical trajectory on workdays on cellular network data. The highlight diagonal entries tell us that trajectories during the workday are periodical. The bottom-right matrix in (b) tells us that trajectories during the weekend are even more periodical than the workday.

4.6 Model Variations

In order to present the efficiency of the historical attention module, we first compare two proposed historical attention modules in terms of prediction accuracy and computation efficiency and then discuss how different sampling strategies in the embedding encode attention module can influence the final results. Finally, we discuss the effect of user embedding and present the effectiveness of our model in describing personal preference.

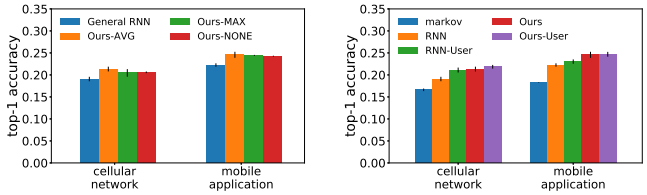
We compare the performance and efficiency of our proposed two historical attention module on two datasets. The results are presented in Table 4. The sequential encode attention module works better than the embedding encode attention module in most of the time especially in mobile application data, while the latter one computes more efficiently. Two reasons may account for the better performance of the sequential encode attention module: 1) it captures sequential information along the lengthy trajectory to some extent, while the embedding encoder cannot; 2) the latent space of output of it is more similar to the current mobility status's because of the similar generation structure.

Besides, we evaluate the system performance of different sampling strategies in the sampling layer of embedding encode attention module. As mentioned in the model section, we design three kinds of sampling strategies in the historical attention module: average sampling, maximum sampling, and none sampling. Figure 11(a) shows the evaluation results of three different samplings in two datasets in terms of top-1 prediction accuracy. In general, the average sampling

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering

strategy works better among three strategies, while the maximum sampling strategy performs a little worse, the result shows us that most people own regular mobility pattern and limited locations to visit. The performance gap of three strategies on cellular network data are larger than the mobile application data.

Finally, we identify every single user with a user ID and add a user ID embedding feature to the model to capture the personality. The results are presented in Figure 11(b). Obvious performance gain can be observed in the general RNN model after adding the user ID embedding. However, the performance gain of our model can be omitted, which demonstrates that our proposed model not only capture deeper periodical pattern but also characterize personal regularities.



(a) sampling strategy (b) user embedding

Fig. 11: Performance variation with the model design.

Dataset	Model	Accuracy	Overhead(s)
cellular network	attention 1	0.22	≈600
	attention 2	0.24	≈1600
mobile application	attention 1	0.24	≈2600
	attention 2	0.27	≈11200

TABLE 4: Efficiency of two attention models.

4.7 Evaluation on User Groups

In order to evaluate the variation of performance gain among different users, we cluster them into different groups based on three rules: mobility entropy [30], explore ratio, the radius of gyration [6]. Mobility entropy calculates the entropy of locations in trajectory, which is related to the regularity level of human mobility. Explore ratio represents the fraction of new locations in test data, which do not exist during the training. Thus, one person with more regular behaviors should have lower mobility entropy and lower explore ratio. The final rule is the radius of gyration which describes the spatial range of the mobility.

The evaluation results are presented in Figure 12, where the vertical axis shows the performance gain compared with the baseline method-general RNN. There are two interesting insights from the result: 1) our model outperforms the baseline method on almost all kinds of users; 2) our model predicts non-regular users better than the baseline method that meets the goal of our historical attention module. For example, in the top-left image of Figure 12, our model's prediction accuracy increases when the mobility entropy of users increases. With the effective usage of lengthy historical

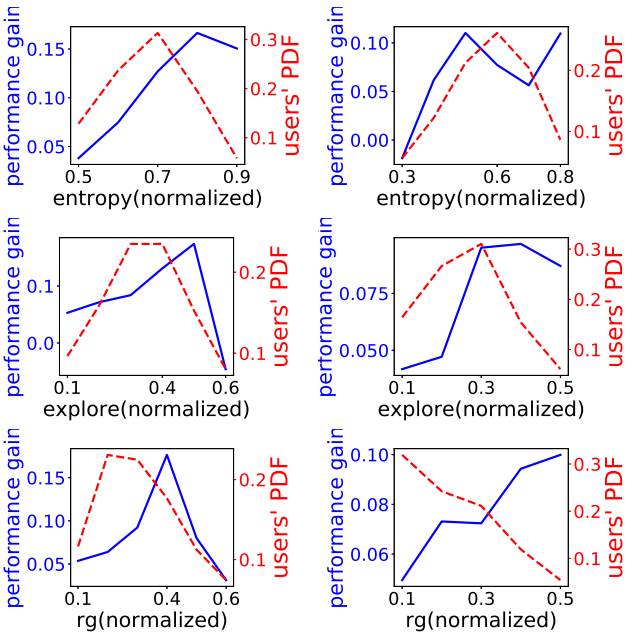


Fig. 12: Performance varies with mobility entropy/explore ratio/radius of gyration (rg) on cellular network data (left) and mobile application data (right).

trajectory, our model captures the underlying and deeper periodical patterns of human mobility.

In summary, extensive results on four mobility datasets demonstrate the superiority of our proposed model compared with the baselines. Specifically, the proposed semantic-based multi-modal and multi-task learning framework further improve the prediction performance of DeepMove on the mobility dataset with abundant semantic information.

4.8 The Effect of Parameters

In order to evaluate the effects of hyper-parameters on the prediction, there are four parameters investigated in Figure 13. Three of them are the embedding size of trajectory input (location, time, user ID) and the left one is the size of the hidden layer. Location embedding size is critical for embedding a one-hot location vector with spatial correlation. As Figure 13(a) shows, we find that the size of 300 performs best among other settings. With the length of location embedding getting longer or shorter, the accuracy of three datasets will decrease. As for time embedding, experimental results from Figure 13(b) show that smaller size has better performance, indicating that time embedding should not take up a large part in the input vector. The curves in Figure 13(c) are flatter, which shows that User ID embedding is relatively stable with embedding size, which manifests a less important status than location and POI embedding. As shown in Figure 13(d), our model is more sensitive to hidden layer size compared with other parameters. The accuracy curve declined sharply with the hidden layer size exceeds 150, which is the optimal value for three datasets.

To further understand the role of location embedding, we check whether the location embedding module in our

model really learns the spatial dependencies. We cluster the embedding weights of the location embedding module of the model in different training stages (before the training and after the training) by the k-means algorithm in the high-dimensional space in neural network. For example, the clustering results of cellular network data are presented in Figure 14, the location belongs to the same cluster in the high-dimensional space will be rendered with the same color. As we can observe from Figure 14(a), the regions close to each other in the physical space always are not rendered with the same color, which indicates that the location embedding module with one-hot input does not know about the spatial dependencies. In Figure 14(b), we can observe remarkable clusters in the physical world, which are similar to the clusters in the high-dimensional space in the neural network. The well-organized clusters in Figure 14(b) demonstrate that our model really learns about the spatial dependencies in the training.

5 RELATED WORK

Related works can be classified into two categories: model-based methods and pattern-based methods. Besides, we introduce related works on RNN and attention model.

Conventional methods Markov model and its variations are one of the most common models of these approaches. In Markov-based models [13, 31], they model the probability of future action by building a transition matrix between several locations based on the past trajectories. To capture the unobserved characteristics between location transition, Mathew et al. [2] cluster the locations from the trajectories and then train a Hidden Markov Model for each user. Considering the mobility similarity between user group, Zhang et al. [3] propose GMove: a group-level mobility modeling method to share significant movement regularity. Different from existing Markov-based models, our model can model time-dependent and high order transitions. Pattern-based methods [8, 9, 12, 28, 32] are another branch of previous works, which first discover the popular sequential mobility patterns from the trajectory, and then try to predict the mobility based on these popular patterns. Matrix factorization can also be regarded as a kind of pattern discovered method. Matrix factorization (MF) [10, 33] emerges from the recommendation system and the basic idea of it is to factorize the users-items matrix into two latent matrices that represent the users and items characteristics. Cheng et al. [11] fuse MF with the geographical influence by modeling the location probability as a multi-center Gaussian Model. Combining Markov model with matrix factorization, Rendle et al. [29] propose the factorized personalized Markov model (FPMC) to do item recommendation. Based on FPMC, Cheng et al. [4] propose a matrix factorization method named FPMC-LR to capture the sequence transition with Markov chain while considering the localized region constraint. Compared with pattern-based methods, our model can not only model the transitional regularities shared by all the users but also model the personal preference based on the user embedding and personal historical trajectory.

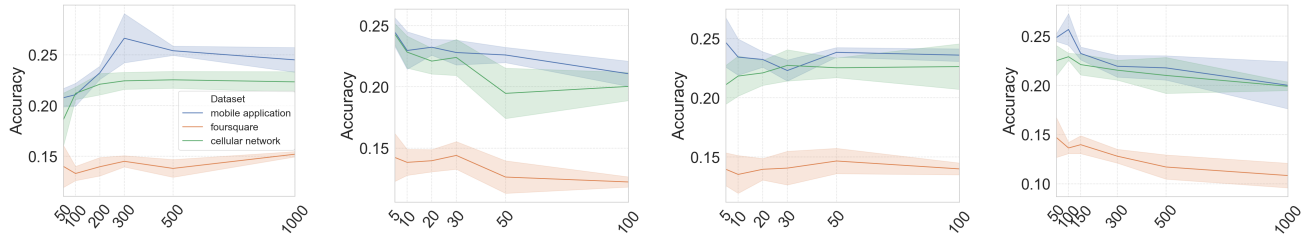
Deep Learning methods Recurrent Neural Networks (RNN) [34, 35] is a powerful tool to capture the long-range dependencies in the sequence and has achieved success in Natural Language Processing (NLP) [21, 36], Image Caption [37], etc. Because of its powerful representation ability, RNN have been applied to many fields like click prediction [38], recommendation system [39, 40], and mobility prediction [5, 41, 42, 43, 44, 45]. Zhao et al. [7, 46] propose Geo-teaser to build temporal POI embedding for better next POI recommendation, which focuses on capturing the temporal variation of different features. Liu et al. [5] propose Spatial-Temporal Recurrent Neural Networks (ST-RNN) to model temporal and spatial contexts. However, the proposed model is too complicated to train and apply with so many parameters. Besides, it can not be applied to the discrete location prediction scene because of its continuous spatial modeling method. Du et al. [41] propose Recurrent Marked Temporal Point Process (RMTPP) to learn the conditional intensity function automatically from the history. However, this model is not specific for the trajectory prediction and does not consider the natural characteristics of trajectory like multi-level periodicity. By coupling convolutional and recurrent neural network, the Yao et al. [47] propose DeepSense: a unified deep learning framework for mobile sensing data. However, this model needs uniform sampling data and also does not consider the multi-level periodicity of trajectory.

After the first try in DeepMove [16], Zhao et al. [48] propose to use Bi-LSTM with attention to better understand the sub-trajectory for destination prediction and Altaf et al. [49] design two independent spatial and temporal attention unit for better location prediction via attention. The attention unit becomes widely used basic trajectory sequence modeling unit [48, 49, 50, 51]. Besides, the role of our historical attention is similar to the user memory network in RUM [52]. Different from RUM, our method focuses on the sequential transitions between mobility. However, the explainable prediction of human mobility is far from the application. In this paper, different from only relying on attention weights, we explore the semantically motivated location prediction as another try for explainable prediction. Multi-task learning [53] leverages useful information contained in multiple related tasks to help improve the generalization performance of all the tasks or one target task. Yang et al. [42] propose to jointly model social networks and mobile trajectories. Liu et al. [54] model multiple types of behaviors in historical sequences and achieved better performance than only modeling a single type of behavior. Different from them, we propose to predict the POI label of location as an auxiliary task for better mobility prediction, which is more generalized and intuitive for human mobility prediction.

6 CONCLUSION

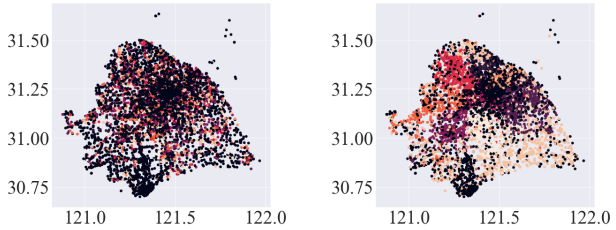
In this paper, we investigate the problem of mobility prediction from the sparse and lengthy trajectories. We propose an attentional mobility model, which enjoys two novel characteristics compared to previous methods: 1) a multi-modal embedding recurrent neural network for capturing multiple factors that govern the transition regularities of human mobility; and 2) a historical attention module for modeling the multi-level periodicity of human mobility.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2020.3006048, IEEE Transactions on Knowledge and Data Engineering



(a) location embedding size (b) time embedding size (c) user embedding size (d) hidden layer size

Fig. 13: The effects of different parameters on different mobility datasets.



(a) before training (b) after training

Fig. 14: The clustering results of location embedding weights after training on cellular network data.

Further, we extend it by modelling the semantic motivation of human mobility by a context adaptor and multi-task learning prediction output. Extensive experiments demonstrate that proposed models significantly outperform all the baselines on four datasets. Meanwhile, the visualization of historical attention weights shows that DeepMove is able to effectively capture meaningful periodicities for mobility prediction.

There are several future directions for our work. First, we currently only predict the next location because we fix the time interval in practice. In the future, we plan to expand the location prediction into the spatiotemporal point prediction by taking the potential duration into consideration. Second, while the proposed usage of historical attention unit and multi-task learning framework in our model enables the implicit interpretable mobility prediction, the explicitly explainable mobility prediction is still a challenging task. Third, we directly use the high-dimensional one-hot representation of location in this paper and how to build a unified and effective location representation for better mobility prediction is also an interesting topic.

REFERENCES

[1] C. Song, Z. Qu, N. Blumm, and A. L. Barabási, "Limits of predictability in human mobility," *Science*, 2010.

[2] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden markov models," in *ACM Conference on Ubiquitous Computing*, 2012.

[3] 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 SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[4] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: successive point-of-interest recommendation," in *International Joint Conference on Artificial Intelligence*, 2013.

[5] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[6] M. C. González, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns," *Nature*, 2008.

[7] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *Proceedings of the 26th international conference on world wide web companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 153–162.

[8] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext:a location predictor on trajectory pattern mining," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July*, 2009.

[9] F. Pinelli, F. Pinelli, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, 2009.

[11] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *AAAI Conference on Artificial Intelligence*, 2012.

[12] C. Zhang, J. Han, L. Shou, J. Lu, and T. La Porta, "Splitter: Mining fine-grained sequential patterns in semantic trajectories," *Proc. VLDB Endow.*, 2014.

[13] M. O. Killijian, "Next place prediction using mobility markov chains," in *The Workshop on Measurement, Privacy, and Mobility*, 2012.

[14] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[15] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, and

J. Han, "Pred: Periodic region detection for mobility modeling of social media users," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017.

[16] 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*.

[17] C. Parent, S. Spaccapietra, C. Renso, G. L. Andrienko, N. V. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. A. F. de Macêdo, N. Pelekis, Y. Theodoridis, and Z. Yan, "Semantic trajectories modeling and analysis," *ACM Comput. Surv.*, vol. 45, 2013.

[18] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies in Computational Intelligence*, vol. 385, 2008.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, 1997.

[20] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Computer Science*, 2014.

[22] H. Wang and Z. Li, "Region representation learning via mobility flow," in *CIKM*, 2017.

[23] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Computer Science*, 2015.

[24] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[26] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2015.

[27] F. Xu, P. Zhang, and Y. Li, "Context-aware real-time population estimation for metropolis," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016.

[28] 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 *International Conference on World Wide Web*, 2017.

[29] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010.

[30] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the gap between physical location and online social networks," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, 2010.

[31] M. Chen, Y. Liu, and X. Yu, "Nlpmm: A next location predictor with markov modeling," vol. 8444, 2014.

[32] J. C. Ying, W. C. Lee, T. C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *ACM Sigspatial International Conference on Advances in Geographic Information Systems*, 2011.

[33] X. Su and T. M. Khoshgoftaar, *A survey of collaborative filtering techniques*. Hindawi Publishing Corp., 2009.

[34] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *Computer Science*, 2015.

[35] A. Graves, "Generating sequences with recurrent neural networks," *Computer Science*, 2013.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," vol. 4, 2014.

[37] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *Computer Science*, 2015.

[38] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T. Y. Liu, "Sequential click prediction for sponsored search with recurrent neural networks," in *28th AAAI Conference on Artificial Intelligence*, 2014.

[39] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *Computer Science*, 2015.

[40] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 66, 2018.

[41] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[42] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, 2017.

[43] H. Mei and J. Eisner, "The neural Hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, 2017.

[44] D. Yao, C. Zhang, J. Huang, and J. bi, "Serm: A recurrent model for next location prediction in semantic trajectories," in *ACM International on Conference on Information and Knowledge Management*, 11 2017.

[45] J. K. X. T. Y. J. S. L. P. G. J. Y. Huaxiu Yao, Fei Wu and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of Thiry-Second AAAI Conference on Artificial Intelligence*, 2018.

[46] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[47] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[48] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, and F. Zhu, "On prediction of user destination by sub-trajectory understanding: A deep learning based approach," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[49] B. Altaf, L. Yu, and X. Zhang, "Spatio-temporal attention based recurrent neural network for next location

prediction," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 937–942.

[50] J. Feng, M. Zhang, H. Wang, Z. Yang, C. Zhang, Y. Li, and D. Jin, "Dplink: User identity linkage via deep neural network from heterogeneous mobility data," in *WWW '19*, 2019.

[51] K. Zhao, J. Feng, Z. Xu, T. Xia, L. Chen, F. Sun, D. Guo, D. Jin, and Y. Li, "Deepmm: Deep learning based map matching with data augmentation," *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019.

[52] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in *Proceedings of the 11th ACM international conference on web search and data mining*. ACM, 2018.

[53] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[54] Q. Liu, S. Wu, and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1254–1267, 2017.

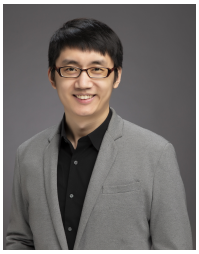
7 BIOGRAPHY

Jie Feng is a Ph.D. candidate at the Department of Electronic Engineering of Tsinghua University, advised by Prof. Yong Li. He received his 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 the spatial-temporal data mining filed to improve the performance of the 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.

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.



Qiang Qiu is currently an engineer of the CAS Key Lab on Network Data Science & Technology, Institute of Computing Technology, CAS. He received the B.S. degree in software engineering from Shandong University, Jinan, China, in 2010 and the Ph.D. degree in computer application technology from University of Chinese Academy of Sciences, Beijing, China, in 2015. His research interests include big data mining, spatial data analysis, geographic information system and parallel computing. He has published over 20 papers in his major and has rich project experiments, including National Science Foundation for Young Scientists of China, National High-tech R&D Program (863 Program) of China, and National Major Projects of China.

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 the 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.