sustainability

Article

Spatiotemporal Prediction of Urban Online Car-Hailing Travel Demand Based on Transformer Network

Shuoben Bi *, Cong Yuan, Shaoli Liu, Luye Wang and Lili Zhang

School of Geographical Sciences, Nanjing University of Information Science & Technology, Nanjing 210044, China
* Correspondence: bishuoben@163.com

Abstract: Online car-hailing has brought convenience to daily travel, whose accurate prediction benefits drivers and helps managers to grasp the characteristics of urban travel, so as to facilitate decisions. Spatiotemporal prediction in the transportation field has usually been based on a recurrent neural network (RNN), which has problems such as lengthy computation and backpropagation. This paper describes a model based on a Transformer, which has shown success in computer vision. The study area is divided into grids, and the structure of travel data is converted into video frames by time period, based on predicted spatiotemporal travel demand. The predictions of the model are closest to the real data in terms of spatial distribution and travel demand when the data are divided into 10 min intervals, and the travel demand in the first two hours is used to predict demand in the next hour. We experimentally compare the proposed model with the three most commonly used spatiotemporal prediction models, and the results show that our model has the best accuracy and training speed.

Keywords: online car-hailing; video frames; spatiotemporal prediction; transformer

Citation: Bi, S.; Yuan, C.; Liu, S.; Wang, L.; Zhang, L. Spatiotemporal Prediction of Urban Online Car-Hailing Travel Demand Based on Transformer Network. *Sustainability* **2022**, *14*, 13568. https://doi.org/10.3390/su142013568

Academic Editor: Marilisa Botte

Received: 6 September 2022
Accepted: 18 October 2022
Published: 20 October 2022

1. Introduction

The global pandemic of the corona virus disease 2019 (COVID-19), as a global public health emergency, has brought major challenges to the entire world. The outbreak of the virus has changed the thinking and lifestyle of residents, and maintaining social distance and avoiding contagion have become a unified perception of residents in public places. It has brought out an unprecedented reduction in public transport demand over the past three years, shifting residents' perceptions of public transport from positive to negative [1]. According to the survey, residents' propensity to use transportation has changed significantly after the COVID-19 outbreak. Public transport has been the most affected, with a remarkable decrease in the number of users, while the use of private cars has increased. However, for people who do not have a private car and do not want to use public transportation, shared mobility is a good option in many cities. Online car-hailing is a kind of shared mobility, which is favored by people for its convenient and reliable service. As the best alternative to private car travel and an important supplement to public transportation, it has gradually become one of the important travel modes [2]. At the same time, making full use of shared travel resources can alleviate various traffic pressures. The online car-hailing platform collects and matches passenger orders and service vehicles in a new way for service sources, creating conditions for the provision of large-scale transportation services [3]. It is helpful to build an intelligent, green, efficient, and safe integrated transportation system and promote the sustainable development of urban transportation [4,5].

According to the national online car-hailing regulatory information interaction platform [6], 263 online car-hailing platform companies are licensed in China, and 4.053 million online car-hailing driving licenses have been issued. Online car-hailing faces several problems. Residents cannot quickly access car-hailing services, and drivers run empty for long times without orders, which wastes energy resources and increases traffic congestion. Therefore, it is of great significance to accurately predict the time and space of online car-hailing travel demand. This can help residents to better understand demand according to regions and times, enabling better decisions and improved travel efficiency. Reasonable scheduling of vehicles and the timely fulfilment of travel needs can reduce the waste of road resources and ease traffic congestion, energy waste, and pollution. The data used in this article come from the Gaiya data plan (where personal information is anonymized) of Didi Taxi. Online car-hailing platforms have almost uniform operation characteristics. Customers manually input the longitude and latitude of the starting and ending points. Orders include information such as an order number and times of boarding and alighting, which provides a basis for the spatiotemporal prediction of online car-hailing travel.

Online car-hailing travel data are spatiotemporal, with obvious periodicity, enabling the prediction of travel demand. Past travel research initially used historical average data for prediction, and focused on time series prediction, with typical models such as Autoregressive Integrated Moving Average Model (ARIMA) and its variants [7,8]. Traditional methods use linear mathematical representation to find the internal characteristics of traffic flow, but this does not apply to online car-hailing, which has more complex nonlinear spatiotemporal sequence data. Machine learning and deep learning models, such as neural networks, perceptrons, and support vector machine, have been used in traffic prediction [9], and perform better and are more accurate than traditional models [10]. Deep learning has formed the basis of many excellent network models in the field of transportation. Convolutional neural networks (CNNs) [11] and recurrent neural networks (RNNs) [12] are the basic structures for traffic prediction research. However, CNNs have a weak ability to capture long-term time dependence, while RNNs require intensive computation. Previous studies considered online car-hailing data as one-dimensional, which has a weak ability to capture spatiotemporal features.

Several models have been proposed to capture the spatiotemporal characteristics of network car trips. Convolutional long short-term memory (ConvLSTM) [13] was originally used for precipitation prediction, and is widely used in transportation due to its spatiotemporal prediction. To predict urban short-term traffic flow, Chen et al. [14] designed a prediction model based on ConvLSTM. Zheng et al. [15] introduced an attention module to a ConvLSTM model that can extract spatial and short-term temporal features to improve prediction accuracy. However, ConvLSTM lacks the characteristics of parallel operation, so training time is slow, and the network lacks long-term dependence on sequences.

Among the above methods, previous studies have treated the information of online car-hailing demand as one-dimensional data, and the ability to capture its spatiotemporal characteristics is weak. Although the emergence of ConvLSTM can fill this gap, due to its lack of parallel operation characteristics, its training time is slow, and there is also a lack of long-term dependence on sequences. In order to overcome the above shortcomings, this paper converts the online car-hailing order data into video frame data containing spatial information and time series, and proposes a predictive Spatiotemporal Transformer (SPTformer) model based on the architecture of the Transformer model, so as to better predict the demand for online car-hailing travel from the temporal and spatial scale. From this perspective, this study also converts the demand forecast problem of online car-hailing into an image extrapolation problem. In recent years, deep learning technology has improved the technical level in many fields with its good performance, especially in computer vision [16]. The Transformer model was originally developed by Vaswani et al. for natural language processing [17], and was widely used in computer vision [18–22]. SPTformer utilizes the encoding part of the Transformer network. At the same time, in order to consider the temporal correlation between sequences, we inject information about the relative or absolute position of the image sequence into the input data by adding position coding, and introduce a 3D convolutional layer (Conv3D) and scaled dot product attention in the self-attention calculation part to capture the short-term and long-term dependencies between sequences. Like the Transformer model, our model can run in parallel, maintaining good performance while speeding up training. As far as we know, this is the first time online car-hailing order data have been converted into video frame data, and that the Transformer network has been used to predict the travel demand of online car-hailing in time and space. The contributions of this paper are as follows:

(1) Based on the Transformer architecture, a spatiotemporal prediction SPTformer model is proposed, and the experimental results prove that our model is competitive for the spatiotemporal prediction of residents' online car-hailing travel.

(2) After the online car-hailing order data are processed into video frame data, they still contain the spatiotemporal information of the online car-hailing trip data, so the model can better predict the online car-hailing demand on the spatiotemporal scale.

The remainder of this paper is organized as follows: in Section 2, we briefly review the existing solutions and models for solving traffic prediction problems. In Section 3, the structure of the model in this paper is described, and each part of the model is explained in detail. At the same time, the special processing method and data structure of online car-hailing data are introduced. In Section 4, extensive experiments are described using Haikou online car-hailing order data, and the existing spatiotemporal prediction methods in the transportation field are used as a comparison to test the effect of the proposed model. The results of the projections are discussed at the end. In Section 5, we summarize this research and discuss further work.

2. Related Work

Deep learning has found application in various fields, and most transportation research depends on it. This paper addresses the spatiotemporal prediction of online car-hailing travel demand.

The CNN and RNN are two basic models in traffic prediction research. In order to better consider the spatiotemporal correlation of ride-hailing travel demand, Zhang et al. proposed an end-to-end multi-task learning beat convolutional neural network (MTL-TCNN), which predicts short-term passenger demand at a multi-regional level based on Didi Chuxing's ride-hailing data in Chengdu, China and taxi data in New York City [23]. To predict short-term traffic flow, Zhang et al. [24] designed a model based on a CNN, with higher accuracy than the traditional model. Chen et al. [25] proposed PCNN, which is based on deep CNNs and can model periodic traffic data and predict short-term traffic congestion. Mou et al. [26] proposed a temporal information augmented LSTM (T-LSTM) model to predict the traffic flow of a single road segment, which can capture the intrinsic correlation between traffic flow and temporal information, thereby improving prediction accuracy. The prediction of traffic flow during peak hours is of great significance to alleviate traffic pressure. Yu et al. [27] designed a traffic flow prediction model based on long short-term memory (LSTM) to predict traffic flow in urban peak hours. Tang et al. [28] proposed ST-LSTM, which extracts spatiotemporal features from data and combines them as input. Gu et al. [29] combined an LSTM neural network and gated recurrent unit (GRU) in a two-layer deep learning model that outperformed a single network model. However, ordinary CNNs weakly capture long-term temporal dependencies, and RNNs do not capture spatial dependencies well, which motivates their combination.

A CNN can capture the spatial basis of traffic flow, while an RNN can mine short-term changes and periodicity. Wu et al. [30] combined these in a deep learning framework, CLTFP, for the spatiotemporal prediction of traffic flow. Similarly, Zhen et al. [31] used a CNN to extract traffic spatial features, and an RNN to predict traffic flow changes. Liu et al. [32] designed a ConvLSTM model based on a CNN, which can extract the spatiotemporal features of traffic flow, and has an end-to-end deep learning architecture. To consider the temporal and spatial characteristics of traffic flow and extract the temporal and spatial correlation and variation law of traffic flow data, Li et al. [33] combined a CNN and bidirectional LSTM (BiLSTM) in Conv-BiLSTM. To extract the spatiotemporal correlation of data from historical traffic data, He et al. [34] designed a spatiotemporal CNN (STCNN) based on a model of convolutional LSTM cells. Wang et al. [35] designed a traffic demand prediction model based on deep spatiotemporal ConvLSTM, which was experimentally shown to outperform traditional models in both accuracy and speed. Huang et al. [36] designed a ConvLSTM-Inception network (CL-IncNet) to make spatiotemporal predictions of traffic flow data. Li et al. [37] constructed a ConvLSTM network to predict taxi demand, which was shown to more accurately process spatial information. Chen et al. [38] proposed a BT-ConvLSTM model to introduce temporal information to a ConvLSTM network, and it was experimentally shown to improve traffic flow prediction accuracy. Di et al. [39] proposed CPM-ConvLSTM, a spatiotemporal model to make short-term predictions of the congestion levels of road segments. To reduce resource requirements, Huang et al. [40] built a sparse convolutional recurrent network utilizing sparse gates in ConvLSTM and ConvGRU. Ranawaka et al. [41] used a ConvLSTM model with Google traffic data to predict traffic flow in the next 20, 30, and 60 min. Although the combination of CNN and RNN can capture the spatiotemporal features of traffic data, they are computationally expensive and slow to train, due to the sequential nature of the recurrent structure.

This study uses a Transformer network to construct a prediction model. Compared with RNN-based methods, Transformer can effectively capture long-term dependencies, can be operated in parallel, has good performance and a fast training speed, and can capture the correlation of each part of an image through self-attention well. In order to consider the different spatial relationships between variables, Grigsby et al. [42] proposed a method called Spacetimeformer, which achieved good results in the field of spatiotemporal prediction. Xu et al. [43] proposed a new paradigm of spatiotemporal transformer networks (STTNs), which exploits dynamic directional spatial dependencies and long-term temporal dependencies to improve the accuracy of long-term traffic flow prediction, and their model performs well in long-term traffic flow prediction. Song et al. [44] proposed a model named TSTNet based on the Transformer architecture, which is a sequence-to-sequence (Seq2Seq) spatiotemporal traffic prediction model, which can be used for urban traffic spatiotemporal flow prediction. Zhang et al. [45] used the Transformer network to propose a novel architecture called a time-fusion transformer (TFT), which can predict short-term highway speeds, which has been experimentally shown to have high accuracy. Cai et al. [46] referred to Google's Transformer machine translation framework to design a network called a Traffic Transformer network that captures the continuity and periodicity of traffic flow time series and models spatial correlations. Girdhar et al. [47] designed an anticipatory video transformer (AVT) based on a Transformer network to predict actions, with an attention module and an end-to-end model architecture. In order to make accurate predictions of autonomous driving trajectories, Zhang et al. [48] designed a Gatformer model based on transformer architecture, which can make more accurate predictions while shortening the forecasting time. Wu et al. [49] proposed an object-centric video transformer (OCVT) to predict video frames, decomposing a scene into tokens suitable to generate video transformers. Farazi et al. [50] designed an end-to-end learnable model, the frequency domain transformer network (FDTN), which can estimate and use signal transforms in the frequency domain. Wang et al. [51] designed a concise and efficient temporal Transformer network with progressive prediction, aggregating observed features, and a lightweight architecture to progressively predict features. Liu et al. [52] proposed a ConvTransformer network for video frame sequence learning and synthesis. Shi et al. [53] proposed Transformer-based video interpolation framework self-attention to compute long-term dependencies. Zheng et al. [54] designed a pure Transformer-based network to predict the next step for a 3D human pose in a video. Tai et al. [55] designed higher-order self-attention, and proposed a higher-order recursive layer design, HORST. Farazi et al. [56] introduced a transformer model that enables local predictions with selectable sparsity.

The Transformer network has achieved great success in computer vision, and provides a theoretical basis for our research, since traffic data are spatiotemporal. To better predict online car-hailing demand, we convert order data into video data with spatiotemporal characteristics according to a fixed time period. To predict video data, we propose a model called the Spatiotemporal Convolution Transformer (SPTformer) based on the architecture of a Transformer. Experiments show that the model is suitable for the research of spatiotemporal prediction, and it performs well.

3. Methodology

3.1. Overview

To predict the time and space of urban car-hailing trips, this study refers to the prediction of video data. We preprocess the data, convert them into frames of video data, and use a video prediction method to generate future frames. The historical travel demand sequence, x $= \{x_0, x_1, \cdots x_n\} \in R^{H\times W\times C}$, has sequence length n, and h, W, and C are the height, width, and number of channels, respectively. Our goal is to use the m frames of sequence data before time t, $\hat{x} = \{\hat{x}_{t-m+1}, \hat{x}_{t-m}, \cdots, \hat{x}_t\}$, to predict the sequence data of k frames after time t, $\hat{x} = \{\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+k}\}$.

This paper proposes a Transformer network, SPTformer. A feature embedding module embeds the input historical sequences, x $= \{x_0, x_1, \cdots x_n\} \in R^{H\times W\times C}$, capturing rough short-term spatial dependencies. After adding batches, the input data are five-dimensional data. The position coding module adds position coding to the feature embedded historical sequence, and the frame feature map after adding position coding is transmitted to the Encoder as input. The space–time dependence between frames in the historical sequence is extracted by a self-attention mechanism and convolution. Linear transformation is used to predict and generate future frames, $\hat{x} = \{\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+k}\} \in R^{H\times W\times C}$, so as to complete the prediction of the demand for online car-hailing in time and space. Figure 1 shows the framework of the model.

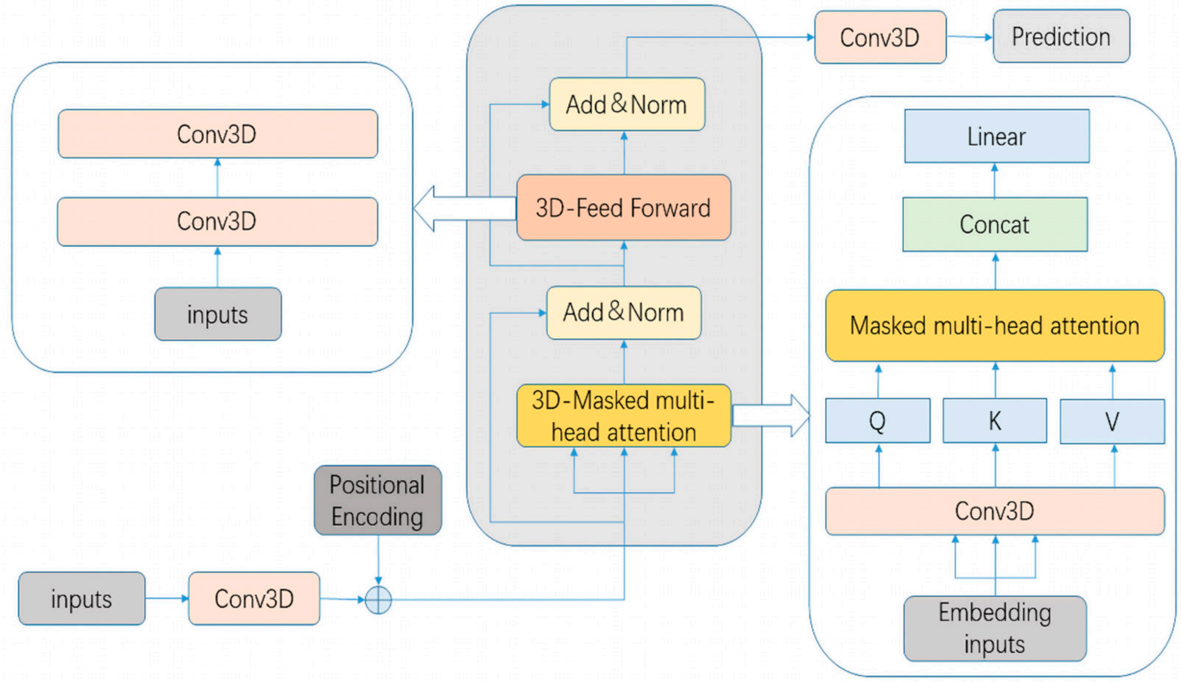


Figure 1. Framework of SPTformer model.

3.2. Data Conversion

People using an online car-hailing platform must input information such as the departure point and destination, which provides a basis for the spatiotemporal prediction of travel. The proposed prediction method converts the data into two-dimensional pictures, and divides them according to fixed time intervals to calculate trips. A fixed number of frames constitutes the experimental data. The city is divided into equal-sized grids, with fixed numbers of rows and columns. Each grid represents a small traffic area, and its order quantity represents the travel demand in that area. The travel data used in this paper cover central Haikou City, in the form

$$D_t = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1j} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2j} \\ \vdots & & & & \vdots \\ X_{i1} & X_{i2} & X_{i3} & \dots & X_{ij} \end{bmatrix}, \tag{1}$$

where D_t represents the demand at time t, and X_{ij} represents the demand at grid coordinates (i, j).

3.3. Model Structure

3.3.1. Spatial Embedding

The data must be spatially encoded before being fed into the decoding block, for which this study adopts the 3D convolution layer of the ReLU activation function, and the number of convolution kernels is expressed as *d_model*. After the data pass through the convolutional layer, their representative features can be extracted, so that the network can learn more effectively. The image frame that needs to be encoded is represented as $x_i^{H \times W \times C}$, the spatially encoded data can be represented as $F_i^{H \times W \times d\_model}$, and their relationship is

$$F_i = \sigma(W_s * x_i),\ i \in [1, n], \tag{2}$$

where W_s is the convolution kernel, "$*$" represents convolution, and σ represents activation.

3.3.2. Positional Encoding

Since the model has no recursive process, our data are spatiotemporal sequence data containing time. For the model to capture the sequential relationship between time series during training, the spatially encoded sequence data must be injected with information about the relative or absolute position of the image sequence. Therefore, before entering the coding block, a layer is added with position coding,

$$\mathrm{Pos}_{\mathrm{Enc}p,h,w,2i} = \sin\left(\frac{p}{10,000^{\frac{2i}{d_{model}}}}\right), \tag{3}$$

$$\mathrm{Pos}_{\mathrm{Enc}p,h,w,2i+1} = \cos\left(\frac{p}{10,000^{\frac{2i}{d_{model}}}}\right), \tag{4}$$

which is computed using the sine and cosine functions of different frequencies. Pos_Enc is the calculated position code, p is the absolute position of the video frame in the sequence, h is the image height, w is the width, i represents the channel dimension, and *d_model* is the number of channel dimensions. The calculation result is then added, element by element, to the spatially encoded video frame data,

$$M_i = F_i \oplus \mathrm{Pos}_{\mathrm{Enc}i},\ i \in [1, n], \tag{5}$$

where "$\oplus$" indicates that the tensor element corresponds to addition.

3.3.3. Encoder Layer

The coding block consists of 3D-Masked multi-head attention, 3D-Feedforward, and Add-Normalize. The 3D-Masked multi-head attention is used as the main body to calculate the spatiotemporal relationship between historical frame sequences. The 3D-Feedforward has two Conv3D layers, which can better capture short-term dependencies between time series. Add-Normalize has residual and normalization layers, which can speed up training and improve stability. Its structure is shown in the decoding block in Figure 1.

Scaled Dot product Attention [17]: The query Q, key K, and value V matrices $\left(Q, K, V \in R^{H \times W \times d_{model}}\right)$ must be calculated from the input data. We compare all keys with their queried representations, and if the query and keys are similar, the corresponding values are assumed to be related. For each Q vector, the attention weight for each value V is computed by taking the dot product of Q with every other K vector. To prevent the instability of gradient calculation when *d_model* is too large and the dot product becomes large, we divide these dot products or attention weights by $\sqrt{\text{dk}}$ (where dk is the channel dimension of the key vector). To avoid seeing future information, model training only relies on the sequence before this time, and cannot rely on the sequence after it. A masking method (Mask) is inserted in the attention calculation process. The results are normalized by the softmax function, and the attention weights of all V vectors are weighted and summed to obtain the final output of scaled dot product attention,

$$\text{Att} = Softmax\left(Mask\left(\frac{QK^T}{\sqrt{dk}}\right)\right)V. \tag{6}$$

A Conv3D layer is used in the calculation of Q, K, and V. Compared with a linear transformation, 3D convolution can capture the short-term correlation of the sequence and extract some features. The formula is

$$Q = \sigma(W_q * M),\ K = \sigma(W_k * M),\ V = \sigma(W_V * M), \tag{7}$$

where W represents the convolution kernel, and "$*$" represents convolution.

Mask Multi-head Self-attention Layer: According to the characteristics of multi-head attention, multiple sets of Q, K, V are used to calculate the attention. Encoded data are calculated by multiple linear projections; multiple groups of Q, K, and V are spliced; and groups calculate attention in parallel, for a better model effect than a set of linear projection methods. Our model uses this feature to build multi-head attention with masks. Information in different positions of the video frame is jointly modeled with multiple heads, each computing scalar dot product attention in parallel, and masks prevent future information leakage. The formula is

$$\text{MultiHead}(Q,\ K,\ V) = \text{Concat}(h_1,\ h_2 \ldots,\ h_n)W, \tag{8}$$

$$\text{where } h_i = \ \text{Attention}(Q_i,\ K_i, V_i),\ i \in [1,\ \ldots,\ n]. \tag{9}$$

This paper use Conv3D instead of linear projection to calculate Q, K, and V, so W is the convolution kernel.

Add-Normalize: This layer consists of residual and normalization layers. To alleviate the problem of gradient disappearance, which leads to degradation in the training of the deep neural network, the encoded data M and the calculated attention data $\hat{M}$ are used to construct the residual layer. At the same time, in order to improve the generalization ability of the model, we use the batch normalization (BN) layer. Batch normalization reduces overfitting by reducing internal covariance drift [57]. It not only improves the training speed, but also speeds up the convergence process. It is also a regularization expression similar to Dropout to prevent over-fitting, and can achieve the same effect as Dropout. The formula is

$$A = BN(\hat{M} \oplus M). \tag{10}$$

3D-Feedforward: Compared with fully connected neural networks, CNNs can successfully capture spatial information in images due to the reduction of parameters and reusable weights. To better capture dependencies between video frame features and timing, this paper describes two Conv3D layers to construct a fully connected layer,

$$f = W_2 * (\sigma(W_1 * A)), \tag{11}$$

where W_1 and W_2 represent the convolution kernel, and A is the output of the normalization layer.

Above is the entire content of the encoding block. When the data added with position coding enter the coding block, the self-attention of the data is first calculated, and the calculated data are then transferred to the Add-Normalize layer to improve the robustness of the model. Next, the data enter the 3D-Feedforward layer to assist in capturing short-term sequence correlation, and are exported after Add-Normalize to complete the Encoder layer block calculation. The entire Encoder layer part of the framework is as follows:

$$Q, K, V = \sigma(W * M), \tag{12}$$

$$\hat{M} = MultiHead(Q, K, V), \tag{13}$$

$$A = BN\left(\hat{M} \oplus M\right), \tag{14}$$

$$f = W_{f2} * \left(\sigma\left(W_{f1} * A\right)\right), \tag{15}$$

$$F = BN(f \oplus A). \tag{16}$$

3.3.4. Prediction Layer

The final prediction layer is a Conv3D layer. To maintain consistent numbers of channels of output and input data, its convolution kernel is set to 1 and the padding is set, so the output and input are of the same order. The formula is

$$X = W_p * F, \tag{17}$$

where W_p represents the convolution kernel, and F is the output of the encoding block.

3.3.5. Optimizer

Mean Squared Error: This paper describes the mean square error as the prediction loss,

$$Loss = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2, \tag{18}$$

where y_i is the real value, $\hat{y}_i$ is the predicted value, and m is the number of predicted frames.

RMSprop: This study chooses the RMSprop algorithm, which can obtain the exponentially weighted moving average of past squared gradient values, as the optimizer, to reduce the swing amplitude of the loss function, for faster convergence.

We analyze each module. During training, this study uses $\{\hat{x}_{t-m+1}, \hat{x}_{t-m}, \cdots, \hat{x}_t\}$ as input and $\{\hat{x}_{t-m+2}, \hat{x}_{t-m+1}, \cdots, \hat{x}_{t+1}\}$ as a prediction label, calculates the training loss from the prediction result, and uses this to adjust the model parameters. To predict the future k frames of sequence data, we use $\{\hat{x}_{i-m+1}, \hat{x}_{t-m}, \cdots, \hat{x}_i\}$ to generate a frame of video $\hat{x}_{i+1}$, add $\hat{x}_{i+1}$ to the new input frame sequence $\{\hat{x}_{i-m}, \hat{x}_{t-m}, \cdots, \hat{x}_{i+1}\}$ as the input of the next round of prediction, and repeat the process to obtain the k-frame sequence $\hat{x} = \{\hat{x}_{t+1}, \hat{x}_{t+2}, \cdots, \hat{x}_{t+k}\}$ after time t. This study uses the test set to test the model results when the training is optimal.

4. Experimental Setup and Analysis

4.1. Study Area and Data

4.1.1. Overview of Study Area

This study takes Haikou City, the capital city of Hainan Province and the central city of the Beibu Gulf urban agglomeration, as an example. It is located from 19°31′–20°04′ north latitude and 110°07′–110°42′ east longitude. It is the political, economic, technological, and cultural center of Hainan Province, and is its largest transportation hub. It is the fulcrum city of China's "One Belt, One Road" strategy [58]. However, Haikou has some problems in transportation, especially in the aspect of public transport development. Compared with the evaluation indicators of public transport in China, Haikou has fewer public transport lines, and the number of buses per 10,000 people is lower than the national standard. Buses are mainly concentrated on trunk roads and their lines are unevenly distributed. In addition, the time interval between bus lines is long, which makes few citizens choose to take the bus. As an important supplement to public transportation, online car-hailing is very popular among residents [59]. In 2012, Didi Chuxing, Shenzhou, Yidao, and other online car-hailing companies began to operate in Haikou City. By the end of 2016, the number of online car-hailing vehicles in Haikou had reached 10,000, including 6000 legal car-hailing drivers [60]. Figure 2 shows an overview of the study area.

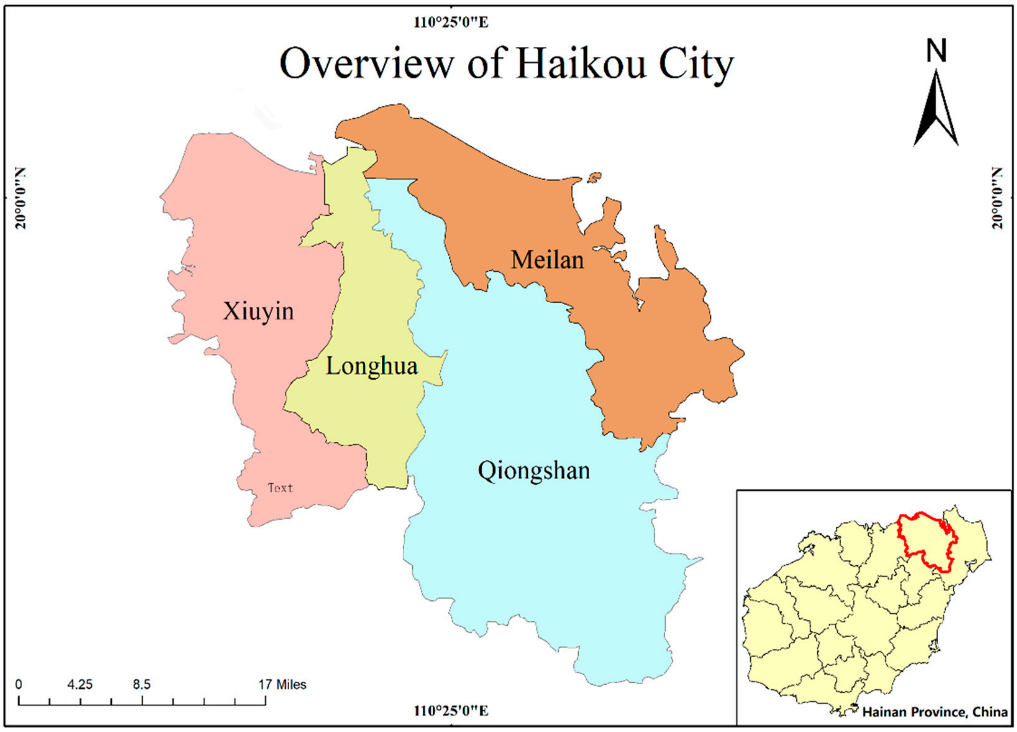


Figure 2. Overview of research area. The map comes from the standard map service published on the website of the Ministry of Natural Resources of China (http://bzdt.ch.mnr.gov.cn/ (accessed on 1 June 2022)).

4.1.2. Ride-Hailing Data

The online car-hailing order data used in this study come from the travel dataset published by Didi Chuxing's Gaia data open plan [61]. This study selected the daily order data of Haikou from 1 May to 31 October 2017, including order ID, order time, order type, traffic type, number of passengers, estimated road distance between departure and destination, arrival time, estimated price, duration, primary business line, and longitude and latitude of the destination and starting point. Personal information was anonymized, and did not affect the research.

4.2. Data Preprocessing

Map data: The online car-hailing data cover the downtown area of Haikou. Therefore, the main study areas of this paper are Jinmao Street, Zhongshan Street, Jinyu Street, Guoxing Street, Heping South Street, Haifu Street, Haixiu Street, Xiuying Street, haixiu Town, Datong Street, Chengxi Town, Haiken Street, Binhai Street, Bailong Street, Lantian Street, Boai Street, Binjiang Street, Fucheng Street, Fengxiang Street, Haidian Street, Renmin Road Street, Xinbu Town, and Baisha Street, covering 19°89′–20°12′ north latitude and 110°10′–110°63′ east longitude.

Research area division: To facilitate future modeling, the area should be divided into small research units in order to avoid the complexity of map matching in large-scale network demand forecasting. We divided the study area into multiple grids for faster analysis. Since the point data are aggregated, the results will be affected by the size or method of grid division. The size of the grid should be comprehensively considered and determined according to actual needs. When selecting a small-scale grid, the travel demand in each grid is low, the network complexity is high, and the actual operation is difficult, but the small-scale grid division describes the demand more finely in terms of spatial granularity. Although the computational complexity is reduced when large-scale grids are selected, the accuracy of large-scale grid description is poor. In this article, the study area is initially divided into 60 × 60 grids, i.e., 0.09 km^2, and the travel demand of each grid according to the time scale is calculated.

Online car-hailing data: The order data are found to have missing information and abnormal problems, such as a missing order ID or a null estimated distance. Invalid information in the historical order data is deleted, including city ID, city area code, secondary district and county, driver sub product line, estimated road distance between departure and destination, estimated price, duration, and primary business line. IDs are randomly generated for missing IDs. Orders whose origin and destination latitude and longitude are outside the scope of the study area are deleted. Only order data appearing for the first time are retained in the case of duplication. The original data include 14,160,162 order data, and 11,255,140 order data are retained after data cleaning.

4.3. Experimental Data Construction

Online car-hailing order data must be converted into video frame data to construct a spatiotemporal matrix, i.e., into different grayscale images according to time periods. Each pixel represents a study area, whose grayscale represents its travel volume. Video frames are combined to construct a video frame dataset, which is imported into the model for spatiotemporal prediction. The method is as follows. The five-month data are arranged before and after the event, and the time scale of historical variables is divided. For more accurate prediction, we divide time slices for research to judge the influence of time divisions on the experimental results, and select the best method. To facilitate data segmentation, we divide the data into time slices of 10, 15, 20, and 30 min. It is then necessary to calculate the number of online car-hailing trips in each grid in each time slice. This study uses the latitude and longitude of each order, determines which grid the order falls in, and records the travel demand in the grid study area. Video frame data of each time slice are obtained, and are arranged in order of time slices, to obtain 26,496 frames of images for 10 min, 17,664 for 15 min, 13,284 for 20 min, and 8832 frames for 30 min. To control the variables, all models use the data of the first two hours to predict the data of the next hour. For three hours of data, the data of the first two hours are for training, and the data of the last hour are used as the label of the spatiotemporal prediction result to calculate the loss and accuracy rate. Image data for periods of 10, 15, 20, and 30 min are processed into video data with 18, 12, 9, and 6 frames, respectively. The first 1400 frames are used as experimental data, with 80% for training, 10% for validation, and 10% for testing. Figure 3 shows the converted data.

4.4. Evaluation Indicators

This study evaluates the quality of model prediction by mean absolute error (MAE) and root mean square error (RMSE),

$$\text{MAE} = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{1}{m}\sum_{i=1}^{m}\left|\hat{y}_k^i - y_k^i\right|\right), \tag{19}$$

$$\text{RMSE} = \frac{1}{n}\sum_{k=1}^{n}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_k^i - y_k^i\right)^2}, \tag{20}$$

where n is the number of predicted frames, k denotes the frame, m is the number of video frame grids, i denotes the research area, $\hat{y}_k^i$ is the predicted value, and y_k^i is the real value. MAE is the real error, which can intuitively reflect the average difference between the predicted and actual values. A lower MAE indicates a more accurate prediction. RMSE reflects the difference between the predicted and real data, magnifying larger errors, and it reflects the maximum error. A smaller RMSE indicates a better prediction result. When calculating MAE and RMSE, y and $\hat{y}$ are gray values. The prediction accuracy can be obtained by calculating each grid of each frame.

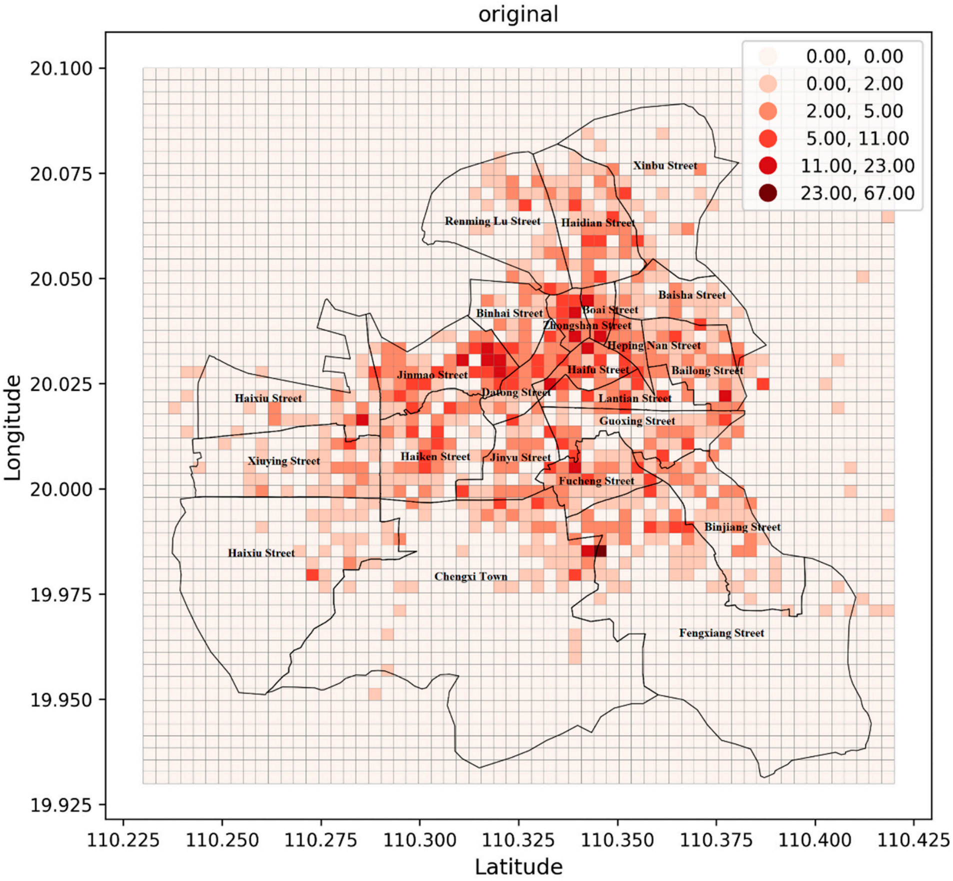


Figure 3. Conversion of online car-hailing order data into video frame data.

4.5. Experimental Analysis

4.5.1. SPTformer

The SPTformer model, as shown in Figure 1, includes a spatiotemporal embedding layer, which encodes and adds locations to the data; an Encoder layer for computing self-attention; and an output layer, which consists of a Conv3D layer that performs the final prediction. The Encoder consists of two Encoder layers in series.

This study uses the video frame datasets divided by different time slices to compare the impact of time division methods on the prediction results.We take the first 1400 data points as experimental data; construct training, validation, and test sets in an 8:1:1 ratio; and select the optimal model parameters after experiments. We set the number of models to 16, the number of attention heads to 4, the size of the convolution kernel of the spatial embedding convolutional layer to (3, 3, 3), the number of convolution kernels to 16, and use a ReLU activation function. In the Encoder block, the convolutional layer settings calculated by Q, K, V are the same. The 3D-Feedforward has two Conv3D layers, with 32 and 16 convolution kernels in the first and second layer, respectively, of size (3, 3, 3), using ReLU activation. The output layer is a Conv3D layer with one convolution kernel of size (1, 1, 1). This study uses MSE to calculate the model training loss, optimized with

RMSprop, with a 0.001 learning rate and 0.9 decay rate, and uses temporal backpropagation feedback to adjust the model. During training, when the number of model iterations reaches a certain level, the loss and accuracy rates change slowly, and the model reaches the optimum, so we set the number of model iterations to 50.

4.5.2. Compared Models

This study adopts a convolutional LSTM network as our baseline model, and utilizes ConvGRU and Self-Attention ConvLSTM (SaConvLSTM) models for comparison.

The ConvLSTM neural network was first used to solve the problem of precipitation nowcasting. This structure can establish temporal relationships between two-dimensional plane data and extract spatial relationships like a CNN [62]. Its principle is similar to that of an LSTM network. There are also forgetting, input, and output gates, but the difference is the addition of convolution between the input and each gate. ConvLSTM has been widely used in the research of spatiotemporal prediction. The formula is as follows:

$$f_t = \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \otimes C_{t-1} + b_f\right), \tag{21}$$

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \otimes C_{t-1} + b_i), \tag{22}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \tag{23}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \otimes C_{t-1} + b_o), \tag{24}$$

$$H_t = o_t \otimes tanh(C_t), \tag{25}$$

where the X_t, H_t, C_t, i_t, f_t, and o_t are all converted from two-dimensional to three-dimensional tensors. Two dimensions represent the rows and columns of the network in which the grid is located, and the other dimension represents the number of features in each grid; i_t, f_t, and o_t represent input, oblivion, and output gates. X_t represents the input of the network at t-time, H_t represents output at t-time, and C_t represents the cellular state at t-time; W and b represent the weights and biases for each gate, respectively. However, W acts like a convolutional kernel, and "$*$" represents convolutional operations; "$\otimes$" represents the Hadamard product as in LSTM.

Since LSTM is slower to train, GRU has made slight modifications to increase the speed. Inspired by this, the LSTM was replaced by a GRU, and the ConvGRU model was proposed. Like the ConvLSTM model, ConvGRU changes the operation between the input and each gate to convolution, and it can perform spatiotemporal prediction. Unlike ConvLSTM, ConvGRU converts LSTM into GRU for computation. Yu et al. [63] found that ConvGRU is faster and has better spatiotemporal prediction results.

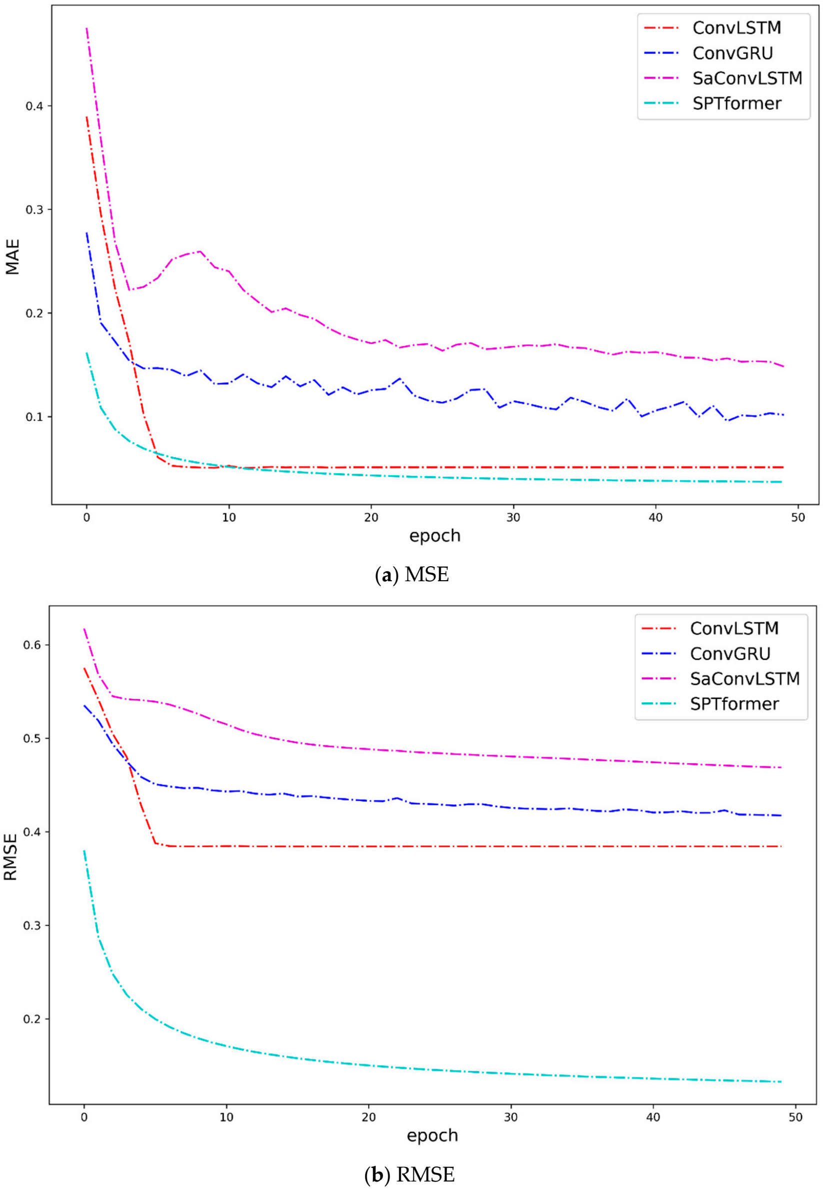
Lin et al. [64] found that SaConvLSTM relies on convolutional layers to capture spatial dependencies, which is locally inefficient, and introduces self-attention to extract spatial features with global and local dependencies and capture features with long-term dependencies in the spatial and temporal domains. Experimental results show that the method achieves better prediction results, with fewer parameters and higher efficiency.

4.5.3. Results

This paper compared the effect of the proposed model to that of other models on the same dataset, using the data of the first two hours to predict the data of the next hour. We used datasets constructed by division in different time periods, taking the first 1400 data points of each part as the experimental data, and constructing training, validation, and test sets, with results as shown in Table 1, from which it can be seen that our model has the highest prediction accuracy on all constructed datasets, with increasing accuracy with finer time scale divisions. Our model has the lowest RMSE and MAE when the dataset is constructed with 10 min time periods, and they gradually decrease with finer time periods. To observe the model training process, we visualized the changes in the accuracy of each model as it was trained. Figure 4 shows the changes in the MAE and RMSE.

Table 1. Prediction results of SPTformer, ConvLSTM, ConvGRU, and SaConvLSTM with different divided data.

	Training 10 min		Training 15 min		Training 20 min		Training 30 min	
MODEL	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
SPTformer	0.038	0.133	0.056	0.194	0.085	0.272	0.160	0.422
ConvLSTM	0.051	0.384	0.093	0.583	0.140	0.787	0.244	1.187
ConvGRU	0.103	0.416	0.163	0.619	0.208	0.812	0.303	1.202
SaConvLSTM	0.157	0.481	0.198	0.650	0.267	0.846	0.357	1.226

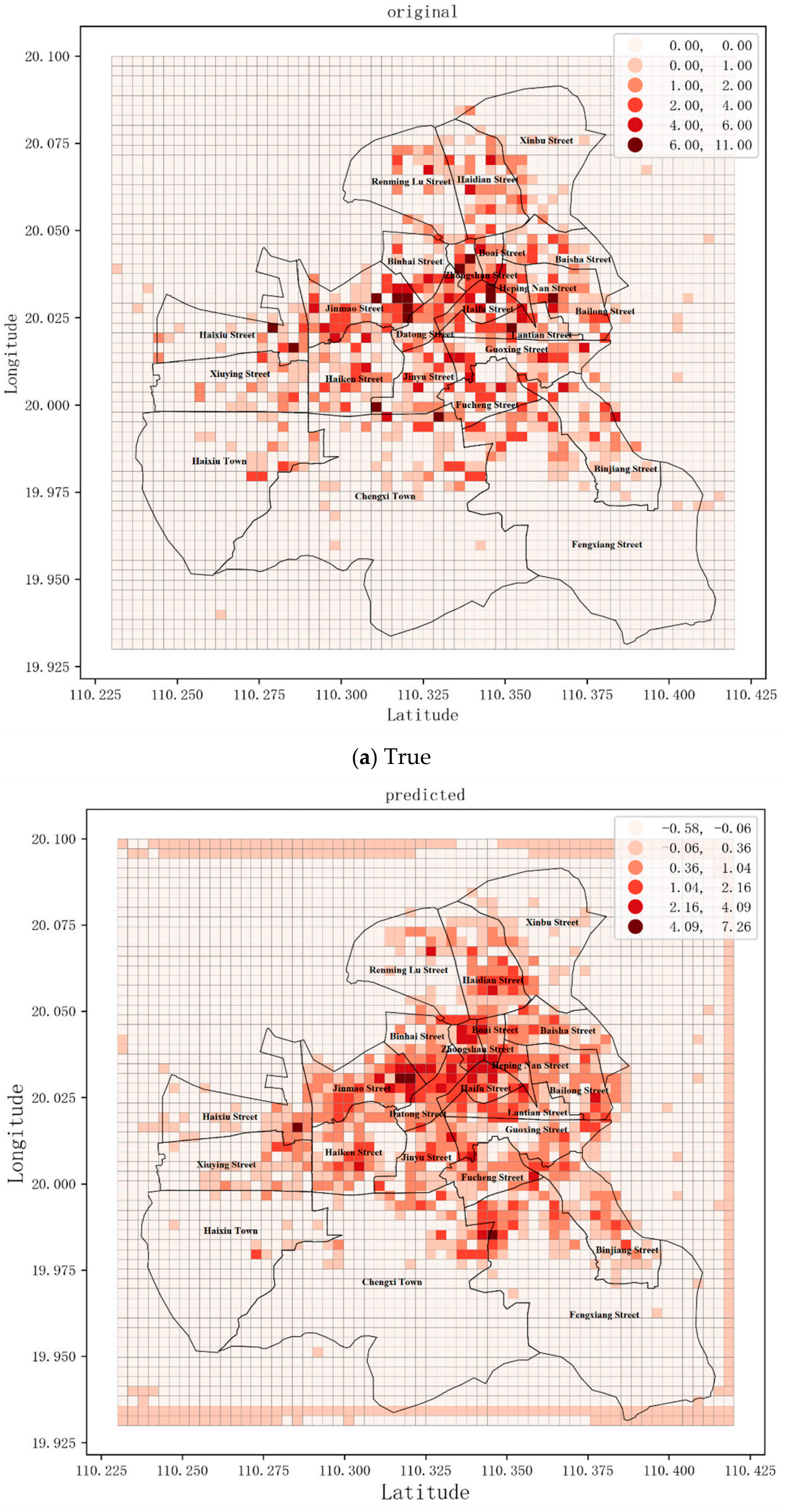


(**a**) MSE

(**b**) RMSE

Figure 4. Validation set training MAE and training RMSE curve changes.

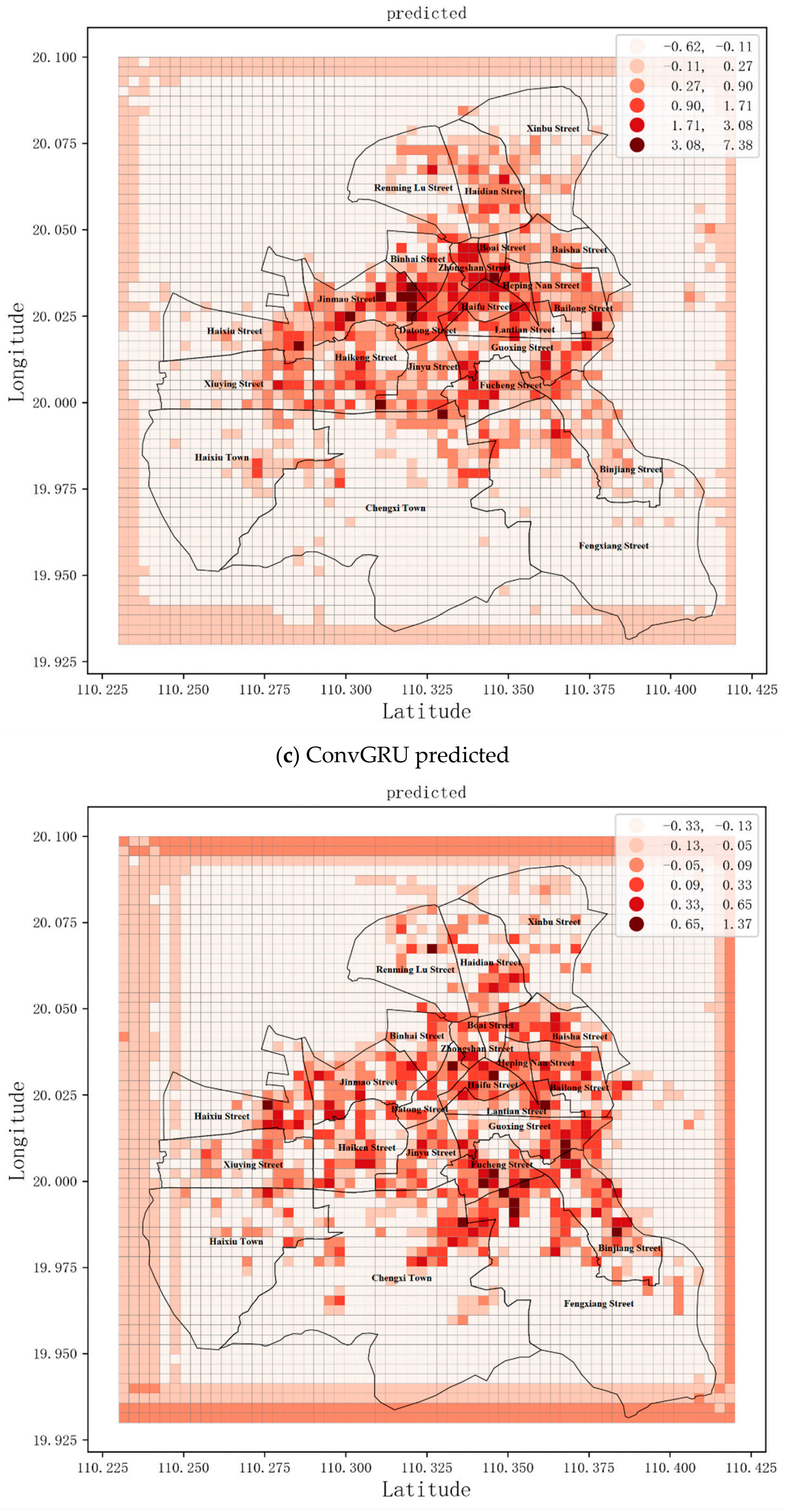
It can be seen from Figure 4 that the proposed model has the best fitting degree, the loss rate and accuracy curves are smoothest, and the accuracy has a rising trend. It can be seen from the accuracy change that the training speed of the proposed model is best in the first 20 rounds of training, and then the accuracy changes slowly and gradually flattens. At 50 training rounds, the fitting degree of the model is best, and the accuracy reaches the maximum. Compared to the other models, the MAE and RMSE curves of the proposed model are always at the bottom, and its training effect is best. The accuracy curve of the reference model is serrated, and the data fit poorly. The reference model has the fastest training speed in the first 10 rounds, and then it slows down, reaching the optimal value at the 30th round. To more intuitively see the performance of the model, we visualize the prediction results in Figure 5.



(**a**) True

(**b**) ConvLSTM predicted

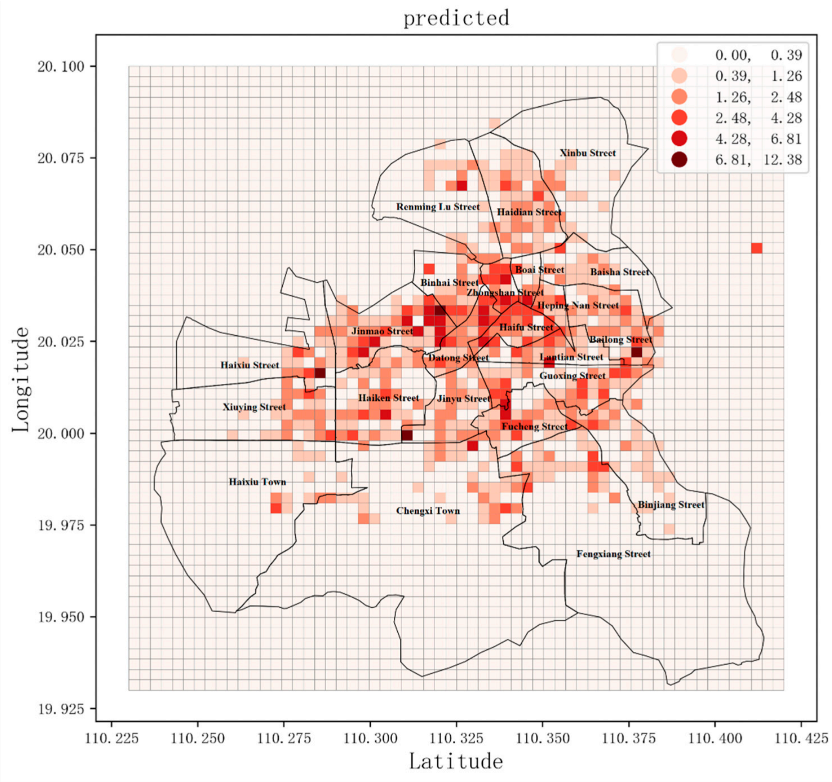
Figure 5. *Cont.*



(**c**) ConvGRU predicted

(**d**) SaConvLSTM predicted

Figure 5. *Cont.*



(**e**) SPTformer predicted

Figure 5. Prediction results of different models: (**a**) visualization of real data; (**b**) prediction result of ConvLSTM; (**c**) prediction result of ConvGRU; (**d**) prediction result of SaConvLSTM; (**e**) prediction result of SPTformer. An instance from the test set. Each pixel represents a study area, whose grayscale represents its travel volume.

This study selected the same dataset and used the trained model to predict it. Figure 5 shows the visualization of the prediction results of different models. Each frame of data has an image data structure, whose gray value is the travel demand. Comparing the real map to the predictions, it can be found that the prediction map of the proposed model is similar to the real map, it has the smallest difference in distribution shape and intensity, and it can best express details. Although the other models predict the general distribution characteristics of the data, there is a big gap in the details, and the predicted value of travel demand differs greatly from the actual value.

After careful observation the prediction results of each model, we found that: ConvLSTM and ConvGRU have similar forecasts, with predictions for travel intensity roughly the same. However, in predicting spatial distribution, ConvLSTM is slightly better. Compared with the first two models, the saConvLSTM has poor prediction performance, which is lower than the ConvLSTM and ConvGRU models in both spatial distribution and travel intensity prediction. Overall, although the three comparison models predicted the spatial distribution of the central area of the study area, they had poor predictions for the marginal area around the study area. Looking at the original map, it can be found that the travel intensity of the edge area around the original map is very low or there are even no travel data, and the compared model makes excessive predictions. In contrast, the prediction results of the model proposed in this paper are better, the prediction of the edge area of the study area is basically the same as the original data, and the prediction of travel intensity is closer to the original data. Thus, our model is more competitive.

Analyzing the prediction results of the model proposed in this paper, it can be found that: our model predicted best for the city center area, but the Xinbu Street, Haixiu Street, Xiuying Street, Haixiu Town, Chengxi Town, Fengxiang Street, and Binjiang Street areas were poor. The model predicts better results in the city center area due to the higher intensity of online car-hailing trips in urban centers and the stronger cyclicality of residents' daily trips. However, in other areas, the prediction results are poor due to the small number of daily trips and the irregular use of online car-hailing. Subsequent studies can analyze this part of the area separately to increase the accuracy of the forecast.

4.6. Discussion

This study used the same dataset to experiment with different models. The experimental results show that our model has the best fitting degree to the travel demand data of online car-hailing. The spatial distribution of the prediction results is closer to the original data, and it can better describe details. At the same time, our model most accurately predicts the demand for car-hailing. In order to capture the spatial relationship between sequences, the contrasting model changes the operation between the input and each gate to the form of convolution, while the CNN receptive field is usually small, which is not conducive to capturing global features. Unlike CNN, Transformer can extract all the information we need from the input and its relations at the same time, thus capturing long-range dependencies.

Using the same data and training batch, the average training times of SPTformer, ConvLSTM, and ConvGRU are 16 s, 33 s, and 20 s, respectively. ConvGRU is faster than ConvLSTM, but its prediction accuracy is poor. The training time of SaConvLSTM is 44 s, so our model has a shorter training time and faster speed. In addition, our model consumes fewer GPU resources. Since the contrastive model utilizes the LSTM and GRU structures to capture the temporal relationship between sequences, this type of neural network has evolved from the RNN structure. Since the RNN was proposed, it has been widely used in time series data problems. Generally speaking, the RNN is a for loop structure, which reuses the results of the previous iteration of the loop. Theoretically, it should be able to remember information seen before many time steps, but in fact, it can hardly learn this long-term dependence. Therefore, the LSTM network has been proposed subsequently. It is a variant of the RNN, which can better learn long-term dependence than the RNN. However, like the RNN, it must process sequence data in sequence, so it has no room for parallelization to accelerate the speed of model training. The working principle of GRU is the same as that of LSTM, with some simplifications and less computation, but its representation ability is not as good as LSTM in terms of prediction results. SPTformer is a deep learning model that utilizes an attention mechanism. Attention mechanisms in neural networks enhance the relevant and important parts of the input and remove the irrelevant parts, and learn which parts are important through training. Compared with a recurrent neural network, SPTformer has the advantage that it does not need to process sequential data in sequence. In the process of model training, there is a larger parallel interval, so the training time is reduced. To sum up, our model is more competitive.

Observing the prediction results of datasets constructed with different time divisions, we found that the more refined the time division, the better the prediction effect. When the time division is finer, the more information the model obtains, and the more accurate the prediction. Therefore, when forecasting travel demand, the data should be divided more finely.

5. Conclusions

To make more accurate spatiotemporal predictions of online car-hailing travel demand, based on the Transformer architecture, this paper proposes a new spatiotemporal prediction model. We utilized positional encoding, an attention mechanism, and a 3D convolutional network to effectively capture the spatiotemporal relationships between data. Based on the parallel mechanism of the Transformer network, our model has a fast training speed. This study processes the car-hailing order data into a video frame sequence, and the processed data are more in line with the spatiotemporal characteristics of online car-hailing travel data. At the same time, the travel intensity of online car-hailing can be directly obtained from the predicted results. Compared with the overall travel forecast for Haikou, our experiment can obtain travel demand in the small study area. This research used the 2017 real online car-hailing order data of Haikou City to test the performance of the proposed model, and the experiment proved the effectiveness of the method proposed in this paper. In real life, the method proposed in this paper can be used to predict the travel demand of online car-hailing in the next hour. For passengers, it is possible to better understand the changing laws of online car-hailing travel demand in different regions and at different times, so as to make more reasonable travel decisions and improve the travel efficiency of residents. For online car-hailing drivers, it is possible to accurately find the hot spots of travel demand, reduce the empty driving rate, and increase the income of online car-hailing drivers. At the same time, for urban management personnel, it can reasonably dispatch vehicles and timely solve traffic travel needs, improve the level of urban traffic management, and reduce urban road traffic pressure. This will provide a reference for the research on shared travel and promote the development of shared mobility.

This paper only considered the impact of historical travel on the future. Although the model achieved good results, differences can be found in some details of performance. In real life, there are many factors that affect residents' travel, such as weather, points of interest, holidays, and differences in travel intensity at different time periods on the same day. Therefore, in the next study, more influencing factors can be comprehensively considered. In addition, the outbreak of large-scale infectious diseases has a greater impact on residents' thinking and travel methods, such as COVID-19. Therefore, follow-up research can analyze and predict the travel characteristics of residents using online car-hailing during COVID-19. In this paper, Haikou City was only divided according to the fixed grid scale, yet the prediction results are different with different division methods. Subsequent research can divide the study area into different scales for comparison to achieve the best results.

Author Contributions: C.Y., S.B. and S.L. conceived and designed the experiments; C.Y. and L.W. performed the experiments; C.Y., S.B. and L.Z. wrote the Chinese paper; C.Y., S.B. and S.L. translated the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (No. 41971340 and No. 41271410).

Institutional Review Board Statement: Approval for the study was not required in accordance with local/national legislation.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, Shuoben Bi, upon reasonable request.

Acknowledgments: We thank LetPub (www.letpub.com (accessed on 1 May 2022)) for its linguistic assistance during the preparation of this manuscript. The authors would like to thank the handling editor and anonymous reviewers for their careful reading and helpful remarks.

Conflicts of Interest: The authors declare that they do not have any competing interest.

References

1. Olayode, I.O.; Severino, A.G.; Campisi, T.; Tartibu, L.K. Comprehensive Literature Review on the Impacts of COVID-19 Pandemic on Public Road Transportation System: Challenges and Solutions. *Sustainability* **2022**, *14*, 9586. [CrossRef]
2. Torrisi, V.; Campisi, T.; Inturri, G.; Ignaccolo, M.; Tesoriere, G. Continue to share? An overview on italian travel behavior before and after the COVID-19 lockdown. *AIP Conf. Proc.* **2021**, *2343*, 090010. [CrossRef]
3. Guo, Y.; Zhang, Y.; Boulaksil, Y.; Tian, N. Multi-dimensional spatiotemporal demand forecasting and service vehicle dispatching for online car-hailing platforms. *Int. J. Prod. Res.* **2022**, *60*, 1832–1853. [CrossRef]
4. Wu, T.; Wang, S.; Wang, L.; Tang, X. Contribution of China's online car-hailing services to its 2050 carbon target: Energy consumption assessment based on the GCAM-SE model. *Energy Policy* **2022**, *160*, 112714. [CrossRef]
5. Ge, H.; Li, S.; Cheng, R.; Chen, Z. Self-Attention ConvLSTM for Spatiotemporal Forecasting of Short-Term Online Car-Hailing Demand. *Sustainability* **2022**, *14*, 7371. [CrossRef]

6. China Internet Network Information Center. *The 47th China Statistical Report on Internet Development*; Office of the Central Cyberspace Affairs Commission, Cyberspace Administration of China: Beijing, China, 2021.
7. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [CrossRef]
8. Zhang, H.; Wang, X.; Cao, J.; Tang, M.; Guo, Y. A multivariate short-term traffic flow forecasting method based on wavelet analysis and seasonal time series. *Appl. Intell.* **2018**, *48*, 3827–3838. [CrossRef]
9. Kanchymalay, K.; Salim, N.; Sukprasert, A.; Krishnan, R.; Hashim, U.R. Multivariate time series forecasting of crude palm oil price using machine learning techniques. *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *226*, 012117. [CrossRef]
10. Zhao, M.; Chang, C.H.; Xie, W.; Xie, Z.; Hu, J. Cloud shape classification system based on multi-channel CNN and improved FDM. *IEEE Access* **2020**, *8*, 44111–44124. [CrossRef]
11. Mozo, A.; Ordozgoiti, B.; Gómez-Canaval, S. Forecasting short-term data center network traffic load with convolutional neural networks. *PLoS ONE* **2018**, *13*, e0191939. [CrossRef]
12. Dalgkitsis, A.; Louta, M.; Karetsos, G.T. Traffic forecasting in cellular networks using the LSTM RNN. In Proceedings of the 22nd Pan-Hellenic Conference on Informatics, Athens, Greece, 29 November–1 December 2018; pp. 28–33. [CrossRef]
13. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28, pp. 802–810.
14. Chen, X.; Xie, X.; Teng, D. Short-term Traffic Flow Prediction Based on ConvLSTM Model. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 846–850. [CrossRef]
15. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6910–6920. [CrossRef]
16. Jin, B.; Cruz, L.; Goncalves, N. Deep facial diagnosis: Deep transfer learning from face recognition to facial diagnosis. *IEEE Access* **2020**, *8*, 123649–123661. [CrossRef]
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 6000–6010.
18. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. Deepvit: Towards deeper vision Transformer. *arXiv* **2021**, arXiv:2103.11886. [CrossRef]
19. Chen, Z.; Xie, L.; Niu, J.; Liu, X.; Wei, L.; Tian, Q. Visformer: The vision-friendly Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 589–598.
20. Zhang, P.; Dai, X.; Yang, J.; Xiao, B.; Yuan, L.; Zhang, L.; Gao, J. Multi-scale vision longformer: A new vision Transformer for high-resolution image encoding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2998–3008.
21. Shu, H.; Wang, J.; Chen, H.; Li, L.; Yang, Y.; Wang, Y. Adder Attention for Vision Transformer. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021.
22. Yuan, Y.; Fu, R.; Huang, L.; Lin, W.; Zhang, C.; Chen, X.; Wang, J. HRformer: High-Resolution Vision Transformer for Dense Predict. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021.
23. Zhang, K.; Liu, Z.; Zheng, L. Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1480–1490. [CrossRef]
24. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transp. A Transp. Sci.* **2019**, *15*, 1688–1711. [CrossRef]
25. Chen, M.; Yu, X.; Liu, Y. PCNN: Deep convolutional networks for short-term traffic congestion prediction. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3550–3559. [CrossRef]
26. Mou, L.; Zhao, P.; Xie, H.; Chen, Y. T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *IEEE Access* **2019**, *7*, 98053–98060. [CrossRef]
27. Yu, R.; Li, Y.; Shahabi, C.; Demiryurek, U.; Liu, Y. Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 777–785. [CrossRef]
28. Tang, Q.; Yang, M.; Yang, Y. ST-LSTM: A deep learning approach combined spatio-temporal features for short-term forecast in rail transit. *J. Adv. Transp.* **2019**, *2019*, 8392592. [CrossRef]
29. Gu, Y.; Lu, W.; Qin, L.; Li, M.; Shao, Z. Short-term prediction of lane-level traffic speeds: A fusion deep learning model. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 1–16. [CrossRef]
30. Wu, Y.; Tan, H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv* **2016**, arXiv:1612.01022. [CrossRef]

31. Zhene, Z.; Hao, P.; Lin, L.; Guixi, X.; Du, B.; Alam Bhuiyan, Z.; Long, Y.; Li, D. Deep convolutional mesh RNN for urban traffic passenger flows prediction. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; pp. 1305–1310. [CrossRef]
32. Liu, Y.; Zheng, H.; Feng, X.; Chen, Z. Short-term traffic flow prediction with Conv-LSTM. In Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017; pp. 1–6. [CrossRef]
33. Li, T.; Ni, A.; Zhang, C.; Xiao, G.; Gao, L. Short-term traffic congestion prediction with Conv–BiLSTM considering spatio-temporal features. *IET Intell. Transp. Syst.* **2021**, *14*, 1978–1986. [CrossRef]
34. He, Z.; Chow, C.; Zhang, J. STCNN: A spatio-temporal convolutional neural network for long-term traffic prediction. In Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM), Hong Kong, China, 10–13 June 2019; pp. 226–233. [CrossRef]
35. Wang, D.; Yang, Y.; Ning, S. Deepstcl: A deep spatio-temporal convlstm for travel demand prediction. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio, Brazil, 8–13 July 2018; pp. 1–8. [CrossRef]
36. Huang, P.; Huang, B.; Zhao, F.; Zhang, Y.; Chen, M. Deep ConvLSTM-Inception Network for Traffic Prediction in Smart Cities. In Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Cuvu, Fiji, 14–16 December 2020; pp. 1211–1218. [CrossRef]
37. Li, P.; Sun, M.; Pang, M. Prediction of taxi demand based on convLSTM neural network. In Proceedings of the International Conference on Neural Information Processing, Siem Reap, Cambodia, 13–16 December 2018; Springer: Cham, Switzerland, 2018; pp. 15–25. [CrossRef]
38. Chen, Z.; Xu, J.; Lin, Y.; Feng, B.; Liao, D.; Lin, H. Traffic Flow Prediction based on Time Information. In Proceedings of the 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Guangzhou, China, 13–15 November 2020; pp. 64–71. [CrossRef]
39. Di, X.; Xiao, Y.; Zhu, C.; Deng, Y.; Zhao, Q.; Rao, W. Traffic congestion prediction by spatiotemporal propagation patterns. In Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM), Hong Kong, China, 10–13 June 2019; pp. 298–303. [CrossRef]
40. Huang, X.; Tang, J.; Peng, Z.; Chen, Z.; Zeng, H. A Sparse Gating Convolutional Recurrent Network for Traffic Flow Prediction. *Math. Probl. Eng.* **2022**, *2022*, 6446941. [CrossRef]
41. Ranawaka, Y. Short-Term Traffic Flow Prediction Using Google Traffic Data. 2021. Available online: https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/4176 (accessed on 1 March 2022).
42. Grigsby, J.; Wang, Z.; Qi, Y. Long-range transformers for dynamic spatiotemporal forecasting. *arXiv* **2021**, arXiv:2109.12218. [CrossRef]
43. Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.J.; Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv* **2020**, arXiv:2001.02908. [CrossRef]
44. Song, X.; Wu, Y.; Zhang, C. TSTNet: A sequence to sequence transformer network for spatial-temporal traffic prediction. In Proceedings of the International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021; Springer: Cham, Switzerland, 2021; pp. 343–354. [CrossRef]
45. Zhang, H.; Zou, Y.; Yang, X.; Yang, H. A Temporal Fusion Transformer for Short-term Freeway Traffic Speed Multistep Prediction. *Neurocomputing* **2022**, *500*, 329–340. [CrossRef]
46. Cai, L.; Janowicz, K.; Mai, G.; Yan, B.; Zhu, R. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Trans. GIS* **2020**, *24*, 736–755. [CrossRef]
47. Girdhar, R.; Grauman, K. Anticipative video Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 13505–13515.
48. Zhang, K.; Feng, X.; Wu, L.; He, Z. Trajectory Prediction for Autonomous Driving Using Spatial-Temporal Graph Attention Transformer. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–11. Available online: https://ieeexplore.ieee.org/abstract/document/9768029 (accessed on 1 March 2022). [CrossRef]
49. Wu, Y.F.; Yoon, J.; Ahn, S. Generative Video Transformer: Can Objects be the Words? In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–14 August 2021; pp. 11307–11318.
50. Farazi, H.; Behnke, S. Frequency domain Transformer networks for video prediction. *arXiv* **2019**, arXiv:1903.00271. [CrossRef]
51. Wang, W.; Peng, X.; Su, Y.; Qiao, Y.; Cheng, J. Ttpp: Temporal Transformer with progressive prediction for efficient action anticipation. *Neurocomputing* **2021**, *438*, 270–279. [CrossRef]
52. Liu, Z.; Luo, S.; Li, W.; Lu, J.; Wu, Y.; Sun, S.; Li, C.; Yang, L. ConvTransformer: A convolutional Transformer network for video frame synthesis. *arXiv* **2020**, arXiv:2011.10185. [CrossRef]
53. Shi, Z.; Xu, X.; Liu, X.; Chen, J.; Yang, M.-H. Video Frame Interpolation Transformer. *arXiv* **2021**, arXiv:2111.13817. [CrossRef]
54. Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; Ding, Z. 3d human pose estimation with spatial and temporal Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 11656–11665.

55. Tai, T.M.; Fiameni, G.; Lee, C.K.; Lanz, O. Higher Order Recurrent Space-Time Transformer for Video Action Prediction. *arXiv* **2021**, arXiv:2104.08665. [CrossRef]
56. Farazi, H.; Nogga, J.; Behnke, S. Local frequency domain Transformer networks for video prediction. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–10. [CrossRef]
57. Zheng, Q.; Yang, M.; Yang, J.; Zhang, Q.; Zhang, X. Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process. *IEEE Access* **2018**, *6*, 15844–15869. [CrossRef]
58. Haikou Statistics Bureau. *2018 Statistical Bulletin of Haikou National Economic and Social Development*; Haikou Statistics Bureau: Haikou, China, 2019.
59. Chen, L.; Zhang, Y. Research on alleviating urban traffic congestion in the context of free trade port construction—Taking Haikou City as an example. *New Orient.* **2021**, *3*, 5–9.
60. Hainan Daily. Hainan Taxi Industry will Usher in "Tripartite Confrontation". 2016. Available online: https://m.sohu.com/a/112699585_162698 (accessed on 1 June 2021).
61. Didi Chuxing. Available online: https://outreach.didichuxing.com/app-vue/HaiKou (accessed on 1 January 2021).
62. Agrawal, S.; Barrington, L.; Bromberg, C.; Burge, J.; Gazen, C.; Hickey, J. Machine learning for precipitation nowcasting from radar images. *arXiv* **2019**, arXiv:1912.12132. [CrossRef]
63. Yu, W.; Li, J.; Liu, Q. Spatial-temporal prediction of vegetation index with a convolutional GRU network. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020. [CrossRef]
64. Lin, Z.; Li, M.; Zheng, Z.; Cheng, Y.; Yuan, C. Self-attention convlstm for spatiotemporal prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34. [CrossRef]