Article

Vehicle Trajectory Prediction Based on Local Dynamic Graph Spatiotemporal-Long Short Term Memory Model

Juan Chen *, Qinxuan Feng, Daiqian Fan

Posted Date: 23 November 2023

doi: 10.20944/preprints202311.1498.v1

Keywords: Vehicle trajectory; Location prediction; Knowledge graph; Points of interest; Graph attention network

Article

Vehicle Trajectory Prediction Based on Local Dynamic Graph Spatiotemporal- Long Short Term Memory Model

Juan Chen [1,2,*], Qinxuan Feng [1] and Daiqian Fan [1]

[1] SHU-UTS SILC Business School, Shanghai University, Shanghai 201899, China; chenjuan82@shu.edu.cn (Q.F.)
[2] Smart City Research Institute, Shanghai University, Shanghai 201899, China
* Correspondence: chenjuan82@shu.edu.cn; Tel.: +86-10-6998-0028

Abstract: Traffic congestion and frequent traffic accidents have become main problems affecting urban traffic. Effective location prediction of vehicle trajectory can help alleviate traffic congestion, reduce the occurrence of traffic accidents, and optimize the urban traffic system. Vehicle trajectory is closely related to the surrounding Points of Interest (POI). POI can be considered as the spatial feature and can be fused with trajectory points to improve prediction accuracy. A Local Dynamic Graph Spatiotemporal- Long Short-Term Memory (LDGST-LSTM) is proposed in this paper to extract and fuse the POI knowledge and realize next location prediction. POI semantic information is learned by constructing the traffic knowledge graph, and spatial and temporal features are extracted by combining Graph Attention Network (GAT) and temporal attention mechanism. Moreover, the weights of POI that influence location prediction are visualized to improve the interpretability of the proposed model. The effectiveness of LDGST-LSTM is verified on two datasets, including Chengdu taxi trajectory data in October 2018 and August 2014. The accuracy and robustness of the proposed model are significantly improved compared with the benchmark models.

Keywords: vehicle trajectory; location prediction; knowledge graph; points of interest; graph attention network

1. Introduction

Traffic congestion and frequent traffic accidents have become serious problems affecting urban traffic with the rapid development of cities [1]. The Intelligent Transportation System (ITS) is proposed to alleviate these traffic problems. Location prediction of the vehicle trajectory is one of the research topics in the ITS [2]. By effective location prediction, the traffic congestion can be alleviated and reduced, which is of great significance to optimize traffic management. The goal of this task is to predict the next location according to the observed trajectory [3]. Deep learning methods has been widely used in vehicle trajectory location prediction. Vehicle trajectory is usually considered as a time series since it is a sequence of trajectory points [1]. Long Short-Term Memory (LSTM) is mainly used to deal with time sequences. Li et al. [4] apply a LSTM to capture temporal features between vehicles and design a convolutional social pooling network to capture spatial features to predict vehicle trajectory. However, complex roads and surrounding buildings could become the factors affecting prediction. Therefore, how to integrate external features with vehicle trajectory to improve the accuracy and robustness of model needs further research.

Knowledge graph, GAT, and time attention mechanism are the latest methods used in the field of transportation and have shown better performance in improving the accuracy and robustness of model [5]. Knowledge graph can help fuse features. GAT can be used as spatial attention mechanism to effectively extract spatial features, and temporal attention mechanism can effectively extract temporal features. The application of knowledge graph has developed rapidly in the field of

doi:10.20944/preprints202311.1498.v1

transportation. A multi-layer traffic knowledge graph model is proposed by Wang [6] to realize destination prediction by modeling the complex traffic network, weighted summing, and fusing different node features. The driving intention in the traffic network is affected by the functional area and the surrounding POI [7]. Therefore, it is important to predict the location of vehicle trajectory by designing traffic knowledge graph. The accuracy can be improved by interacting the vehicle trajectory with the surrounding environment [7]. The driving intention of vehicles is closely related to the surrounding Points of Interest (POI). POI can be considered as the spatial feature and can be fused with trajectory points to improve prediction accuracy [8]. Trajectory may change when there are traffic jams or traffic accidents, and the POI knowledge around trajectory points has an impact on the future trend. By integrating POI knowledge and vehicle trajectory, the prediction can be improved to help optimize traffic management. Graph Attention Network (GAT) [9] and various temporal attention mechanisms [10] are also widely used for extract spatial and temporal features. A new GAT is proposed by Su et al. [11] to allocate attention to explain and express the spatial correlation of road network, as well as effectively capture the dynamic update of adjacent transformation matrix. LSTM combined with the attention mechanism are proposed by Ali et al. [12] to capture temporal features. However, how to fuse the surrounding environment features such as POI, and the synergy of knowledge graph, GAT, as well as temporal attention mechanism on location prediction task still lack research.

A Local Dynamic Graph Spatiotemporal- Long Short-Term Memory (LDGST-LSTM) is proposed in this paper to predict the next location of vehicle trajectory. POI knowledge is extracted and fused by constructing a traffic knowledge graph. GAT and temporal attention mechanism are combined to improve the accuracy, robustness, and interpretability of the prediction model. The main contributions of this paper are as follows.

- Knowledge graph and spatiotemporal attention mechanism are combined in this paper to predict the vehicle location at the next moment. POI is integrated with historical trajectory, and the POI weights that affect the prediction are visualized additionally. Regions that have a great impact on the prediction is explored, and the interpretability of model is enhanced.
- A global traffic knowledge graph is constructed to learn and represent POI semantic information. POI nodes are considered as the entity, and the connection between POI is considered as the relationship. The representation vector of each node is obtained by Translate Embedding (TransE) algorithm and is considered as the feature vector for vehicle location prediction.
- A spatiotemporal attention mechanism is designed to allocate weights for spatial and temporal features, thus enhancing the interpretability and accuracy of model. The weight distribution of spatial features is achieved through GAT to obtain the corresponding graph representation vector. LSTM combined with multi-head attention mechanism is used to allocate weights of trajectory points at different timestamps to improve the prediction accuracy.

2. Related work

This section will provide a literature review on related works, including vehicle trajectory location prediction, knowledge graph, and attention mechanism.

2.1. Vehicle trajectory location prediction

Deep learning method is widely used in location prediction of vehicle trajectory. A vehicle trajectory prediction model combining deep encoder-decoder and neural network is designed by Fan et al. [13]. Multiple segments are processed and corrected by the deep neural network model, which improves the accuracy of the prediction and realizes long-term prediction [14]. Kalatan and Farooq [15] propose a new multi-input LSTM to fuse sequence data with context information of the environment. It can reduce prediction error but more external features that could improve the prediction performance are ignored. Graph Convolutional Network (GCN) combined with its variant model are proposed by An et al. [16] to realize vehicle trajectory prediction in urban traffic system.

Through this conjunction, model can effectively capture the spatiotemporal characteristics, therefore improving the effectiveness of vehicle trajectory prediction.

Methods based on deep learning algorithm can effectively improve the prediction accuracy and reduce error. Through feature mining and extraction, deep learning model can effectively improve its accuracy and robustness. However, the impact of surrounding buildings, or to say, Point-of-Interest (POI), on prediction under complex traffic environment is usually ignored in existed research. Therefore, how to better integrate external factors such as POI features needs further research to help improve the prediction performance.

The efficiency and accuracy of vehicle trajectory location prediction can be improved by combining vehicle trajectory with road network through data preprocessing. For example, original vehicle trajectory points are replaced with marked nodes or roads of road network by Fan [13] to realize location prediction. However, only the location information is considered in the proposed method, external factors such as driving intentions or preferences that have significant impacts on prediction results are ignored.

2.2. Knowledge graph

Knowledge graph is one of the most popular representation methods, which describes entities and their relationships in a structured triple. Each triple is composed of a head entity, a tail entity, and their relationship. The knowledge graph can be combined with deep learning models to learn the representation of the entities and relationships to realize prediction tasks. It can make contributions in the fields of medical treatment, national defense, transportation, and network information security [17,18].

Translating Embedding (TransE) is one of the typical models of knowledge graph [19]. A TransGraph model based on TransE is proposed by Chen et al. [20] to learn the structural features. Moreover, the knowledge graph is also widely used in machine learning. An open-source tool DGL-KE that can effectively calculate the embedded representation of the knowledge graph is proposed by Zheng et al. [21] to execute TransE algorithm. This tool can speed up the training of entities and relationships in the knowledge graph through multi-processing, multi-GPU, and distributed parallelism. A traffic mode knowledge graph framework based on historical traffic data is proposed by Ji and Jin [22] to capture the traffic status and congestion propagation mode of road segments. This framework has pioneering significance for the knowledge graph in the prediction of traffic congestion propagation. Wang et al. [23] propose a method of embedding the temporal knowledge graph through a sparse transfer matrix. Based on static and temporal dynamic knowledge graphs, the global and local embedding knowledge are captured respectively, which alleviates the problem of inconsistent parameter scalability when model learning embedding from different datasets.

More semantic knowledge can be considered in the knowledge graph to enhance the interpretability of complex models [18]. In addition, how to learn entities and relationships with low frequency better, how to integrate context into graph embedding learning better, and how to combine the graph embedding algorithm with other algorithms are future research directions.

2.3. Attention mechanism

Graph Attention Network (GAT) is widely used in the field of transportation to obtain the spatial correlation between nodes. Wang et al. [24] propose a trend GAT model to predict traffic flow. By constructing a spatial graph structure, the transmission among similar nodes is realized and the problem of spatial heterogeneity is solved. A spatio-temporal multi-head GAT model is proposed by Wang and Wang [25] to realize traffic prediction. Spatial features are captured through multi-head attention mechanism and temporal features are captured through full-volume transformation linear gated unit. The spatiotemporal correlation between traffic flow and traffic network is integrated to reduce the prediction error. Wang et al. [26] propose a dynamic GAT model to realize traffic prediction. Spatial feature is extracted by a node embedding algorithm based on the dynamic attention mechanism, and temporal feature is extracted by gated temporal CNN in this model.

Temporal attention mechanism is also widely applied in traffic research. The time correlation between research objectives can be better learned by using the time attention mechanism. A temporal attention perception dual GCN is proposed by Cai et al. [27] to realize air traffic prediction. The historical flight and time evolution pattern are characterized through temporal attention mechanism. Yan et al. [28] design a gated self-attention mechanism module to realize the interaction between the current memory state and the long-term relationship. Chen et al. [29] apply multi-head attention mechanism to mine the features of fine-grained spatiotemporal dynamics. A one-dimensional convolution LSTM model based on the attention mechanism is proposed by Wang et al. [30] to realize traffic prediction. Multi-source data are fused as various features through attention mechanism.

Spatiotemporal attention mechanism can obtain reasonable weights of the features under complex traffic environment, therefore improving the interpretability of model. GAT can be considered as a spatial attention mechanism to capture spatial features and the multi-head attention mechanism can be considered as temporal attention mechanism to capture temporal features. Therefore, it is meaningful and important to combine GAT and temporal attention mechanisms when realizing vehicle trajectory location prediction.

3. Problem Statement

In this section, some basic concepts and notations in this paper are first introduced and then the studied problem are formally defined.

Definition 1 (Raw trajectory $\boldsymbol{T_i}$). A raw vehicle trajectory is usually represented by a sequence of points continuously sampled by the Global Positioning System (GPS). Given a raw trajectory dataset $Traj = \{T_1, T_2, \dots T_n\}$, the i^{th} trajectory $T_i = \{P_{i1,}P_{i2}, \dots, P_{ik}\}$ is defined as a sequence of sampled points P_{ij}, where $P_{ij} \in \mathbb{R}^2$, $i \in (1,2, \dots, n), j \in (1,2, \dots, k)$. A sampled point P_{ij} is defined as a tuple (lng_{ij}, lat_{ij}) which represents the longitude and latitude of the GPS point at timestamp j. Due to the different sampling rate, trajectory data may have the characteristics of irregular distribution, such as being sparse in one area while dense in another area.

Definition 2 (Point of Interest $\boldsymbol{Q_u}$). Point-Of-Interests (POI) can be denoted as spatial representation of urban infrastructure such as school, hospital, restaurant and so on. It can reflect land use and urban functional characteristics, and has potential semantic information. Its distribution influences the intentions of travel. POI can be classified into different types. Given a POI dataset with h types $POI = \{Q_1, Q_2, \dots, Q_h\}$, the u^{th} type of POI $Q_u = \{I_{u1}, I_{u2}, \dots I_{um}\}$ is defined as a set of points I_{uj}, where $u \in (1, \dots, h), j \in (1, \dots, m)$. A POI $I_{uj} = (name_{uj}, address_{uj}, lng_{uj}, lat_{uj}, attribute_{uj})$ is a five-tuple, which represents the semantic information including the name, address, longitude, latitude, as well as attribute of the j^{th} POI. 11 types of POI information are considered in this paper, including shopping, entertainment, food, hotel, scenic spot, finance, government, company, hospitals, life services, and sports.

Definition 3 (Road network $\boldsymbol{G}$). Vehicle trajectory combined with POI are matched to a global map in this paper. A road network is defined as a graph $G = (V, E, A)$, where V is the set of nodes, each v_i denotes a road segment. E is the set of edges connecting nodes and each $e_{i,j} = (v_i, v_j)$ denotes the connectivity between road segment v_i and v_j. $A \in \mathbb{R}^{|V| \times |V|}$ is an adjacency matrix, and semantic information of various POI classes are considered as features. Each element $a_{i,j}$ in this matrix is a binary value, which is 1 when road segment v_i is adjacent to v_j and 0 otherwise.

Definition 4 (Normalized trajectory $\boldsymbol{NT_i}$). Raw trajectory T_i will be processed through a data conversion layer in this paper and converted into a normalized trajectory NT_i. Due to the measurement error, it is not appropriate to directly input raw trajectory to the model. Given a raw trajectory dataset $Traj = \{T_1, T_2, \dots T_n\}$, the i^{th} trajectory $T_i = \{P_{i1,}P_{i2}, \dots, P_{ik}\}$. Match each point P_{ij} to the nearest node of road network. A normalized point NP_{ij} is defined as a tuple $(vlng_{ij}, vlat_{ij})$ which represents the longitude and latitude of j^{th} node. A normalized trajectory NT_i is then defined as a sequence of road segments after projection.

Definition 5 (Normalized POI $\boldsymbol{NQ_u}$). Match each POI I_{uj} to the nodes of road network which has the shortest projection distance. A normalized POI NI_{uj} is defined as the corresponding road segment after projection and POI semantic information is assigned to the matched nodes as node features. NQ_u refers to the normalized POI of type u and NI_{uj} represents semantic information of the j^{th} normalized POI of u^{th} type.

Definition 6 (POI Knowledge graph KG^{POI}). Defining a knowledge graph $KG^{POI} = (E^{POI}, R^{POI}, T^{POI})$, where $E^{POI} = \{e_1, e_2, e_3, \dots e_m\}$ is the POI entity set, where m is the number of the POI entities. $R^{POI} = \{r_1, r_2, r_3, \dots r_n\}$ is the POI relationship set, where n is the number of the POI relationship. $T^{POI} = \{(h_{POI}, l_{POI}, t_{POI})\}$ is a triple set, which refers to head entity, relation, and tail entity respectively.

Problem (Vehicle trajectory location prediction). Given a raw trajectory dataset $Traj = \{T_1, T_2, \dots T_n\}$ and a road network G, for current point P_{ij} of the i^{th} trajectory T_i, the task aims to predict the next location P_{ij+1}, which is a tuple (lng_{ij+1}, lat_{ij+1}) consisting of the longitude and latitude of GPS point at timestamp $j+1$.

4. Methodology

Vehicle trajectory is a sequence of continuously-sampled points and its intention are closely related to POI. The semantic information of POI nearby each trajectory point may have an impact on the next location prediction. Therefore, a Local Dynamic Graph Spatiotemporal- Long Short-Term Memory model (LDGST-LSTM) is proposed in this paper to realize location prediction for vehicle trajectory and explore the impact weight of the nearby POI.

Raw vehicle trajectory and POI points are first matched to the traffic network through the data conversion layer in the proposed model. Then global POI knowledge graph is constructed to obtain the representation vectors of the POI entities in the global POI knowledge extraction layer. Based on the global knowledge graph, Local graph related to each trajectory point is generated and the graph representation vector is captured through GAT in the local dynamic graph generation module. Finally, trajectory points with the related graph representation vectors are input into LSTM with multi-head attention mechanism in the trajectory prediction layer to predict the next location.

In this section, details of the proposed model LDGST-LSTM are provided, which consists of four major components: a data conversion layer, a global POI knowledge extraction layer, a local dynamic graph generation module, and a trajectory prediction layer. The overall framework is first described in section 4.1. Then, each component in this model is specifically introduced in section 4.2 to 4.5.

4.1. Overall framework

The overall framework of LDGST-LSTM is shown in Figure 1. There are four major components in the proposed model: (1) *a data conversion layer*, (2) *a global POI knowledge extraction layer*, (3) *a local dynamic generation module*, and (4) *a trajectory prediction layer*. Additionally, the local dynamic generation module is concluded in the trajectory prediction layer. The observed vehicle trajectory $T_i = \{P_{ij-t}, P_{ij-t+1}, \dots, P_{ij}\}$ is considered as the input of the model, and the predicted next location P_{ij+1} of the trajectory at timestamp $j+1$ is considered as the output.

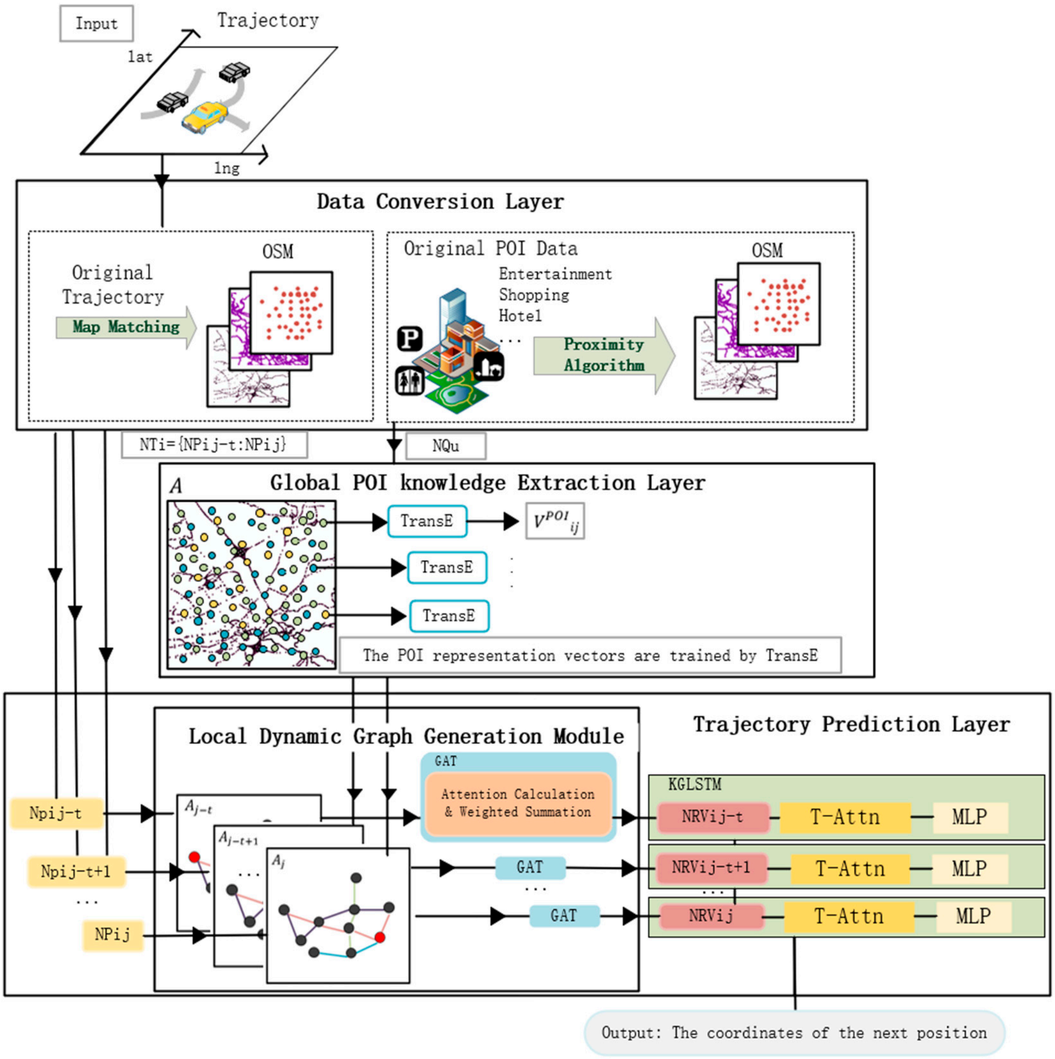


Figure 1. The overall framework of LDGST-LSTM.

As shown in Figure 1, vehicle trajectory T_i is considered as the input of the model. Firstly, normalized trajectory NT_i and normalized POI NQ_u is obtained through map matching and proximity algorithm respectively in the data conversion layer. Then the representation vector $V^{POI}{}_{ij}$ of every POI is trained through TransE algorithm based on the knowledge graph in the global POI knowledge extraction layer. It will be considered as semantic features of trajectory points. In the trajectory prediction layer, local graphs $G'_{ij-t:j}$ related to trajectory points $P_{ij-t:j}$ are first generated, and the corresponding graph representation vectors $NRV_{ij-t:j}$ are obtained through GAT in the local dynamic graph generation module. Normalized trajectory points $NP_{ij-t:j}$ and corresponding graph representation vectors $NPV_{ij-t:j}$ are then concatenated and input into LSTM with multi-head attention mechanism. The model finally outputs the predicted next location P_{ij+1} of the trajectory. The overall framework can be denoted by the following formula (1)-(4), where $P_{j-t:j} = (P_{j-t}, P_{j-t+1}, \dots, P_j)$.

- Data conversion layer: (1)

$$T_i \xrightarrow{map\ matching} NT_i, Q_u \xrightarrow{proximity\ alg.} NQ_u$$

- Global POI knowledge extraction layer: (2)

$$[KG^{POI}, NQ_u] \xrightarrow{TransE} V^{POI}{}_{ij}$$

- Local dynamic graph generation module: (3)

$$P_{ij-t:j} \xrightarrow{V^{POI}{}_{ij}} G'{}_{ij-t:j} \xrightarrow{GAT} NRV_{ij-t:j}$$

- Trajectory prediction layer: (4)

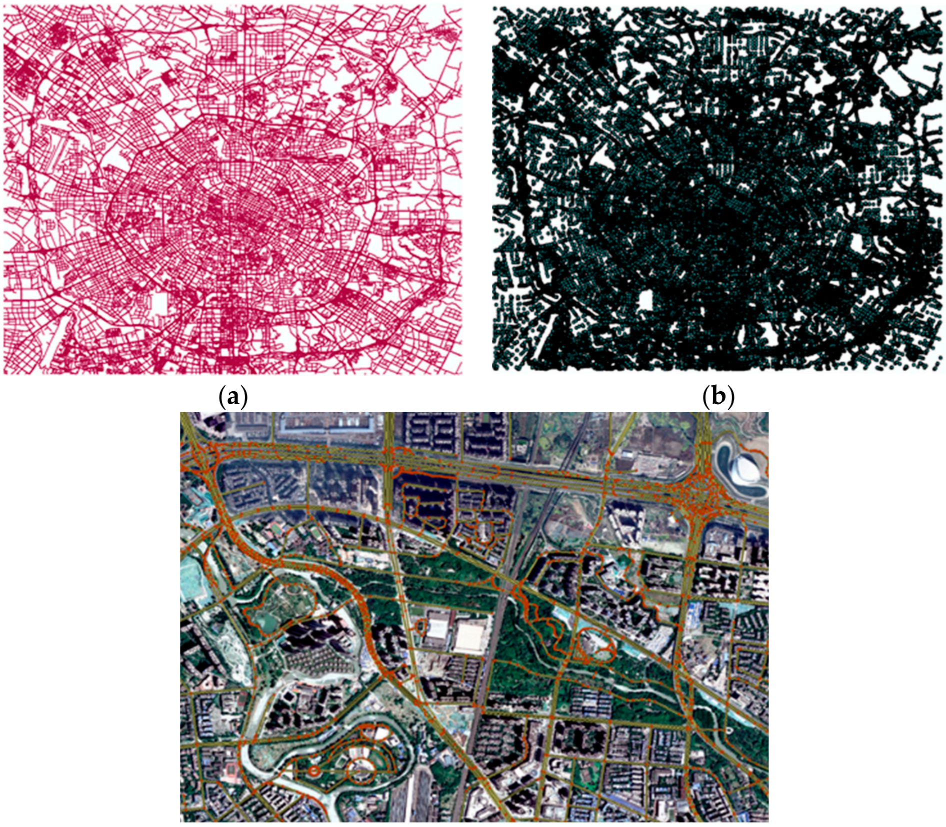
$$NP_{ij-t:j}\,; NRV_{ij-t:j} \xrightarrow{LSTM} P_{ij+1}$$

4.2. Data conversion layer

This section will introduce road network data, trajectory data conversion and POI data conversion specifically.

4.2.1. Road network

The urban road network data used in this paper is downloaded from the Open Street Map (OSM), an open-source map website. Taking Chengdu as an example, the road network is shown in Figure 2. This paper only maintains the roads and nodes within the Third Ring Road of Chengdu as the research scope, which can mainly represent the urban district.



(**a**) (**b**)

(**c**)

Figure 2. Road network of Chengdu: (**a**) Roads vector map; (**b**) Nodes vector map; (**c**) Satellite projection map.

4.2.2. Trajectory data conversion

In the data conversion lay, raw trajectory data is converted into normalized trajectory. Each point is represented by the latitude and longitude of matched road network node. Original trajectory coordinate system (GCJ-02) is converted to the road coordinate system (WGS84) in this paper. Based on the previous map-matching work of Lou et al. [31], a certain threshold is considered as the radius to search for candidate points, and the candidate points with the shortest distance are selected as the normalized trajectory points. Trajectory points that are not successfully matched to the corresponding nodes will be deleted as redundant points.

4.2.3. POI data conversion

POI data of Chengdu in 2018 is used in this paper. The proximity algorithm is used in ArcGIS to search and select the nearest nodes in the road network for each POI point based on distance. The semantic information of POI points is allocated to the corresponding nodes as the normalized POIs. As shown in Figure 3, POI points of different types are matched to the road network nodes, taking life services, food, and hospitals as examples.

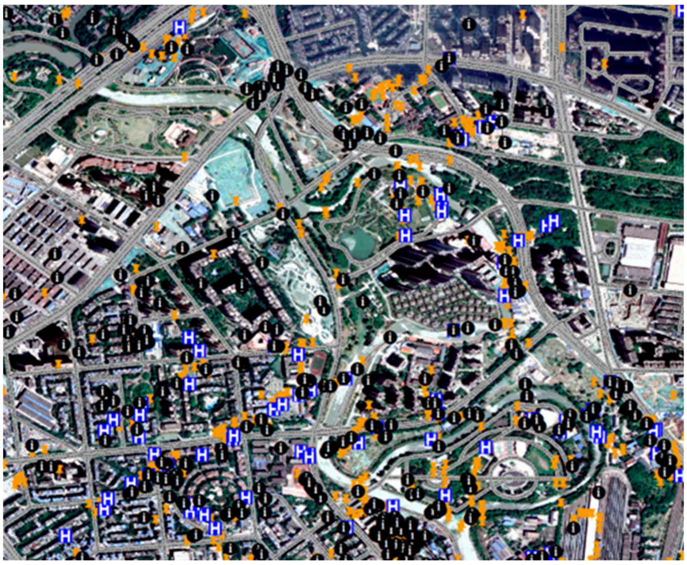


Figure 3. Partial visualization of normalized POI.

4.3. Global POI knowledge extraction layer

A global graph $G = (V, E, A)$ and a knowledge graph $KG^{POI} = (E^{POI}, R^{POI}, T^{POI})$ are constructed in the global POI knowledge extraction layer, as shown in Figure 4. Entity set, relationship set and related triples are defined to study the representation vectors of each POI entity through TransE algorithm. In the defined global graph, each node in the global graph contains the related POI knowledge through the data conversion layer.

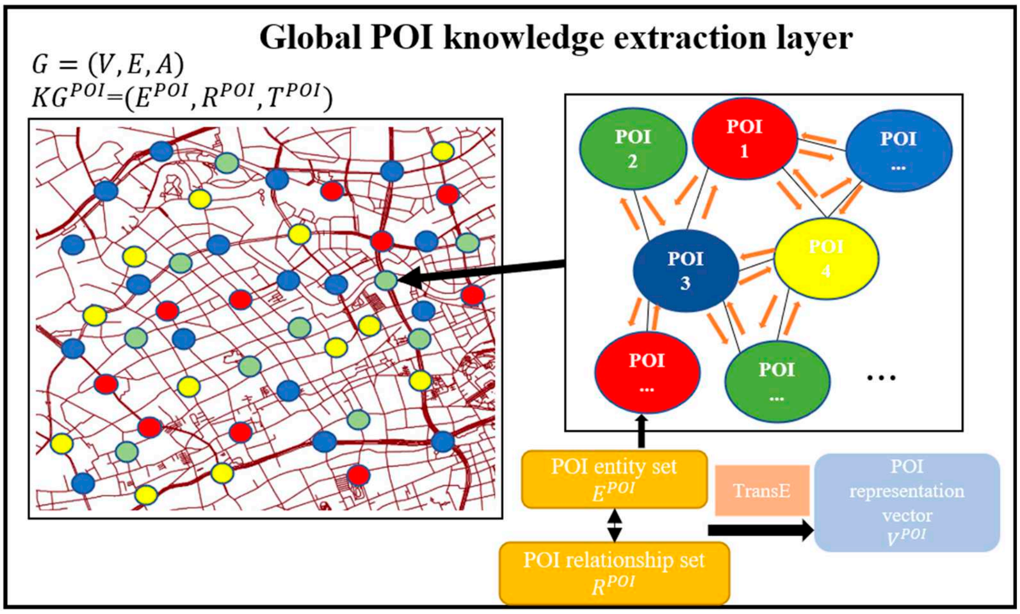

Global POI knowledge extraction layer
$G = (V, E, A)$
$KG^{POI}=(E^{POI}, R^{POI}, T^{POI})$
POI 2
POI 1
POI ...
POI 3
POI 4
POI ...
POI ...
...
POI entity set E^{POI}
TransE
POI representation vector V^{POI}
POI relationship set R^{POI}

Figure 4. Global POI knowledge extraction layer.

As shown in Figure 4, the normalized POI is considered as the entity, and the link between the normalized POI is considered as the relationship, which in this paper denotes that there is a connection between two POI nodes. Moreover, the triplet $T^{POI} = \{(h, l, t)\}$ is considered as the training set of TransE algorithm, where h represents the head entity, t represents the tail entity, and l represents the relationship; h and t belongs to the entity set E^{POI}, and l belongs to the relationship set R^{POI}. The target of the TransE algorithm is to consider the relationship as the translation from the head entity to the tail entity, or to say, to make $h + l$ equal to t as much as possible. The potential energy of a triplet is defined by the L2 norm of the difference between $h + l$ and t, as shown in formula (5), where N is the number of the triplets, and i represents the i^{th} triplet.

$$f(h_i, l_i, t_i) = \left\|h_i + l_i - t_i\right\|_2 = \sqrt{\sum_{i=1}^{N}(h_i + l_i - t_i)^2} \qquad (5)$$

Wrong triplets are identified and considered as negative samples in TransE algorithm for uniform sampling. A negative sample is generated when any one of the factors in a positive sample is replaced by the other entities or relationship randomly. The potential energy of the positive samples is reduced and the potential energy of the negative samples are increased in TransE algorithm. The objective function is defined as below.

$$L = \sum_{(h,l,t)\in\Delta} \sum_{(h',l',t')\in\Delta'} \left[\left(f_r(h,t) + \gamma - f_{r'}(h',t')\right)\right]_+ \qquad (6)$$

Where, Δ is the set of positive samples, and Δ' is the set of negative samples. γ is a constant, usually is set as 1, that represents the distance between positive and negative samples. (h, l, t) and (h', l', t') are the triplets of positive and negative samples respectively. $f(\cdot)$ is the potential energy function and $[\cdot]_+$ is $max(0,\cdot)$.

The distributed representation vector of the current head and tail entities are considered as the representation vectors $V^{POI}{}_{ij} \in \mathbb{R}^d$. Pseudocode of the global POI knowledge extraction layer is shown in Table 1.

Table 1. Pseudocode of the global POI knowledge extraction layer.

```
Algorithm 1: Global POI knowledge extraction layer
Input entity, relation, and training sets $E^{POI}, R^{POI}, T^{POI} = \{(h, l, t)\}$, embedding dim $k$
normalize $l \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $l \in R^{POI}$
          $l/\|l\|$ for each $l \in R^{POI}$
          $e \leftarrow$ uniform $(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $e \in E^{POI}$
loop
  $e/\|e\|$ for each $e \in E^{POI}$
  $S^{POI}_{batch} \leftarrow$ sample $T^{POI}, b)$ //sample by size $b$
  $T^{POI}_{batch} \leftarrow \emptyset$ //initialize the triplets
  for $(h, l, t) \in S^{POI}_{batch}$ do
    $(h^{'}, l^{'}, t') \leftarrow$ sample $S'^{POI}_{(h,l,t)}$ //extract negative samples
    $T^{POI}_{batch} \leftarrow T^{POI}_{batch} \cup \{\left((h, l, t), (h^{'}, l^{'}, t^{'})\right)\}$ //extract positive and negative samples randomly
  end for
Update embeddings $\sum_{(h,l,t)\in\Delta} \sum_{(h^{'},l^{'},t^{'})\in\Delta^{'}} \left[\left(f_r(h,t) + \gamma - f_{r^{'}}(h^{'},t^{'})\right)\right]_+$ //L
end loop
Output representation vector $V'^{POI}_{ij}$ of the current entity
```

4.4. Local dynamic graph generation module

Local graphs are generated for each trajectory point in the local dynamic graph generation module. Graph Attention network (GAT) is used as a spatial attention mechanism to allocate weight and update the parameters of every trajectory point and its neighbors. Corresponding graph representation vector NRV_{ij} can be obtained through GAT.

As shown in Figure 5, every node in the global graph is embedded with the POI representation vector ${V^{POI}}_{ij}$ based on the POI knowledge graph in section 4.3. The feature matrix $F \in \mathbb{R}^{N\times k}$ is constructed by all the embeddings of nodes, $F_{ij} = {V^{POI}}_{ij}$, where N is the number of the graph nodes, k is the embedding dimension of the feature, F_{ij} represents the feature matrix of the j^{th} trajectory point of the i^{th} trajectory. The related local graphs ${G'}_{ij-t:j} = (V_{ij-t:j}, E_{ij-t:j}, {A'}_{ij-t:j})$ are generated for each normalized trajectory point $NP_{ij-t:j}$ in the normalized trajectory NT_i, where V_{ij} is the set of the current point NP_{ij} and its neighbor nodes, E_{ij} is the set of edges among the current point and its neighbors, ${A'}_{ij}$ is the local adjacency matrix of the current point, and it concludes the features of both the current point and its neighbor nodes.

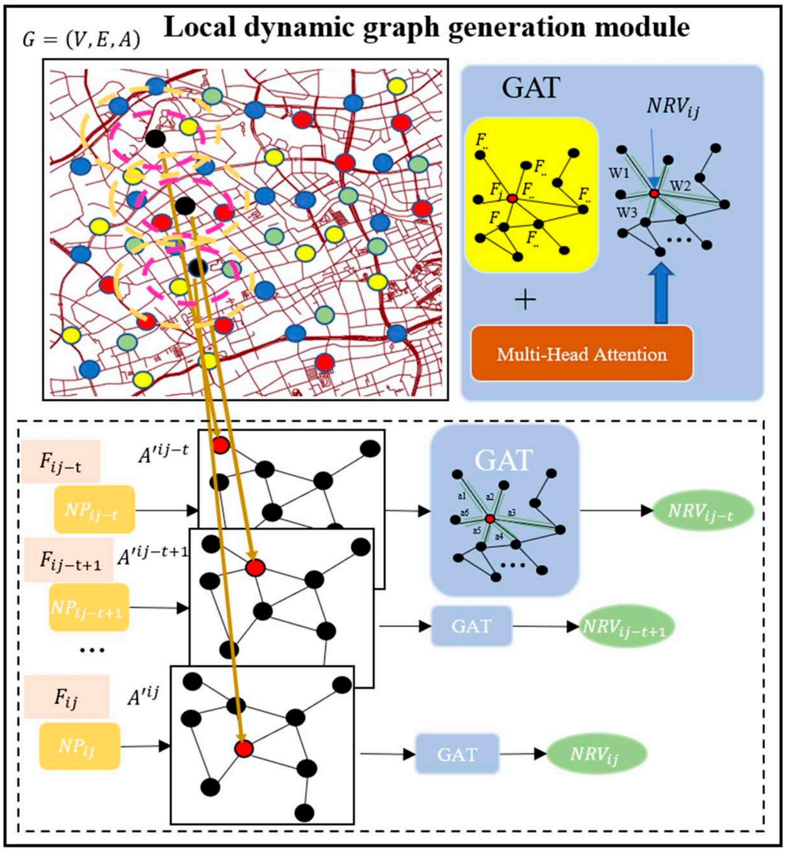


Figure 5. Local graph generation module.

GAT is used to calculate the attention weight between the trajectory points and its neighbor nodes and fuse the features for the local graphs. The adjacency matrix is used to check whether there is connection among nodes, and the resource is allocated by calculating the weights of the neighbor nodes in GAT. It can be considered as a spatial attention mechanism to enhance the interpretability of the proposed model. The definition of attention mechanism is shown as formula (7).

$$Attention(query, source) = \sum similarity(query, key_m) * value_m \qquad (7)$$

where $source$ is the original information, and it is formed by the $key - value$ pairs. $Attention(query, source)$ represents the information extraction through weight allocation from the $source$ under the condition of $query$. The aim of the GAT is to learn the relevance among target nodes and their neighbor nodes through the parameter matrix. In this paper, $query$ is set as the feature vector F_{ij} of the current point NP_{ij}, $source$ is set as the feature vectors of all the neighbor nodes of NP_{ij}, key_m and $value_m$ are respectively the m^{th} neighbor node and its feature vector. The relevance coefficients among every trajectory point and its neighbor nodes are calculated in GAT.

$$e_{jm} = LeakyReLU\left(w^T[WF_{ij}||WF_{jm}]\right) \tag{8}$$

Where $||$ represents concatenation, F_{ij} is the feature vector of the j^{th} point of the i^{th} trajectory, and F_{jm} is the feature vector of the m^{th} neighbor node of the j^{th} point. W is a learnable weight parameter. w is a learnable weight of the linear layer and $LeakyReLU$ is the activation function. Moreover, all the coefficients are normalized by the function $softmax$ in GAT to obtain the attention weight. a_{jm} is the attention weight between node j and its m^{th} neighbor node, where $\mathcal{N}_{ij}$ denotes the neighborhood of node j in the i^{th} trajectory.

$$a_{jm} = softmax\left(e_{jm}\right) = \frac{\exp\left(e_{jm}\right)}{\sum_{n\in\mathcal{N}_{ij}}\exp(ejn)} \tag{9}$$

The aggregated and updated feature vector is then calculated, as shown in formula (10). Multi-head attention mechanism is used to enhance the feature fusion of the neighbor nodes.

$$NPV_{ij} = ||_{h=1}^{H_1}\sigma(\sum\nolimits_{m\in\mathcal{N}_{ij}} a_{jm}{}^{(h)}W^{(h)}F_{jm}) \tag{10}$$

where H_1 is the number of the heads, σ is the activation function and NPV_{ij} is the graph representation vector. Pseudocode of the local dynamic graph generation module is shown in Table 2.

Table 2. Pseudocode of the local dynamic graph generation module.

```
Algorithm 2: Local dynamic graph generation module
Input normalized trajectory points $NP_{ij-t:j}$, POI feature vectors $F_{ij} = V_{ij}^{POI}$
    $G'_{ij-t:j} \leftarrow (V_{ij-t:j}, E_{ij-t:j}, A'_{ij-t:j})$ for each trajectory point    // generate local graphs
    Target node vector $F_{ij} \leftarrow V_{ij}$ and neighbor node vector $F_{jm} \leftarrow V_{jm}$ // according to local graphs
    $e_{jm} \leftarrow LeakyReLU\left(w^T[WF_{ij}||WF_{jm}]\right)$
    $a_{jm} \leftarrow softmax\left(e_{jm}\right)$
    weighted sum:
    for $h \leq H_1$ do
        $NPV_{ij} \leftarrow NPV_{ij}||\sigma(\sum\nolimits_{m\in\mathcal{N}_{ij}} a_{jm}{}^{(h)}W^{(h)}F_{jm})$
    end for
Output graph representation vector $NPV_{ij}$ for each trajectory point
```

4.5. Trajectory prediction layer

In the trajectory prediction layer, trajectory points and corresponding graph representation vectors are input, and the coordinate of the next location are obtained after going through LSTM with a multi-head attention mechanism, as shown in Figure 6.

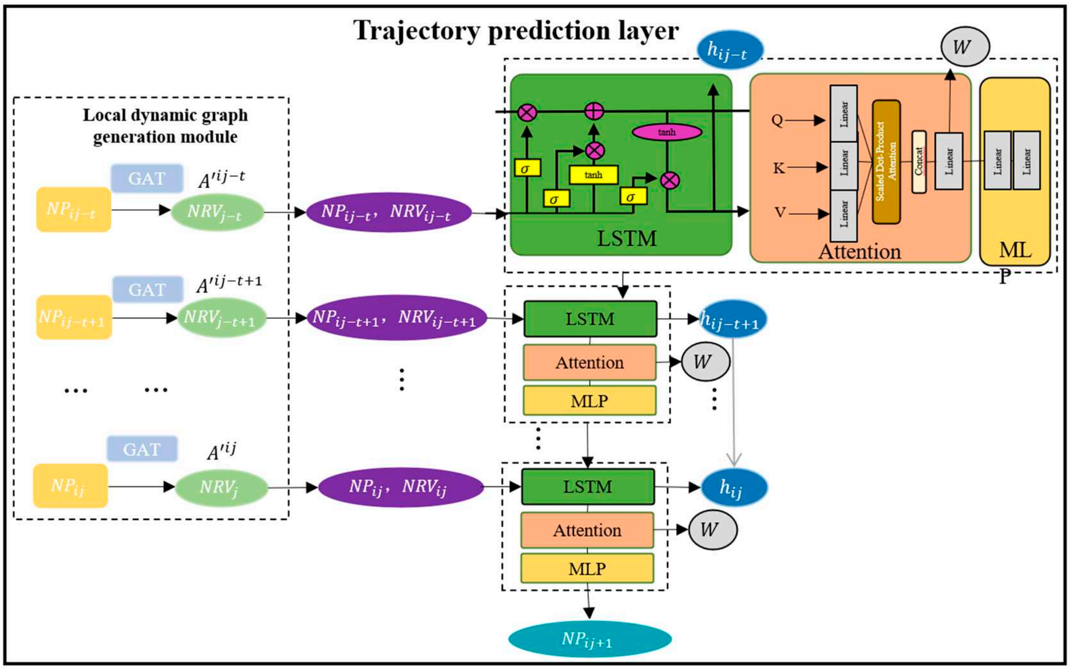


Figure 6. The research framework of the trajectory prediction layer.

As shown in Figure 6, the trajectory points $NP_{ij-t:j}$ and the corresponding graph representation vectors $NPV_{ij-t:j}$ are concatenated and input into LSTM, as shown in formula (11)-(17).

$$x_j = \{NP_{ij}, NPV_{ij}\} \tag{11}$$

$$f_j = \sigma(W_f \cdot [h_{ij-1}, x_j] + b_f) \tag{12}$$

$$i_j = \sigma(W_i \cdot [h_{ij-1}, x_j] + b_i) \tag{13}$$

$$\tilde{C}_j = \tanh(W_C \cdot [h_{ij-1}, x_j] + b_C) \tag{14}$$

$$o_j = \sigma(W_o \cdot [h_{ij-1}, x_j] + b_o) \tag{15}$$

$$C_j = f_j * C_{j-1} + i_j * \tilde{C}_j \tag{16}$$

$$h_{ij} = o_j * \tanh(C_j) \tag{17}$$

where W_f, W_i, W_C and W_o are respectively the weight of the forget gate, the input gate, the cell state, and the output gate. b_f, b_i, b_C and b_o are the corresponding bias. The trajectory points and corresponding graph representation vectors are concatenated as the input x_j. x_j goes through the forget, input and output gates, and generate the cell state C_j. Necessary information is processed by the input gate and the updated information is activated by the function $tanh$ to obtain the $\tilde{C}_j$. The current cell state C_j is shown in formula (16), where f_j and i_j are respectively the output of the forget gate and the input gate. C_{j-1} is the cell state of the previous vector and $\tilde{C}_j$ is the activated state updated by the input gate. The hidden state h_{ij} of the current vector is obtained by multiplying the activated cell state C_j and the output of the output gate o_j.

A multi-head attention mechanism is used based on LSTM to allocate weight and enhance the interpretability of the proposed model.

$$A_h(Q_h, K_h, V_h) = softmax\left(\frac{Q_h \cdot {K_h}^T}{\sqrt{d}}\right)V_h \tag{18}$$

$$MultiAtt(Q,K,V) = \left(A_1 || \dots || A_{H_2}\right) W^O \tag{19}$$

Where $Q_h = W_i^Q\, h_{ij}$, $K_h = W_i^K, h_{ij-t:j}$, $V_h = W_i^V h_{ij-t:j}$. h_{ij} is the hidden state of the current vector and is considered as the query, $h_{ij-t:j}$ denotes the hidden states of the vectors $x_{j-t:j}$ and is considered as the key and value, W_i^Q, W_i^K, W_i^V, W^O are learnable weight parameters, d is the dimension, H_2 is the number of the heads.

The predicted result is obtained by adding a multilayer perceptron composed of two full-connected layers.

$$\hat{Y}_j = W_F^2 ReLU\left(W_F^1 h_j{}' + b_F^1\right) + b_F^2 \tag{20}$$

where $\hat{Y}_j$ is the predicted result. W_F^1, $W_F^2{}'$, b_F^1, and b_F^2 are respectively the weights and biases of the two full-connected layers. $h_j{}'$ is the output calculated by the multi-head attention mechanism. Mean Square Error (MSE) is considered as the loss function to calculate the difference between predicted results and truth, where N is the number of trajectories and K is the length of the trajectory.

$$L = \sum_{i=1}^{N} \sum_{j=1}^{K} \left(\hat{Y}_j - NP_{ij}\right)^2 \tag{21}$$

The pseudocode of the trajectory prediction layer is shown in Table 3.

Table 3. Pseudocode of the trajectory prediction layer.

```
Algorithm 3: Trajectory prediction layer
Input normalized trajectory points $NP_{ij-t:j}$, graph representation vectors $NPV_{ij-t:j}$
LSTM Module:
loop
    $x_j \leftarrow \{NP_{ij}, NPV_{ij}\}$
    $f_j, i_j, \widetilde{C}_j, o_j \leftarrow$ calculated by forget gate, input gate and output gate
    $C_j, h_{ij} \leftarrow$ cell state and hidden state are calculated by $f_j, i_j, \widetilde{C}_j, o_j$
    Temporal attention mechanism:
    $Q_h \leftarrow W_i^Q\, h_{ij},\ K_h \leftarrow W_i^K, h_{ij-t:j},\ V_h \leftarrow W_i^V h_{ij-t:j}$
    for $h \leq H_2$ do
        $A_h \leftarrow softmax\left(\frac{Q_h \cdot K_h{}^T}{\sqrt{d}}\right) V_h$
        $h_j{}' \leftarrow h_j^{'} || A_h W^O$
    end for
    MLP:
    $\hat{Y}_j \leftarrow W_F^2 ReLU\left(W_F^1 h_j{}' + b_F^1\right) + b_F^2$
end loop
Output coordinate $\hat{Y}_j$ of the next location
```

4. Experiments

This section will demonstrate the details of experiments, including datasets, experiment settings and result analysis. Accuracy and robustness experiment are conducted to evaluate the performance of the proposed model compared with benchmarks. Additionally, ablation experiment is set to explore the effectiveness of the proposed model by filtering the spatial and temporal attention mechanisms. Moreover, the POI weights that influence the prediction results are visualized on the map to demonstrate the significance of urban functional regions to vehicle trajectories.

4.1. Datasets

Trajectory dataset: Chengdu taxi trajectory data in October 2018 from Didi Gaiya is used in this paper. It concludes attributes of driver ID, order ID, timestamp, longitudes, and latitudes. Although it collects trajectory data for one month, it has over 380k trajectories, and 270k are generated on holidays while 110k are trajectories of working days. Concerning the data distribution, the dataset covers mainly urban area of Chengdu, from 30.65283 to 30.72649°N, and 104.04210 to 104.12907° E. The data sampling frequency of each trajectory is 4s, which has relatively high data density on the city road network.

Road Network dataset: POI and trajectory information are combined to construct a global traffic graph. The road network data of Chengdu is downloaded from OSM. The node vector data is composed of road node ID and its corresponding longitude and latitude, and each road vector data is composed of road ID and sequence of the node coordinates.

POI dataset: POI data of 11 categories of Chengdu in AutoNavi map is obtained from the Chinese Software Developer Network (CSDN). The original data format is POI name, longitude and latitude coordinates, address, district, and POI class.

4.2. Experimental settings

Experimental settings in this paper are shown as follows: CPU used in the experiment is AMD RYZEN 75800h, and GPU is Nvidia Geforce RTX 3060. Windows 10 is used as the operating system, Python is used as the development language, and PyTorch is used as the deep learning development framework. Moreover, the parameter settings in this paper are shown as follows: The input data are divided into the training, validation, and test set according to the proportion of 7:2:1. Batch size is set as 16, the initial learning rate is set as 0.0001, and Adam is used as the optimizer in the training process.

4.2.1. Benchmark models

Five benchmark models for comparative experiments are mentioned, including LSTM [32], GRU [33], BiLSTM [34], AttnLSTM [35], AttnBiLSTM [36].

LSTM: It is usually applied in time series prediction. Compared with the traditional RNN models, LSTM makes up for the defect that traditional RNN models only consider the recent state and cannot deal with long-term memory. It can determine which states are retained or forgotten through the internal cell state and forget gate. Moreover, LSTM can avoid some gradient vanishing problems of traditional RNN models.

GRU: Compared with LSTM, GRU only has two gates and three full connection layers, which reduces the amount of calculation to a certain extent and reduces the risk of overfitting. GRU has update and the reset gate, which can determine the output of information, and delete irrelevant information while saving historical information.

BiLSTM: Based on LSTM, the input in BiLSTM will have both forward and backward propagation, therefore each timestamp in the input sequence can save both the future and past historical information at the same time.

AttnLSTM: Attention mechanism can assign greater weight to more important tasks when the amount of computation is limited. It is a resource allocation scheme to solve information overload. Combining LSTM with attention mechanism, the model performance can be improved by increasing the weight of the most important features, or filtering useless feature information.

AttnBiLSTM: BiLSTM combined with the attention mechanism can fuse information by increasing the weight of important features, and bi-directional propagation. It can improve the calculation efficiency and accuracy of the model with abundant semantic information.

4.2.2. Evaluation Metrics

Evaluation metrics used in this paper are Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Haversine function (HSIN), and Accuracy. The identification and formulas of these metrics are shown as follows. Y_i and $\hat{Y}_i$ is true and predicted value respectively, and N is the predicted number.

doi:10.20944/preprints202311.1498.v1

MAE: It represents the average of the absolute error between the real and the predicted value, which can be calculated as follows.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \hat{Y}_i\right| \tag{22}$$

MSE: It represents the expected value of the square of the difference between the predicted and the real value, which can be calculated as follows.

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\mathrm{Y_i} - \hat{\mathrm{Y}}_\mathrm{i}\right)^2 \tag{23}$$

RMSE: It represents the sample standard deviation of the difference between the real and the predicted value. RMSE can represent the dispersion degree of the sample. For nonlinear model fitting, the smaller the RMSE is, the better the model fits. Compared with MAE, RMSE has more penalties for high differences. It can be calculated as follows, where $\mathrm{SSE} = \sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{Y_i} - \hat{\mathrm{Y}}_\mathrm{i})^2$ represents Sum of the Squares of Error.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{SSE}{N}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2} \tag{24}$$

HSIN: It represents the distance difference between the predicted points and the ground truth. The smaller the distance error, the higher the accuracy of the model prediction. It can be calculated as follows, where $Y_i = (lat_i, lon_i)$ is the truth, and $\hat{Y}_i = \left(\widehat{lat}_i, \widehat{lon}_i\right)$ is the predicted longitude and latitude.

$$HSIN = 2r\ argsin\left(\sqrt{sin^2\left(\frac{lat_i - \widehat{lat}_i}{2}\right) + \cos(lat_i)\cos\left(\widehat{lat}_i\right) sin^2\left(\frac{lon_i - \widehat{lon}_i}{2}\right)}\right) \tag{25}$$

Accuracy: It indicates the proportion of the accurate predicted output in the total output. The higher the accuracy rate, the better the model training effect. It can be calculated as follows, TP is the correctly predicted samples, NP is the incorrectly predicted samples, and $count(\cdot)$ represents the number of the objects.

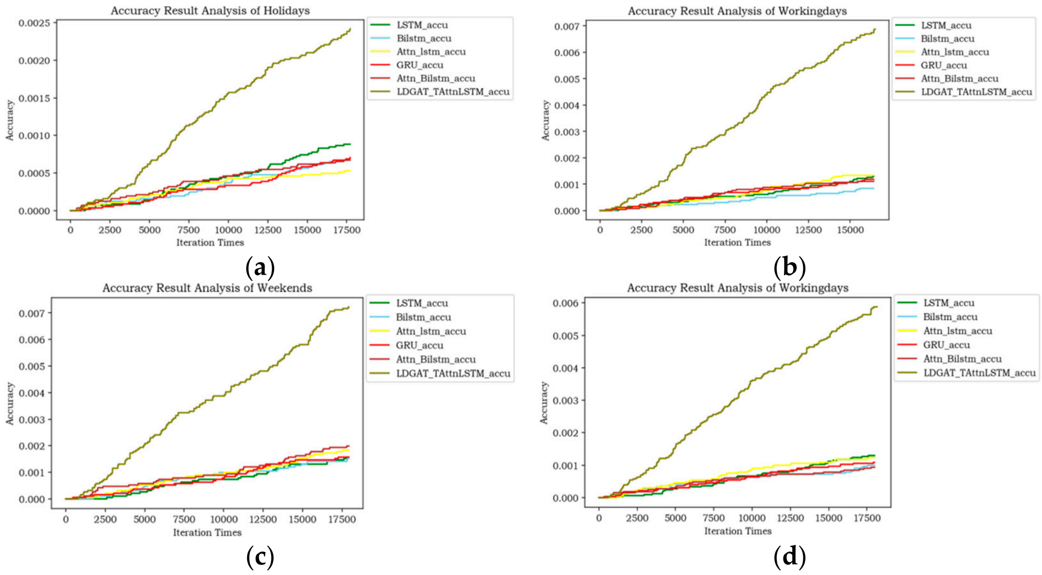
$$Accuracy = \frac{count(TP)}{count(TP) + count(NP)} \tag{26}$$

4.3. Result analysis

The accuracy experiment and the robustness experiment are discussed in this paper. The LDGST-LSTM model is compared with the benchmark models to explore the performance on accuracy and robustness. The ablation experiment is discussed to verify the spatial and temporal attention mechanisms of the proposed model to enhance the interpretability. Moreover, the POI weights calculated by the temporal attention mechanism are visualized on the map. The POI is considered as the feature of the vehicle trajectory and the influence of the POI on the vehicle position prediction is revealed.

4.3.1. Accuracy experiment

All models are trained and the accuracy comparison results on training set are shown as Figure 7. It can be seen that in both holidays and working days, the accuracy of LDGST-LSTM is significantly higher than the benchmarks.



(a) (b)
(c) (d)

Figure 7. Accuracy comparison of different models on training set: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

As shown in Figure 7a,b, the accuracy of the proposed model in both holidays and working days are nearly 4.5 times higher than the benchmarks in the dataset of October 2018. As shown in Figure 7c,d, the accuracy of the proposed model in both weekends and working days are nearly 3.5-4.5 times higher than the benchmarks in the dataset of August 2014. Therefore, the accuracy of location prediction can be effectively improved by considering POI knowledge and combing GAT and temporal attention mechanism.

The accuracy comparison results of different models are shown in Table 4.

Table 4. Accuracy comparison of different models: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

(**a**)

Model	Acuraccy (%)
LSTM	0.09
GRU	0.07
BiLSTM	0.06
Attn-LSTM	0.05
Attn-BiLSTM	0.06
LDGST-LSTM	**0.30**

(**b**)

Model	Acuraccy (%)
LSTM	0.10
GRU	0.08
BiLSTM	0.07
Attn-LSTM	0.14
Attn-BiLSTM	0.09
LDGST-LSTM	**0.75**

(**c**)

Model	Acuraccy (%)
LSTM	0.12
GRU	0.12
BiLSTM	0.11
Attn-LSTM	0.15
Attn-BiLSTM	0.16
LDGST-LSTM	**0.80**

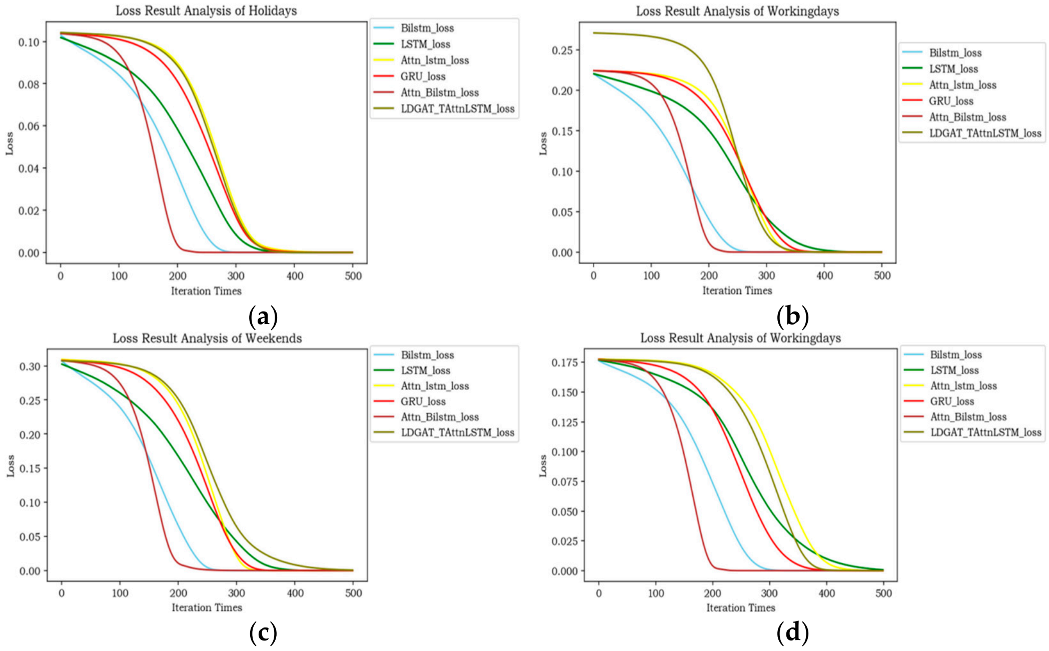
(**d**)

Model	Acuraccy (%)
LSTM	0.10
GRU	0.07
BiLSTM	0.05
Attn-LSTM	0.09
Attn-BiLSTM	0.05
LDGST-LSTM	**0.57**

In the October 2018 dataset, the accuracy of LDGST-LSTM is 21% higher compared to LSTM during holidays and 61% higher compared to Attn-LSTM during weekdays. In the August 2014 dataset, LDGST-LSTM improved by 64% compared to Attn-BiLSTM on weekends. Therefore, compared to the benchmarks, the model proposed in this paper can improve performance by selecting appropriate POI categories on weekdays and holidays, respectively. More importantly, integrating POI knowledge and combining spatial and temporal attention mechanisms can greatly improve the accuracy of vehicle trajectory location prediction.

4.3.2. Robustness experiment

The results of robustness experiment are discussed in this section. The convergence speed and the metrics after convergence of the proposed model and benchmarks are compared to analyze the robustness. The results of convergence speed are shown in Figure 8.



(a) (b)

(c) (d)

Figure 8. Convergence speed comparison of different models: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

As shown in Figure 8a,b, the loss of LDGST-LSTM drops slowly compared with the benchmark models before the 200th iterations on the dataset of October 2018. The convergence speed of the proposed model becomes faster from the 250th to the 300th iterations, and it converges at around the 350th iteration. As shown in Figure 8c,d, the loss of LDGST-LSTM drops slowly compared with the benchmark models before the 200th iterations on the dataset of August 2014. And the convergence speed of the proposed model becomes faster from the 250th to the 350th iterations and it converges at around the 450th iteration.

The performance on MAE, MSE, RMSE and HISN of different models are shown in Table 5. As shown in Table 5a,b, the performance of the proposed model on evaluation metrics are the best compared with benchmarks when using the dataset of Octorber 2018. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 7.73%, 28.57%, 11.11% and 15.72% lower than LSTM that is the most robust among the benchmars in the holidays. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 6.19%, 33.33%, 27.27% and 41.79% lower than GRU that is the most robust among the benchmarks in the working days. As shown in Table 5c,d, the performance of the proposed model on evaluation metrics are still the best when using the dataset of Chengdu August 2014. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 6.09%, 33.33%, 15.64% and 41.66% lower than GRU in the weekend. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 4.88%, 28.57%, 18.55% and 15.99% lower than LSTM in the working days. Therefore, it can be seen that the robustness of the proposed model are the best compared with all the benchmark models.

Table 5. Performance comparison of different models: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

(**a**)

Model	**MAE**	**MSE**	**RMSE**	**HISN**
LSTM	0.0207	0.0007	0.0234	2.4433
GRU	0.0218	0.0008	0.0243	2.8943
BiLSTM	0.0302	0.0012	0.0329	4.7287
Attn-LSTM	0.0234	0.0009	0.0258	3.7313
Attn-BiLSTM	0.0492	0.0035	0.0567	8.4362
LDGST-LSTM	**0.0191**	**0.0005**	**0.0208**	**2.0591**

(**b**)

Model	**MAE**	**MSE**	**RMSE**	**HISN**
LSTM	0.0212	0.0009	0.0295	3.5067
GRU	0.0210	0.0009	0.0275	3.4957
BiLSTM	0.0305	0.0010	0.0305	3.1503
Attn-LSTM	0.0220	0.0008	0.0250	3.4728
Attn-BiLSTM	0.0345	0.0012	0.0355	3.7504
LDGST-LSTM	**0.0197**	**0.000**	**0.0200**	**2.0348**

(**c**)

Model	**MAE**	**MSE**	**RMSE**	**HISN**
LSTM	0.0202	0.0007	0.0305	2.9507
GRU	0.0197	0.0006	0.0275	2.6054
BiLSTM	0.0255	0.0008	0.0335	3.0504
Attn-LSTM	0.0199	0.0006	0.0290	2.6595
Attn-BiLSTM	0.0301	0.0012	0.0403	3.7955
LDGST-LSTM	**0.0185**	**0.0004**	**0.0232**	**2.0634**

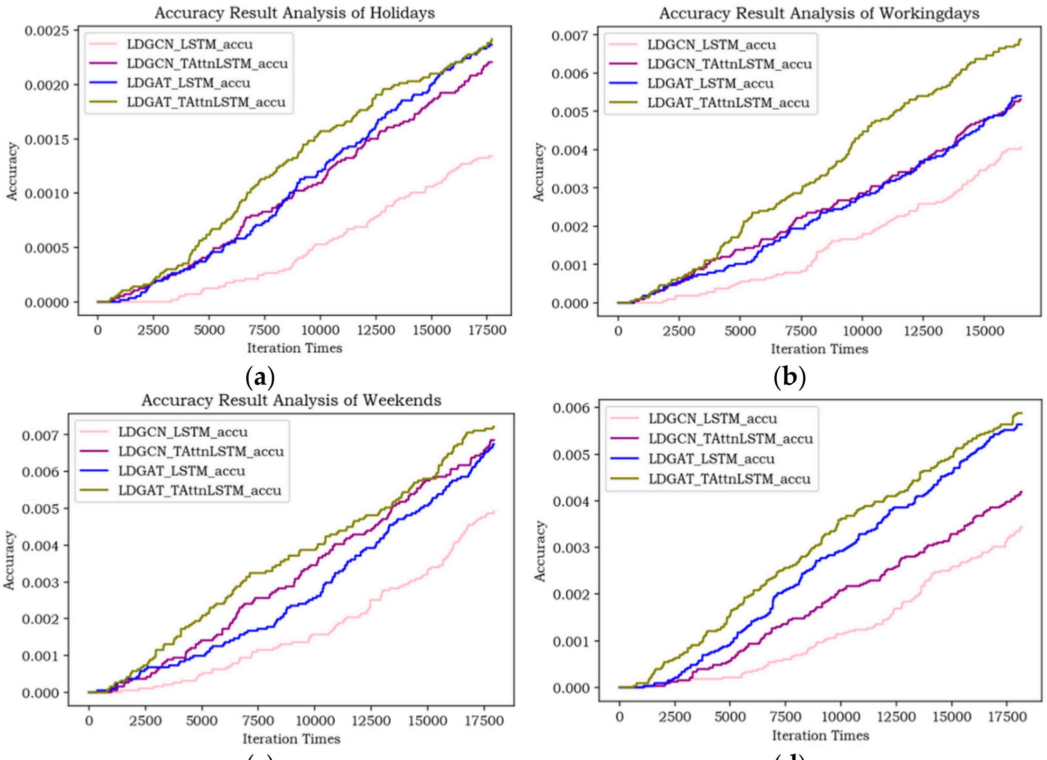
(**d**)

Model	**MAE**	**MSE**	**RMSE**	**HISN**
LSTM	0.0205	0.0007	0.0275	2.5047
GRU	0.0227	0.0009	0.0294	2.8643
BiLSTM	0.0269	0.0011	0.0327	3.0457
Attn-LSTM	0.0235	0.0009	0.0310	3.0137
Attn-BiLSTM	0.0312	0.0014	0.0343	3.3189
LDGST-LSTM	**0.0195**	**0.0005**	**0.0224**	**2.1042**

4.3.3. Ablation experiment

Ablation experiment is discussed to analyze the effect of major components in the proposed model by filtering GAT and temporal attention mechanism. In addition to the proposed model, three ablation models are analyzed, including (1) Local Dynamic Graph Convolutional Network- Long Short-Term Memory (LDGCN-LSTM), (2) Local Dynamic Graph Convolutional Network- Temporal Attention Long Short-Term Memory (LDGCN-TAttnLSTM), and (3) Local Dynamic Graph Attention- Long Short-Term Memory (LDGAT-LSTM).

As shown in Figure 9, the accuracy of LDGST-LSTM is the highest among the other three ablation models in both two datasets. The accuracy of LDGST-LSTM is almost the same as the accuracy of LDGAT-LSTM. It may be the reason that in holidays, the intention of the taxis is more focuesd on some functional regions, and the importance of spatial overweighs the temporal features.



(**a**) (**b**)

(**c**) (**d**)

Figure 9. Accuracy comparison of different ablation modules: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

Table 6. Ablation results of the proposed model: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

(a)

Model	MAE	MSE	RMSE	HISN	Accuracy (%)
LDGST-LSTM	**0.0191**	**0.0005**	**0.0208**	**2.0591**	**0.25**
LDGAT-LSTM	0.0212	0.0007	0.0239	2.4357	0.24
LDCN-TAttnLSTM	0.0225	0.0008	0.0244	2.6329	0.20
LDGST-LSTM	0.0195	0.0005	0.0224	2.1042	0.13

(b)

Model	MAE	MSE	RMSE	HISN	Accuracy (%)
LDGST-LSTM	0.0197	0.0006	0.0200	2.0348	0.70
LDGAT-LSTM	0.0205	0.0007	0.0257	2.5473	0.50
LDCN-TAttnLSTM	0.0220	0.0008	0.0295	2.6904	0.45
LDGST-LSTM	0.0237	0.0010	0.0305	2.9754	0.40

(c)

Model	MAE	MSE	RMSE	HISN	Accuracy (%)
LDGST-LSTM	0.0185	0.0004	0.0232	2.0634	0.70
LDGAT-LSTM	0.0253	0.0007	0.0301	2.8751	0.64
LDCN-TAttnLSTM	0.0248	0.0006	0.0284	2.5323	0.65
LDGST-LSTM	0.0297	0.0009	0.0328	2.9107	0.50

(**d**)

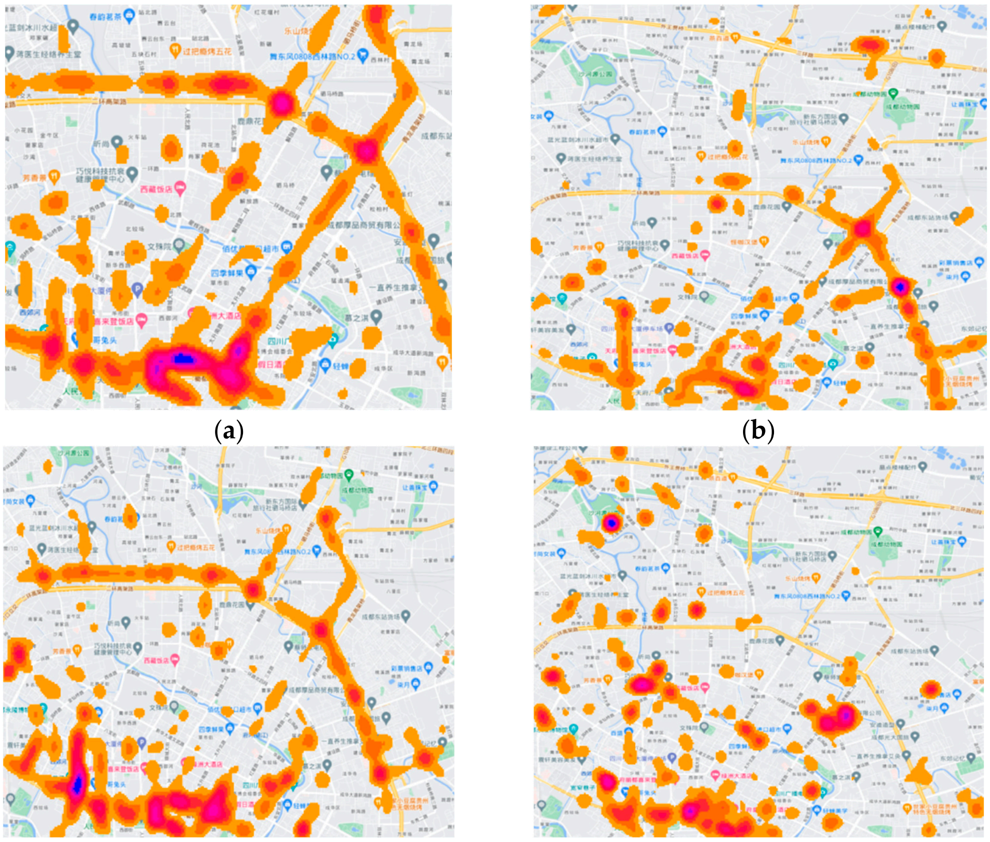
Model	MAE	MSE	RMSE	HISN	Accuracy (%)
LDGST-LSTM	0.0195	0.0005	0.0224	2.1042	0.57
LDGAT-LSTM	0.0218	0.0007	0.0254	2.3490	0.55
LDCN-TAttnLSTM	0.0261	0.0009	0.0278	2.5983	0.40
LDGST-LSTM	0.0284	0.0009	0.0290	2.9841	0.30

The ablation results is shown in Figure 6. As shown in Figure 6a,b, the evaluation metrics of LDGST-LSTM are the best among the ablation models when using dataset in 2018. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 9.91%, 28.57%, 12.97% and 15.46% lower, and the accuracy is 4.17% higher than LDGAT-LSTM that performances the best among ablation models in the holidays. The MAE, MSE, RMSE and HSIN of the proposed model are respectively 3.9%, 14.29%, 22.18% and 20.12% lower, and the accuracy is 40% higher than LDGAT-LSTM in the working days. As shown in Figure 6c,d, the performances of LDGST-LSTM are also the best in 2014 dataset. The MAE, MSE, RMSE and HSIN of LDGST-LSTM are respectively 25.40%, 33.33%, 18.31% and 18.52% lower, and the accuracy is 7.69% higher than LDGAT-LSTM that performances the best among ablation models in the weekends. The MAE, MSE, RMSE and HSIN of LDGST-LSTM are respectively 10.55%, 28.57%, 11.81% and 10.42% lower, and the accuracy is 3.64% higher than LDGAT-LSTM in the working days. In conclusion, the proposed model performances the best compared with other ablation models. Therefore, the combination of GAT and temporal attention mechanism can enhance the interpretability and also improve the accuracy and robustness of model.

4.3.4. POI weights visualization

The predicted coordinates of the next location and corresponding weight are calculated by the proposed model. The visualization of POI weights is realized through the nuclear density analysis in the ArcGIS.

The visualization of the POI weight has the positive significance to the vehicle trajectory planning, traffic optimazation and vehicle location prediction. As shown in Figure 10, there are some regions of POI information having effects on the vehicle location prediction. For example , the western south regions in Figure 10a,c, and the right and the top side in Figure 10b,d. It can be seen that in holidays and weekends, POI in the western south regions can be considered as the important information that influences the vehicle trajectories. It may denote that these regions are close to the center of the city, and has high traffic flow in holidays and weekends. Therefore, it is important to plan the driving path in these regions in holidays and weekends. Moreover, POI regions that influence the location prediction is more dispersed in working days, and the trajectory decision can be more flexible.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Visualization of the POI weights: (**a**) holidays in Oct. 2018; (**b**) working days in Oct. 2018; (**c**) weekends in Aug. 2014; (**d**) working days in Aug. 2014.

5. Conclusions

A Local Dynamic Graph Spatiotemporal- Long Short-Term Memory (LDGST-LSTM) model is proposed in this paper to predict the next location of vehicle trajectory. Data conversion layer, POI global knowledge extraction layer, local dynamic graph generation module and trajectory prediction layer are major components in the proposed model. Raw taxi trajectory and POI semantic information are matched to the road network through map matching algorithm and proximity algorithm in the data conversion layer. The representation vectors of POI are learned through TransE algorithm by constructing knowledge graph in the POI global knowledge extraction layer. Based on the global knowledge graph, Local graph related to each trajectory point is generated and the graph representation vector is captured through GAT in the local dynamic graph generation module. Finally, trajectory points with the related graph representation vectors are input into LSTM with multi-head attention mechanism in the trajectory prediction layer to predict the next location.

However, there are limitations in this paper. Firstly, since the non-uniform sampling of GPS and the existence of GPS signal shielding areas, accurate trajectory recovery can be the future research direction. Moreover, only POI knowledge is considered as the external feature in this paper, while the vehicle trajectory can also be affected by other external features, such as morning and evening peak, and weather. Therefore, it is essential to integrate multi-source external features to predict the next location. Furthermore, scenario in this paper is the macro traffic roads, therefore more specific traffic scenarios, such as intersections, can be further studied in the future.

Author Contributions: Funding acquisition, J.C.; methodology, J.C.; project administration, J.C.; software, Q.F. and XX.; supervision, J.C.; validation, XX.; visualization, Q.F. and XX.; writing—original draft, J.C.; writing—review & editing, Q.F. and XX. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 61104166.

Data Availability Statement: The research data is unavailable due to privacy or ethical restrictions.

Acknowledgments: The authors would like to thank the reviewers for useful suggestions.

Conflicts of Interest: The authors declare there are no conflict of interest regarding the publication of this paper. The authors have no financial and personal relationships with other people or organizations that could inappropriately influence our work.

References

1. Guo, L. Research and application of location prediction algorithm based on deep learning, Doctoral thesis, Lanzhou University, Lanzhou, China, 2018.
2. Havyarimana, V.; Hanyurwimfura, D.; Nsengiyumva, P.; Xiao, Z. A novel hybrid approach based-SRG model for vehicle position prediction in multi-GPS outage conditions. *Inf. Fusion* **2018**, *41*, 1-8.
3. Wu, Y.; Hu, Q.; Wu, X. Motor vehicle trajectory prediction model in the context of the Internet of Vehicles, *Journal of Southeast University (Natural Science Edition)*, **2022**, *52*, 1199-1208.
4. Li, L.; Xu, Z. Review of the research on the motion planning methods of intelligent networked vehicles, *Journal of China Highway and Transport*, **2019**, *32*, 20-33.
5. Wang, K.; Wang, Y.; Deng, X.; et al.: Review of the impact of uncertainty on vehicle trajectory prediction, *Automotive Technology*, **2022**, *7*, 1-14.
6. Wang, L. Trajectory destination prediction based on traffic knowledge map, Doctoral thesis, Dalian University of Technology, Dalian, China, 2021.
7. Guo, H.; Meng, Q.; Zhao, X.; et al.: Map-enhanced generative adversarial trajectory prediction method for automated vehicles, *Information Sciences*, **2023**, *622*, 1033-1049.
8. Xu, H.; Yu, J.; Yuan, S.; et al.: Research on taxi parking location selection algorithm based on POI, *High-tech Communication*, **2021**, *31*, 1154-1163.
9. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio', P.; Bengio, Y. Graph Attention Networks. **2017**, ArXiv, abs/1710.10903.
10. Li, L.; Ping, Z.; Zhu, J.; et al.: Space-time information fusion vehicle trajectory prediction for group driving scenarios, *Journal of Transportation Engineering*, **2022**, *22*, 104-114.
11. Su, J.; Jin, Z.; Ren, J.; Yang, J.; Liu, Y. GDFormer: A Graph Diffusing Attention based approach for Traffic Flow Prediction. *Pattern Recognit. Lett.*, **2022**, *156*, 126-132.
12. Ali, A.; Zhu, Y.; Zakarya, M. Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. *Inf. Sci.*, **2021**, *577*, 852-870.
13. Fan, H. Research and implementation of vehicle motion tracking technology based on Internet of Vehicles, Doctoral thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2017.
14. Hui, F.; Wei, C.; Shangguan, W.; Ando, R.; Fang, S. Deep encoder-decoder-NN: A deep learning-based autonomous vehicle trajectory prediction and correction model. *Physica A: Statistical Mechanics and its Applications*. **2022**.
15. Kalatian, A.; Farooq, B. A context-aware pedestrian trajectory prediction framework for automated vehicles, *Transportation Research Part C: Emerging Technologies*, **2022**, *134*.
16. An, J.; Liu, W.; Liu, Q.; Guo, L.; Ren, P.; Li, T. DGInet: Dynamic graph and interaction-aware convolutional network for vehicle trajectory prediction, *Neural Networks*, **2022**, *151*, 336-348.
17. Yang, D.; He, T.; Wang, H., et al.: Research progress in graph embedding learning for knowledge map, *Journal of Software*, **2022**, *33*, 21.
18. Xia, Y.; Lan, M.; Chen, X., et al.: Overview of interpretable knowledge map reasoning methods, *Journal of Network and Information Security*, **2022**, *8*, 1-25.
19. Zhang, Z.; Qian, Y.; Xing, Y.; et al.: Overview of TransE-based representation learning methods, *Computer Application Research*, **2021**, *3*, 656-663.
20. Chen, W.; Wen, Y.; Zhang, X.; et al.: An improved TransE-based knowledge map representation method, *Computer Engineering*, **2020**, *46*, 8.
21. Zheng, D.; Song, X.; Ma, C.; Tan, Z.; Ye, Z.; Dong, J.; Xiong, H.; Zhang, Z.; Karypis, G. DGL-KE: Training Knowledge Graph Embeddings at Scale. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. **2020**.
22. Ji, Q.; Jin, J. Reasoning Traffic Pattern Knowledge Graph in Predicting Real-Time Traffic Congestion Propagation. *IFAC-PapersOnLine*, **2020**, *53*, 578-581.

23. Wang, X.; Lyu, S.; Wang, X.; Wu, X.; Chen, H. Temporal knowledge graph embedding via sparse transfer matrix. *Inf. Sci.*, **2022**, *623*, 56-69.
24. Wang, C.; Tian, R.; Hu, J.; Ma, Z. A trend graph attention network for traffic prediction. *Inf. Sci.*, **2023**, *623*, 275-292.
25. Wang, B.; Wang, J. St-Mgat:Spatio-Temporal Multi-Head Graph Attention Network for Traffic Flow Prediction. *SSRN Electronic Journal*. **2022**.
26. Wang, T.; Ni, S.; Qin, T.; Cao, D. TransGAT: A dynamic graph attention residual networks for traffic flow forecasting. *Sustain. Comput. Informatics Syst.* **2022**, *36*, 100779.
27. Cai, K.; Shen, Z.; Luo, X.; Li, Y. Temporal attention aware dual-graph convolution network for air traffic flow prediction. *Journal of Air Transport Management*. **2023**, *106*.
28. Yan, X.; Gan, X.; Wang, R.; Qin, T. Self-attention eidetic 3D-LSTM: Video prediction models for traffic flow forecasting. *Neurocomputing,* **2022**, *509*, 167-176.
29. Chen, L.; Shi, P.; Li, G.; Qi, T. Traffic flow prediction using multi-view graph convolution and masked attention mechanism. *Comput. Commun.*, **2022**, *194*, 446-457.
30. Wang, K.; Ma, C.; Qiao, Y.; Lu, X.; Hao, W.; Dong, S. A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction. *Physica A-statistical Mechanics and Its Applications,* **2021**, *583*, 126293.
31. Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y. Map-matching for low-sampling-rate GPS trajectories. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2009.
32. Cai, Y. Research on vehicle trajectory prediction based on RNN-LSTM network, Doctoral thesis, Jilin University, Jilin, China, 2021.
33. Zhang, H., Huang, C., Xuan, Y., et al.: Real time prediction of air combat flight trajectory using gated cycle unit. *Systems engineering and electronic technology*, **2020**, *42*, 7.
34. Guo, Y.; Zhang, R.; Chen, Y. Vehicle trajectory prediction based on potential characteristics of observation data and bidirectional short-term and long-term memory network. *Automotive Technology*, **2022**, *3*.
35. Liu, C.; Liang, J. Vehicle trajectory prediction based on attention mechanism. *Journal of Zhejiang University (Engineering Edition)*, **2020**, *54*, 8.
36. Guan, D. Research on Modeling and prediction of vehicle moving trajectory in the Internet of vehicles, Doctoral thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.