

AttnMove: History Enhanced Trajectory Recovery via Attentional Network

Tong Xia¹, Yunhan Qi¹, Jie Feng¹, Fengli Xu¹, Funing Sun², Diansheng Guo², Yong Li^{1*}

¹Beijing National Research Center for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University.

²Tencent Corporation.

Emails: xia-t17@tsinghua.org.cn, liyong07@tsinghu.edu.cn

Abstract

A considerable amount of mobility data has been accumulated due to the proliferation of location-based service. Nevertheless, compared with mobility data from transportation systems like the GPS module in taxis, this kind of data is commonly sparse in terms of individual trajectories in the sense that users do not access mobile services and contribute their data all the time. Consequently, the sparsity inevitably weakens the practical value of the data even it has a high user penetration rate. To solve this problem, we propose a novel attentional neural network-based model, named AttnMove, to densify individual trajectories by recovering unobserved locations at a fine-grained spatial-temporal resolution. To tackle the challenges posed by sparsity, we design various intra- and inter-trajectory attention mechanisms to better model the mobility regularity of users and fully exploit the periodical pattern from long-term history. We evaluate our model on two real-world datasets, and extensive results demonstrate the performance gain compared with the state-of-the-art methods. This also shows that, by providing high-quality mobility data, our model can benefit a variety of mobility-oriented down-stream applications.

Introduction

Widely adopted location-based services have accumulated large-scale human mobility data, which have great potential to benefit a wide range of applications, from personalized location recommendations to urban transportation planning (Zheng et al. 2014). Nevertheless, since users may not allow the service provider to collect their locations continuously, individual trajectory records in such data are extremely sparse and unevenly distributed in time, which inevitably harms the performance of downstream applications, even when it has a notable user penetration rate and covers a long period of time (Gao et al. 2019). For example, with a limited number of trajectory records per day, it is difficult to predict an individual’s next location and recommend proper points-of-interests (Xi et al. 2019). As for collective mobility behaviour, because personal location records are missing for most of the time, it is also hard to produce exact estimates of hourly crowd flow in cities for emergency

response (Li, Li, and Li 2013). Therefore, it is of great importance to rebuild individual trajectories at a fine-grained spatial-temporal granularity by imputing missing or unobserved locations.

One commonest solution to this problem is to impute the missing value by treating individual trajectories as two-dimensional time series with latitude and longitude at each timestamp (Alwan and Roberts 1988; Moritz and Bartz-Beielstein 2017). As such, smoothing filters (Alwan and Roberts 1988; Moritz and Bartz-Beielstein 2017) and LSTM-based models (Cao et al. 2018; Wang et al. 2019a) have been proposed. Their performance is acceptable when only a small percentage of locations are missing due to limited movement during a short time span. However, in highly sparse scenarios, their performance degrades significantly, since they fail to effectively model complex mobility regularity. Another line of study is to model users’ transition regularity among different locations, so as to generate the missing locations according to the highest transition probability from observed location (Liu et al. 2016; Wang et al. 2019a). But this kind of strategy is still insufficient in the sense that the observed records are unevenly distributed in time in LBS data, and thus transition regularity is incapable of inferring those locations which are continuously unobserved.

Luckily, human mobility has some intrinsic and natural characteristics like periodicity and repeatability, which can help to better rebuild the trajectory (Gonzalez, Hidalgo, and Barabasi 2008; Schneider et al. 2013; Cho, Myers, and Leskovec 2011). In this regard, a promising direction is to leverage long-term mobility history, i.e., mobility records prior to the targeted trajectory to be recovered, considering both that daily movements are spatially and temporally periodic, and that collecting long-term data is quite achievable in LBS. In addition, previous work has also shown that explicitly utilizing historical trajectory can help with next location prediction (Feng et al. 2018), which is similar to our task. All of this has inspired us to design a history enhanced recovery model.

However, trajectory recovery is still challenging for the following reasons: first of all, the high sparsity of both targeted day and historical trajectories hinder us from inferring the missing locations by spatial-temporal constraints, i.e., how far a user can move in the unobserved periods. Because of the high uncertainty between two consecutive

*This is the corresponding author.

records in a sparse trajectory, a framework which can better model the mobility pattern and reduce the number of potentially visited locations is needed. The second is how to distill periodical features from huge historical data effectively, considering that real-world historical data includes a great deal of noise. Researchers have proposed detecting the locations of the home and the workplace from historical trajectories, in order to build the basic periodic pattern. This is straightforward but insufficient because other locations are neglected (Chen et al. 2019). Another way is to directly exploit the most frequently visited location at the targeted time slot from multiple historical trajectories as imputation (Liu et al. 2016). However, historically more popular locations may not be the ones which are missing on any targeted day because mobility is influenced by many factors and thus some locations can only be visited occasionally. Thus, deciding how well to rely on history is the third challenge.

Keeping the above challenges in mind, we aim to propose a novel attentional neural network-based mobility recovery model named **AttnMove**, which is dedicated to handling sparsity and exploiting long-term history. For clarity, we define the targeted day’s trajectory as the current trajectory and any trajectories before the targeted day as historical trajectories. The proposed AttnMove model can be broken down into three key components, which address the main challenges accordingly. Firstly, to capture the mobility pattern and indicate the most likely visited areas for the missing records, we design a *current processor* with *intra-trajectory attention* mechanism to initially fill in the blanks of the current trajectory. We choose attention mechanism rather than RNN structure for the reason that sparse trajectories have few sequential characteristics and all the observed locations should be considered equally regardless of the visited order. For example, a user’s locations at 6 AM, 9 AM, 3 PM and 6 PM are observed: to recover the location at 9 PM, we should give priority to the location at 6 AM and the adjacent area (i.e., home area) instead of at 6 PM (i.e., commuting time). For the second challenge, we design a *history processor* with another *intra-trajectory attention* to distill periodical features from multiple historical trajectories. By aggregating more information from the long-term history can be leveraged for recovery. Finally, to fuse the features from current and history processors and generate missing locations, we propose a *trajectory recovery* module with *inter-trajectory attention* and *location generation attention* mechanisms. To be specific, the former mechanism yields attention weights to select locations from history based on current mobility states, and the later further considers spatial-temporal constraints from observed locations to better rebuild the trajectory.

Overall, our contributions can be summarized as follows:

- We study the problem of rebuilding individual trajectories with a fine-grained spatial-temporal granularity from sparse mobility data by explicitly exploiting long-term history, which brings high-quality mobility data to facilitate a wide range of down-stream applications.
- We propose a novel attentional neural network model AttnMove for missing location recovery, behind which the

key idea is to utilize attention mechanism to model mobility regularities and extract periodical patterns from multiply trajectories of different days, and then fuse them to rebuild the trajectories.

- We conduct extensive experiments on two real-life mobility datasets. Results show that our AttnMove model significantly outperforms state-of-the-art baselines in terms of improving recovery accuracy by 4.0%~7.6%¹.

Preliminaries

In this section, we introduce the definition and notations we use in this paper.

Definition 1 (Trajectory). We define a trajectory as a user’s time-ordered location sequence in one day. Let $\mathcal{T}_u^n : l_u^{n,1} \rightarrow l_u^{n,2} \dots \rightarrow l_u^{n,t} \dots \rightarrow l_u^{n,T}$ denote user u ’s n -th day’s trajectory, where $l_u^{n,t}$ is the location of t -th time slot for a given time interval (e.g., every 10 minutes). Note that if the location of time slot t is unobserved, $l_u^{n,t}$ is denoted by null, named missing location.

Definition 2 (Current and Historical Trajectory). Give a targeted day n and user u ’s trajectory \mathcal{T}_u^n , we define \mathcal{T}_u^n as the user’s current trajectory, and the historical trajectories are defined as u ’s trajectories in the past days, i.e., $\{\mathcal{T}_u^1, \mathcal{T}_u^2, \dots, \mathcal{T}_u^{n-1}\}$.

When most of the locations in current day’s trajectory \mathcal{T}_u^n are missing, exploiting history is beneficial for the recovery. For this reason, we are motivated to formulate the investigated problem as:

Problem (History Enhanced Trajectory Recovery). Given user u ’s trajectory \mathcal{T}_u^n with the historical trajectories $\{\mathcal{T}_u^1, \mathcal{T}_u^2, \dots, \mathcal{T}_u^{n-1}\}$, recover the missing locations, i.e., \forall null in \mathcal{T}_u^n , to rebuild the current day’s complete trajectory.

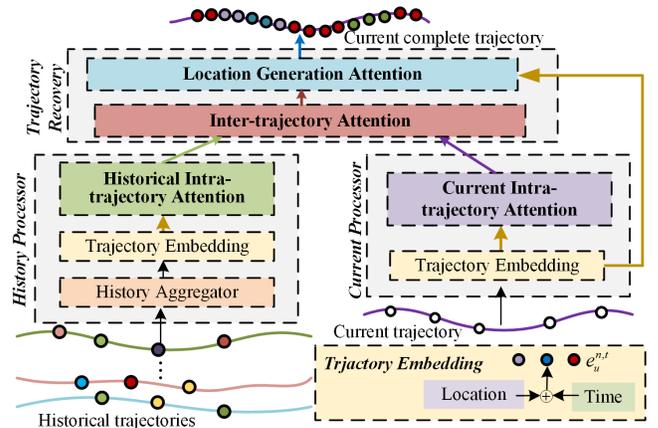


Figure 1: Main architecture of the proposed AttnMove model, where current and historical trajectories are first processed separately, and then are fused to generate locations for missing time slots.

¹Codes are available in <https://github.com/XTxiatong/AttnMove>

AttnMove: Model Design

To solve the above defined problem, we devise a novel model AttnMove, which first processes historical and current day’s trajectories separately, and then integrates their features to comprehensively determine the location to be recovered. The architecture of AttnMove is illustrated in Fig. 1. In our AttnMove, to projects sparse location and time representations into dense vectors which are more expressive and computable, a *trajectory embedding module* is designed as a pre-requisite component for other modules. Then, in order to extract periodical patterns, multiply historical trajectories are feed into a *history processor* to be aggregated. Paralleled with *history processor* is a *current processor*. We design it to enhance spatial-temporal dependence to better model mobility regularities. Finally, to fuse historical and current trajectories from the above modules and generate locations as recovery, a *trajectory recovery module* is proposed as the finally component. In the following, we elaborate on the details of those four modules.

Trajectory Embedding Module

To represent the spatial-temporal dependency, we jointly embed the time and location into dense representations as the input of other modules. Specifically, for each location $l \in \mathcal{L}$ including the missing one *null*, we set up a trainable embedding vector $e_l \in \mathbb{R}^d$. All location embeddings are denoted as a matrix $E_l \in \mathbb{R}^{|\mathcal{L}+1| \times d}$. As for mobility modeling, temporal information is also important, therefore we also set up embeddings for time slots. Following (Vaswani et al. 2017), for each time slot t , we generate its embedding as follows,

$$\begin{cases} e_t(2i) = \sin(t/10000^{2i/d}), \\ e_t(2i+1) = \cos(t/10000^{2i/d}), \end{cases} \quad (1)$$

where i denotes the i -th dimension. The time embedding vectors have the same dimension d with location embedding. Finally, for each time slot t , we sum the time and location embedding vectors into a single one, denoted by $e_u^{n,t} \in \mathbb{R}^d$ as follows,

$$e_u^{n,t} = e_l + e_t, \quad (2)$$

which also benefits the follow-up computation as a lower dimensional vector than the original one-hot ones.

History Processor

When exploiting the historical trajectory, information provided by a single historical trajectory with one day’s observed locations is insufficient to capture periodical characters as each day’s records are sparse. To utilize multiple trajectories jointly, we design a *history aggregator* as follows,

$$\mathcal{T}_u^p = \mathcal{T}_u^1 \oplus \mathcal{T}_u^2 \dots \oplus \mathcal{T}_u^{n-1}, \quad (3)$$

where \oplus means to extract the location with highest visiting frequency in the corresponding time slot. Then, \mathcal{T}_u^p is embedded to e_u^p with each time slot represented by $e_u^{p,t}$ according to (2). As such, \mathcal{T}_u^p is less sparse than any single historical trajectory. However, as it is possible that user u did not generate any record at a certain time slot previously,

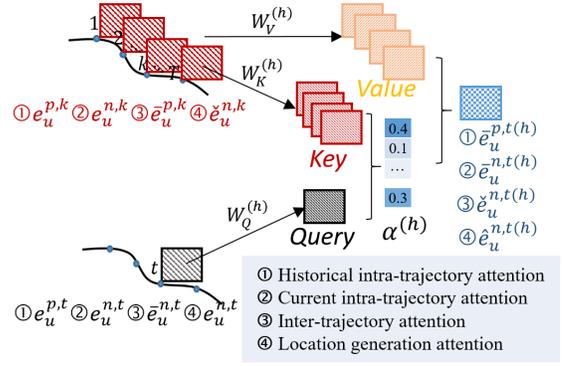


Figure 2: The architecture of proposed trajectory attention mechanisms under one head, where the new representation is a combination of embeddings of value time slots conditioned on attention weights $\alpha_{t,k}^{(h)}$ from query and key time slots. Note that superscript p denotes history while n is for the current trajectory.

locations can still be missing after aggregating. Therefore, we further propose the following mechanism to infill \mathcal{T}_u^p :

Historical Intra-trajectory Attention. Intuitively, the missing location can be identified by its time interval and distance to other observed locations. Take a simple example, $l_u^{p,4}$ is most likely in the geo-center of $l_u^{p,3}$ and $l_u^{p,5}$. In the embedding space, this can be expressed by $e_u^{p,4} = 0.5e_u^{p,3} + 0.5e_u^{p,5}$. However, the actual scenario would be much more complex than this as people do not move uniformly. Therefore, we use a multi-head attentional network to model the spatial-temporal relation among trajectory points. Specifically, we define the correlation between time slot t and k under head h as follows,

$$\alpha_{t,k}^{(h)} = \frac{\exp(\phi^{(h)}(e_u^{p,t}, e_u^{p,k}))}{\sum_{g=1}^T \exp(\phi^{(h)}(e_u^{p,t}, e_u^{p,k}))}, \quad (4)$$

$$\phi^{(h)}(e_u^{p,t}, e_u^{p,k}) = \langle W_Q^{1(h)} e_u^{p,t}, W_K^{1(h)} e_u^{p,k} \rangle, \quad (5)$$

where $W_Q^{1(h)}, W_K^{1(h)} \in \mathbb{R}^{d' \times d}$ are transformation matrices and $\langle \cdot, \cdot \rangle$ is the inner product function. Next, we generate the representation of time slot t under each head via combining all locations in other time slots guided by coefficient $\alpha_{t,k}^{(h)}$:

$$\tilde{e}_u^{p,t(h)} = \sum_{k=1}^T \alpha_{t,k}^{(h)} (W_V^{1(h)} e_u^{p,k}), \quad (6)$$

where $W_V^{1(h)} \in \mathbb{R}^{d' \times d}$ is also a transformation matrix. Furthermore, we make use of different combinations which models different spatial-temporal dependence and collect them as follows,

$$\tilde{e}_u^{p,t} = \tilde{e}_u^{p,t(1)} \parallel \tilde{e}_u^{p,t(2)} \parallel \dots \parallel \tilde{e}_u^{p,t(H)}, \quad (7)$$

where \parallel is the concatenation operator, and H is the number of total heads. To preserve the representation of raw locations, we add a standard residual connections in the network. Formally,

$$\bar{e}_u^{p,t} = \text{ReLU}(\tilde{e}_u^{p,t} + W^1 e_u^{p,t}), \quad (8)$$

where $W^1 \in \mathbb{R}^{d^H \times d}$ is the projection matrix in case of dimension mismatching, and $ReLU(z) = \max(0, z)$ is non-linear activation function.

With the historical intra-trajectory attention layer, the representation of history trajectory e_u^p is updated into a more expressive form \bar{e}_u^p , which is shown in Fig. 2. We can stack multiple such layers with the output of the previous layer as the input of the next one. By doing this, we can extract more complex mobility features from historical trajectories.

Current Processor

When recovering the trajectory for user u , it is necessary to consider the locations visited before and after the missing time slots, which enclosure the geographical area of the missing locations. Since locations can be missed for several consecutive time slots, the spatial constraint is weak. Therefore, we conduct intra-trajectory attention on the current trajectory \mathcal{T}_u^n to capture the current day's mobility pattern:

Current Intra-trajectory Attention. The first step is to embed \mathcal{T}_u^n into a dense representation e_u^n via the aforementioned mobility embedding module. Next, we conduct an attention mechanism on e_u^n , which has same network structure as given in (4)-(8) with input history pattern e_u^p replaced by current trajectory representation e_u^n . We denote the relevant projection matrices as W_Q^2, W_V^2, W_K^2, W^2 , respectively. We also stack this layer for multiple times to fully capture the spatial-temporal correlation. Then, after updating, we obtain an enhanced current trajectory represented by $\bar{e}_u^{n,t}$.

Trajectory Recovery Module

After extracting the historical and current features, the problem is how much to depend on the interpolation via current observed locations or rely on the candidates generated from historical trajectories. Intuitively, a good solution is to compare the current mobility status with historical one, and combine the history information with current interpolation results according to their similarity. To achieve this, we propose the follow mechanism:

Inter-trajectory Attention. We define the similarity of current and historical trajectory denoted by α as the correlation between the enhanced representation in corresponding time slots, i.e., between $\bar{e}_u^{n,t}$ and $\bar{e}_u^{p,k}$ ($\forall t, k = 1, 2, \dots, |T|$). Then we combine history candidates by the similarity α , followed by a residential connection to remain the raw interpolation results. Finally, the fused trajectory \bar{e}_u^n is generated from $\bar{e}_u^{n,t}$ and $\bar{e}_u^{p,k}$, which can be expressed as follows,

$$\alpha_{t,k}^{(h)} = \frac{\exp(\phi^{(h)}(\bar{e}_u^{n,t}, \bar{e}_u^{p,k}))}{\sum_{g=1}^T \exp(\phi^{(h)}(\bar{e}_u^{n,t}, \bar{e}_u^{p,g}))}, \quad (9)$$

$$\phi^{(h)}(\bar{e}_u^{n,t}, \bar{e}_u^{p,k}) = \langle W_Q^3(h)\bar{e}_u^{n,t}, W_K^3(h)\bar{e}_u^{p,k} \rangle, \quad (10)$$

$$\bar{e}_u^{p,t(h)} = \sum_{k=1}^T \alpha_{t,k}^{(h)} (W_V^3(h)\bar{e}_u^{p,k}), \quad (11)$$

$$\bar{e}_u^{p,t} = \bar{e}_u^{p,t(1)} \parallel \bar{e}_u^{p,t(2)} \parallel \dots \parallel \bar{e}_u^{p,t(H)}, \quad (12)$$

$$\bar{e}_u^{n,t} = ReLU(\bar{e}_u^{p,t} + W^3\bar{e}_u^{n,t}), \quad (13)$$

where $W_Q^3(h), W_K^3(h), W_V^3(h), W^3$ are the projection matrices.

With the fused trajectory that contains both history mobility information and current spatial-temporal dependence, we are ready to recovery the missing location. We use the following mechanism to generate the representation of the missing location, and then use it to identify the specific location.

Location Generation Attention. To generate the final representation denoted by $\hat{e}_u^{n,t}$, we define a temporal similarity among the current trajectory represented by e_u^n and the fused trajectory represented by $\bar{e}_u^{n,t}$ as α , which can be derived by (9)-(10) with $\bar{e}_u^{n,t}$ and $\bar{e}_u^{p,k}$ replaced by $\bar{e}_u^{n,t}$ and $e_u^{n,t}$, respectively. Then, $\hat{e}_u^{n,t}$ for time slot t is a combination of $\bar{e}_u^{n,t}$ according to α , which is the same as (11)-(13). The projection matrices are denoted by W_Q^4, W_V^4, W_K^4, W^4 , respectively.

Once we obtain $\hat{e}_u^{n,t}$, we are able to compute the probability that user u visits location l at time slot t as follows,

$$P_u^{n,t}(l) = \frac{\langle \hat{e}_u^{n,t}, e_l \rangle}{\sum_{k \in \mathcal{L}} \langle \hat{e}_u^{n,t}, e_k \rangle}, \quad (14)$$

where $\langle \cdot, \cdot \rangle$ is the inner product function, and $P_u^{n,t} \in \mathbb{R}^{|\mathcal{L}|}$ denotes the normalized probabilities of all location visited at time slot t . In practice, the location with maximum probability is identified as the missing location.

Training

Overall, the parameters of AttnMove include projection matrices and location embedding matrix, denoted by $\theta = \{E_l, W_Q^i, W_V^i, W_K^i, W^i, i = 1, 2, 3, 4\}$. To train the model, we use *Cross entropy* as the loss function:

$$L(\theta) = - \sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}^M} \langle y_u^{n,t}, \log(P_u^{n,t}) \rangle + \lambda \|\theta\|^2, \quad (15)$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $y_u^{n,t}$ is the one-hot representation of user u 's location in n -th day's t -th time slot, \mathcal{T}^M denotes the missing time slots, and λ is a parameter to control the power of regularization. Training algorithm is illustrated in Algorithm 1, and the process is done through stochastic gradient descent over shuffled mini-batches across Adam optimizer (Kingma and Ba 2014). In addition, our model is implemented by Python and Tensorflow (Abadi et al. 2016). We train our model on a linux server with a TITAN Xp GPU (12 G Memory) and a Intel(R) Xeon(R) CPU @ 2.20GHz.

Performance Evaluation

Datasets

- **Tencent**²: This data is collected from the most popular social network and location-based service vendor Tencent in Beijing, China from June 1st~30th, 2018. It records the GPS location of users whenever they request the localization service in the applications.
- **Geolife**³: This open data is collected from Microsoft Research Asia Geolife project by 182 users from April 2007

²<https://lbs.qq.com/>

³<https://www.microsoft.com/en-us/research/project/geolife-building-social-networks-using-human-location-history/>

Algorithm 1: Training Algorithm for AttnMove

Input: Trajectory sets $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_U\}$;
Output: Trained Model θ .
//construct training instances: $\mathcal{D} \leftarrow \emptyset$
for $u \in \{u_1, \dots, U\}$ **do**
 for $n \in \{1, 2, \dots, \mathcal{N}\}$ **do**
 Put a training instance $(u, n, \mathcal{T}^{\mathcal{M}})$ into \mathcal{D}
// Train the model: initialize the parameters θ
for $i \in \{1, 2, \dots, EPOCH\}$ **do**
 Select one batch \mathcal{D}_b from \mathcal{D} ;
 Update θ by minimizing the objective $L(\theta)$ with \mathcal{D}_b ;
 Stop training when criteria is met;
Output trained model θ

to August 2012 over all the world. Each trajectory is represented by a sequence of time-stamped points, containing longitude and altitude (Zheng et al. 2010).

Table 1: Basic statistics of mobility datasets.

Dataset	City	Duration	#Users	#Traj.	#Distinctive Loc.
Tencent	Beijing	1 month	4265	39,422	8,998
Geolife	Beijing	5 years	40	896	3,439

Pre-processing: To represent the location, we crawl the road network of Beijing from online map², and divide the area into 10,655 blocks. Each blocks is treated as a distinctive location with the size of 0.265 km^2 on average. Following (Chen et al. 2019), we set time interval as 30 minutes for both two datasets. For model training and testing, we filter out the trajectories with less than 34 time slots (i.e., 70% of one day) and the users with less than 5 day’s trajectories for Tencent, and filter out the trajectories with less than 12 time slots and the users with less than 5 day’s trajectories for Geolife. The final detailed statics are summarized in Table 1.

Baselines

We compare our AttnMove with several representative baselines. Among them, the first four are directly based on our knowledge about human mobility regularity and the last four are the state-of-the-art deep learning models which can extract more complex mobility features:

- **Top** (Liu et al. 2016): It is a simple counting-based method. The most popular locations in the training set are used as recovery for each user.
- **History** (Li et al. 2019): In this method, the most frequently visited locations of each time slot in historical trajectories are used for recovery.
- **Linear** (Hoteit et al. 2014): This is also a rule-based method. It recovers the locations by assuming that users are moving straightly and uniformly.
- **RF** (Li et al. 2019): RF is a feature-based machine learning method. Entropy and radius of each trajectory, the missing time slot, the location before and after the missing time slot are extracted as features to train a random

forest classifier for recovery.

- **LSTM** (Liu et al. 2016): It models the forward sequential transitions of mobility by recurrent neural network, and use the prediction for next time slot for recovery.
- **BiLSTM** (Zhao et al. 2018): It extends LSTM by bi-directional RNN to consider the spatial-temporal constraints given by all observed locations.
- **DeepMove** (Feng et al. 2018): Besides modeling sequential transitions, DeepMove incorporates the historical trajectories by attention mechanism for next location prediction. We also use the prediction result for recovery.
- **Bi-STDDP** (Xi et al. 2019): This is the latest missing location recovery method, which jointly models user preference and the spatial-temporal dependence given the two locations visited before and after the targeted time slot to identify the missing ones.

Apart from those baselines, to evaluate the effectiveness of our designed mechanism to exploit history, we also compare **AttnMove** with its simplified version **AttnMove-H**, where history processor and inter-trajectory attention layer are removed.

Experimental Settings

To evaluate the performance, we mask some time slots as ground-truth to recover. Since about 20% locations are missing in the raw datasets on average, we randomly mask 30 and 10 time slots per day for Tencent and Geolife. We sort each user’s trajectories by time, and take the first 70% as the training set from the fourth day (to guarantee that each trajectory has at least three days as history), the following 10% as the validation set and the remaining 20% as the test set. Linear, Top and History are individual models, while other models are shared by the users in one dataset. The regularization factor λ is set as 0.01.

We employ the widely used metrics *Recall* and Mean Average Precision (*MAP*) (Wang et al. 2019a; Liu et al. 2016). *Recall* is 1 if the ground-truth location is recovered with maximum probability; otherwise is 0. The final *Recall* is the average value over all instances. *MAP* is a global evaluation for ranking tasks, so we use it to evaluate the quality of the whole ranked lists including all locations. The larger the value of those two metrics is, the better the performance will be. We also make use of the metric of *Distance*, which is the geographical distance between the center of recovered location and the ground-truth. The smaller the *Distance* is, the better the performance will be.

Experiment Results

Overall Performance We report the overall performance in Table 2 and have the following observations:

1) Rule-based models fail to achieve high accuracy with *Recall* and *MAP* lower than 0.66 in Tencent and lower than 0.39 in Geolife. Although intuitively, frequently visited locations, historically visited locations and moving constraints are helpful for recovery, simply utilizing them cannot achieve favourable performance because these methods cannot model complex mobility regularity.

2) RNN-based models are insufficient but bidirectional ones can reduce the *Distance* error. A plausible reason is

Table 2: Overall performance comparison. The best result in each column is in bold, while the second is underlined.

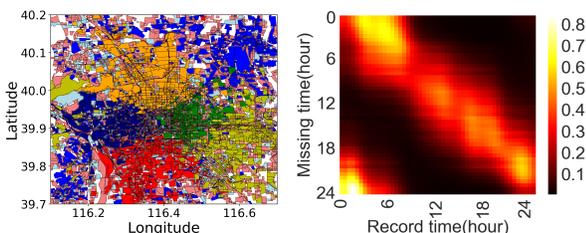
Model	Tencent			Geolife		
	Recall	MAP	Dis.(m)	Recall	MAP	Dis.(m)
TOP	0.5879	0.6123	2530	0.2757	0.2879	5334
History	0.4724	0.4937	1613	0.2505	0.2648	5116
Linear	0.6234	0.6567	1145	0.3642	0.3889	2383
RF	0.4848	0.4912	8128	0.2551	0.2540	6144
LSTM	0.6084	0.6722	3759	0.2725	0.3314	5864
BiLSTM	0.7090	0.7805	1371	0.3471	0.4148	5097
DeepMove	<u>0.7259</u>	<u>0.7872</u>	1322	0.3391	0.4015	4912
Bi-STDDP	0.7037	0.7831	<u>1168</u>	<u>0.3701</u>	<u>0.4510</u>	4041
AttnMove-H	0.7358	0.7999	1083	0.3853	0.4501	4129
AttnMove	0.7646	0.8249	934	0.3982	0.4691	3886

spatial-temporal constraints from all the observed locations is crucial instead of the sequential regularity for recovery. In addition, our AttnMove achieves further performance gain over the best RNN methods. It indicates the superiority of attention mechanisms.

3) AttnMove outperforms all the baselines for all the metrics except for *Distance* in Geolife. Specifically, *Recall* and *MAP* of AttnMove outperforms the best baselines by 4.0%~7.6%. *Distance* is also reduced by 20% in Tencent. It is possible that Geolife is a small dataset for training, therefore *Distance* of AttnMove cannot outperform Linear. But higher *MAP* of AttnMove shows that although the generated location with the highest probability does not hit the ground-truth, which leads to higher *Distance*, the correct location is on the top of recovered locations list.

4) When comparing DeepMove with LSTM, and AttnMove with AttnMove-H, we can observe that history plays its role in reducing the uncertainty and improving recovery accuracy.

To conclude, our AttnMove model achieves preferable results compared with both rule-based and machine learning-based methods. This justifies our model’s effectiveness in capturing mobility pattern and exploiting history, and therefore it is a powerful model to rebuild fine-grained trajectories.



(a) Location embedding with color denoting cluster. (b) Attention weight.

Figure 3: Visualization results.

Visualization Analysis Besides the above quantitative analysis, we also make some visualization study. First, to investigate whether our learned trajectory embedding can learn the spatial correlation automatically, we cluster the regions by using their embedding e_t as features via k-means

with Euclidean distance. Figure 3 (a) shows their geographic distribution with clustering results. We can observe that adjacent locations generally share the same color indicating they are also closed in the embedding space, demonstrating that the spatial adjacent relation has been modeled. Second, we visualize the attention weights of each head in each layer for different users, and present the average attention weights of all trajectories in the final current intra-trajectory attention layer in Figure 3(b). Overall, the diagonal entries are highlighted indicating that the target location more likely depends on the locations of adjacent time slots. More importantly, the bright part in bottom-left corner indicates apart from adjacent time, morning and night locations are highly related as well. It demonstrate that our attentional network can movement regularity and periodical characters simultaneously.

Ablation Analysis We analyze the effects of each trajectory attention mechanism. We create ablations by removing them one by one, i.e., using the embedding of the corresponding time slot directly instead of using a weighted combination. We report the results in Table 3. As expected, AttnMove outperforms all the ablations, indicating each attention can improve recovery performance. Specifically, when removing current intra-trajectory attention, the performance declines most significantly. This is because the attention can effectively strengthen the spatial constraints for the missing locations, without which even utilizing history, the improvement is limited.

Table 3: Impact of attention mechanisms on Tencent dataset, where δ denoted the performance decline.

Ablation	Recall(Δ)	MAP(Δ)	Dis.(Δ)(m)
Historical intra-trajectory attention	0.7457(2.5%)	0.8074(2.1%)	986(6.3%)
Current intra-trajectory attention	0.7002(8.4%)	0.7793(5.6%)	1636(75.2%)
Inter-trajectory attention	0.7434(2.8%)	0.8119(1.6%)	1063(13.8%)
Location generation attention	0.7552(1.2%)	0.8152(1.2%)	962(3.0%)

Robustness Analysis We also conduct experiments to evaluate the robustness of AttnMove when applied in different scenarios. Firstly, we study the recovery performance w.r.t missing ratio, i.e., the percentage of missing locations. The results is presented in Table 4. As missing ratio rises from 50% to almost 100%, it is more intractable to recover correctly, while our model can maintain a significant gap (i.e., more than 3%) with the state-of-the-art baseline. Second, we present the performance in different of a day and of trajectories with a different number of locations visited. As we can observe from Figure 4, during the day time when people are more likely to move or for these trajectories where numerous locations are visited, it is more difficult to recover correctly and thus *Recall* declines. Nevertheless, our *Recall* can always outperform baselines by more than 5%. Those results demonstrate the robustness of our proposed model in different scenarios.

We finally investigate the sensitivity of dimension d , the head number H , and the number of layers N , which determine the representation ability of the model. Figure 5(a) presents the performance on different embedding dimension

Table 4: Performance *w.r.t* missing ratios on Tencent dataset.

Missing Rate		60%-70%	70%-80%	80%-90%	90%-100%
Percentage		19.2%	30.2%	41.8%	8.8%
BiSTDDP	Recall	0.7637	0.7366	0.6636	0.6501
	Distance(m)	755	1022	1358	1667
AttnMove	Recall	0.8064	0.7820	0.7465	0.6993
	Distance(m)	651	848	1050	1294

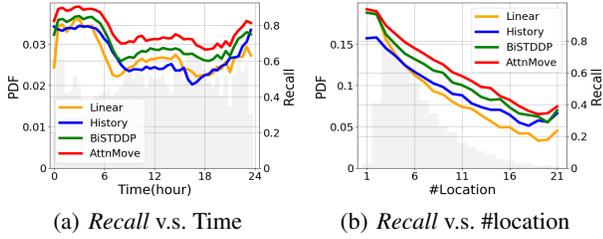


Figure 4: Model performance in different contexts, where the solid lines denote *Recall* and the shades show the distribution.

values. We can observe that as the dimension increases, the performance is gradually improved, and when it larger than 128, the performance becomes stable. This is why we select embedding size as 128. Then, we conduct a grid research for H and N . Figure 5(b) partly shows the results. We find that a larger number of layers general achieves better performance while the impact of head is not significant. Considering that more layers makes the model more expressive but requires more computational cost, to make a compromise between performance and efficiency, we finally fix the number of layer and head as 4 and 8, respectively.

Related Work

Trajectory Recovery. Recovering the missing value for time series is an important problem, for which deep learning models have achieved promising performance (Luo et al. 2018). However, they are insufficiently precise to be applied with mobility data due to their inability to model spatial-temporal dependence and users’ historical behaviours. Apart from these recovery methods, some works for mobility prediction can also be adopted for recovery. Feng *et al.* (Feng et al. 2018; Xia et al. 2020) incorporated the periodicity of mobility learned from history with the location transition regularity modeled by recurrent neural network (RNN) to predict the next location. However, the performance declines because it cannot model the spatial-temporal dependence from the sparse trajectory. As such, models for mobility data recovery have been particularly studied. Li *et al.* (Li, Li, and Li 2013) used entropy and radius such trajectory-specific features for spatial-temporal dependence modeling, while it failed to exploit history. By presenting a user’s long-term history as a tensor with day-of-the-month, time-of-the-day, and location-of-the-time three dimensions, a tensor factorization-based method was proposed (Chen et al. 2019). However, for this to work, it is required that the tensor is low-rank, and thus cannot model the randomness and complexity of mobility. Recently, Bi-STDDP was designed to represent users’ history by a vector and combine it the spatial-temporal correlations for recovery (Xi et al. 2019). However, the expressiveness of the vector is limited as it is

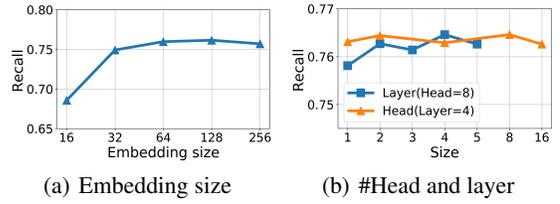


Figure 5: Key hyper-parameter tuning.

unable to reflect the dynamic importance of history. Overall, both the general time series recovery methods, trajectory prediction methods, and the existing mobility data recovery methods are incapable of tackling the instinctive challenges of the mobility recovery problem. By contrast, we propose an attentional neural network-based model, which can better model spatial-temporal dependence for sparse trajectory and exploit history more efficiently.

Attentional Neural Network. The recent development of attention models has established new state-of-the-art benchmarks in a wide range of applications. Attention is first proposed in the context of neural machine translation (Bahdanau, Cho, and Bengio 2014) and has been proved effective in a variety of tasks such as question answering (Sukhbaatar et al. 2015), text summarization (Rush, Chopra, and Weston 2015), and recommender systems (He et al. 2018; Song et al. 2019). Vaswani et al. (Vaswani et al. 2017) further proposed multi-head self-attention, renewed as *Transformer*, to model complicated dependencies between words for machine translation. It makes significant progress in sequence modeling, as it uses fully attention-based architecture, which discards RNN but outperforms RNN-based models. Researchers also showed consistent performance gains by incorporating attention with RNN for mobility modeling, such as location prediction (Feng et al. 2018), and personalized route recommendation (Wang et al. 2019b), where attention can make up RNN’s limitation in capturing long-term temporal dependence. Distinct from previous researchers, we are the first to adopt the fully attentional neural network, *Transformer* alike, to tackle the mobility data recovery problem.

Conclusion and Future Work

In this paper, we proposed an attentional neural network-based model AttnMove to recover user’s missing locations at a fine-grained spatial-temporal. To handle the sparsity, we designed an intra-trajectory mechanism to better model the mobility regularity. To make full use of history and distill helpful periodic characters, we proposed to integrate relatively long-term historical records. Finally, we also designed an inter-trajectory mechanism to effectively fuse the mobility regularity and the historical pattern. Extensive results on real-worlds datasets demonstrate the superiority of AttnMove compared with the state-of-the-arts.

In the future, we plan to extend our framework by incorporating some features like the function of location or the points of interest users visited to achieve semantic-aware trajectory interpolation.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China under grant 2020AAA0106000, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

- Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 265–283.
- Alwan, L. C.; and Roberts, H. V. 1988. Time-series modeling for statistical process control. *Journal of Business & Economic Statistics* 6(1): 87–95.
- Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; and Li, Y. 2018. BRITS: bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, 6775–6785.
- Chen, G.; Viana, A. C.; Fiore, M.; and Sarraute, C. 2019. Complete trajectory reconstruction from sparse mobile phone data. *EPJ Data Science* 8(1): 30.
- Cho, E.; Myers, S. A.; and Leskovec, J. 2011. 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*, 1082–1090.
- Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*, 1459–1468. International World Wide Web Conferences Steering Committee.
- Gao, C.; Huang, C.; Yu, Y.; Wang, H.; Li, Y.; and Jin, D. 2019. Privacy-preserving cross-domain location recommendation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3(1): 1–21.
- Gonzalez, M. C.; Hidalgo, C. A.; and Barabasi, A.-L. 2008. Understanding individual human mobility patterns. *nature* 453(7196): 779–782.
- He, X.; He, Z.; Song, J.; Liu, Z.; Jiang, Y.-G.; and Chua, T.-S. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30(12): 2354–2366.
- Hoteit, S.; Secci, S.; Sobolevsky, S.; Ratti, C.; and Pujolle, G. 2014. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks* 64: 296–307.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Li, L.; Li, Y.; and Li, Z. 2013. Efficient missing data imputation for traffic flow by considering temporal and spatial dependence. *Transportation research part C: emerging technologies* 34: 108–120.
- Li, M.; Gao, S.; Lu, F.; and Zhang, H. 2019. Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data. *Computers, Environment and Urban Systems* 77: 101346.
- Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- Luo, Y.; Cai, X.; Zhang, Y.; Xu, J.; et al. 2018. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, 1596–1607.
- Moritz, S.; and Bartz-Beielstein, T. 2017. imputeTS: time series missing value imputation in R. *The R Journal* 9(1): 207–218.
- Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.
- Schneider, C. M.; Belik, V.; Couronné, T.; Smoreda, Z.; and González, M. C. 2013. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface* 10(84): 20130246.
- Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; and Tang, J. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1161–1170.
- Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Wang, J.; Wu, N.; Lu, X.; Zhao, X.; and Feng, K. 2019a. Deep Trajectory Recovery with Fine-Grained Calibration using Kalman Filter. *IEEE Transactions on Knowledge and Data Engineering*.
- Wang, J.; Wu, N.; Zhao, W. X.; Peng, F.; and Lin, X. 2019b. Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 539–547. ACM.
- Xi, D.; Zhuang, F.; Liu, Y.; Gu, J.; Xiong, H.; and He, Q. 2019. Modelling of Bi-Directional Spatio-Temporal Dependence and Users’ Dynamic Preferences for Missing POI Check-In Identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5458–5465.

Xia, T.; Li, Y.; Feng, J.; Jin, D.; Zhang, Q.; Luo, H.; and Liao, Q. 2020. DeepApp: Predicting Personalized Smartphone App Usage via Context-Aware Multi-Task Learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11(6): 1–12.

Zhao, J.; Xu, J.; Zhou, R.; Zhao, P.; Liu, C.; and Zhu, F. 2018. 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*, 1413–1422. ACM.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5(3): 38.

Zheng, Y.; Xie, X.; Ma, W.-Y.; et al. 2010. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33(2): 32–39.