

Research Article

Multigraph Aggregation Spatiotemporal Graph Convolution Network for Ride-Hailing Pick-Up Region Prediction

Cong Li ^{1,2}, Huyin Zhang ¹, Zengkai Wang ³, Yonghao Wu ¹ and Fei Yang ¹

¹School of Computer Science, Wuhan University, Wuhan, China

²Department of Information Engineering, Wuhan Institute of City, Wuhan, China

³College of Information Science and Engineering, Jiaying University, Jiaying, China

Correspondence should be addressed to Huyin Zhang; zhy2536@whu.edu.cn

Received 8 April 2022; Revised 16 May 2022; Accepted 18 May 2022; Published 8 June 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Cong Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prediction of pick-up regions for online ride-hailing can reduce the number of vacant vehicles on the streets, which will optimize the transportation efficiency of cities, reduce energy consumption and carbon emissions, and increase the income of online ride-hailing drivers. However, traditional studies have ignored the temporal and spatial dependencies among pick-up regions and the effects of similarity of POI attributes in different regions in modelling, making the features of the model incomplete. To address the above problems, we propose a new multigraph aggregation spatiotemporal graph convolutional network (MAST-GCN) model to predict pick-up regions for online ride-hailing. In this paper, we propose a graph aggregation method to extract the spatiotemporal aspects and preference features of spatial graphs, order graphs, and POI graphs. GCN is used on the aggregated graphs to extract spatial dimensional features from graph-structured data. The historical data are sequentially divided into temporal granularity according to the period, and convolution operations are performed on the time axis to obtain the features in the temporal dimension. The attention mechanism is used to assign different weights to features with strong periodicity and strong correlation, which effectively solves the pick-up region prediction problem. We implemented the MAST-GCN model based on the PyTorch framework, stacked with a two-layer spatiotemporal graph convolution module, where the dimension of the graph convolution is 64. We evaluate the proposed model on two real-world large scale ride-hailing datasets. The results show that our method provides significant improvements over state-of-the-art baselines.

1. Introduction

As of June 2021, according to China's online ride-hailing regulatory information interaction platform, the scale of China's online ride-hailing passengers reached 397 million, with an average daily order number of more than 21 million units. Demand for online ride-hailing services has led to an increase in the popularity of ride-hailing among residents. Although online ride-hailing services have many advantages, some concerns have emerged against the backdrop of their increasing adoption. It was found that as more and more passengers choose to travel by online ride-hailing, this in turn has increased urban traffic congestion. A study of online ride-hailing and urban congestion reveals that instead of complementing public transportation, online ride-hailing

is diverting passenger traffic. There has been some disagreement among researchers as to whether or not online ride-hailing has added to traffic congestion in some cities [1, 2]. According to the statistical analysis of relevant researchers [3, 4], the daily mileage of taxi or online ride-hailing is about 400 km, while the average empty rate of cabs is about 40%. This means that almost half of the time cabs operate inefficiently, leading to more transportation resources consumption and environmental pollution. Despite the fact that cabs are mostly vacant, many citizens still struggle with the difficulty of taking a cab. The most pressing issue that needs to be addressed is reducing the number of miles traveled by online ride-hailing cabs when they are vacant in the process of looking for passengers. The usual method for taxi drivers to identify potential passengers in a traditional taxi service is

to drive about the city and wait at “hot spots,” such as the gates of railway stations, hotels, restaurants, and shopping malls. However, while there is a large data processing server that monitors consumer requests and distributes them to drivers for ride-hailing services, problems similar to those experienced by normal taxi services still persist [5]. According to statistics from the Online ride-hailing Regulatory Information Exchange Platform, the average waiting time of online ride-hailing in China is about 8 minutes. Customers’ demands may still be far away from the driver’s location, accumulating a high distance of empty cars, resulting in huge fuel consumption and redundancy.

Some ride-hailing companies, such as Lyft in the U.S. and DiDi in China, offer passengers a convenient ride-hailing service. These ride-hailing platforms collect an extensive amount of ride-hailing operation data, including passengers’ travel time, pick-up region, destination, waiting time, and other valuable information. Mining and analyzing the relevant valuable information in these data can provide better services for drivers and passengers. Pick-up region prediction can help drivers effectively improve the efficiency of looking for passengers and reduce carbon dioxide emissions and energy consumption caused by no-load cruising [6].

The development of intelligent transportation provides unprecedented opportunities for pick-up region prediction, but it also faces new challenges. First of all, the increasing operation data of ride-hailing will consume a lot of storage resources and reduce the forecasting efficiency [7]. Secondly, many cities are constantly building or renovating roads. If they use too much historical trajectory data, noise data will be introduced instead, leading to the decline of performance. Therefore, the noise data can be greatly reduced if the order data can reflect the law of driver’s choice of passengers. However, the passenger carrying area reached by ride-hailing vehicles is very limited every day, which makes this method face the problem of data sparsity [8]. In addition, as the geographical location changes, the driver’s focus on the passenger area will also change dynamically, such as the number and type of POI (point of interest) in different areas, which will also affect the order number [9].

According to the urban planning, the city is divided into various functional regions, such as industrial region, commercial region, residential region, and leisure region [5]. In general, people return to their residential areas on weeknights or go shopping downtown on weekends, as well as commuting between functional regions. People engage in a range of social activities related to these areas. According to the study [1], most of the areas where residents ride belong to a small area within a certain point of interest. Therefore, we mine the data of ride-hailing orders for potential information about passengers’ pick-up regions. These potential messages are characterized by spatial dependence, temporal dependence, and preference of POIs.

- (1) Spatial dependence: passenger travel has a certain spatiality. According to the planning of the city, the number of orders in the central city is more than the number of orders in the suburbs. Residential

areas, commercial areas, and other functional areas have more orders than industrial areas. Passengers frequently visit specific areas and their neighborhoods.

- (2) Temporal dependence: passenger travel is temporal in nature [10]. During the morning peak, the boarding area is generally near the residential area, while the alighting area is generally near the office area. There is temporality in passenger travel, and the boarding area shows a cyclical change.
- (3) Preference of POIs: passengers’ trips also show different preferences when they are in different areas or at different times of the day [11]. For example, passengers may frequently visit shopping-related POIs on weekends and leisure places near residential areas in the evening. Thus, the pattern of passengers in functional urban areas is relatively stable compared to mobility [12], as shown in Figure 1.

In summary, pick-up region prediction faces more challenges in providing prediction due to spatiotemporal dependencies compared to traditional ride-hailing order prediction. In addition, the movement of online vehicles between functional regions involves different movement patterns of passengers and the order data in a region contains POIs for multiple functions, making it more complex to analyze.

To solve the aforementioned issues, we offer a multigraph aggregation spatiotemporal graph convolution network model in this research. First, a multigraph aggregation method is proposed to fuse geospatial data, ride-hailing order data, and POI data to mine the potential information of passenger boarding area to provide a solution to the data sparsity problem. Second, we use a multigraph structure to obtain data representations from various perspectives and adopt a spatiotemporal graph convolution structure to simultaneously capture data in time, space, and behavioral preferences. Finally, we capture the dynamic spatial correlation and dynamic temporal correlation between different regions by using the spatiotemporal attention mechanism to improve the prediction performance.

Our contributions are highlighted as follows:

- (i) We present a graph convolutional network model with multiple graph aggregation and three separate graphs, spatial graph, order graph, and POI graph. A method for aggregating multiple graphs is designed so that graph convolution can be applied simultaneously among multiple heterogeneous data
- (ii) A spatiotemporal graph convolution structure is proposed to model the temporal dependence, spatial dependence, and spatiotemporal dependence among three kinds of graph. A spatiotemporal attention-based mechanism is used to selectively obtain more valuable information to enhance the prediction
- (iii) An encoder-decoder structure with LSTM units is constructed to extract the temporal dependence of

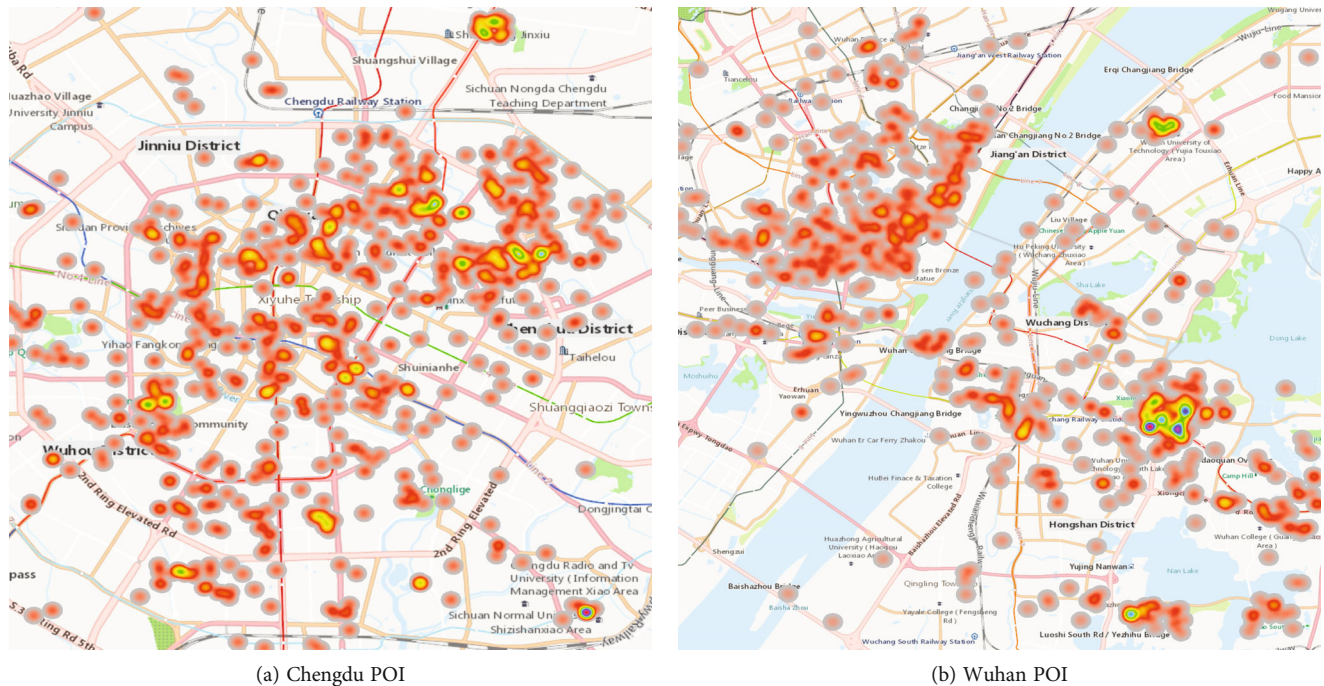


FIGURE 1: POI information for Chengdu and Wuhan.

the multigraph and predict the pick-up region of the ride-hailing

- (iv) Extensive experiments on the model proposed are conducted on the Chengdu DiDi Chuxing dataset and the Wuhan taxi dataset. On three evaluation metrics, MSE, RMSE, and MAE, the relevant experimental results reveal that our strategy outperforms the state-of-the-art baseline

2. Related Work

Ride-hailing pick-up region prediction is one of the research hotspots in recent years. The use of a ride-hailing pick-up region prediction approach can help to organize vehicle flow, increase vehicle utilization, decrease waiting time, and alleviate traffic congestion; an overview of relevant research work is shown in Table 1.

The development of smart transportation offers great opportunities for employing data mining methods for demand forecasting of cabs. Several researchers have used spatial cluster analysis methods to address taxi pick-up and drop-off region prediction. [13] used spatial point cluster analysis to perform point clustering of cab pick-up and drop-off points and obtain popular pick-up and drop-off areas to recommend the best pick-up points to cab drivers. However, it is difficult to quickly identify clusters with irregular shapes when the amount of data on pick-up and drop-off points is large. Also, how to determine the similarity coefficient of data with multiple attributes is one of the current difficulties. [14] clustered the trajectories between the drop-off hotspots and the pick-up hotspots to get the optimal path with the most customer-seeking potential. However, overall-based trajectory clustering ignores the detailed

information of subtrajectories, while segment-based trajectory clustering segments the trajectories, thus ignoring the similarity measure in the spatiotemporal dimension. Researchers also use time-series methods to analyze taxi trajectory data; [15] propose a Taxi-RS method to search for frequent pattern subsequences of trajectories and construct a frequent trajectory graph model, which can calculate the best pick-up region prediction results. [16] have used automatic ARIMA models for time-series analysis to predict hotspot areas for passengers. However, time-series methods are used to analyze one-dimensional data, and there are many limitations in applying them to two-dimensional data.

In recent years, researchers have tried to solve this problem using machine learning methods. [17] have used population genetic algorithms for the shortest path calculation to implement a taxi dispatching model and also to recommend the best area for taxi drivers to carry passengers. [18] develop a taxi path optimization model and solve the taxi path optimization model by using an improved genetic algorithm. The genetic algorithm uses a heuristic search, which is easy to parallelize, but may be premature and inefficient when dealing with large-scale data. To forecast the demand for taxi, [19] combined local geographical elements of taxi demand and meteorological data into a convolutional LSTM (ConvLSTM). The model not only establishes the same temporal characteristics as the traditional LSTM model but also depicts the local spatial features like CNN. [20] propose an integrated model based on LSTM method, a selected pass recurrent unit network (GRU), and an extreme gradient advancement (XGBOOST) model, combined with point of interest (POI) data to predict taxi demand. The above methods can capture the nonlinear characteristics of time-series data. Nevertheless, it is difficult for time-series prediction which is for a single node to describe the interactions

TABLE 1: Overview of related work to pick-up region prediction.

Method	Technology	Related model
Traditional methods	Spatial cluster methods	Point cluster [13]
		Trajectory cluster [14]
	Time series methods	Taxi-RS [15] ARIMA [16]
Machine learning methods	Genetic algorithm	PGA [17] HGA [18]
	LSTM	ConvLSTM [19]
		LSTM and GRU [20]
Deep learning methods	GCN	GCN [21, 22] ST-ED-RMGC [23]
	Attention mechanism	MRA [24]

between nodes. Meanwhile, recurrent networks for sequence learning require iterative training, which introduces a gradual accumulation of errors.

Due to the heterogeneity, spatial and temporal characteristics of traffic data, many methods, and models based on graph convolutional networks have emerged to achieve the extraction of spatial and temporal features of traffic data. [21] studied the operation mode of urban cabs, divided the road network into grids, and realized the traffic flow prediction while using graph convolution to achieve taxi demand prediction. [22] constructed various graph structure-based representations using adjacent neighborhoods and similar functional areas as nodes, based on which they used multiple groups of GCNs for spatial correlation modeling to achieve demand forecasting for online ride-hailing. [23] propose a new deep learning model ST-ED-RMGC to construct multiple graphs for OD (Origin Destination) prediction of cabs by spatial distance as well as semantic correlation and use RMGC network to decode the compressed vectors into OD graphs and finally predict the future OD demand. Some researchers have introduced attention mechanisms into GCNs. [24] proposed a multirange attention mechanism for two-component graph convolution. The model first constructs the node graph and edge graph separately and designs a two-layer graph convolution model for modelling edge-node interactions, considers the influence of surrounding nodes on the target table nodes, and proposes the use of a multi-range attention mechanism to aggregate information of neighboring nodes to dynamically understand the importance of different aggregation ranges. GCN has become a fundamental model for traffic prediction research and a benchmark method for experiments. Since the traffic data itself is time-series data, how to mine the time-series features and fuse them with spatial features for prediction has become the focus of improvement of the models used for traffic prediction. The above GCN model relies on the eigenvalues of the Laplacian matrix, which makes it difficult to abstract the convolution operation from the whole static graph structure. At the same time, the information carried by a single graph is increasingly difficult to meet the needs of traffic prediction. Therefore, the traffic prediction prob-

lem relies on well-defined graph structure information to efficiently extract spatial and temporal features and model them more finely.

As mentioned in the review, the pick-up region is affected by the space, time, and POI of that region. Therefore, we construct three heterogeneous graphs based on historical data and propose a graph aggregation method to fuse the temporal and spatial features as well as preference features of the three graph structures. GCN is used on the aggregated graphs to extract spatial dimensional features from the graph structure data. The historical data are sequentially divided into temporal granularity based on periodicity, and convolution operations are performed on the time axis to obtain the features in temporal dimension. The pick-up region prediction problem is effectively solved by using the spatiotemporal attention mechanism to assign different weights to features with strong periodicity and strong correlation.

3. Definitions and Preliminaries

In this section, we first provide important definitions of geospatial graphs, ride-hailing order graphs, and POI graphs. This paper adopts the classic processing method in the field of transportation [25] and divides urban regions to be processed into multiple grids on average, dividing relevant study regions into $i \times j$ grids equally. The spatial area on the i^{th} row and j^{th} column of the grid region is denoted as $R(i, j)$ in the paper. With these region grids, we can transform the geospatial data into a region grid matrix. The grid matrix is the most suitable data input format for graph convolutional network models. In this manner, the total number of ride-hailing orders in each small grid area is studied. Then, each grid is regarded as a vertex of the graph, which is used to construct the graph model [26].

Definition 1. In spatial graph, we define the regional grids as an undirected graph $G_s = (V_s, E_s, A_s)$, where V_s is the centre point of each grid region, E is the distance between the centre points of each grid region, and the centre point of each grid is regarded as the geographic location centre of the grid.

The distance from the centre is regarded as the edge weight of the geographic graph structure. A is the spatial adjacency matrix representing the distance dependence between nodes [27]. The calculation function of the centre distance between the two nodes is $\text{dist}(\cdot)$. The closer the distance is, the smaller the weight is. The online ride-hailing demand between the two will also be similar to a certain extent. The geographical map range set can be defined as

$$\psi_i = \{r_j | \text{dist}(r_i, r_j) \leq D\}, \quad (1)$$

where D is the settable threshold.

Definition 2. In order graph, the number of ride-hailing orders in a region is a key factor affecting the prediction of passenger pick-up regions. Therefore, based on the similarity of regional ride-hailing demands, we constructed an order graph of spatial regions to represent the correlation between regions and called it an order graph. We define the order graph as an undirected graph $G_o^{(t)} = (V_o, E_o^{(t)}, A_o^{(t)})$ at time interval t . V_o is the set of nodes, and $E_o^{(t)}$ is the set of edges [28]. The graph's nodes are the region grids, and the graph's edges represent the links between them. The spatial adjacency matrix $A_o(t)$ represents the reliance between nodes. If an order is placed within the range of regional grid i , it will be recorded as a visit to grid i by ride-hailing. In this study, the number of passengers in the region is used to approximate the ride-hailing pick-up region, and the Dynamic Time Warping (DTW) algorithm [29] is used to computational similarity φ_{ij} of the demand time-slot between grid i and grid j , as shown in Equation (2). As long as there is a demand for online ride-hailing between any two vertices, they are related. At the same time, the order graph will be affected by the time factor, because the order information between the two regions is often different in different time periods, so the changes of the order graph at different time should be taken into account in the modeling. We divide the orders of the day into 48 segments equal in time. The grid attribute values are updated once every 0.5 hours to adapt to the preference changes of ride-hailing in different time periods.

$$\varphi_{ij} = \text{DTW}(i, j). \quad (2)$$

At this point, the adjacency matrix A_o is calculated by

$$A_o = \begin{cases} 0, & \varphi_{ij} > \varphi, \\ 1, & \varphi_{ij} \leq \varphi, \end{cases} \quad (3)$$

where φ is the hyperparameter, indicating the similarity threshold of whether the two regions are related.

Definition 3 POI Graph. In this study, POI in some areas of Chengdu was climbed and classified into activity types of POIs in each grid. The activity attributes of the grid were consistent with those of POIs with the largest number of same attributes [30]. We define POI graph denoted as G_p

$= (V_p, E_p, A_p)$; the DTW algorithm was used to quantify the similarity between grids with similar POI activity attributes, as show in

$$P_{i,j} = \begin{cases} \exp(-\text{DTW}(O_i, O_j)), & i \neq j, \\ 1, & i = j, \end{cases} \quad (4)$$

where $O_i, O_j \in \mathbb{R}^{1 \times T}$ represents the ride-hailing orders vector of the i th grid and T is the length of the vector, determined by the selected control time scale. After the matrix P is obtained, the weight of POI activity attribute graph can be obtained by normalization.

4. Proposed Method

4.1. The Framework of the Proposed Method. In this section, we formalize the learning problem of spatiotemporal prediction of ride-hailing pick-up region and describe how to model spatiotemporal correlation using the proposed multi-graph aggregation spatiotemporal graph convolution network (MAST-GCN). Figure 2 shows the system architecture which consists of four main components: the graph construction block, the GCN block, the LSTM block, and the prediction block. In the graph construction module, we divide the spatial area by a fixed-size grid, map the POI information and ride-hailing order information to the corresponding grid, and then aggregate the relevant information by the grid to form the aggregation graph. The GCN module uses the above aggregated multigraph to make the input of the graph convolutional network can have one graph feature and multigraph structure feature description matrices, thus fusing multiple spatial relationship matrices (graph structure feature description matrix) and temporal feature (graph feature). The time-series data with spatial features are used as the input to the LSTM module by a two-layer graph convolution operation. The encoded LSTM in the LSTM module is used to capture the position vector sequence, and the decoded LSTM is used to predict the pick-up point vector sequence. The spatiotemporal attention mechanism between encoding and decoding is to dynamically capture the dependency between the location to be predicted and the sequence of location vectors. The output of the model is a recommended sequence of ride-hailing pick-up regions.

4.2. Multigraph Aggregation. If each graph model is trained separately, the complexity of the algorithm will be greatly improved. In order to avoid this shortcoming, this study improves the traditional aggregation function and designs a graph aggregator by comprehensively considering the different influence degrees of the three graph models on the prediction results [21], as shown in Figure 3. The aggregation mode of spatial graph is shown in

$$S_t^i = \delta \left(W_s \cdot \left(f_t^i + \sum_{r_j \in \psi_i} \frac{\text{dist}^2(r_i, r_j)}{\sum \text{dist}^2(r_i, r_j)} f_t^j \right) \right), \quad (5)$$

where s_t^i represents the spatial graph embedding vector at

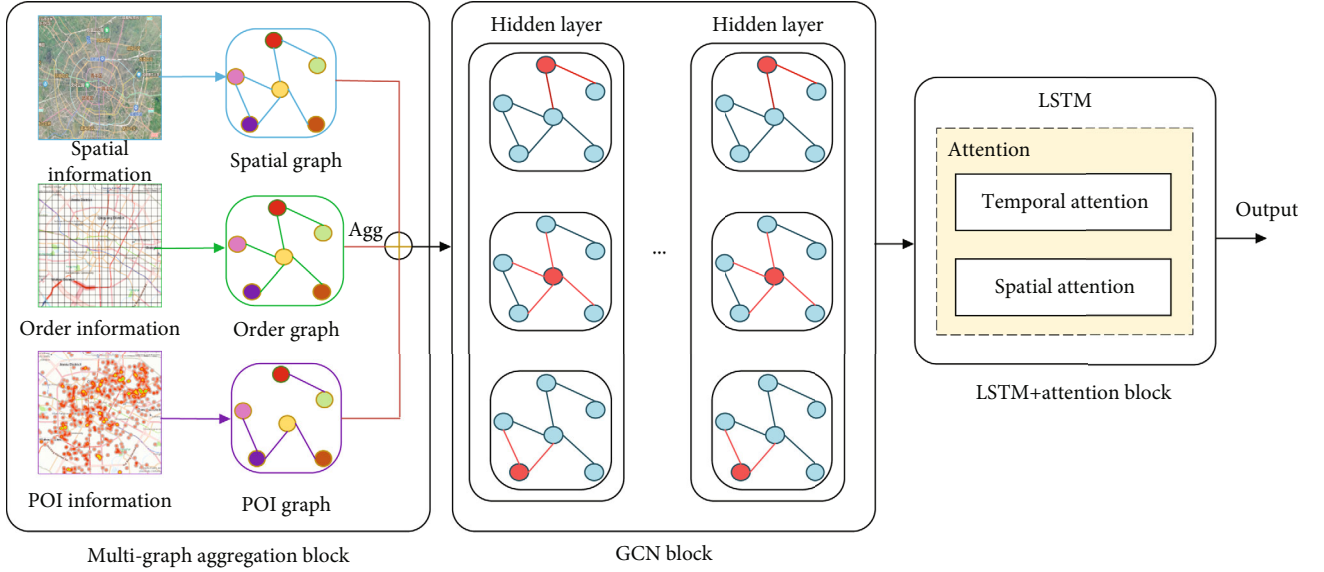


FIGURE 2: The framework of the proposed multigraph aggregation spatiotemporal graph convolutional network model (MAST-GCN).

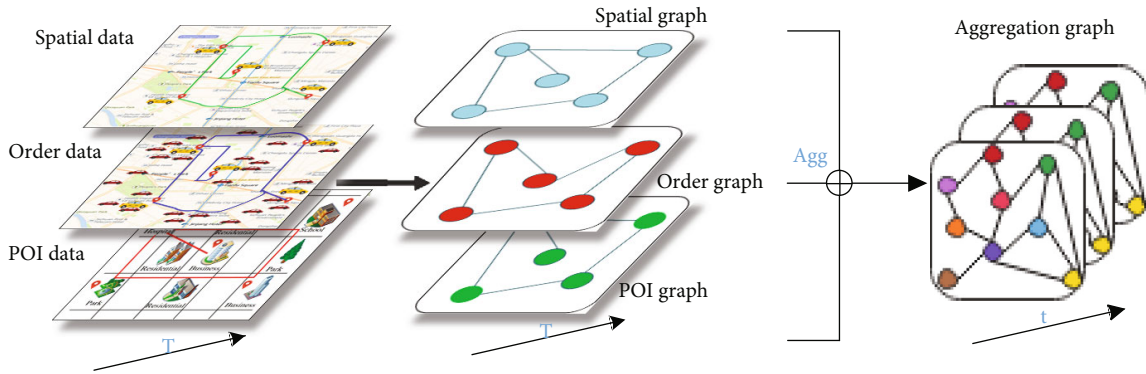


FIGURE 3: Illustration of graph aggregation of spatial, order, and POI graphs.

time t , W_s is the trainable weight matrix, and f_t^i and f_t^j are characteristics of r_i and r_j before spatial aggregation operation, respectively. Similarly, the features of order graph can be aggregated as shown in

$$O_t^i = \delta \left(W_O \cdot \left(f_t^i + \sum \frac{\text{num}^2(r_i)}{\sum \text{num}^2(r_j)} f_t^j \right) \right), \quad (6)$$

where $\text{num}(m_i)$ represents the number of orders in region r_i , W_O is the trainable weight matrix, O_t^i represents the order graph embedding vector at time t , and f_t^i and f_t^j are characteristics of r_i and r_j before the number of order aggregation operation, respectively. The features of POI graph can be aggregated as shown in

$$P_t^i = \delta \left(W_P \cdot \left(f_t^i + \sum \frac{\text{sim}^2(r_i, r_j)}{\sum \text{sim}^2(r_i, r_j)} f_t^j \right) \right), \quad (7)$$

where $\text{sim}(r_i, r_j)$ represents POI activity attribute similarity

of r_i and r_j , P_t^i represents the POI graph embedding vector at time t , W_P is the trainable weight matrix, and f_t^i and f_t^j are characteristics of r_i and r_j before the activity attribute of POI aggregation operation, respectively.

We embed vectors S_t^i , O_t^i , and P_t^i carry spatial information, order quantity of grid, and POI attribute information, respectively. Different neighborhood contexts and spatial features are used to learn the knowledge simultaneously. The final representation \mathcal{F}_t^i of grid G_i at time t is calculated in the last stage of grid embedding by merging grid embedding vectors from three aspects as shown in

$$\mathcal{F}_t^i = [S_t^i, O_t^i, P_t^i], \quad (8)$$

where $[\cdot]$ represents the concatenation of three vectors.

4.3. Graph Convolution Network. After graph aggregation construction, we adopt the GCN module to capture the spatiotemporal dependencies. GCN uses a neighborhood aggregation scheme to compute a new feature vector for each node by iteratively aggregating and transforming the feature

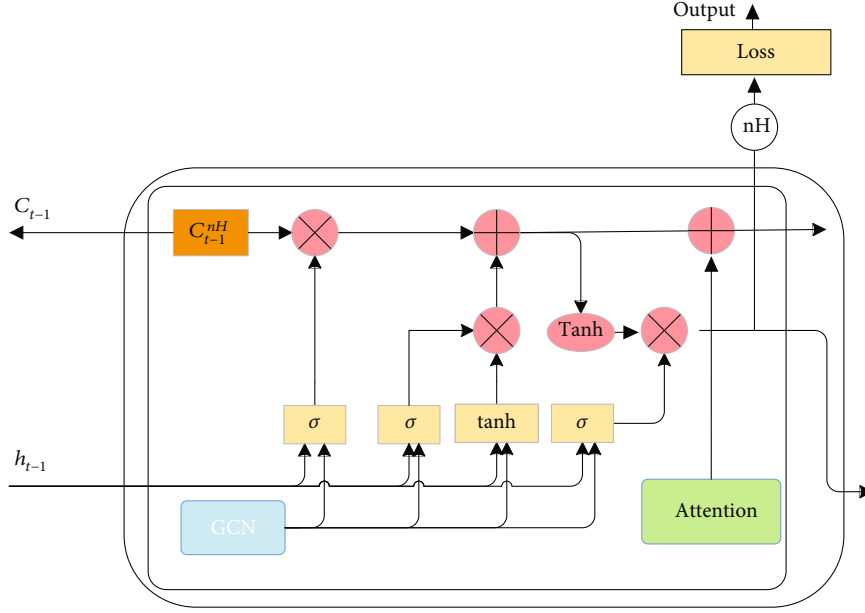


FIGURE 4: The architecture of LSTM and Attention Block.

vectors of its neighboring nodes. The aggregate operation of GCN performs a traversal of the neighboring nodes for each node, aggregating the feature vectors of the neighboring nodes. According to [25], the graph convolution operation method based on Chebyshev polynomial approximation is defined as follows:

$$\mathcal{F}' = \sigma \left(\sum_{i=0}^{n-1} \alpha_i L^i \mathcal{F} \right), \quad (9)$$

where \mathcal{F} is the input feature, $(\alpha_0, \alpha_1, \dots, \alpha_{n-1})$ is learnable coefficients, L^i is the i -th power graph Laplacian matrix, and $\sigma(\cdot)$ denotes the activation function. The graph Laplacian matrix is calculated as follows:

$$L = I - D^{-1/2} A D^{-1/2}, \quad (10)$$

where I is the identity matrix, D denotes the degree matrix of the graph, and A is the adjacency matrix of the graph.

Based on the GCN theory proposed by the [31], the equation of GCN layer can be expressed as

$$\mathcal{F}' = \sigma \left(W \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \mathcal{F} \right), \quad (11)$$

where $\tilde{A} = A + I$ is expressed as the renormalized matrix, $\tilde{D} = \sum_i \tilde{A}_{ij}$.

Based on the spatiotemporal graph structure, the model can obtain the first-order neighbor information through the graph structure during a graph convolution operation, that is, temporal dependence and spatial dependence. In the second convolution operation, the model can obtain the information of the second-order neighbor, that is, the spatiotemporal dependence.

4.4. LSTM Module. Although the spatiotemporal graph convolution network has captured the spatiotemporal relation, the time-series information cannot be captured because the graph convolution will ignore the sequence of nodes before and after capturing the time information. Therefore, the output of the spatiotemporal graph convolutional network is input into LSTM to capture the sequence information between time nodes, as shown in Figure 4.

The LSTM model has proved to be very effective in processing time-series data with long temporal dependent features. In a spatiotemporal graph convolution network, each grid G_i has a feature vector of time t , such as $\{v_1^i, v_2^i, \dots, v_t^i\}$. Taking a time-series $\{v_t^i\}_{t=1}^T$ as input, LSTM encodes $\{v_t^i\}_{t=1}^T$ into hidden states $\{h_t^i\}_{t=1}^T$ via $h_t^i = f(v_t^i, h_{t-1}^i)$; the formula is shown as follows:

$$\begin{aligned} f_t^i &= \sigma \left(W_{xf} v_t^i + W_{hf} h_{t-1}^i + b_f \right), \\ i_t^i &= \sigma \left(W_{xi} v_t^i + W_{hi} h_{t-1}^i + b_i \right), \\ o_t^i &= \sigma \left(W_{xo} v_t^i + W_{ho} h_{t-1}^i + b_o \right), \end{aligned} \quad (12)$$

$$c_t^i = f_t^i \odot c_{t-1}^i + i_t^i \odot \tanh \left(W_{xc} v_t^i + W_{hc} h_{t-1}^i + b_c \right),$$

$$h_t^i = o_t^i \odot \tanh \left(c_t^i \right),$$

where f_t^i, i_t^i and o_t^i , respectively, forgotten gate, update gate, and output gate and c_t^i and h_t^i , respectively, are cell memory state vector and hidden state vector. In these equations, σ is the sigmoid function, \odot is element wise product, and v_t^i is the input vector. W and b is the weight and bias in the training process. We simplify the LSTM representations in

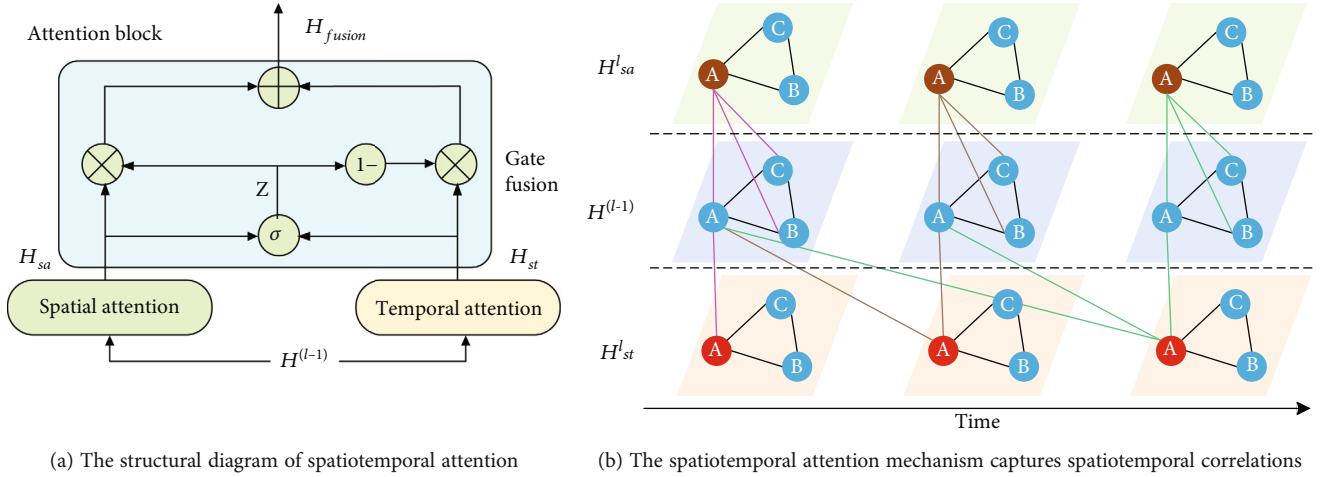


FIGURE 5: Spatiotemporal attention mechanism.

Equation (13).

$$h_t^i = \text{LSTM}\left(v_t^i, h_{t-1}^i, c_{t-1}^i\right). \quad (13)$$

4.5. Spatiotemporal Attention Mechanism. The problem of predicting pick-up regions lies in only considering the number of orders in the neighboring grid or in the current time window, which is insufficient. Regions farther away from the grid regions and the orders in the period before and after the grid regions should receive relatively more or less receive attention. We take into consideration the uneven characteristics of passenger travel activity levels in different periods and the mobility characteristics in different POIs and pay more attention to the correlation between key spatial regions and POI regions and between key periods and prediction regions. Although the spatiotemporal features are obtained by the GCN module, the spatial regions have been changing dynamically with time, so the attention weights for each node are different at different time points. At each moment, global attention should be paid to each node instead of a few nodes individually. Therefore, the dynamic spatial correlation between different regions is captured by using the spatial attention mechanism and the dynamic temporal correlation between different times is captured by using the temporal attention mechanism to improve the prediction performance.

We used an attention mechanism to modify the initial LSTM to use the incoming spatiotemporal information. First, for a single time step, spatial weights are dynamically applied to the input features. Then, at each time step, the temporal attention weights are allocated to the hidden states by making full use of the hidden states at each LSTM phase. The input and output of the LSTM cells are affected by spatial and temporal attention weights. We can dynamically alter the attention weights while enhancing the LSTM cell's performance with the help of the spatial and temporal attention modules.

We denote the input of the l^{th} layer as $H^{(l-1)}$, where the hidden state of vertex v_i at time step t_j is denoted as h_{v_i, t_j}^{l-1} .

The outputs of the temporal attention mechanism in the l^{th} layer are denoted as H_s^l and H_t^l , respectively, where the hidden state of vertex v_i at time step t_j is denoted as h_{v_i, t_j}^{l-1} and h_{v_i, t_j}^l , respectively. After gated fusion, we obtain the output $H^{(l)}$ of the l^{th} layer, as shown in Figure 5(a).

4.5.1. Spatial Attention Mechanism. In the spatial dimension, there are significant differences in the interactions between different regions. For example, most people leave home in the morning to go to work, indicating a clear influence relationship between residential and industrial areas. However, the influence relationship between certain unrelated places is relatively weak. So, passengers' pick-up regions affect each other and have a strong dynamic. Therefore, we use spatial attention to dynamically assign different weights to different vertices and adaptively obtain the spatial association relations of nodes in the spatial dimension, as shown in Figure 5(b). The spatial attention mechanism assigns different weights to the spatial features of nodes at time t . Then, the attention weights of each node relative to all nodes are fused through the full connection layer. The formula for calculating spatial attention is as follows:

$$H_{sa}^t = \text{ReLU}\left(\sum_{i=1}^N S_i^t h_i^t\right),$$

$$S_i^t = V_s \tanh\left(W_s' [h_{t-1}; c_{t-1}] + W_s h_t + b_s\right), \quad (14)$$

$$S_{i,j} = \frac{\exp(S_{i,j})}{\sum_{j=1}^N \exp(S_{i,j})},$$

where H_{sa}^t represents the aggregation of spatial information of the node at time t , S_i^t is the weight of the hidden state vector of nodes, and $[h_{t-1}; c_{t-1}]$ is the hidden state and cell state of the previous unit at time $t-1$. The value of an element $S_{i,j}$ in S_i^t semantically represents the correlation strength between node i and node j . where $V_s, b_s \in \mathbb{R}^{m \times 1}$, $W_s' \in \mathbb{R}^{m \times 2n}$, and $W_s \in \mathbb{R}^{m \times n}$ are learnable parameters.

TABLE 2: Comparison of prediction performances in different models.

Dataset	Methods	Weekday		Weekend		
		RMSE	MAE	RMSE	MAE	
Chengdu	Traditional models	HA	53.25	32.18	54.67	33.58
		ARIMA	41.72	29.86	43.32	30.67
		LSTM	40.81	26.35	42.73	28.49
	Deep models	GRU	36.56	25.22	37.55	26.62
		MGCN	32.39	23.21	34.68	24.94
		STGCN	29.82	21.24	30.99	22.57
	Our model	MAST-GCN	20.57	13.29	21.82	14.12
Wuhan	Traditional models	HA	55.49	33.55	56.72	34.25
		ARIMA	43.89	31.87	45.76	32.36
		LSTM	42.13	27.57	45.18	29.77
	Deep models	GRU	37.52	26.47	38.37	28.17
		MGCN	33.89	24.31	34.94	25.39
		STGCN	30.76	21.52	31.8	23.71
	Our model	MAST-GCN	22.49	14.68	23.68	15.31

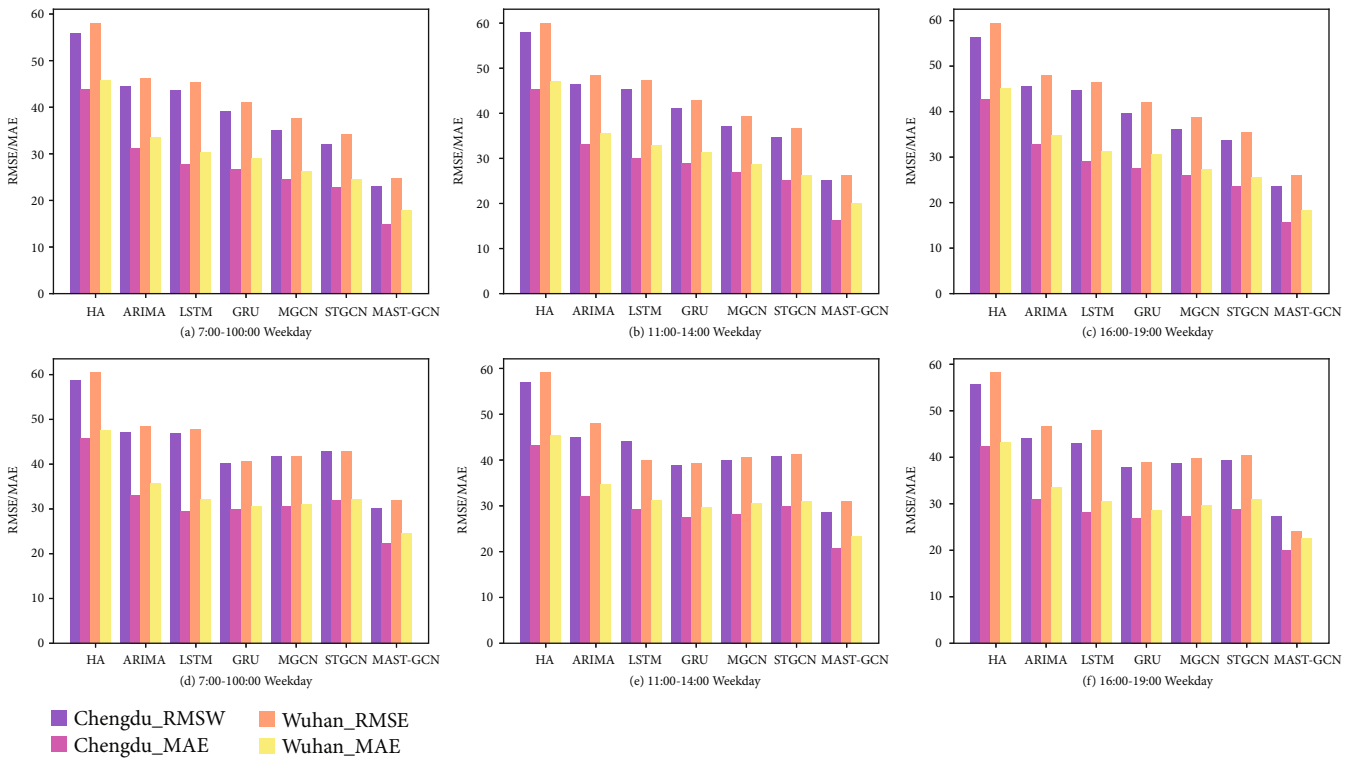


FIGURE 6: Comparison of average performance of different methods.

4.5.2. Temporal Attention Mechanism. During the prediction process, we find that some destinations are highly correlated during certain periods of time. Most of the destinations of online ride-hailing during the morning rush hour (7:00-9:00) are companies, schools, hospitals, and so on. In the evening peak (16:00-19:00), there are more residential clusters. The temporal attention mechanism learns the long time-dependent characteristics of historical data and assigns

higher weights to more relevant destinations over a particular time period. Therefore, we can grasp the underlying movement patterns by analyzing destination weights and making the attention model easier to interpret. As shown in Figure 5(b), temporal attention is used to adaptively capture dynamic temporal correlations between different time periods in order to best manage temporal information with periodicity. Assign various priority weights to each hidden

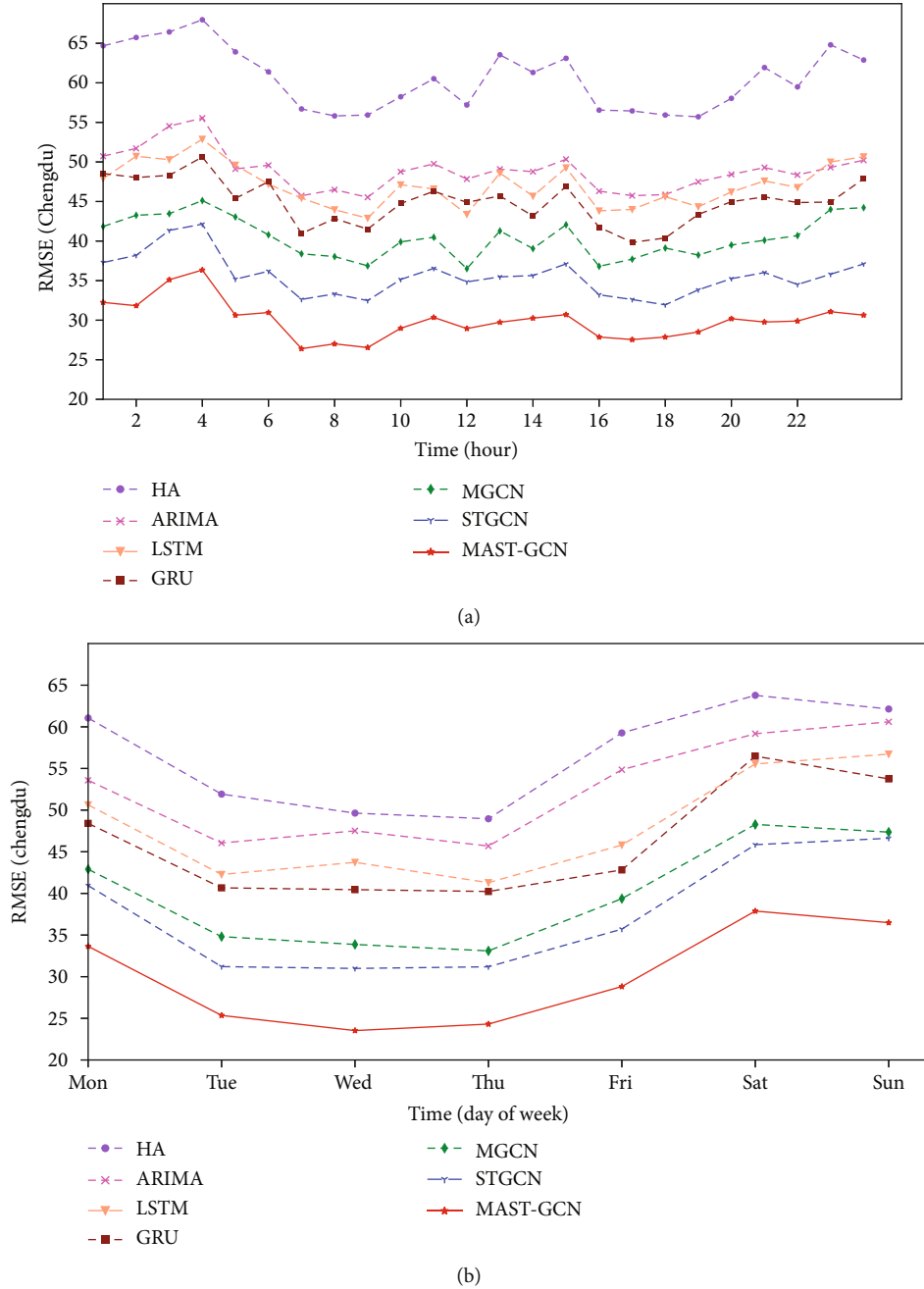


FIGURE 7: Time-based performance: (a) performance across different hours of the day and (b) performance across different days of the week.

state on an adaptive basis. The following is how we compute the temporal attention of each hidden state at the output:

$$\begin{aligned}
 H_{st}^t &= \text{ReLU} \left(\sum_{i=1}^T E_{ij}^t h_i^t \right), \\
 E_i^t &= V_e \tanh (W_{e'} [h_{t-1}; c_{t-1}] + W_e h_t + b_e), \\
 E_{i,j}^t &= \frac{\exp (E_{i,j})}{\sum_{j=1}^T \exp (E_{i,j})},
 \end{aligned} \tag{15}$$

where H_{st}^t represents temporal attention weights are jointly determined by historical states, E_i^t is the weight of the hidden state vector of nodes, and $[h_{t-1}; c_{t-1}]$ is the hidden state and cell state of the previous unit at time $t - 1$. The value of an element $E_{i,j}^t$ in E_i^t semantically represents the correlation strength between time i and time j . where $V_e, b_e \in \mathbb{R}^{m \times 1}$, $W_{e'} \in \mathbb{R}^{m \times 2n}$, and $W_e \in \mathbb{R}^{m \times n}$ are learnable parameters.

4.5.3. Gated Fusion. We use a gated fusion unit to fuse spatiotemporal features by adaptively controlling the effect of spatiotemporal attention at each time slot [32]. As shown

TABLE 3: Experiment results for ablation analysis.

Dataset	Method	RMSE	MAE
Chengdu	MAST-GCN	21.32	15.26
	MAST-GCN-V1	23.32	16.04
	MAST-GCN-V2	24.12	16.31
	MAST-GCN-V3	22.73	15.99
Wuhan	MAST-GCN	23.65	16.31
	MAST-GCN-V1	24.41	17.21
	MAST-GCN-V2	25.96	17.44
	MAST-GCN-V3	25.09	16.92

in Figure 5(a), the hidden features are fused by fusion gate z , including spatial features h_{sa} and temporal features h_{st} .

$$H_{\text{fusion}} = z \odot H_{sa} + (1 - z) \odot H_{st}, \quad (16)$$

$$z = \sigma(H_{sa} W_{z,1} + H_{st} W_{z,2} + b_z), \quad (17)$$

where \odot indicates the element-wise product; $\sigma(\cdot)$ indicates the sigmoid activation; $W_{z,1}$, $W_{z,2}$, and b_z are parameters that can be learned; and z is the fusion gate which is calculated by Equation (17).

5. Experiments

5.1. Datasets and Preprocessing. Multigraph data construction is performed on the data of two real datasets according to Definitions 1, 2, and 3 in Section 3.

5.1.1. Spatial Data. In this paper, the selected datasets are map data of Chengdu and Wuhan cities in China. The study area was set as a rectangular range region based on the distribution of the urban areas and the origin and destination of the ride-hailing trips. The longitude range [104.032, 104.132] and latitude range [30.615, 30.685] are for the Chengdu Urban areas. The longitude range [114.146, 114.475] and latitude range [30.474, 30.737] are for the Wuhan Urban areas. We determined the area size based on the size of the downtown neighborhoods so that the pick-up and drop-off in each area is related to the nearby POIs. According to Definition 1, the Chengdu Urban areas are divided into $40 * 40$ grids and the Wuhan Urban areas are divided into $50 * 50$ grids so that the boarding and alighting in each area are related to the nearby POIs [33].

5.1.2. Ride-Hailing Order Data. The ride-hailing order records use the 2016 Chengdu China dataset and the Wuhan taxi dataset released by Didi Chuxing. There are about 200,000 order records in the Chengdu dataset. Each order record contains seven fields: order ID, order start time, order stop time, pick-up point longitude and latitude, and drop-off point longitude and latitude. Wuhan online taxi dataset contains a dataset of more than 1200 taxis from June 1, 2018, to December 31, 2018. The dataset contains approximately 2.9 million records. The dataset includes taxi ID, location time, longitude, latitude, direction, speed, empty/heavy vehicles, and other information. According to Definition 2, passenger

demand data for a region grid within a period of time can be obtained by summing up order requests for that region within that period of time.

5.1.3. POI Data. We used POI data obtained from the Gaode API, a total of 351,216 POIs items in the entire study region. Each POI data retrieved contains five fields: POI name, longitude, latitude, category, and address. The pick-up regions are highly correlated with the category, quantity and distribution of nearby POIs, and the size of all grids is consistent, so the density of POIs is classified according to the number of POIs [34]. According to the POI classification information in the grid, the POI graph is generated through Definition 3.

5.2. Experimental Setting. The training and test sets for the prediction modelling were partitioned into two sets: the first 23 days of data as the training dataset and the last 7 days of data as the test dataset.

The development computer was equipped with AMD Ryzen 9 3900X, 128G RAM and Nvidia 3090 graphics card. We implemented the MAST-GCN model based on the PyTorch framework. By referring to the experimental parameter settings in [22, 25], we set the ranges of the relevant experimental parameters. The dimensional reference values of the graph convolution module (16, 32, 64), the historical time window reference values (30 min, 60 min, 90 min), the learning rate reference values (0.1, 0.01, 0.001, 0.0001), the dropout reference values (0.1, 0.2, 0.3, 0.4, 0.5), the batch size reference values (16, 32, 64, 128), and optimization are chosen from SGD and Adam. We find the optimal parameters in the validation by implementing a grid search strategy. It is determined that the experimental model consists of a two-layer graph convolution module (dimension of the graph convolution is 64) and a three-layer LSTM (dimension of the LSTM is 128) stacked. The values of the relevant hyperparameters are set as follows: Dropout is set to 0.5, learning rate is set to 0.001, batch size is 64, and Adam is chosen as the optimizer of the model. 60 minutes is used as the historical time window for all tests; i.e., 12 observed data points are used to predict the pick-up region for the next 15, 30, and 60 minutes. In our experiments, we adopt the Mean Square Error (MSE), Rooted Mean Square Error (RMSE), and Mean Absolute Error (MAE) as the metrics to evaluate the performance of all methods.

MSE is the loss function we used, and we train our model by minimizing it as follows:

$$\text{LOSS} = \text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (18)$$

where n is the sum of all samples, \hat{y}_i represents predicted value, and y_i represents ground truth.

Finally, RMSE and MAE were used as model error analysis metrics to evaluate the prediction performance of each

model. The error metrics were calculated as follows:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \\ \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \end{aligned} \quad (19)$$

where n is the sum of all samples, \hat{y}_i represents predicted value, and y_i represents ground truth.

5.3. Baseline Methods. To further demonstrate that the MAST-GCN model presented in this paper is effective, we compare it with the following models:

- (i) HA (Historical average method): we used the average of the last 100 time slices pick-up regions to predict the value of the next time value.
- (ii) ARIMA (Autoregressive Integrated Moving Average) [16]: we used this method for time-series analysis to predict passenger pick-up region; we also use it as the prediction baseline of this paper.
- (iii) LSTM (Long-Short-Term Memory) [35]: because LSTM has more advantages in the learning of long sequences, it is natural to consider using it for the prediction task of this paper and as a comparison.
- (iv) GRU (Gated Recurrent Unit network) [20]: GRU can be regarded as a variant of LSTM in order to solve the problem of gradient disappearance in standard RNN.
- (v) MGCN (Multigraph Convolution Network) [24]: the Euclidean correlations between adjacent regions of the MGCN model space are explicitly modeled using multigraph convolution. This model enhances the ability of the recursive neural network to predict future value through a context attention mechanism.
- (vi) STGCN (spatiotemporal graph convolutional network) [25]: STGCN proposes a new GCN structure composed of spatiotemporal blocks for traffic prediction.

5.4. Comparison with Baselines. Table 2 and Figure 6 show the results of MAST-GCN and other baseline models based on these two datasets. On different datasets, the MAST-GCN model outperforms the other models in both metrics, demonstrating the effectiveness of our model in the ride-area prediction task. Based on Table 2, we summarize three conclusions as follows:

- (1) The relatively low prediction accuracy of traditional statistical forecasting methods such as HA and ARIMA indicates that traditional statistical methods are not well suited to the task of spatiotemporal prediction in non-Euclidean space

- (2) Machine learning-based LSTM and GRU models have better prediction accuracy than traditional statistical models. For example, compared with the HA model for the Chengdu and Wuhan datasets, the GRU model reduces the RMSE by about 16.9% and 17.97% on weekdays and by about 17.12% and 18.35% on weekends, respectively. The performance of LSTM is slightly lower than that of GRU using temporal-dependent
- (3) Among the various baselines, the MAST-GCN model performs best by capturing the spatial characteristics, temporal dependence, and behavioral preferences of the spatial, order, and POI maps simultaneously. For example, compared with the MGCN model based on the Chengdu and Wuhan datasets, there is a reduction of 11.82% and 11.4% in RMSE for weekdays and 12.86% and 11.26% for weekends, respectively. The lower prediction accuracy of the MGCN model is mainly due to the fact that it models only spatial correlation and ignores important temporal features. Compared with the STGCN model, the MAST-GCN model performs about 9.25% and 8.27% reduction in RMSE for weekdays and 9.17% and 8.12% reduction in RMSE for weekends in Chengdu and Wuhan datasets, respectively. The experimental results show that MAST-GCN can capture multigraph hidden spatial correlation, temporal correlation, and behavioural preferences and improve the prediction performance

Figure 6 shows the RMSE and MAE values of the seven different methods for the test data of 7-10, 11-14, and 16-19 on weekdays and weekends on both Chengdu and Wuhan datasets. From the figure, it seems that our proposed MAST-GCN method has the best results in both RMSE and MAE conditions. Specifically, the LSTM and GRU models have better performance compared to the traditional methods of HA and ARIMA. The graph convolutional network model using fused geographic information outperforms the LSTM and GRU models. Further spatiotemporal dependencies are taken into account by STGCN and MGCN deep learning methods. Compared with the above two methods, the MAST-GCN method proposed in this paper achieves better performance by considering temporal and spatial information in addition to the POI information. Specifically, MGCN only embeds static graphs into vectors when extracting global spatial features, which has limited impact on spatial modeling. The STGCN model models non-Euclidean correlations between regions, which proves its rationality and necessity. However, due to the use of traditional GCNs, they only correlate different neighboring regions without considering the importance of regularity in time and space. The MAST-GCN model proposed in this paper considers the temporality and spatiality of online ride-hailing trips and the regularity of passenger trips, proposes a spatial graph modeling approach with multiple graph aggregation, and uses a graph convolutional network model to deal with spatiotemporal dependencies and finally uses a spatiotemporal attention mechanism to assign higher

weights to temporality and preferences, thus achieving better performance.

5.5. Time-Based Performance. In addition, we also evaluate the prediction performance for each hour of the day and each day of the week, as shown in Figures 7(a) and 7(b). Due to space limitation, only the RMSE of the Chengdu dataset is demonstrated here. In Figure 7(a), we can see that the maximum error of these methods occurs in the early morning hours when the demand for ride-hailing is small and passengers are scattered in the boarding locations, which means that the prediction in the pick-up region is more difficult. In the morning and evening peak hours, the demand for ride-hailing is high and the pick-up locations of passengers are mostly in residential areas. In Figure 7(b), we can observe that the error on weekends is higher than that on weekdays, almost 10% worse than that on weekdays. This means that it is more difficult to predict the pick-up region of ride-hailing on weekends. This may be because most people have more fixed places in their pick-up region on weekdays, such as in residential areas and schools, while they have more choices on weekends, resulting in a more dispersed pick-up region, such as commercial areas or short trips to other places.

5.6. Ablation Analysis. In this section, we will conduct further MAST-GCN ablation experiments. The experiments are performed by reducing the relevant modules for comparative analysis of MAST-GCN to measure the performance gain of different modules in MAST-GCN. For this purpose, three comparative versions of MAST-GCN were constructed:

- (1) Remove the graph aggregation module and use only the single graph model of the order graph, and remove spatial dependencies and preferences
- (2) Remove the graph convolution module, and use LSTM for pick-up region prediction
- (3) Remove the spatiotemporal attention module to remove the focus spatial and temporal weights

As shown in Table 3, firstly, we can observe the performance of MAST-GCN-V1 after removing the graph aggregation module, which only uses the order graph for graph convolution operation, the performance of the model has a certain degree of degradation, and we can find that both spatial information and POI information are important in the pick-up region prediction. Secondly, MAST-GCN-V2 removes the GCN module and the performance decreases the most, so we can find that GCN has some advantages in processing spatiotemporal data. Finally, MAST-GCN-V3 removes the spatiotemporal attention module and eliminates the weights given to the focused spatiotemporal data and POI locations and the performance also decreases, indicating that passengers' travel is somewhat spatiotemporally dependent and preferential. The ablation experiment illustrates that each of the submodules in the model has a posi-

tive effect on the improvement of the prediction performance.

6. Conclusion

In this paper, we propose a new multigraph aggregation spatiotemporal graph convolutional network model to predict the pick-up region in online ride-hailing. We design three heterogeneous graphs to model the prediction of online ride-hailing pick-up region: spatial graph, order graph and POI graph. We propose a graph aggregation method to extract the spatiotemporal features and preference features of the three graphs. The network treats the regional grid as the vertices of the graph and combines geospatial data, online ride-hailing order data, and POI data to build a spatial multigraph model. After graph aggregation construction, we adopt the GCN module to capture the spatiotemporal dependencies. We introduce an attention mechanism and assign different weights to different nodes so that the pick-up region of online ride-hailing can be effectively predicted. The experiment on ride-hailing pick-up region prediction shows that our proposed model achieved significantly better results than the state-of-the-art baselines.

Although we have used a variety of heterogeneous data for modeling, the online ride-hailing pick-up region prediction is also influenced by many external factors. Research shows that the number of online ride-hailing orders is affected by such special factors such as emergencies, holidays, and unfavorable weather. In the future, we will add more external factors to the input features of the online ride-hailing pick-up region prediction model to adapt the prediction model to more scenarios and further explore the spatiotemporal dependence of multisource data. What is more, although the model obtains better results on the relevant dataset, there are still other important parameters to be considered in future work, such as the size of the grid division, the classification of POI categories, and the spatial geographic information features. We introduce geographic laws in the model to further optimize the model and make it more interpretable.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

All authors declare that they have no conflict of interest.

Authors' Contributions

Cong Li contributed to the conception of the study and wrote the manuscript; Huiying Zhang contributed significantly to the analysis and manuscript preparation; Yonghao Wu performed the experiment; Zengkai Wang collated the experimental data; Fei Yang visualized the results of the experimental data.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61772386), the Natural Science Foundation of Hubei Province (Grant No. 2020CFB795), and the Zhejiang Provincial Natural Science Foundation of China General Project (Grant No. LY18F020021).

References

- [1] G. D. Erhardt, S. Roy, D. Cooper, B. Sana, M. Chen, and J. Castiglione, "Do transportation network companies decrease or increase congestion?," *Science Advances*, vol. 5, no. 5, article eaau2670, 2019.
- [2] A. Tirachini, "Ride-hailing, travel behaviour and sustainable mobility: an international review," *Transportation*, vol. 47, no. 4, pp. 2011–2047, 2020.
- [3] A. Henao and W. E. Marshall, "The impact of ride-hailing on vehicle miles traveled," *Transportation*, vol. 46, no. 6, pp. 2173–2194, 2019.
- [4] A. Tirachini, E. Chaniotakis, M. Abouelela, and C. Antoniou, "The sustainability of shared mobility: Can a platform for shared rides reduce motorized traffic in cities?," *Transportation Research Part C: Emerging Technologies*, vol. 117, article 102707, 2020.
- [5] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 712–725, 2015.
- [6] B. Qu, W. Yang, G. Cui, and X. Wang, "Profitable taxi travel route recommendation based on big taxi trajectory data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 653–668, 2020.
- [7] X. Wan, H. Ghazzai, and Y. Massoud, "A generic data-driven recommendation system for large-scale regular and ride-hailing taxi services," *Electronics*, vol. 9, no. 4, p. 648, 2020.
- [8] R. Jia, Z. Li, Y. Xia et al., "Urban road traffic condition forecasting based on sparse ride-hailing service data," *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 668–674, 2020.
- [9] J. Zhao, D. Peng, C. Wu et al., "Incorporating semantic similarity with geographic correlation for query-POI relevance learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1270–1277, 2019.
- [10] S. Wang and R. B. Noland, "Variation in ride-hailing trips in Chengdu, China," *Transportation Research Part D: Transport and Environment*, vol. 90, article 102596, 2021.
- [11] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, and J. Han, "PRED: periodic region detection for mobility modeling of social media users," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ACM, pp. 263–272, Cambridge United Kingdom, 2017.
- [12] H. Zhang, L. Zhang, F. Che, J. Jia, and B. Shi, "Revealing urban traffic demand by constructing dynamic networks with taxi trajectory data," *IEEE Access*, vol. 8, pp. 147673–147681, 2020.
- [13] M. Zhang, J. Liu, Y. Liu, Z. Hu, and L. Yi, "Recommending pick-up points for taxi-drivers based on spatio-temporal clustering," in *2012 Second International Conference on Cloud and Green Computing*, pp. 67–72, Xiangtan, Hunan, China, 2012.
- [14] J. Sun, C. Guan, and M. Jin Hong, "Exploiting optimization mechanism for pick-up points recommendations," in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, IEEE, pp. 1180–1184, Dalian, 2017.
- [15] X. Xu, J. Y. Zhou, Y. Liu, Z. Z. Xu, and X. W. Zha, "Taxi-RS: taxi-hunting recommendation system based on taxi GPS data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1716–1727, 2015.
- [16] M. S. Jamil and S. Akbar, "Taxi passenger hotspot prediction using automatic ARIMA model," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, IEEE, pp. 23–28, Bandung, 2017.
- [17] J. Huang, B. Mao, Y. Bai, T. Zhang, and C. Miao, "An integrated fuzzy C-means method for missing data imputation using taxi GPS data," *Sensors*, vol. 20, no. 7, p. 1992, 2020.
- [18] C. Ma, R. He, and W. Zhang, "Path optimization of taxi car-pooling," *PLoS One*, vol. 13, no. 8, article e0203221, 2018.
- [19] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial-temporal network for taxi origin-destination demand prediction," 2019, <http://arxiv.org/abs/1905.06335>.
- [20] U. Vanichrujee, T. Horanont, W. Pattara-atikom, T. Theeramunkong, and T. Shinozaki, "Taxi demand prediction using ensemble model based on RNNs and XGBOOST," in *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*, IEEE, pp. 1–6, Khon Kaen, 2018.
- [21] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 1227–1235, Anchorage AK USA, 2019.
- [22] X. Geng, Y. Li, L. Wang et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3656–3663, 2019.
- [23] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 122, article 102858, 2021.
- [24] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 3529–3536, 2020.
- [25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," 2018, <http://arxiv.org/abs/1709.04875>.
- [26] H. Xu, W. Ding, W. Shen, J. Wang, and Z. Yang, "Deep convolutional recurrent model for region recommendation with spatial and temporal contexts," *Ad Hoc Networks*, vol. 129, article 102545, 2022.
- [27] L. Zhao, Y. Song, C. Zhang et al., "T-GCN: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [28] Z. Xiong, W. H. Jian Li, and H. Wu, "Understanding operation patterns of urban online ride-hailing services: a case study of Xiamen," *Transport Policy*, vol. 101, pp. 100–118, 2021.
- [29] X. Kong, J. Zhang, X. Wei, W. Xing, and W. Lu, "Adaptive spatial-temporal graph attention networks for traffic flow

- forecasting,” *Applied Intelligence*, vol. 52, no. 4, pp. 4300–4316, 2022.
- [30] H. Han, M. Zhang, M. Hou et al., “STGCN: a spatial-temporal aware graph learning method for POI recommendation,” in *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 1052–1057, Sorrento, Italy, 2020.
- [31] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017, <http://arxiv.org/abs/1609.02907>.
- [32] Y. Wu, G. Chen, Z. Li et al., “HSTA: a hierarchical spatio-temporal attention model for trajectory prediction,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11295–11307, 2021.
- [33] Y. Shen, L. Zhao, and J. Fan, “Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces,” *Information*, vol. 6, no. 2, pp. 134–151, 2015.
- [34] B. Zhang, S. Chen, Y. Ma, T. Li, and K. Tang, “Analysis on spatiotemporal urban mobility based on online car-hailing data,” *Journal of Transport Geography*, vol. 82, article 102568, 2020.
- [35] Y. Lai, K. Zhang, J. Lin, F. Yang, and Y. Fan, “Taxi demand prediction with LSTM-based combination model,” in *2019 IEEE Intl Conf on parallel & distributed processing with applications, big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, IEEE, pp. 944–950, Xiamen, China, 2019.