

## Article

# Human Mobility Prediction with Calibration for Noisy Trajectories

Qing Miao <sup>1</sup>, Min Li <sup>1</sup>, Wenhui Lin <sup>2</sup>, Zhigang Wang <sup>2</sup>, Huiqin Shao <sup>2</sup>, Junwei Xie <sup>2</sup>, Nanfei Shu <sup>2</sup>  
and Yuanyuan Qiao <sup>1,\*</sup> 

<sup>1</sup> Research Center for Intelligent Perception and Computing, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>2</sup> Aisino Corporation, Beijing 100195, China

\* Correspondence: yyqiao@bupt.edu.cn

**Abstract:** Human mobility prediction is a key task in smart cities to help improve urban management effectiveness. However, it remains challenging due to widespread intractable noises in large-scale mobility data. Based on previous research and our statistical analysis of real large-scale data, we observe that there is heterogeneity in the quality of users' trajectories, that is, the regularity and periodicity of one user's trajectories can be quite different from another. Inspired by this, we propose a trajectory quality calibration framework for quantifying the quality of each trajectory and promoting high-quality training instances to calibrate the final prediction process. The main module of our approach is a calibration network that evaluates the quality of each user's trajectories by learning their similarity between them. It is designed to be model-independent and can be trained in an unsupervised manner. Finally, the mobility prediction model is trained with the instance-weighting strategy, which integrates quantified quality scores into the parameter updating process of the model. Experiments conducted on two citywide mobility datasets demonstrate the effectiveness of our approach when dealing with massive noisy trajectories in the real world.

**Keywords:** human mobility; spatio-temporal prediction; noisy trajectories; neural network



**Citation:** Miao, Q.; Li, M.; Lin, W.; Wang, Z.; Shao, H.; Xie, J.; Shu, N.; Qiao, Y. Human Mobility Prediction with Calibration for Noisy Trajectories. *Electronics* **2022**, *11*, 3362. <https://doi.org/10.3390/electronics11203362>

Academic Editors: Vasco N. G. J. Soares, Joel J. P. C. Rodrigues, João M. L. P. Caldeira and Jaime Galán-Jiménez

Received: 22 September 2022

Accepted: 13 October 2022

Published: 18 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

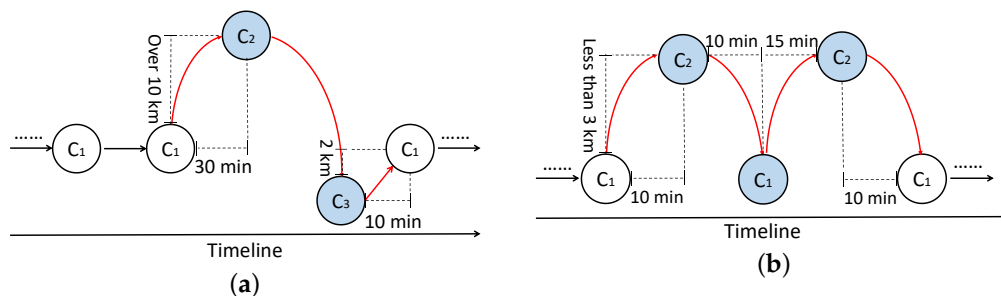
Predicting human mobility on a citywide level is vital to studies and applications in city-related areas, such as urban planning, traffic engineering, and epidemic prevention and control. Though human movements have a high degree of freedom and variation, several remarkable works [1–3] have proven that regularity and periodicity dominate the main pattern of human movements and play a critical role in mobility prediction. Therefore, extracting mobility patterns from people's historical trajectories and predicting their next location has attracted much attention.

Traditional mobility prediction methods are mostly pattern-based. They employ machine learning methods such as matrix factorization to discover movement patterns from successive trajectories and predict the next location based on these formalized pattern representations. In addition, a large number of studies apply variants of Markov models due to their advantage in modeling sequential transitions, including hidden-Markov [4], weighted-Markov [5], attentional-Markov [6], Bayesian nets [7] and some other hybrid models.

In recent years, with the rapid development of deep learning, neural network models have shown promising performance in spatio-temporal sequence modeling such as Recurrent Neural Networks (RNN) [8] and Transformer [9]. Liu et al. [10] propose Spatial Temporal Recurrent Neural Networks (ST-RNN) to model the temporal and spatial context. Based on RNN, DeepMove [11] employs an attention mechanism to extract multi-level periodicity from long historical trajectories. Dang et al. [12] proposed a dual-attentive network to capture the long-range sequential dependency within a trajectory and the correlation between different trajectories. Given trajectory data, existing works usually train a learnable module to predict the next Point of Interest (POI) or corresponding region,

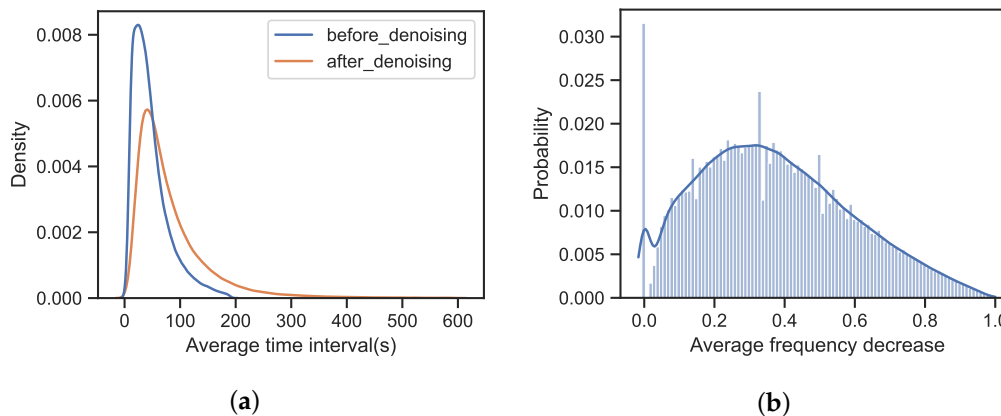
aiming to consider the more complex context of spatio-temporal sequences and other related factors (such as social network structure [3] and location semantics [13]) to improve prediction accuracy.

However, the above methods require a large amount of mobility data to train the model to learn the regularity and periodicity of mobility patterns. Unfortunately, most mobility data with large scale and long duration contain noisy data which could reduce the prediction accuracy. Noise trajectories are often generated due to technical deviations from positioning devices and methods. Taking cellular mobility datasets [14] as an example, usually, a user connects to the cell tower closest to them, so the location of the cell tower to which the user is connected is used as an estimate of the user’s location. When there is a load balancing or signal strength change in cell towers [15], even if the user is not moving, the recorded locations switch between two or more cell towers, which is called oscillation [16–18]. Two real examples of such oscillations in our dataset are depicted in Figure 1. With the help of various vehicles, many trips represented by oscillations are theoretically possible [19]. What is more, researchers often lack ground-truth information for validation to identify and clean these noisy trajectories [20].



**Figure 1.** Two real examples of oscillations. (a) Cells jump to a far away tower then return back; (b) cells swift between two nearby towers.

We use heuristic-based approaches [19] to remove significant device noise from the data used in our experiments and roughly consider the average frequency decrease in recording after denoising for each user as the proportion of device noise in the user trajectory. Figure 2 shows that device noise is prevalent in trajectory data. Most users’ trajectories have a small proportion of noise, ranging from 0 to 0.6, and a few users have a large proportion of noise. This distribution indicates heterogeneity in the quality of user trajectories. Therefore, it is crucial to quantify the importance and quality of each user’s trajectory in massive noisy mobile data and to reflect this in the prediction process.



**Figure 2.** (a) Average time interval of user recordings before and after denoising; (b) average frequency decrease in user recording by denoising .

There have been some works that successfully learn with noisy data in dialogue systems for NLP. To evaluate the quality of training dialogues and generate more reasonable conversations, several methods [21,22] concentrate on quantifying the relatedness between queries and replies and train conversation models through an instance-weighting strategy.

Inspired by these, we propose the trajectory quality calibration framework, which automatically estimates the quality of each training trajectory through a pre-trained calibration network and promotes high-quality training instances to calibrate the final prediction process. Several recurrent neural sub-networks with shared weights are used in the proposed calibration network to capture and quantify the consistency of the user's mobility transitions each day. Additionally, the quality score of that user's trajectory is evaluated by measuring the correlation between daily features. Then, in the unsupervised pre-training process of the calibration network through a negative sampling strategy, the quality score will be gradually updated by comparing the relatedness between the original trajectory and the corresponding trajectory injected with random noise. The impact of each training instance on the parameters of the final prediction model can be quantified as a weight value by a normalization process on their quality scores. Finally, the prediction model takes advantage of these normalized weight values to calibrate the training process by an instance-weighting method, which multiplies the weight values to loss functions and gradient descents when updating corresponding parameters. In summary, the main contributions of this work are:

- We propose a calibration framework for mobility prediction based on the instance-weighting strategy to automatically evaluate the quality of training trajectories and differentiate their impact on the training process of the mobility prediction model. To the best of our knowledge, we are the first to explore mobility prediction problems on massive noisy data through an instance-weighting method.
- We employ an unsupervised method to estimate the parameters of the proposed calibration network, which liberates human energy on label tagging and makes it possible to apply the network to massive citywide mobility data. Additionally, the proposed approach is model-independent and can be combined with any massive data-driven neural prediction network.
- We conducted our experiments on citywide cellular network datasets collected from two metropolises that covered more than two million people, respectively. The experimental results show that evaluating the quality of user trajectories and calibrating the training process can effectively improve the performance of neural prediction models.

The rest of this paper is organized as follows. Section 2 reviews the related works of mobility prediction and instance weighting in NLP. Section 3 introduces the proposed calibration framework for mobility prediction in detail and gives an overview of our dataset and methods for data pre-processing and selection. Section 4 illustrates the experimental results. Finally, we analyze the experimental results and present the limitations of our work in Section 5.

## 2. Related Work

### 2.1. Mobility Prediction

Former studies on human mobility prediction can be generally classified into two categories: pattern-based methods and model-based methods.

The pattern-based methods [23–26] focus on discovering intrinsic mobility patterns from sequential trajectories and predicting next locations based on these formalized pattern representations. Most works are based on matrix factorization and are often assisted by feature engineering. In addition, embedding techniques can also be regarded as the pattern-mining method, such as POI2Vec [27] and Personalized Ranking Metric Embedding (PRME) [28].

The model-based methods [4,29–31] predict the user's next visit by modeling the statistical relationship of contexts in the trajectories. The Markov model and its variations are common methods in earlier research. They model the probability of a transition matrix

between relevant positions based on different data assumptions. The Hidden Semi-Markov Model (HSMM) [32] is designed to remove the constant or geometric distributions of the state durations assumed in the Hidden Markov Model (HMM). Yan et al. [5] propose a weighted Markov model for different user classifications. However, the association between Markov process and mobility dynamics has been questioned [33] in recent years because human mobility may exhibit scale-invariant long-term dependency, which contrasts with the initial Markov assumption [34].

Recently, deep learning techniques have been widely applied to model human mobility. Many works use RNNs to capture long-term dependencies in trajectories. Spatial–Temporal Recurrent Neural Networks (ST-RNN) [10] model temporal and spatial contexts at each time interval. DeepMove [11] employs an attention mechanism to capture periodic features in historical trajectories. The Bidirectional Long Short-Term Memory–Convolutional Neural Network (BiLSTM-CNN) [35] passes the output of RNN to CNN to capture the overall spatial and temporal patterns. Transformer [9] entirely relies on the attention mechanism to model the global dependencies of the sequence and breaks through the limitation that RNN cannot be parallelized. Deep Wide Spatio-Temporal Transformer Network (DWSTTN) [36] uses two attention mechanisms to extract relevant information in time and space, respectively. Graph Convolutional Dual-Attentive Networks (GCDAN) [12] design a dual-attention mechanism within and between trajectories and use graph convolution to extract spatial features in the embedding layer. With the help of sufficient training data sources, neural-based models can parameterize different kinds of mobility transitions instead of fixed hypothetical representations, making it possible to capture more complex regularities and model sequential trajectories more accurately.

However, these proposed models treat every training trajectory equally importantly, regardless of the uneven data quality distribution, which significantly influences mobility prediction accuracy [37].

## 2.2. Instance Weighting

Instance weighting is a training strategy that assigns different weight values to training instances when reflecting their influence on the updating process of model parameters. Some recent works have adopted this method for domain adaptation tasks in NLP, especially for dialogue systems. Jiang et al. [38] analyze and characterize the domain adaptation problem from a distributional view and propose a general instance-weighting framework for domain adaptation.

In common domain adaptation tasks, noisy data are identified and easy to distinguish when involving the uneven distribution of data quality. However, the noisy data in training a conversation model for the dialogue system are not that easy to identify, due to their high diversity based on characteristics of human language. Wang et al. [39] adopt the instance-weighting strategy to address the noisy label issue during data processing. Tao et al. [22], Lison et al. [21], and Shang et al. [40] propose relatedness-based evaluation metrics and matching networks to quantify and measure the relationship between queries and replies in conversations. The above problems are similar to the regularity and periodicity between daily mobility transitions in our task. Then, the data quality evaluation results assist the conversation model in focusing on meaningful training dialogues and generating more intelligent answers.

In the proposed trajectory quality calibration framework, we build a calibration network to estimate the quality of each trajectory and integrate the quality scores into parameters updating in the mobility prediction model. Therefore, this work mainly investigates how the performance of existing prediction models can be improved when applying them to massive, noisy trajectories that widely exist in the real world.

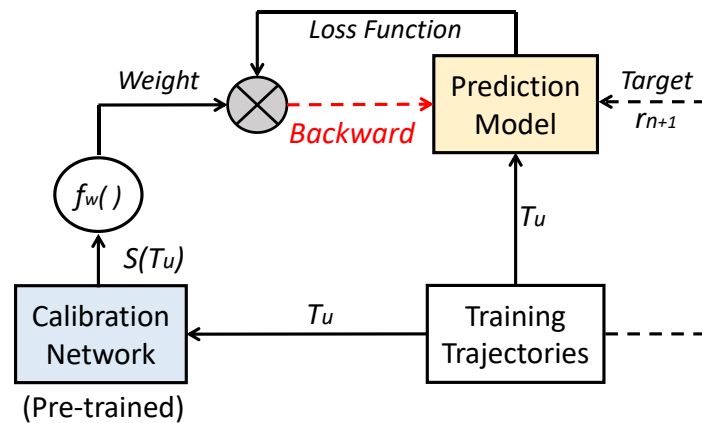
### 3. Materials & Methods

#### 3.1. Preliminaries

The problem of mobility prediction is usually defined as follows: *given the current trajectory and the corresponding trajectory history of a user, train a model to predict the next spatio-temporal point in the trajectory* [11,41,42]. In our experiments, the time interval between the spatio-temporal records is quantified into a fixed value, along with a spatial granularity of street blocks segmented by road networks. Thus, the mobility prediction of a mobile user is simplified to predict the next street block ID in the next time interval.

The spatio-temporal record  $r$  in a trajectory can be denoted as a tuple  $r = (l, t)$ , where  $l$  is the location identification number (street block ID in our experiments) and  $t$  is the time stamp.  $T_u$  denotes trajectories generated by user  $u$ , represented as a spatio-temporal sequence  $T_u = r_1 r_2 \dots r_n$ . We define a general training data set  $T$  with a corresponding user set  $U$  of size  $N$ , then  $T = \{T_u | u \in U\}$ . To model daily movements, we denote  $T_u = T_u^1 T_u^2 \dots T_u^m$ , representing the user trajectories within the duration of  $m$  days. In some scenarios, to capture the periodic trends in a typical week,  $T_u$  can also represent the trajectories on a specific day of the week. For example,  $T_u^{Mon}$  then denotes the whole trajectories generated by user  $u$  on all Mondays in the duration of training data  $T$ . As a result, a prediction model is trained to predict the next  $r_{n+1}$  and maximize the probability  $P(T_u | r_{n+1})$  for each training instance  $T_u$ .

Instead of considering each training instance equally, our goal is to train a calibration network that learns a scoring function  $S(T_u)$  to represent the quality of each training instance  $T_u$ . Then, the weight value  $f_w(S(T_u))$ , normalized based on other instances in the same training batch, will be utilized for the weighted updates of parameters in the prediction model. An overview of the proposed calibration framework for mobility prediction is depicted in Figure 3.



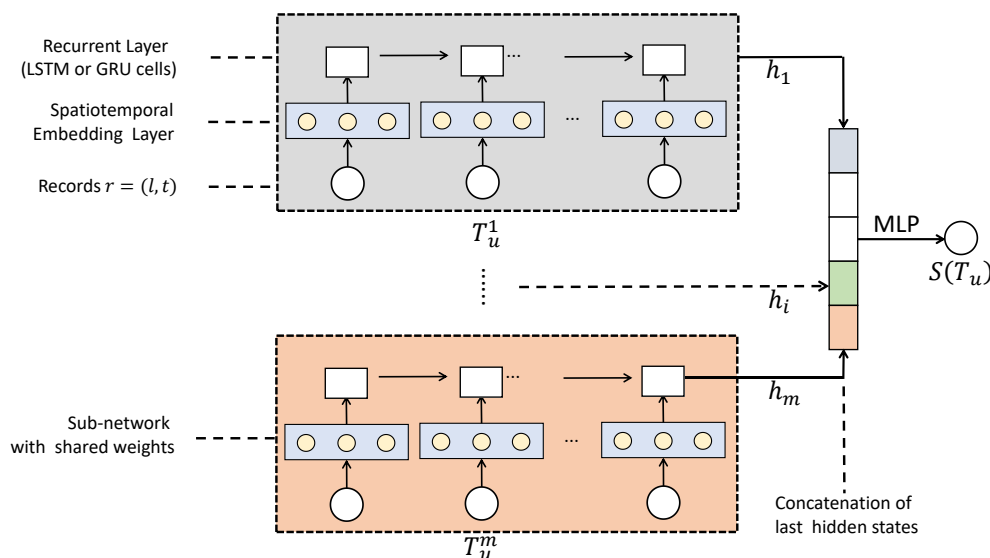
**Figure 3.** An overview of the trajectory quality calibration framework with a pre-trained calibration network to evaluate the data quality of training trajectories and integrate it into the parameter updating process of the main prediction model.

#### 3.2. Calibration Network

The calibration network is an RNN-based model, the input of which is  $T_u = T_u^1 T_u^2 \dots T_u^m$ , and the output is a numerical score  $S(T_u)$  to represent data quality and regularity. The detailed architecture of this neural network is depicted in Figure 4.

**Spatio-temporal embedding.** The input sequences of  $r = (l, t)$  are first fed into a spatio-temporal embedding layer. Some former works such as ST-RNN and STF-RNN [43] apply complex methods to utilize the spatio-temporal factors of human trajectories, such as considering geographical distances between locations in the representations of their RNN hidden states. However, mobility transitions can be complex and high-order [3,44], making it hard to quantify by artificially defined formulas. Instead, the neural network can automatically learn the numerical representations of spatio-temporal contexts without prior knowledge

or predefined features [11,42,45]. To achieve such representation, as introduced in the Preliminaries, all the available trajectory features can be numbered in our experiments, such as location ID and timestamp ID. Then, the numbered spatio-temporal features are translated into one-hot vectors and fed into the spatio-temporal embedding layer after a concatenation operation, which outputs their low-dimensional representations [46]. Additionally, we consider the user ID of a trajectory as unnecessary information, since the calibration network and the prediction model mainly focus on the mobility patterns and historical preferences regardless of user identity (without a user ID embedding layer in practice).



**Figure 4.** Detailed architecture of the calibration network, taking  $T_u = T_u^1 T_u^2 \dots T_u^m$  as input and returning a score  $S(T_u)$  to represent its data quality.

Neural architecture. The calibration network is composed of  $m$  recurrent sub-networks with shared weights to model daily mobility. In each sub-network, the spatio-temporal embeddings are jointly fed into an RNN layer with LSTM or GRU cells to capture the regularity of daily transitions. Considering the acceptable amount of records  $r$  in a daily trajectory  $T_u^i$ , we did not adopt complicated methods such as the attention mechanism, which mainly focuses on handling vanishing gradient problems in RNN and capturing long-term dependency with higher consumption of computation. Such a recurrent sub-network can be seen as a form of daily trajectory embedding with Equation (1),

$$\begin{cases} h_1 = RNN(T_u^1), \\ \dots \\ h_m = RNN(T_u^m), \end{cases} \quad (1)$$

where  $RNN(\cdot)$  denotes the final hidden state of the recurrent sub-network. We take these final hidden states  $S_1 \dots S_m$  as embedding representations of daily movements  $T_u^1 \dots T_u^m$ . Then, a scoring function is trained to quantify the regularity and periodicity between them, which is given as Equation (2),

$$S(T_u) = \text{sigmoid}(\text{MLP}([h_1; h_2 \dots; h_m])), \quad (2)$$

where  $\text{MLP}(\cdot)$  is a Multi-Layer Perceptron (MLP) with  $\tanh$  activation function, and  $[\cdot]$  denotes a concatenation operation. An additional  $\text{sigmoid}$  function is added at the last layer of MLP to guarantee the final score falls in the range of  $[0,1]$ , representing the quality of  $T_u$ .

Average sampling. It should be noticed that the number of recurrent sub-networks  $m$  depends on the time duration of  $T$ . For the mobility dataset lasting within a week,  $m$



can be the number of days in the duration of  $T$ . However, when  $T$  is collected during an extended period, such as a whole month, directly employing  $m = 30$  would lead to high computation complexity and poor performance. Here, we apply a similar method proposed in [11] to adopt an average sampling strategy depicted in Figure 5. The trajectory vectors are reorganized into a two-dimensional matrix: a fixed-length temporal dimension and a spatial dimension. In the temporal dimension, we align all the time stamps of trajectories into one week ( $m = 7, T_u = T_u^{Mon} \dots T_u^{Sun}$ ) to simulate the periodical nature of human mobility. In the spatial dimension for each user, all the visited locations appearing in the same period are collected in a location set for every time slot. We add all the location embedding vectors and compute the mean values as representations. Then, the reorganized sequences of average spatio-temporal embeddings are fed into the recurrent layer.

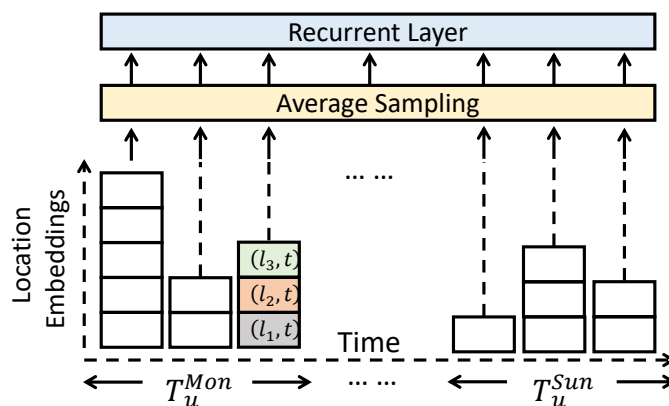


Figure 5. The average sampling strategy, designed to deal with long-time-period mobility datasets.

### 3.3. Parameter Estimation

A common strategy to train the calibration network is to provide the neural model with positive examples of high-quality trajectories and negative examples randomly sampled from  $T$ . The most accurate and direct way to provide such positive examples is to have human raters manually select training instances of high quality. However, training a calibration network to estimate data quality of  $T$  with  $N > 2,000,000$  needs a massive amount of positive examples, which makes manually labeling the dataset almost impossible. Therefore, it is crucial to design the parameter updating process in an unsupervised manner.

Here, we employ an unsupervised training strategy through negative sampling to train our calibration network. Specifically, for a training instance  $T_u = T_u^1 T_u^2 \dots T_u^m$ , we randomly sample another user  $v$  from  $U$  with  $T_v = T_v^1 T_v^2 \dots T_v^m$ . Then, we replace a random  $T_u^i, i \in \{1 \dots m\}$  with the corresponding  $T_v^i$  to break its original feature. The mixed new trajectory is denoted as  $T_u^- = T_u^1 \dots T_v^i \dots T_u^m$ . Such a replacement will affect the consistency and relatedness between daily trajectory embeddings. Thus, the training objective is that  $S(T_u)$  should be greater than  $S(T_u^-)$  with at least  $\theta$  threshold. Through continuous iteration and updating, the calibration network will be able to correct  $S(T_u)$  gradually. Then, the loss function is given as Equation (3).

$$\mathcal{L}_w = \max(\theta + S(T_u^-) - S(T_u), 0) \tag{3}$$

### 3.4. Instance Weighting

Once the parameter of the calibration network is estimated, we apply it to the whole dataset  $T$  to obtain  $S(T_u)$  for each  $u \in U$ . Then, we utilize these scores in the neural prediction model. Usually, the mobility prediction task with a size-limited target location

list can be considered a multi-classification problem and take the cross-entropy loss  $L$  as the loss function. Then, the overall loss function is given by Equation (4).

$$\mathcal{L}_p = \sum_{u \in U} L(T_u, r_{n+1}). \quad (4)$$

Differently from conventional prediction models, which treat each training instance equally, we take the quality scores from the pre-trained calibration network into account when calculating the loss function in the mobility prediction model as Equation (5),

$$\mathcal{L} = \sum_{u \in U} f_w(S(T_u))L(T_u, r_{n+1}), \quad (5)$$

where  $f_w(\cdot)$  is a normalized function that transforms  $S(T_u)$  into a normalized weight, defined as Equation (6),

$$f_w(S(T_u)) = \frac{S(T_u)}{\frac{1}{b} \sum_{v \in U_b} S(T_v)}, \quad (6)$$

where  $b$  is the batch size and  $U_b$  is the user set of the mini-batch to which  $u$  belongs. In our method, the impact of  $T_u$  on the final loss function is determined by not only its score  $S(T_u)$  but also other instances in the same training batch. Then, through our instance-weighting strategy, the parameters of the prediction model are finally updated by Equation (7),

$$W_{t+1} = W_t + \frac{lr}{b} \sum_{u \in U_t} f_w(S(T_u)) \nabla \mathcal{L}_p(T_u, r_{n+1}), \quad (7)$$

where  $lr$  is the learning rate of the prediction model, and  $W_t$  and  $U_t$  represent all variable parameters of the prediction model and the corresponding user set at the  $t$ -th mini-batch, respectively. The training instance associated with a greater score or weight value has a larger influence on the gradient update steps in Backward Propagation Through Time (BPTT). Compared with another calculation method  $\mathcal{L} = \sum_{u \in U} S(T_u)L(T_u, r_{n+1})$  which directly multiplies  $S(T_u)$  to the raw loss function, the weight normalization in Equation (6) successfully controls the extent of impact that  $S(T_u)$  have on the gradient updates. Specifically, after the transformation, half of the weight values  $f_w(S(T_u))$  will be larger than 1 and another half smaller. Note that we have  $\sum_{u \in U_b} f_w(S(T_u)) = b$ ; such transformation guarantees the same learning strength compared with the original prediction model.

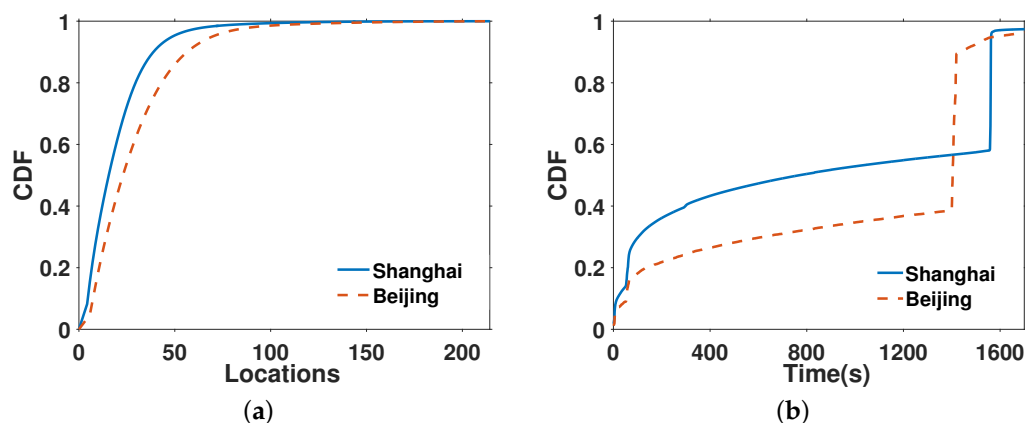
### 3.5. Dataset

We used the cellular network datasets, which were collected by collaborating with a major cellular network operator in China from two metropolises, Shanghai and Beijing, with a one-week duration. To fully protect user privacy, we only use the data authorized by the user, and all data are stored on the cloud platform. We are allowed to run the model and view the statistics and analysis results but have no right to view the specific content of the data. The cellular network data record the spatio-temporal information of mobile users when they access the cellular network for communication. Each record is characterized by an anonymous user ID for privacy protection, time stamp, ID type of user event (including call, text, or heartbeat event), and the location of the connected cellular tower.

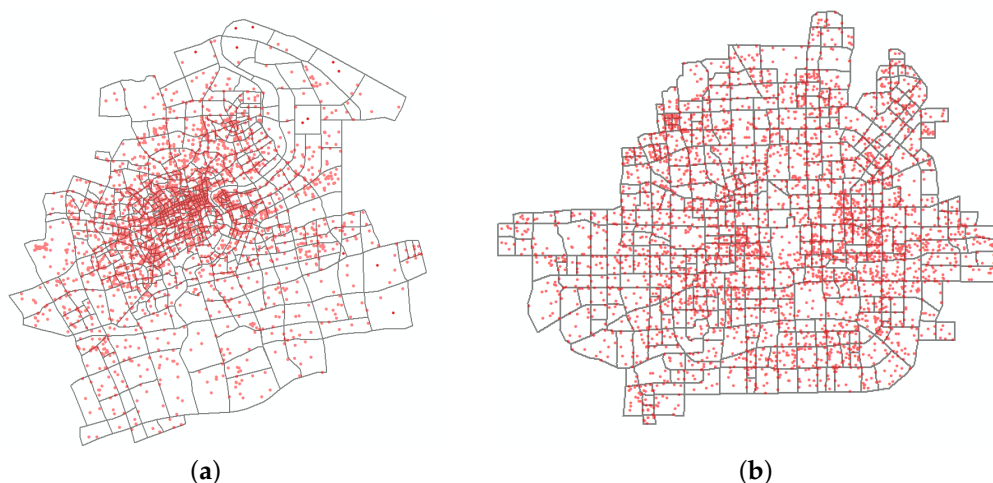
Calls and texts of user events are user-active communication behaviors, while heartbeat events are automatic location updates from cell towers. More specifically, if a mobile subscriber leaves no communication traces for about half an hour, the operator backend automatically records the current location based on the base stations that the user connects to. Figure 6 shows some other statistics of our dataset. It shows that heartbeat events make up most of the dataset and can significantly affect the distribution of time intervals. In fact, for each user, the interval between two consecutive heartbeat records does not strictly accord with a fixed half-hour gap but takes 30 min as an upper limit.



The spatial granularity of our datasets is the cellular base station, which is usually considered to cover a region formed by Thiessen Polygon [47] with a radius ranging from a few hundred meters to a few kilometers. However, urban regions segmented by road networks (street blocks) are the basic geographic units of residents’ daily lives [48], which are more appropriate as the basic units for the study of mobility prediction. Here, we crawl the road network data from the online map service and analyze each urban region. Then, we project the locations of cellular base stations into these street blocks. Finally, the main urban areas of Shanghai and Beijing are divided into hundreds of non-overlapping regions, as shown in Figure 7. After primary data cleaning and oscillation resolution [15], both datasets contain over two million mobile users and thousands of base stations.



**Figure 6.** Spatio-temporal features of two mobility datasets, with Cumulative Distribution Function of (a) different locations that users visit and (b) time intervals between consecutive records.



**Figure 7.** Visualization of road network segmentations in (a) Shanghai and (b) Beijing. The red dots represent locations of cellular base stations.

Different from the sparse social media check-in data [49], which are collected when users check in on mobile applications, the time interval between two consecutive records of one user is usually around 30 min in cellular network data. In practice, many users always stay in a street block for several hours during the day. Without deleting the duplication in raw trajectories, the prediction model can achieve excellent but meaningless results by repeating the current location. Thus, we take 30 min as the essential temporal granularity to obtain meaningful trajectories from cellular network records and split one day into 48 pieces. Then, we aggregate the records in the same time slot and remove the duplication of locations with keeping the first and last records during this location-stable period. For other detailed information on our datasets refer to Table 1.

**Table 1.** Detailed information of cellular network datasets.

Dataset	Metrics	Numerical Value
Shanghai	Duration	Jan 14th to 20th, 2019
	Amount of Mobile Users	2,1728,35
	Amount of Records	1,463,855,232
	Amount of Street Blocks	719
	Amount of Cellular Towers	4,126
	Percentage of Heartbeat	84.25%
	Percentage of Call Event	8.13%
Beijing	Percentage of Text Event	7.62%
	Duration	Mar 5th to 11th, 2018
	Amount of Mobile Users	2,8436,31
	Amount of Records	1,992,757,036
	Amount of Street Blocks	594
	Amount of Cellular Towers	3,844
	Percentage of heartbeat	86.94%
Percentage of call event	6.69%	
	Percentage of text event	6.57%

### 3.6. Data Selection

A common strategy for training and testing the mobility prediction model is to select each user’s first 80% sub-trajectories as the training set and the remaining 20% as testing data [11,42]. Considering that citizens often present different traffic patterns in weekdays and weekends [50], we perform a minor correction on the choice of train-test mode: we denote the whole dataset as  $T = \{T_u | u \in U\}$  with user set  $U$  of size  $N$ , each user’s training trajectories as  $T_u^{train} = T_u^{Mon} \dots T_u^{Thu} T_u^{Sat}$ , and take  $T_1 = \{T_u^{train} | u \in U\}$  as the training set. The remaining trajectories  $T_u^{test} = T_u^{Fri} T_u^{Sun}$  are used as the preliminary testing set.

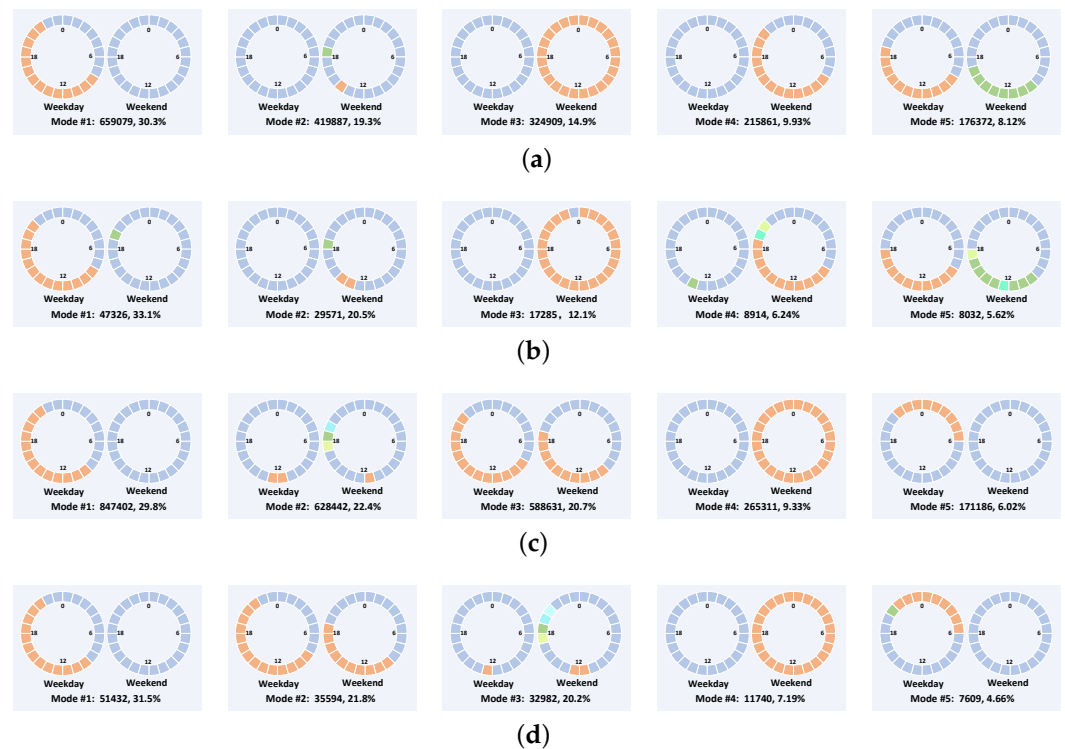
As depicted in Figure 1, there are some doubtful sub-trajectories in cellular network data whose correctness is hard to judge. To guarantee the accuracy of location labels in testing data, we filter out all these unidentified trajectories among  $T_u^{test}$ , along with corresponding users whose trajectories are too sparse to utilize after the selection: only 142,853 mobile users in Shanghai and 163,277 users in Beijing remain. We assume that a strict filter is the only way to guarantee the accuracy of testing data without ground-truth validation. We denote the set of these selected users as  $U_2$  and take trajectory set  $T_2 = \{T_u^{test} | u \in U_2\}$  as final testing data. It is clear that  $U_2 \in U$  and  $T_u^{train} \cap T_u^{test} = \emptyset, T_u = T_u^{train} \cup T_u^{test}$ .

To evaluate the effect of our selection strategy of testing data, we randomly extract 500 users from  $U$  and  $U_2$  and project their trajectories  $T_u^{test}$  on the geographic map with the help of an online map API. Five human judges are asked to rate the interpretability and rationality of these  $T_u^{test}$  on a 3-point scale: - 0 means trajectories of this user are disorganized and can only be roughly identified for residential and place of work (totally abnormal trajectories have already been washed in former pre-processing); 1 indicates that the user’s mobility shows a regular pattern to some extent, though some sub-trajectories are still hard to understand; and 2 is for almost reasonable trajectories. For each group of samples, we calculate the ratio of each score (0, 1, 2) by combining all the human annotations and taking their average score as the human evaluation result. The details of human evaluation are presented in Table 2. Fleiss’ Kappa [51] is often used as a statistic measure for assessing the reliability of agreement between human annotators. In our cases, the Fleiss’ Kappa is computed all around 0.34, which demonstrates a fair agreement between human raters. We can see that the trajectory samples of  $U_2$  show superior interpretability to those of the original  $U$ , which we believe guarantees accurate location labels in testing data.

**Table 2.** Human evaluation of data quality of sampling test data.

Samples	City	0-Ratio	1-Ratio	2-Ratio	Avg-Score
$T_2$	Shanghai	0.072	0.304	0.624	1.552
$T_1$	Shanghai	0.220	0.612	0.168	0.948
$T_2$	Beijing	0.038	0.336	0.626	1.588
$T_1$	Beijing	0.196	0.580	0.224	1.028

A natural doubt is that our selection strategy of testing data may wash the noisy and normal trajectories altogether, leading to an uneven distribution of user types. In other words, we do not know whether our data selection strategy will have an unfair preference for specific user groups. The distribution of users' traffic patterns in  $T_2$  may differ from those in  $T_1$ . It is essential to check the distribution of user groups when making data selections because we cannot train trajectories from a group of office workers to predict the mobility of another group of taxi drivers. This discrepancy will certainly disturb the training process and affect the fairness of the final results of mobility prediction. Here, we employ a method to detect popular temporal modes mentioned in [52]. More specifically, we extract users' traffic representations via time partition from their mobility records and group them through unsupervised clustering. Then, the center representations of main clusters can be seen as the popular temporal modes. We generally list five main temporal modes in our analysis as their corresponding users jointly account for more than 80% of total users in  $U$  and  $U_2$ , from Shanghai and Beijing, respectively. The results shown in Figure 8 prove that the decrease in users in  $U_2$  caused by our data selection method does not change the composition of user groups, which protects the effectiveness and fairness of the mobility prediction model. Finally, the trajectory quality calibration framework takes  $T_1$  as training data and  $T_2$  as the testing set.



**Figure 8.** Visualization of popular temporal modes. (a–d) correspond to 5 main modes in  $T_1$  and  $T_2$ , from Shanghai and Beijing, respectively. Each mode is composed of two annular representations, for weekday patterns and weekend patterns. An annular representation contains 24 time slices, aiming to describe users' traffic patterns on a typical day. Time slices with the same colors mean that the user tends to stay in the same location during these periods. (a) Five main popular temporal modes in Shanghai ( $T_1$ ); (b) five main popular temporal modes in Shanghai ( $T_2$ ); (c) five main popular temporal modes in Beijing ( $T_1$ ); (d) five main popular temporal modes in Beijing ( $T_2$ ).

### 4. Results

In this section, we conduct experiments on two citywide cellular network datasets to compare the performance of our approach against several baselines. Our trajectory quality calibration framework is designed to be model-independent, so we mainly compare the performance of the state-of-art mobility prediction model with and without employing our

calibration network and instance-weighting approach. In addition, some classical methods, such as Markov models, are also tested to evaluate their performance on massive cellular data, though they are not adaptive to the instance-weighting strategy.

#### 4.1. Baselines and Metrics

- **Markov Chain**—A Markov Chain is used to predict human mobility for a long time. It regards the visited locations as states and builds a transition matrix to capture the first-order transition probabilities between them. A Markov Chain is unsuitable for adopting the instance-weighting strategy as it owns a different process of parameter updates.
- **PMM**—The Periodic Mobility Model (PMM) [3] assumes that mobility trajectories follow a spatio-temporal mixture model and predict the next locations with periodicity taken into consideration. Like the Markov Chain, PMM is essentially a two-state mixture of Gaussians with a time-dependent state prior and is not adaptive for a combination of the calibration model.
- **RNN**—ST-RNN is a popular recurrent model in location prediction, which focuses on modeling the continuous spatio-temporal information within the framework of RNN. Here, we adopt a variant ST-RNN (RNN in short) to our scene where only anonymous location ID is known instead of detailed geographic information.
- **DeepTransport**—The deep learning module of DeepTransport [53] can be regarded as a multi-layer LSTM network: two hidden layers share the same parameters to capture the long-term temporal dependency of human mobility and transportation patterns, along with one encoding layer for separated input sequence and one decoding layer for separated output sequences. LSTM is kept for the basic recurrent module of DeepTransport as the design in the original version.
- **DeepMove**—DeepMove is the first historical attention method for learning human mobility from current and historical trajectories. It can be considered a complex version of baseline RNN with a historical attention module.

To make fair comparisons, we use Top@k as the standard performance metrics of evaluation. More specifically, we rank the candidate locations by the probabilities generated from each model and check whether the ground-truth location appears in the top-k candidate locations. To prepare for the following experiments, for recurrent sub-networks in our calibration model and other baselines, we test using both LSTM and GRU cells. However, the results show no noticeable differences in accuracy. As GRU cells are faster to train than LSTM cells, we utilize GRU as the default recurrent module of our models in the following experiments. Additionally, considering the randomness of negative sampling in the training process of the calibration model, for the RNN, DeepTransport, and DeepMove with calibration, we repeat our experiments three times with different random seeds, with average values shown in Table 3. The neural models are implemented on the Pytorch platform and speeded up using a Tesla T4 GPU. Other experimental settings are shown in Table 4.

**Table 3.** Comparison of overall prediction accuracy on datasets in Shanghai and Beijing. Methods combined with our calibration approach are shown in bold.

Method	Shanghai			Beijing		
	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10
Markov	13.03%	22.50%	23.94%	14.74%	24.35%	26.08%
PMM	14.34%	23.52%	25.69%	13.67%	24.72%	26.79%
RNN	18.28%	29.37%	33.50%	19.14%	31.45%	34.83%
<b>RNN with Calibration</b>	<b>21.73%</b>	<b>32.65%</b>	<b>35.33%</b>	<b>22.07%</b>	<b>33.18%</b>	<b>36.40%</b>
DeepTransport	17.84%	28.79%	32.96%	18.66%	28.90%	32.34%
<b>DeepTransport with Calibration</b>	<b>20.13%</b>	<b>31.08%</b>	<b>34.37%</b>	<b>21.95%</b>	<b>31.19%</b>	<b>35.23%</b>
DeepMove	19.16%	33.05%	36.27%	20.57%	33.89%	36.03%
<b>DeepMove with Calibration</b>	<b>22.81%</b>	<b>35.45%</b>	<b>38.90%</b>	<b>23.11%</b>	<b>35.70%</b>	<b>38.92%</b>

**Table 4.** Experimental settings of calibration network.

Training Setting	Value	Model Setting	Value
learning rate ( $lr$ )	5e-4	hidden size	300
threshold $\theta$	0.3	embedding size (time)	16
dropout rate	0.5	embedding size (location)	256
L2 penalty	1e-5	batch size	128

#### 4.2. Experimental Results

Table 3 shows the overall performance comparison of our datasets from Shanghai and Beijing. Deep neural network based methods present much higher prediction accuracy than traditional feature-based models. Compared with Markov Chain and PMM, RNN-based models can capture more users' transitional regularities under massive noisy trajectories. Adequate training data help perfect the parameter updates of the recurrent network and spatio-temporal context embedding.

RNN, DeepTransport, and DeepMove use the recurrent network to capture long-term mobility dependency. In our experiments, there is no significant difference between the performance of these three methods. Compared with RNN and DeepTransport, the increase in prediction accuracy of DeepMove is almost offset by its large time consumption in training. This is probably because our datasets' limited time duration constrains the attention mechanism's performance, as a common GRU or LSTM network can also work well regardless of the gradient vanishing problem [9]. This situation may change when dealing with longer-duration mobility data or denser daily tracks. Additionally, the design architecture of multi-layer recurrent networks in DeepTransport shows no apparent superiority to a common RNN in our experiments.

With the help of the instance-weighting strategy, the variant RNN, DeepTransport, or DeepMove combined with our calibration model outperforms the corresponding baselines, demonstrating our approach's adaptiveness and effectiveness. We hypothesize that the recurrent neural sub-network captures the consistency between daily spatio-temporal transitions—mainly reflecting the regularity of human mobility, and the concatenation of trajectory embeddings measures the correlation extent of trajectories on different dates—which probably reflects the periodicity of human mobility. Trajectories with high-level noises then obtain poor scores on these metrics. Thus, these scores successfully represent data quality and affect the original prediction model. Our trajectory quality calibration framework significantly outperforms the conventional methods without a quality-estimation module when learning from massive and noisy mobility data.

## 5. Discussion

In our proposed framework, we elaborate a calibration model to estimate the quality of user trajectories. The calibration model is designed to give a quality score in the range of  $[0,1]$ . In this section, we analyze these numerical scores in detail. The distribution of the scores  $S(T_u^{train}), u \in U$  is visualized in Figure 9. The scores concentrate on  $[0.3, 0.6]$ , with an average of 0.489 in Shanghai and 0.501 in Beijing, which accords with the threshold settings in Table 4. Additionally, the slight increase in the estimated qualities in the Beijing dataset may explain the corresponding increase in prediction accuracy compared with the Shanghai dataset. In addition, around 15.3% of users' trajectories obtain a score lower than 0.3, which indicates that almost 15% of training cases are estimated as low-quality data. There are also more than 16.2% of the training cases with scores larger than 0.7, generally evaluated as high-quality trajectories. Therefore, it is essential to adopt our instance-weighting approach to control the effect of the low-quality data on the final prediction model.

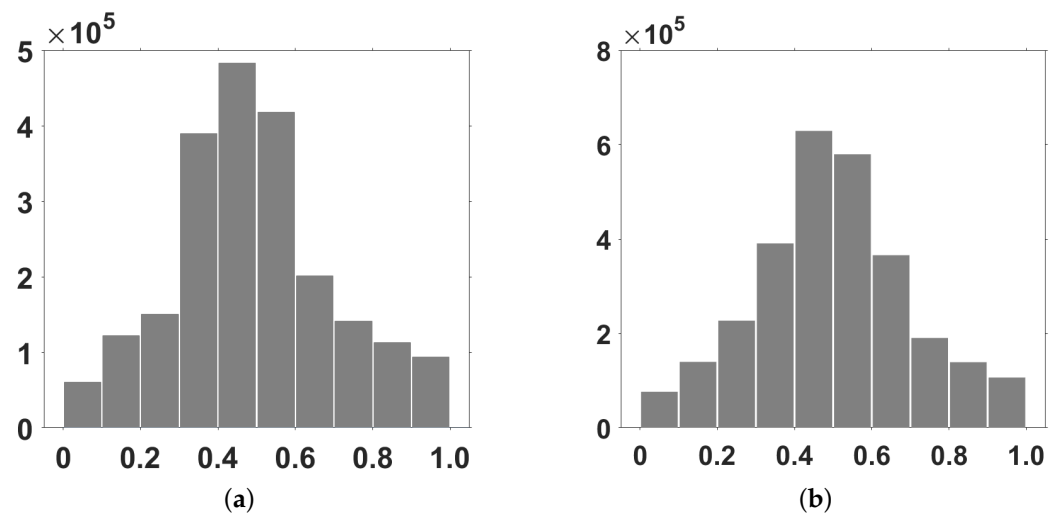


Figure 9. Score distribution of trajectories in  $T_1$ , with (a) Shanghai and (b) Beijing.

To further investigate the performance of our calibration model, we adopt a human evaluation method similar to that in *Data Selection*, where five human judges are asked to rate the interpretability of trajectories of each user on a 3-points scale from incomprehensible to reasonable. We randomly sample 100 mobile users from high-quality ( $S(T_u^{train}) > 0.7$ ) and low-quality ( $S(T_u^{train}) < 0.3$ ) instances, respectively. The statistics of human evaluation are presented in Table 5. The Fleiss' kappas of human annotations on these four sample sets are all around 0.38, which demonstrates the fairness of inter-human agreements. As we can see, though human raters usually prefer to give a score of 1 due to cautiousness and conservativeness, the quality scores estimated from our calibration model accord with human evaluation in general, proving the effectiveness of our quality estimation module.

Table 5. Human evaluation on data quality of sampling data tagged by calibration network.

Samples	City	0-Ratio	1-Ratio	2-Ratio	Avg-Score
High-quality	Shanghai	0.016	0.414	0.570	1.554
Low-quality	Shanghai	0.328	0.626	0.046	0.718
High-quality	Beijing	0.022	0.386	0.592	1.570
Low-quality	Beijing	0.446	0.526	0.028	0.582

In conclusion, we investigate the problem of predicting citywide human mobility from noisy mobility trajectories. The proposed trajectory quality calibration framework aims to protect the mobility prediction model from noises of oscillation. A neural calibration network automatically estimates the data quality, while the prediction model utilizes the quality scores for the weighted update of its parameters. Extensive experiments on real-life cellular mobility datasets prove the effectiveness of our instance-weighting approach.

Our work also has limitations, so our future work will focus on several directions. First, our method is suitable for massive noisy data, while sparse or short-duration trajectory data may not complete the correlation calculation between daily mobility in the calibration module. Second, the limits on our dataset constrain the performance and validation of several related methods, such as the average sampling strategy and the relatedness calculation among daily mobility. In the future, we plan to validate and improve our approach to other data sources with longer duration. Finally, this work concentrates on the problem of data quality. When dealing with mobility data from different cities, we currently train their prediction model, which leads to high computation consumption. From the view of domain adaptation, we plan to investigate the problem of transferring our trajectory quality calibration framework between different cities in the future, which will undoubtedly benefit the application of mobility prediction.



**Author Contributions:** Conceptualization, Q.M., M.L. and Y.Q.; methodology, Q.M.; software, Q.M.; validation, Q.M., W.L. and Z.W.; formal analysis, Q.M.; investigation, Q.M. and M.L.; resources, J.X.; data curation, H.S.; writing—original draft preparation, Q.M.; writing—review and editing, M.L.; visualization, Q.M.; supervision, N.S.; project administration, Q.M.; funding acquisition, Y.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded in part by the funds of the major science and technology projects of Anhui province (202203a05020025) and in part by the national social science fund research project of ideological and political theory course in colleges and universities “Research on the application of mobile app in the teaching of ideological and political theory course in colleges and universities” (20VSZ042).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GRU	Gate Recurrent Unit
GPU	Graphic Processing Unit
POI	Point of Interest
PRME	Personalized Ranking Metric Embedding
ST-RNN	Spatial Temporal Recurrent Neural Networks
HSMM	Hidden Semi-Markov Model
HMM	Hidden Markov Model
BiLSTM-CNN	Bidirectional Long Short-Term Memory–Convolutional Neural Network
DWSTTN	Deep Wide Spatio-Temporal Transformer Network
GCDAN	Graph Convolutional Dual-Attentive Networks
PMM	Periodic Mobility Model

### References

1. Song, C.; Qu, Z.; Blumm, N.; Barabási, A.L. Limits of predictability in human mobility. *Science* **2010**, *327*, 1018–1021. <https://doi.org/10.1126/science.1177170>.
2. Lu, X.; Wetter, E.; Bharti, N.; Tatem, A.J.; Bengtsson, L. Approaching the limit of predictability in human mobility. *Sci. Rep.* **2013**, *3*, 2923. <https://doi.org/10.1038/srep02923>.
3. Cho, E.; Myers, S.A.; Leskovec, J. 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, San Diego, CA, USA, 21–24 August 2011; pp. 1082–1090. <https://doi.org/10.1145/2020408.2020579>.
4. Mathew, W.; Raposo, R.; Martins, B. Predicting future locations with hidden Markov models. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 911–918. <https://doi.org/10.1145/2370216.2370421>.
5. Yan, M.; Li, S.; Chan, C.A.; Shen, Y.; Yu, Y. Mobility Prediction Using a Weighted Markov Model Based on Mobile User Classification. *Sensors* **2021**, *21*, 1740. <https://doi.org/10.3390/s21051740>.
6. Wang, H.; Li, Y.; Jin, D.; Han, Z. Attentional Markov Model for Human Mobility Prediction. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2213–2225. <https://doi.org/10.1109/JSAC.2021.3078499>.
7. Ma, Z.; Rana, P.K.; Taghia, J.; Flierl, M.; Leijon, A. Bayesian estimation of Dirichlet mixture model with variational inference. *Pattern Recognit.* **2014**, *47*, 3143–3157. <https://doi.org/10.1016/j.patcog.2014.04.002>.
8. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. *arXiv* **2015**, arXiv:1506.00019.
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
10. Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
11. Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; Jin, D. Deepmove: Predicting human mobility with attentional recurrent networks. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1459–1468. <https://doi.org/10.1145/3178876.3186058>.

12. Dang, W.; Wang, H.; Pan, S.; Zhang, P.; Zhou, C.; Chen, X.; Wang, J. Predicting Human Mobility via Graph Convolutional Dual-attentive Networks. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Houston, Texas, USA, 21–25 February 2022; pp. 192–200. <https://doi.org/10.1145/3488560.3498400>.
13. Wang, H.; Yu, Q.; Liu, Y.; Jin, D.; Li, Y. Spatio-Temporal Urban Knowledge Graph Enabled Mobility Prediction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–24. <https://doi.org/10.1145/3494993>.
14. Jang, H.S.; Baek, J.H. Mobility Management Scheme with Mobility Prediction in Wireless Communication Networks. *Appl. Sci.* **2022**, *12*, 1252. <https://doi.org/10.3390/app12031252>.
15. Qi, L.; Qiao, Y.; Abdesslem, F.B.; Ma, Z.; Yang, J. Oscillation resolution for massive cell phone traffic data. In Proceedings of the First Workshop on Mobile Data, Porto, Portugal, 13–16 June 2016; pp. 25–30. <https://doi.org/10.1145/2935755.2935759>.
16. Bayir, M.A.; Demirbas, M.; Eagle, N. Mobility profiler: A framework for discovering mobility profiles of cell phone users. *Pervasive Mob. Comput.* **2010**, *6*, 435–454. <https://doi.org/10.1016/j.pmcj.2010.01.003>.
17. Calabrese, F.; Di Lorenzo, G.; Liu, L.; Ratti, C. Estimating origin-destination flows using mobile phone location data. In Proceedings of the IEEE Pervasive Computing, Seattle, WA, USA, 21–25 March 2011; pp. 36–44.
18. Wu, W.; Wang, Y.; Gomes, J.B.; Anh, D.T.; Antonatos, S.; Xue, M.; Yang, P.; Yap, G.E.; Li, X.; Krishnaswamy, S.; et al. Oscillation resolution for mobile phone cellular tower data to enable mobility modelling. In Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management, Brisbane, QLD, Australia, 15–18 July 2014; Volume 1, pp. 321–328. <https://doi.org/10.1109/MDM.2014.46>.
19. Wang, F.; Chen, C. On data processing required to derive mobility patterns from passively-generated mobile phone data. *Transp. Res. Part C Emerg. Technol.* **2018**, *87*, 58–74. <https://doi.org/10.1016/j.trc.2017.12.003>.
20. Xu, Y.; Li, X.; Shaw, S.L.; Lu, F.; Yin, L.; Chen, B.Y. Effects of Data Preprocessing Methods on Addressing Location Uncertainty in Mobile Signaling Data. *Ann. Am. Assoc. Geogr.* **2021**, *111*, 515–539. <https://doi.org/10.1080/24694452.2020.1773232>.
21. Lison, P.; Bibauw, S. Not all dialogues are created equal: Instance weighting for neural conversational models. *arXiv* **2017**, arXiv:1704.08966.
22. Tao, C.; Mou, L.; Zhao, D.; Yan, R. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
23. Monreale, A.; Pinelli, F.; Trasarti, R.; Giannotti, F. Wherenext: a location predictor on trajectory pattern mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 637–646. <https://doi.org/10.1145/1557019.1557091>.
24. Giannotti, F.; Nanni, M.; Pinelli, F.; Pedreschi, D. Trajectory pattern mining. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Nanjing, China, 22–25 May 2007; pp. 330–339. <https://doi.org/10.1145/1281192.1281230>.
25. Peng, C.; Jin, X.; Wong, K.C.; Shi, M.; Liò, P. Collective human mobility pattern from taxi trips in urban area. *PLoS ONE* **2012**, *7*, e34487. <https://doi.org/10.1371/journal.pone.0034487>.
26. Noulas, A.; Scellato, S.; Lathia, N.; Mascolo, C. Mining user mobility features for next place prediction in location-based services. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining, Brussels, Belgium, 10–13 December 2012; pp. 1038–1043. <https://doi.org/10.1109/ICDM.2012.113>.
27. Feng, S.; Cong, G.; An, B.; Chee, Y.M. Poi2vec: Geographical latent representation for predicting future visitors. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017. <https://doi.org/10.1609/aaai.v31i1.10500>.
28. Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y.M.; Yuan, Q. Personalized ranking metric embedding for next new poi recommendation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
29. Gambs, S.; Killijian, M.O.; del Prado Cortez, M.N. Next place prediction using mobility markov chains. In Proceedings of the First Workshop on Measurement, Privacy, and Mobility, Bern, Switzerland, 10–12 April 2012; pp. 1–6. <https://doi.org/10.1145/2181196.2181199>.
30. Chen, M.; Liu, Y.; Yu, X. Nlpm: A next location predictor with markov modeling. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Tainan, Taiwan, 13–16 May 2014; pp. 186–197. [https://doi.org/10.1007/978-3-319-06605-9\\_16](https://doi.org/10.1007/978-3-319-06605-9_16).
31. Amirrudin, N.A.; Ariffin, S.H.; Malik, N.A.; Ghazali, N.E. User’s mobility history-based mobility prediction in LTE femtocells network. In Proceedings of the 2013 IEEE International RF and Microwave Conference (RFM), Penang, Malaysia, 9–11 December 2013; pp. 105–110. <https://doi.org/10.1109/RFM.2013.6757228>.
32. Yu, S.Z.; Kobayashi, H. A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking. *Signal Process.* **2003**, *83*, 235–250. [https://doi.org/10.1016/S0165-1684\(02\)00378-X](https://doi.org/10.1016/S0165-1684(02)00378-X).
33. Kulkarni, V.; Garbinato, B. 20 Years of Mobility Modeling & Prediction: Trends, Shortcomings & Perspectives. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 5–8 November 2019; pp. 492–495. <https://doi.org/10.1145/3347146.3359110>.
34. Kulkarni, V.; Mahalunkar, A.; Garbinato, B.; Kelleher, J.D. Examining the limits of predictability of human mobility. *Entropy* **2019**, *21*, 432. <https://doi.org/10.3390/e21040432>.

35. Bao, Y.; Huang, Z.; Li, L.; Wang, Y.; Liu, Y. A BiLSTM-CNN model for predicting users' next locations based on geotagged social media. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 639–660. <https://doi.org/10.1080/13658816.2020.1808896>.
36. Abideen, Z.U.; Sun, H.; Yang, Z.; Ahmad, R.Z.; Iftekhhar, A.; Ali, A. Deep wide spatial-temporal based transformer networks modeling for the next destination according to the taxi driver behavior prediction. *Appl. Sci.* **2020**, *11*, 17. <https://doi.org/10.3390/app11010017>.
37. Zhang, H.; Dai, L. Mobility prediction: A survey on state-of-the-art schemes and future applications. *IEEE Access* **2018**, *7*, 802–822. <https://doi.org/10.1109/ACCESS.2018.2885821>.
38. Jiang, J.; Zhai, C. Instance weighting for domain adaptation in NLP. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 23–30 June 2007; pp. 264–271.
39. Wang, R.; Utiyama, M.; Liu, L.; Chen, K.; Sumita, E. Instance weighting for neural machine translation domain adaptation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1482–1488. <https://doi.org/10.18653/v1/D17-1155>.
40. Shang, M.; Fu, Z.; Peng, N.; Feng, Y.; Zhao, D.; Yan, R. Learning to Converse with Noisy Data: Generation with Calibration. In Proceedings of the International Joint Conferences on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4338–4344.
41. Yang, H.; Liu, T.; Sun, Y.; Bertino, E. Exploring The Interaction Effects for Temporal Spatial Behavior Prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2013–2022. <https://doi.org/10.1145/3357384.3357963>.
42. Gao, Q.; Zhou, F.; Trajcevski, G.; Zhang, K.; Zhong, T.; Zhang, F. Predicting human mobility via variational attention. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2750–2756. <https://doi.org/10.1145/3308558.3313610>.
43. Al-Molegi, A.; Jabreel, M.; Ghaleb, B. STF-RNN: Space Time Features-based Recurrent Neural Network for predicting people next location. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–7. <https://doi.org/10.1109/SSCI.2016.7849919>.
44. Yuan, Q.; Zhang, W.; Zhang, C.; Geng, X.; Cong, G.; Han, J. 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, Cambridge, UK, 6–10 February 2017; pp. 263–272. <https://doi.org/10.1145/3018661.3018680>.
45. Gao, Q.; Zhou, F.; Zhang, K.; Trajcevski, G.; Luo, X.; Zhang, F. Identifying Human Mobility via Trajectory Embeddings. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, VIC, Australia, 19–25 August 2017; Volume 17, pp. 1689–1695.
46. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
47. Brassel, K.E.; Reif, D. A procedure to generate Thiessen polygons. *Geogr. Anal.* **1979**, *11*, 289–303. <https://doi.org/10.1111/j.1538-4632.1979.tb00695.x>.
48. Yuan, N.J.; Zheng, Y.; Xie, X.; Wang, Y.; Zheng, K.; Xiong, H. Discovering urban functional zones using latent activity trajectories. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 712–725. <https://doi.org/10.1109/TKDE.2014.2345405>.
49. Belcastro, L.; Cantini, R.; Marozzo, F. Knowledge Discovery from Large Amounts of Social Media Data. *Appl. Sci.* **2022**, *12*, 1209. <https://doi.org/10.3390/app12031209>.
50. Agarwal, A. A Comparison of Weekend and Weekday Travel Behavior Characteristics in Urban Areas. Master's Thesis, University of South Florida, Tampa, FL, USA, 2004.
51. Fleiss, J.L.; Cohen, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ. Psychol. Meas.* **1973**, *33*, 613–619. <https://doi.org/10.1177/001316447303300309>.
52. Xu, F.; Xia, T.; Cao, H.; Li, Y.; Sun, F.; Meng, F. Detecting popular temporal modes in population-scale unlabelled trajectory data. *Proc. Acm Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–25. <https://doi.org/10.1145/3191778>.
53. Song, X.; Kanasugi, H.; Shibasaki, R. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level. In Proceedings of the International Joint Conferences on Artificial Intelligence, New York, New York, USA, 9–15 July 2016; Volume 16, pp. 2618–2624.