Long-term Origin-Destination Demand Prediction with Graph Deep Learning

Xiexin Zou, Shiyao Zhang, *Member, IEEE*, Chenhan Zhang, James J.Q. Yu, *Senior Member, IEEE*, and Edward Chung

Abstract—Accurate long-term origin-destination demand (OD) prediction can help understand traffic flow dynamics, which plays an essential role in urban transportation planning. However, the main challenge originates from the complex and dynamic spatial-temporal correlation of the time-varying traffic information. In response, a graph deep learning model for long-term OD prediction (ST-GDL) is proposed in this paper, which is among the pioneering work that obtains both short-term and long-term OD predictions simultaneously. ST-GDL avoids the conventional multi-step forecasting and thus prevents learning from prediction errors, rendering better long-term forecasts. The proposed method captures time attributes from multiple time scales, namely closeness, periodicity, and trend, to study the features with temporal dynamics. In addition, two gate mechanisms are introduced over the vanilla convolution operation to alleviates the error accumulation issue of typical recurrent forecast in long-term OD prediction. A method based on graph convolution is proposed to capture the dynamic spatial relationship, which projects the transportation network into a graphical time-series. Finally, the long-term OD prediction results are obtained by combining the extracted spatio-temporal features with external features from the meteorological information. Case studies on practical datasets show that the proposed model is superior to existing methods in long-term OD prediction problems.

Index Terms—Long-term OD prediction, graph deep learning, gate mechanism, graph convolution

1 Introduction

Reliable forecast of future origin-destination demand (OD) data is crucial for traffic management and taxi company operation [1]. Based on the predicted information, the authorities can dynamically adjust traffic lights; the public transport operators can assign passenger carriers to various regions to achieve the optimal demand balance for maximizing profit guaranteed by the entire company fleet. Besides, road users can select fastest routes plans for their trips by considering the level of traffic congestion. Therefore, the OD forecasting task is worth discussing, and a plethora of research effort has been devoted in the past few decades [2].

Currently, the research community is witnessing increasing efforts in adopting deep learning techniques in OD demand prediction. Because the focus of OD demand forecasting is to capture the temporal and spatial dependence between data. And deep learning models can capture complex non-linear relationships by using distributed and layered feature representations and is excellent at processing massive data. Most of the current OD demand forecasting works rely on Recurrent Neural Network (RNN) and its variant Long Short-Term Memory (LSTM) to capture time correlation [3]–[5], while some works on traffic demand prediction apply convolution in the time dimension [6]–[8]. Additionally, Convolutional Neural Network (CNN) and Graph Convolutional Network (GCN) are utilized to capture spatial dependencies based on geographic distance [9], [10].

While the above approaches are primarily designed for short-term prediction, they can adopt the recursive forecasting technique (multi-step prediction) to perform long-term OD prediction. In this paradigm, the one-step predicted OD data are considered the ground truth for the next-step prediction. Besides, there is current work focusing on introducing some attention mechanisms to make the model learn more information about long-term trends [11], [12]. However, such techniques inevitably suffer from the error accumulation issue, where the prediction errors in the previous forecasting steps are strengthened in the following ones. As far as we are concerned, this issue is not adequately addressed in the existing OD prediction literature. Furthermore, related transport engineering research proposes that spatial relationships should be related to the attributes of the regions rather than their geographic location and change dynamically over time [13], [14]. The most recent work [15] extracted different types of regional relationships based on traffic data, and constructed multiple static traffic graphs for subway OD prediction tasks with fixed routes. Additional POI information is also introduced to obtain the POI similarity to construct traffic graphs [16]. However, these works are either based on static graphs (invariant relationships) or based on additionally collected information which is not available in every data. Dynamic characteristic is not accounted for in the previous urban taxi OD prediction work.

To close the research gap in existing long-term OD prediction approaches, we propose a novel long-term OD prediction approach based on graph deep learning, which combines CNN and GCN for better transport feature exploitation. The periodicity of changes is particularly important in the long-term forecast. In the existing works of predicting the total inflow/outflow of each region, it has been proved that the addition of different types of time slices can help long-term spatiotemporal prediction to obtain this characteristic [8], [12], [17]. Although OD prediction faces more sparse data, it is also a time-space prediction problem. Works [8], [12], [17] provide long-term OD prediction with an idea of how to select more effective inputs from the same available historical OD data. Therefore, in the proposed model, we extract features from different time scales and regional attributes to obtain the dynamically changing time-space relationship between the demands of all OD pairs, as well as the daily and weekly cycle dependence. In this method, considering that directly learning the global spatial relationship of OD data will bring about the complexity of $(N \times N) \times (N \times N)$, we separately learn the relationship between the origins and destinations of OD pairs, which is with $N \times N$ complexity, and then combine the two relationships. This is called Multi-perspective Modeling in the paper, which is different from other works [18] that extract the primary features from the OD matrix and its transpose and combine them into the subsequent network. Specifically, each region is regarded as a departure or arrival point, such that the OD prediction task is treated as a multi-dimensional feature generation task for each departure/arrival region pair, and the dimension of the feature is the number of all regions. The relevant features can be learned from those combined perspectives. Moreover, to alleviate the accumulation of prediction errors, convolutions with gating mechanisms are applied to extract spatio-temporal features and jointly predict multiple OD demands. Besides, graph convolution is introduced to learn dynamic global spatial changes for further improving the long-term prediction accuracy.

In summary, the contributions of this paper are as follows:

- An ST-GDL model is proposed to jointly predict OD demands of multiple time segments in the future, which is among the pioneering work on joint multi-step OD prediction.
- A ResNet-based block ST-Conv with gating mechanisms CA-gate and SA-gate are proposed to capture dynamic space-time transformation.
- A deep learning building block S-GCN is proposed to obtain the adjacency matrix based on the current input to develop the dynamic global spatial relation, which can also be applied to other deep learning models.
- A series of experiments on real-world OD datasets NYC-TOD and DiDi Chuxing show that the proposed model is superior to other approaches in long-term OD prediction.

The rest of this paper is organized as follows. First, we review related literature in Section 2 and present the preliminaries in Section 3. Then, the architecture of the proposed ST-GDL model is elaborated in Section 4. Extensive evaluation experiments are conducted in Section 5. Finally, Section 6 concludes this study.

2 Related Work

In this section, the related work of long-term traffic prediction is summarized with an emphasis on deep learning-based approaches. Early parametric models use Auto-Regressive Integrated Moving Average (ARIMA), Kalman Filter, and their variants for traffic prediction [19]–[24]. However, these methods typically assume that temporal traffic dynamics have a linear correlation. Additionally, they calculate the future traffic flow of areas separately without considering the spatial relationship, which is fundamental for traffic prediction [25]. Although non-parametric modeling methods, such as K-Nearest Neighbor (KNN) [26], [27] and Support Vector Regression (SVR) [28], [29], can predict more accurately, they do not perform well for long-term prediction since they require careful feature engineering or exhaustive calculation time to make inferences. Furthermore, traditional methods naturally predict traffic flows from the data of each specific region; no consideration is given to the correlation between the predicted values [30]. Therefore, most researchers develop deep learning models to handle traffic prediction tasks thanks to their outstanding computing performance and the ability to automatically process large datasets and non-linear functions, which have been shown by the successful application in computer vision and natural language processing tasks [8], [31]–[34]. Next, we first review several typical deep learning methods in long-term traffic prediction tasks and then summarize existing OD forecasting models.

2.1 Long-term Traffic Demand Prediction

Challenges of long-term traffic prediction are mainly the capture of temporal and spatial relationships, especially long-term temporal trends and global spatial relationships. To obtain the time relationship, the most popular network in long-term traffic prediction models based on deep learning is the Long Short-Term Memory (LSTM) [35], which is a popular variant of the Recurrent Neural Networks (RNN) [36] and is widely used in time series prediction problems. However, the prediction accuracy of the LSTM-based models decreases significantly as the prediction time range increases. This is because the ground-truth input is given as the preliminary for subsequent prediction during training. Therefore, in the operation of long-term prediction, the output of the network must be routed back to itself. The distribution of the network output is not entirely consistent with the ground truth value. So in the process of long-term forecasting using the learned model, a new situation that has not been learned is encountered. To solve this problem, some researchers have worked to improve the structure and input data of LSTM. He *et al.* proposed an encoder-decoder architecture based on LSTM units for long-term traffic prediction, introduced a spatial attention model to enhance RNN and LSTM units by considering the dynamic contribution of each route to the whole traffic network. A temporal attention model is then designed to find important hidden states to provide more useful input to the LSTM in the decoder [30]. Besides, Yao *et al.* developed Spatial-Temporal Dynamic Network (STDN) to combine different LSTM structures to solve this problem. A periodic shifting attention mechanism was introduced in LSTM to deal with long-term periodic dependencies [34]. Zheng *et al.* introduced a transition attention mechanism into LSTM to model the relationship between history and future time steps, alleviating the accumulation of errors in multi-step predictions [37]. For spatial relationships, the challenge comes from capturing global spatial relationships. Because the spatial range captured by CNN is relatively small, and the traffic transitions are related to the regional functions [7]. Some researchers introduce additional information to explore the relationship between regions [7], [16]. POI information was included in a new ConvPlus network where the pooling operation and fully connected layer were adopted to capture the long-distance spatial dependence [7]. Additionally, Zhang *et al.* proposed a multi-task deep learning framework to simultaneously predict the demand of each node (region) and each link (route) and combined their hidden states for collaborative training to capture deep spatial information [6]. Yao *et al.* devised a gated local CNN to model the dynamic similarity of positions [34].

The above methods capture the temporal relation by LSTM and the spatial correlation by CNN. Traditional CNN is limited to processing gridded spatial structures similar to images. Therefore, in these works, the traffic network is transformed into a grid map according to geographic proximity, where each grid represents the road condition information of the rectangular area. Given that traffic data is usually sampled in non-Euclidean space, researchers in the field have begun to deploy GCN thanks to its expertise in dealing with irregular structures.

Li *et al.* employed a directed graph to represent pair-wise spatial correlations and performed bidirectional random walk on the graph to capture spatial dependence. They also use encoder-decoder architecture with scheduled sampling to capture temporal correlation for multi-step prediction [9]. Yu *et al.* utilized the graph convolution obtained by the approximate spectral convolution [38] as the spatial convolution operation, and one-dimensional convolution operation to summarize information in the time dimension. In the proposed model, GCN and CNN alternately formed a spatio-temporal convolution block [39]. Cui *et al.* further proposed traffic graph convolutional long short-term memory neural network (TGC-LSTM), which was based on physical distance modeling and additionally added the L1 norm of graph convolution weights and the L2 norm of graph convolution features to the loss function to identify critical links in the transportation network [13]. Subsequently, Temporal Graph Convolutional Network (T-GCN) was devised by Zhao *et al.* by combining a concise GCN model and Gated Recurrent Unit (GRU) to predict the long-term traffic flow [40]. Guo *et al.* proposed Attention Based Spatial-Temporal Graph Convolutional Networks (ASTGCN), which introduces the attention mechanism in the k-order Chebyshev graph convolution and one-dimensional convolution to capture the dynamic spatial and temporal relationships, respectively [12]. Although these studies use GCNs to model traffic graphs, the edges and connection strengths on the graphs are assumed to remain unchanged. This is due to that the construction of graphs is based on the adjacency of the geographic locations. Edges connect regions whose geographic distances are within a custom threshold, and the connection strength is determined according to their geographic distance. They are not feasible for the practical application where the regional relationship changes over time. Therefore, Xu *et al.* proposed the spatiotemporal multi-graph convolution network (ST-GGCN). They constructed different types of graphs based on geographic location, POI similarity, and traffic density to learn different spatial relationships [16]. Chen *et al.* proposed Multi-Range Attentive Bicomponent GCN (MRA-BGCN) that introduces bi-component graph convolution to combine features learned from nodes and edges. They also utilized a multi-range attention mechanism to aggregate the information in different neighborhood ranges [14]. Zheng *et al.* designed an encoder-decoder framework composed of multiple spatio-temporal attention blocks to learn dynamic spatio-temporal correlation [37].

2.2 Long-term OD Prediction

Unlike traffic demand predictions that only need to predict the number of demands in each area, OD forecasting is more challenging because it also requires forecasts to provide the starting and ending point for each demand request. The intensity of demand of regional pairs in each time slice can better provide road authorities with more traffic light planning information, and support service providers optimally allocating their resources. Most of the previous work on OD prediction is for railway transportation or subways with fixed routes, stations, and departure times. Fu *et al.* proposed Expressway OD Prediction Neural Network (EODPNN) based on Bi-LSTM to predict highway OD demand, which confirmed that neural networks are superior to traditional models for OD prediction [3]. Spatio-Temporal Long Short-term Memory Network (STLSTM) was proposed for OD prediction in rail transit by redesigning hidden layers and neurons, introducing temporal state Ctime and spatial state Cspace to improve the structure of LSTM to preserve long-term status [4]. In order to obtain the global spatial relationship, Liu *et al.* designed the Physical-Virtual Collaboration Graph Network (PVCGN) for subway OD prediction. According to the similarity and correlation of passenger flow between stations and the actual topology, three complementary graphs were established for more accurate OD prediction [15]. However, its prediction targets are actually only the inflow and outflow traffic volume of each station at each moment. Moreover, [3] and [4] showed that the attribute information embedding of locations brings noticeable performance improvement.

However, compared to rail transit, urban transportation networks are more complicated. Subway stations and operating time are relatively fixed, while taxis requests may occur at any place and time, resulting in uneven distribution of OD demand and a very sparse OD matrix. So a more complex deep network is needed for urban OD prediction. Wang *et al.* used GCN to aggregate spatial information of geographic and semantic neighbors, and a multi-task learning network is utilized to jointly predict both inflow and outflow for accurate OD demand prediction [41]. Duan *et al.* combined OD information and travel time and studied their implicit correlation based on ConvLSTM to improve the prediction accuracy. They also refined the OD embedding to the road network level to equip data with road attributes [5]. Besides, Chu *et al.* proposed a new method of data embedding to reduced OD data to 2-D while preserving the geographic relevance as much as possible, making the data more suitable for processing with CNN [10]. Furthermore, a multi-scale ConvLSTM network was applied to obtain

TABLE 1
Key notation

Symbol	Description
N	number of all regions ($= 75$)
N_O	number of all origins in the OD data ($N_O = N = 75$)
N_D	number of all destinations in the OD data ($N_D = N = 75$)
$x_{ij,t}$	traffic demand from regions i to j at time t
X_t	traffic demand of all OD pairs at time t
k	number of time segments that need to be predicted ($k = 12$)
G	traffic graph
V, E	vertices and edges corresponding to G
A	adjacency matrix corresponding to G
D	diagonal degree matrix corresponding to A

different ranges of space-time relationships. To obtain the spatial correlation of the global range, based on the ConvLSTM, Liu *et al.* calculated the feature similarity of regional pairs to represent the feature of each region as a weighted sum of all regional features [18].

Nonetheless, existing long-term OD prediction methods are primarily based on LSTM. Therefore, the issue of error accumulation is not resolved. Moreover, for the dynamic spatial relationship, OD data need to be tiled into 2D or 3D tensors when being processed by CNN, during which spatial information is partially lost. When applying GCN, a fixed graph is generally used to express the spatial relationship in existing methods without considering the dynamic change. Regarding these problems, we propose a new long-term OD prediction model based on graph deep learning. The model concerns different temporal data segments and regional attributes and applies graph convolution operation with gating mechanisms to capture the dynamic spatial-temporal relationship within OD demand data.

3 Preliminaries

The important notations used in this paper are described in Table 1.

3.1 Problem Definition

In this paper, the objective of long-term OD prediction is formulated as simultaneously forecasting short-term and long-term future travel demand between regional pairs. Following the previous OD prediction literature [8], [10], [18], the investigated area is divided into N non-overlapping grids[1]. Let $x_{ij,t}$ denotes the traffic demand from regions i to j at time t, and $X_t = \{x_{ij,t}, i, j \in [0, N-1]\}$ represents traffic demand data of all explored OD pairs at t. The problem can be formulated as developing a function $\mathrm{Pred}(\cdot)$ to simultaneously forecast short- and long-term OD demand in the future k time slices with the help of historical data.

As for different temporal data segments, we employ the historical data of three distinct past time slices, i.e., *closeness*, *period,* and *trend*, to jointly predict OD demand for $k = 12$ future time slices at one time, where each time slice corresponds to 30 minutes. Among theses temporal data segments, *closeness* denotes the OD demand of the first three most recent time slices before the current one; *period* and *trend* represent the OD data of 12 time slices one day and one week before the current one, respectively. Therefore, the prediction function can be written as

$$\begin{aligned}&\{\mathbf{X}_t, \mathbf{X}_{t+1}, \cdots, \mathbf{X}_{t+11}\}\\ &= \mathrm{Pred}(\underbrace{\mathbf{X}_{t-1}, \cdots, \mathbf{X}_{t-3}}_{\text{closeness}}, \underbrace{\mathbf{X}_{t-37}, \cdots, \mathbf{X}_{t-48}}_{\text{period}},\\ &\underbrace{\mathbf{X}_{t-325}, \cdots, \mathbf{X}_{t-336}}_{\text{trend}}).\end{aligned} \tag{1}$$

3.2 Road Graph and GCN

In the proposed model, graph convolutional networks are used to model spatial relationships, and the graph convolution based on spectral theory is adopted on each time slice to extract spatial correlation of each one.

3.2.1 *Road Graph*

Based on the divided N regions, we build an undirected characteristic graph $G = (V, E, A)$ regarding the correlation of latent attributes of regional pairs. Each region is treated as a node, where V is a finite set of nodes, E represents a set of edges, indicating the connectivity between regions. $A \in R^{N \times N}$ denotes its weighted adjacency matrix, e.g., the proximity of any pair of nodes is set to 0 when their geographic distance exceeds the set threshold, otherwise the smaller the geographical distance, the higher the proximity.

3.2.2 *GCN*

The use of standard convolution is limited to regular grids, which is not suitable for general graphs with non-Euclidean structure and no translation invariance. Therefore, GCN is proposed to extract features from graph data. The graph convolution is the kernel of GCN, which is defined by

$$H^{(l+1)} = f(H^{(l)}, A) = \sigma(g_\theta * H^{(l)}) \tag{2}$$

where $H^{(l)}$ is the feature of the l^{th} layer, and when $l = 0$, it is the input data. σ is the activation function $\mathrm{Sigmoid}$, and $g_\theta*$ represents a graph convolution operation.

In typical GCN, the core of spectrogram convolution is to analyze its symmetric normalized Laplacian matrix and its eigenvalues. However, when the scale of the graph is large, it is costly to perform eigenvalue decomposition directly on the Laplacian matrix. Therefore, it is simplified by using first-order ChebNet [38] for simple calculation. Although the approximation only covers first-order neighbor nodes, the perception domain of graph convolution can be enlarged by stacking multiple GCN layers due to its flexibility [38]:

$$g_\theta * x = \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x, \tag{3}$$

where D is the diagonal degree matrix corresponding to A, $D_{ii} = \sum_j A_{ij}$. Eigenvalues of $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ are in the range of $[0, 2]$, if many layers are stacked, the input information is always an incremental input, which may lead to numerical instability and gradient explosion problems

[1] In the case studies, $N = 75$ which is in accordance with the previous literature.

[38]. To solve the problem, the renormalization trick is commonly adopted:

$$\widetilde{A} = A + I_N, \tag{4a}$$

$$\widetilde{D_{ii}} = \sum_{j} \widetilde{A}_{ij}, \tag{4b}$$

$$g_\theta * x = \theta \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} x, \tag{4c}$$

where $A + I_N$ enable nodes to consider their own features in the process of information dissemination, and $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ is to obtain a symmetric and normalized matrix to maintain the original distribution of the previous layer's feature. The final applied graph convolution is shown as follows:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{5}$$

where $W^{(l)}$ is the matrix for parameter θ. To conclude, each node is updated according to its nearest neighbors. Additional neighbor information is considered through the superposition of multiple graph convolutional layers.

4 Proposed Methodology

An OD prediction model needs to consider the interdependence of data at different time slices and the spatial interaction between different regions. The OD demand of each region is not only related to the historical demand of itself but also affected by others in the same time slice. For example, locations with the same residential properties often have the same peaks. Moreover, the interactions of regional pairs differ when they are considered as origins or arrival points.

Based on the above principles, we propose the ST-GDL model based on deep learning. In this section, we will introduce the composition of the model.

4.1 Overview

Unlike short-term OD forecasts, long-term OD data exhibits varying degrees of time pattern changes, such as recent changes in a few hours, mid-term changes in a day, and long-term changes in a week.

Fig. 1 shows the overall framework of the proposed ST-GDL model composed of four independent components. "Conv" and "Conv3D" blocks represent the two-dimensional and three-dimensional convolution operations, respectively. "Conv/Conv3D-BN" indicates a convolutional layer followed by one BatchNorm2D/3D layer. Furthermore, "reshape" introduces or removes the singleton dimension from the input data, "permute" reorders the dimension, that is, dimension swap of high-order tensors. And "repeat" repeats input tensor along a specific dimension. Besides, $\mathbb{R}$ in the figure represents the output dimensionality of the previous module, $\oplus$ and $\otimes$ are element-wise addition and multiplication, respectively. "T" in the circle in Fig. 1 represents a $\tanh$ layer, which is used to map the output to the range $[-1, 1]$.

The first component of Fig. 1 embeds the influence of different weather information on the OD value on the corresponding time slice. The other three components learn potential spatio-temporal information from different temporal data segments, namely close, daily, and weekly. Denote their outputs as f_C, f_P, f_T, respectively. Next, the weighted sum f_{OD} of these spatio-temporal features is the feature extracted from historical OD data.

$$f_{OD} = \omega_C \times f_C + \omega_P \times f_P + \omega_T \times f_T \tag{6}$$

where ω_C, ω_P and ω_T are learnable parameters in the network, and $\times$ denotes the point by point multiplication.

For the external data, we take the corresponding meteorological data of the predicted 12 moments as $\mathbb{R}^{(12,29)}$ input and feed it into two stacked 2-dimensional convolution layers. The first convolutional layer has 10 (1, 29) convolution kernels, which merge the corresponding weather information for each time moment, the second one gets the output of size $\mathbb{R}^{(12,1,1)}$. "Permute" is to perform dimension conversion to ensure that the channel does not represent time information, and the convolution size for the time dimension is 1, so that weather information at different time slices does not affect each other. To facilitate the external features to assist in the prediction of the main features obtained from the OD data, we keep the dimensions of the extra features and the main features obtained from the OD data at the same dimensions. That is, by stacking along the 2^{nd} and 3^{rd} axes to obtain a new $\mathbb{R}^{(12,75,75)}$ tensor, a value on each time slice (1^{st} axes) of the original tensor is copied to all points on the corresponding time slice of the new tensor. Subsequently, the new $\mathbb{R}^{(12,75,75)}$ tensor is fed into a $\mathrm{Sigmoid}$ function to multiply the feature f_{OD} extracted from the OD demand data,

$$f_{All} = f_{OD} \times f_{ext} \tag{7}$$

where f_{ext} is the output of the $\mathrm{Sigmoid}$ function. Finally, f_{All} is sent to a convolution layer and a $\tanh$ layer to produce the prediction.

Next, we will describe the inputs and composition of the MP Model in detail.

4.2 Multi-perspective Modeling Block

The vital component of extracting information from OD historical data is the Multi-Perspective Modeling block ("MP Model").

OD data can be expressed as a series of time series, and usually, 3 to 5 recent historical time slices are used for OD prediction. Most works apply LSTM-based models to make historical time slices contribute differently to predictions. However, the calculation of LSTM is time-consuming and brings about error accumulation problems. Therefore, we use a CNN-based model to capture the different contributions of historical time slices. For historical OD data, we select those of the first 3 time slices of the predicted one, and corresponding 12 time slices of the previous day and week as input and refer to them as close ($\mathcal{C}$), period ($\mathcal{P}$), and trend ($\mathcal{T}$), respectively. And the inputs of several MP modules in Fig. 1 is the $\mathbb{R}^{(12,75,75)}$ tensors from $\mathcal{C}/\mathcal{P}/\mathcal{T}$, respectively.

$\mathcal{C}$ needs unique processing before being sent to the MP module, because the OD demand of the closest time slice is more relevant to the predicted ones, and these 3 time slices have different influences on the future 12 time slices. Considering this, as shown in Fig. 1, the input data is first reshaped to be a $\mathbb{R}^{(3,75,75)}$ tensor, where each channel represents a specific time slice. 12 $\mathbb{R}^{(3,1,1)}$ 3D convolution

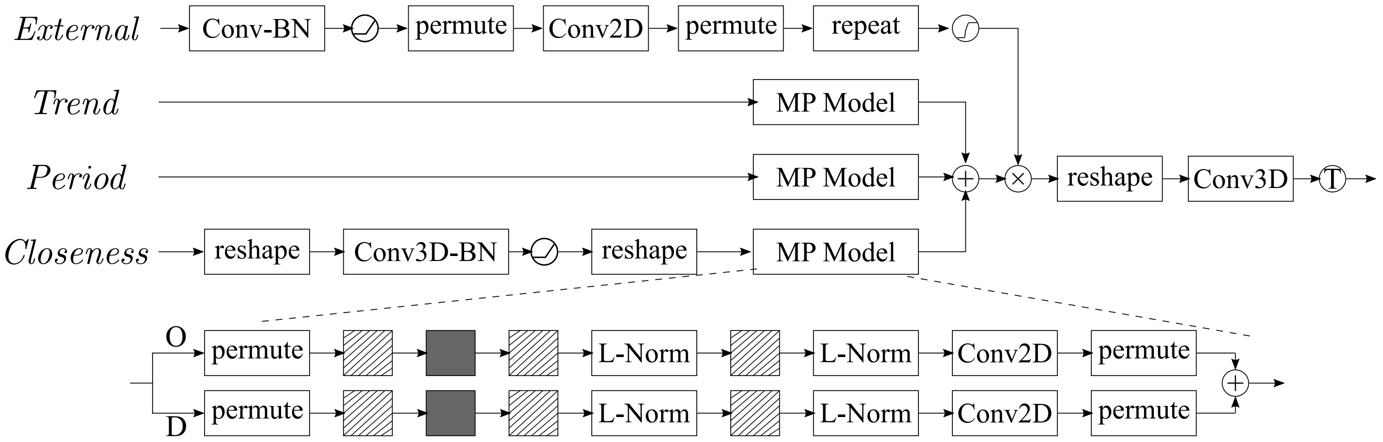

Fig. 1. Overview of the proposed model

kernels are set to aggregate historical time slices of each OD pair according to different combination rules through three-dimensional convolution (3D-Conv), followed by one BatchNorm3D layer and an activation function ReLU. The output 12 $\mathbb{R}^{(1,75,75)}$ tensors are subsequently reshaped into $\mathbb{R}^{(12,75,75)}$. This operation not only transforms the input dimension into the desired output dimension $\mathbb{R}^{(12,75,75)}$ but also explores the different contributions of OD data of the last few time slices to the predicted ones. For $\mathcal{P}/\mathcal{T}$, their time slices and the predicted ones possess one-to-one correspondence, and the dimensionalities are also the same. So no additional convolution operation is required before sending them to the MP module.

In the OD forecasting problem, in addition to time dynamics, spatial relationships are also particularly important. N regions correspond to $N \times N$ OD pairs. Therefore, the acquisition of the global spatial relationship of all OD pairs requires $(N \times N) \times (N \times N)$ complexity. To reduce the complexity, we decompose the problem into two simple sub-problems. Consider that each OD pair contains a corresponding origin and destination. To understand the degree of mutual influence of two OD pairs, we intuitively explore the similarity of the corresponding departure regions and the similarity of the arrival ones separately, each only requires $(N \times N)$ complexity. Furthermore, combining these two relationships to obtain the OD pair relationship.

Therefore, we split the task of extracting the similarity of origins/destinations of regional pairs into two simple subtasks:

(1) Regarding all regions as origins, each region has a feature map of size (T, N_D), i.e., the demand information from the corresponding region to all N regions at selected time slices. This sub-problem is to generate new N_D-dimensional features for each region on each time slice that needs to be predicted based on historical OD data. (2) Similarly, all regions are regarded as destinations, and the feature of each is the demand information from all areas to the specific area, and N-dimensional features are generated for each destination on each time slice.

As shown in Fig. 1, we conduct joint modeling from the perspective of departure and arrival and combine the features extracted from these two aspects. Take the origin perspective as an example, which is represented by O in the figure. First, a dimensional transformation is performed on the input $\mathbb{R}^{(12,N_O,N_D)}$ tensors to get $\mathbb{R}^{(N_D,12,N_O)}$ tensors, so that each region has a 75-dimensional feature on each time slice. N_O and N_D indicate 75 origins and 75 destinations, respectively. Subsequently, tensors are sent to ST-Conv (represented as black squares in Fig. 1) and S-GCN (shaded squares) blocks, which employ vanilla and graph convolutions to extract spatio-temporal features, respectively. Two stacked ST-block are used to extract the space-time information of the original perspective, and the white rectangle between them indicates layer normalization (LN) to prevent overfitting. Since the distribution of OD demands of adjacent time slices is similar, LN performs normalization along the channel dimension. Therefore, the dimension rearrangement is performed before sending the tensors to the LN layer to ensure that the time dimension is normalized as the channel dimension. The processing of the destination perspective differs only in the initial dimensional transformation, the dimensionality of its input tensors is transformed from $\mathbb{R}^{(12,N_O,N_D)}$ to $\mathbb{R}^{(N_O,12,N_D)}$, and then has the same process as the origin perspective. The output of the origin perspective and destination perspective are $\mathbb{R}^{(N_D,12,N_O)}$ and $\mathbb{R}^{(N_O,12,N_D)}$, respectively. N_D,12 and N_O represent information of the destination, time, and origin. Then, perform different dimensional transformations on these two outputs to obtain $\mathbb{R}^{(12,N_O,N_D)}$ tensors for facilitating feature fusion. To conclude, the output results obtained from the two perspectives were weighted and summed to obtain the multi-perspective modeling results. The outputs of the three MP modules from bottom to top in Fig. 1 are $\mathbb{R}^{(12,75,75)}$ tensors f_C, f_P, f_T, which are features extracted from different magnitudes of temporal patterns.

The proposed "MP Model" is composed of two base blocks, i.e., ST-Conv and S-GCN. In the following subsections, we elaborate on the structure of ST-Conv and S-GCN.

4.3 ST-Conv Block

The ST-Conv unit is used to capture temporal and local spatial relationships by CNN, which consists of multiple Conv-BN blocks, a CA-gate, and an SA-gate. Its structure is shown in Fig. 2. For conciseness, we demonstrate and

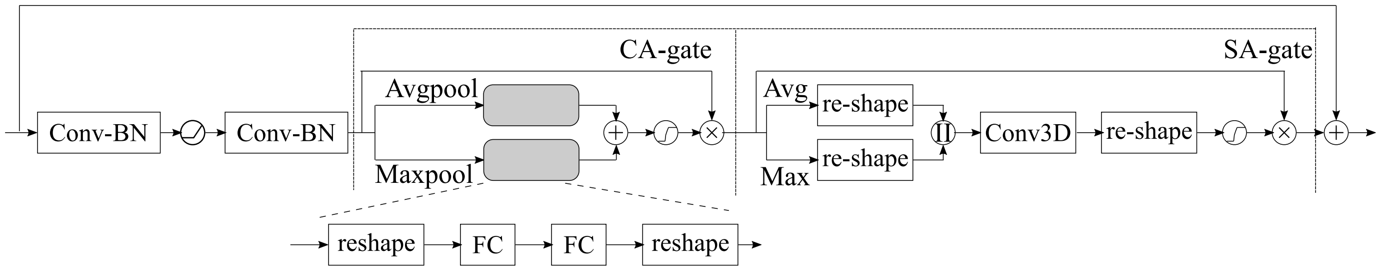

Fig. 2. ST-Conv block

discuss the motivation and architecture using the origin data process.

As explained in Section 4.2, inputs of the origin perspective are $\mathbb{R}^{(N_D,12,N_O)}$ tensors. OD data of a time slice is strongly correlated with one of its adjacent time slices and its neighbors. Therefore, as shown in Fig. 2, we first use two stacked 2D-Conv layers, followed by BatchNorm and the ReLU activation function. Specifically, convolution operation with the kernel of size $(3, 3)$ is applied for feature extraction of *Closeness*, to simultaneously fuse the information of adjacent time and the local spatial relationship. Since time slices of *Period*/*Trend* and the predicted ones possess one-to-one correspondence, kernels of size $(1, 3)$ are utilized to maintain this correspondence. Besides, the channel represents destination information, and some regions intuitively have significantly more arrival demands than others, so certain channels should have more effective information. Therefore, we utilize CA-gate to capture hidden information about the preferences of destinations.

"MaxPool" in Fig. 2 means compressing the third dimension of the input tensors to 1, and using the maximum value of all data in the channel as the new value. In contrast "AvgPool" uses the average value. This operation can be seen as compressing $\mathbb{R}^{(N_D,12,N_O)}$ OD tensors into $\mathbb{R}^{(N_D)}$ arrival demand vectors to directly analyze whether each area is likely to be selected as the destination. Then, two fully connected layers (represented as "FC" in Fig. 2) are stacked to learn from all channel information. Based on the consideration of information integration, the units number of the first layer is set to 15 for the network to learn the most useful information in the extracted destination features. And the second is to make the size of output the same as the input size. Besides, the addition of "reshape" is to rearrange the output from $\mathbb{R}^{75}$ vector to $\mathbb{R}^{(75,1,1)}$ tensor to facilitate subsequent element-wise multiplication. Subsequently, tensors obtained from the two branches of CA-gate are added and sent into the $\mathrm{Sigmoid}$ function to map its value between 0 and 1. Finally, the CA-gate output and the features extracted by the previous "Conv-BN" block are multiplied point by point to equip the learned features with destinations preference information and eliminate noise with minimal adversarial impact.

Similarly, for each studied time slices, the OD departure demand from some regions are significantly more than the rest. The second and third dimensions of current input represent time and Origin information, so an SA-gate is introduced to capture the probability of starting from different areas from the feature map. The first dimension of input tensors is compressed to 1 by maximizing/averaging (represented by Max/Avg in Fig. 2), that is, converting the OD demand $\mathbb{R}^{(N_D,12,N_O)}$ into the departure demand information on each time slice. The first "Re-shape" in Fig. 2 means to rearrange the 75-dimensional vectors representing the total N_O origins into $\mathbb{R}^{15,5}$ matrices according to the grids' location. The matrix is therefore converted to $\mathbb{R}^{12,15,5}$ tensor to facilitate the use of three-dimensional convolution to capture more accurate temporal and spatial information. Then, feature maps with size $\mathbb{R}^{(12,N_O)}$ obtained by the two compression methods are concatenated (represented as "II" in Fig. 2) and sent to a convolutional layer with kernel $(5, 5, 5)$ to summarize large-scale origin preferences. And the second "Re-shape" represents to restore the size from $\mathbb{R}^{(12,15,5)}$ to $\mathbb{R}^{(12,N_O)}$. Extracted preference information of the origins is equipped to the learned features just as the CA-gate. Finally, referring to the idea of the residual network, we add the final output with the initial input of ST-Conv, then a ReLU function is applied to calculate the output of the ST-Conv block with size $\mathbb{R}^{(75,12,75)}$.

4.4 S-GCN Block

To further extract global spatial features, as elaborated in Section 3.2, we project the traffic network onto a series of graphs $G = (V, E, A)$. Traditional GCN builds static graphs and extracts spatial relationships based on the assumption that the closer the geographical location between regions, the tighter the connection. However, in terms of transport networks, the regional correlation is dynamic, and some geographically distant regions have similar demand patterns. In the real world, there is a functionality label for each region, such as residential areas, CBD, school, etc. Primarily, this information can help express the spatial correlation of regional pairs. Specifically, areas with the same or similar labels may have stronger spatial correlation, and their variations over time are similar. For example, for districts with residential labels, the peak time of their departure demand is usually between 8-9 am. Previous work on traffic forecasting considered this type of relationship, but they utilized additional POI information [16], which is not equipped in most OD dataset. There is also work on railway OD prediction (predicting the volume of inflows and outflows at each station) using DTW to obtain time-series correlation for a static relationship [15]. This is similar to our idea, but the strength of DTW is to deal with unaligned sequences. For traffic data, the difference in the same time slice may be more effective for judging regional functionality. Therefore, the Pearson correlation coefficient is chosen in S-GCN. Based on this consideration, the traffic network is projected to a series graph, on which all regions are the corresponding vertex. And the strength of the corresponding edge is represented by the attribute similarity between regions.

Unlike static graphs of traditional GCN, the proposed S-GCN extracts dynamic information from graphs with constantly changing edges. Specifically, from the departure/arrival perspective, the edge represents the functional similarity of regions when they serve as origins/destinations, which is dynamically obtained based on the calculation of the time segments currently processed by the network. Then, the dynamic spatial relationship is extracted through graph convolution.

As described in Section 4.1, for all regions (vertices), the input data $\mathcal{M} = \{\mathcal{C}, \mathcal{P}, \mathcal{T}\}$ has 27 total time slices. From the departure perspective, the adjacency matrix in S-GCN describes the functional similarity of regions(origins), illustrated by the changing trends of the departure demand over time. The calculation process of the adjacency matrix A is as follows:

$$\mathcal{M} \in \mathbb{R}^{(T,O,D)} \overset{\text{permute}}{\longrightarrow} \mathcal{M}' \in \mathbb{R}^{(D,O,T)}, \tag{8a}$$

$$\mathcal{M}_{\mathcal{O}}(i,t) \in \mathbb{R}^{(O,T)} = \sum_{k=0}^{D} \mathcal{M}'(k,i,t), \tag{8b}$$

After transforming the dimension of the input data $\mathcal{M}$, all OD pairs corresponding to each origin on each time slice are summarized into the relevant departure demand $\mathcal{M}_{\mathcal{O}}(i,t)$, which indicates the value of all departure requirements for region i in a time slice t.

$$A_{\mathcal{M}}(i,j) \in \mathbb{R}^{(O,O)} = F_{corr}(\mathcal{M}_{\mathcal{O}}(i), \mathcal{M}_{\mathcal{O}}(j)), \tag{9a}$$

$$A'(i,j) = exp(|A_{\mathcal{M}}(i,j)| - 1), \tag{9b}$$

where F_{corr} denotes the function to calculate the correlation coefficient, which refers to the Pearson correlation coefficient. $\mathcal{M}_{\mathcal{O}}(i)$ is a time series representing the departure demand of area i. By calculating the correlation of the time series of an arbitrary regional pair, their functional similarity during this period is obtained. The larger the absolute value, the stronger the relationship. $A_{\mathcal{M}}(i,j)$ is the functional similarity of region i and j, $A_{\mathcal{M}}(j,i) = A_{\mathcal{M}}(i,j)$. Besides, Eq. (9b) is used to limit the range of values in the adjacency matrix A' to $(0, 1]$.

Subsequently, A' is filtered to remove edges of feeble connection to simplify graphs. All values less than 0.3 are reset to 0, and the corresponding adjacency matrix A of the current input graph is obtained.

Finally, the edges of the obtained graph represent the functional similarity of regions (origins or destination). This graph is a symmetric undirected graph, so we only need to apply a simple and effective graph convolution operation mentioned in Section 3.2 to extract the current spatial correlation on each time slice.

5 Case Studies

We first describe the dataset dedicated to OD prediction—NYC-TOD—created from a widely employed benchmark NYC dataset from the traffic domain. Subsequently, the evaluation metrics and related hyperparameters settings. Next, the comparison between the performance of the proposed model and the state-of-the-art approaches in long-term prediction is presented. To demonstrate the effectiveness of each component, we implement some variants and measure the related performance. Then, to further certificate the superiority of the proposed model in long-term prediction, we use the trained model and predicted value to perform the OD demand prediction for every 30 minutes of the future 6-12 hours and evaluate. Finally, we compare the performance of the proposed method with CSTN on the Didi Travel Haikou dataset to show its performance on other taxi OD demand dataset.

5.1 Dataset and Experiment Settings

5.1.1 NYC-TOD

NYC-TOD is among the most welcoming benchmark datasets for urban taxi OD prediction based on the NYC Taxi dataset and meteorological data of New York Central Park from Wunderground[2] [18]. Specifically, it includes records of yellow taxi journeys in Manhattan in 2014 and the weather data for the same period. The investigated area is evenly divided into 15 × 5 non-overlapping grids, each with a geographic size of about $0.75km \times 0.75km$. Besides, the length of a time slice is set to 30 min, NYC-TOD contains OD data of size $\mathbb{R}^{(17520,75,75)}$ where $365 \times 48 = 17520$ time segments. Moreover, Wunderground offers 29 types of meteorology data, namely temperature, wind chill, humidity, visibility, wind speed, precipitation, and 23 one-hot encoded weather conditions. The first 6 types of data are pre-processed by min-max linear normalization. Finally, NYC-TOD contains the pre-processed meteorological data with a size of $\mathbb{R}^{(17520,29)}$.

5.1.2 Didi Chuxing GAIA Data

To further illustrate the effectiveness of the proposed model and exclude the randomness of the NYC-TOD dataset, we utilize the proposed method to conduct experiments on the DiDi travel data set provided by DiDi's GAIA plan and compared the results with CSTN. Didi Chuxing GAIA Data contains the daily real order data in Haikou City from May 1st to October 31st, 2017. We first performed data cleansing, deleted empty orders and orders outside Haikou according to the latitude and longitude. Items whose order time equals 0 are also excluded. Besides, the surveyed area is equally divided into 16 × 8 non-overlapping areas, and the time slice length is set to 30 min.

5.1.3 Pre-processing and Hyperparameters Settings

Following existing researches on OD prediction and traffic prediction [8], [10], [11], [18], we apply the min-max normalization to scale the value of OD demand to the range $[-1, 1]$. In the paper, we utilize the data of the last 60 days as the testing data, and all previous data for training. Specifically, for the performance comparison of different methods on the NYC-TOD dataset, we all use the same training data and test data. As mentioned in Section 3 and Section 4, $\{\mathcal{C}, \mathcal{P}, \mathcal{T}\}$ is

[2] https://www.wunderground.com/

selected to predict $k = 12$ future OD demand at the same time, and the length of time slices $\mathcal{C}, \mathcal{P}, \mathcal{T}$ is $3, 12, 12$, respectively. Mini-batch training applied in the training phase, the size of a batch is set to 32, and optimizer Adam [42] selected. Besides, we adaptively change the learning rate. The initial value is set to 0.01, and it will be one-tenth of the previous value for every 20 training epochs until 80 epochs elapsed.

5.1.4 Metrics

Referring to most studies on OD demand prediction [8], [10], [11], [18], two commonly utilized performance metrics are employed in the paper, namely, Rooted Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE):

$$\mathrm{RMSE}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{t}) = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\mathbf{y}_{t,i,j} - \hat{\mathbf{y}}_{t,i,j})^2}, \quad (10a)$$

$$\mathrm{MAPE}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{t}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{|\hat{\mathbf{y}}_{t,i,j} - \mathbf{y}_{t,i,j}|}{\mathbf{y}_{t,i,j} + \epsilon}, \quad (10b)$$

where $\epsilon = 10^{-3}$ to prevent the denominator being zero, $\mathbf{y}$ is the ground truth value, and $\hat{\mathbf{y}}$ is the predicted result. Besides, it is worth noting that following the researches on OD prediction and traffic demand prediction [5], [8], [10], [11], [18], [30], screening operation is carried out in the calculation of MAPE since the tiny demand in real life has no practical significance. If the ground truth is less than 5, the corresponding OD demand at the specific time slice will not be included.

5.2 Comparison with the State-of-the-art Approaches

The proposed method is compared with the state-of-the-art approaches and several classical models as follows:

5.2.1 Compared Models

- Convolutional Neural Network (CNN) [31]: CNN is a popular deep learning model for spatio-temporal data, which is widely used to extract features from regular structure data. Therefore, many classic traffic forecasting models are based on it. We set the number of convolution kernels of each convolution layer to $32, 32, 32, 32, 32$ and the kernel size $(3, 3)$.
- Spatial-Temporal Residual Network (ST-ResNet) [8]: ST-ResNet is a more representative method in the research of traffic demand prediction, which introduces shortcut connections and CNNs to extract spatio-temporal information for prediction.
- Bi-directional Recurrent Neural Network (BRNN) [43]: For each time t, the input of BRNN will be provided to two RNNs in opposite directions at the same time, and the output is determined by the two one-way RNNs while obtaining information from the backward and forward states.
- Convolutional LSTM (ConvLSTM) [44] uses the output of the convolutional network as the input of the LSTM, so that the data passing through the convolutional network can use the recurrent neural network LSTM to have a good performance on the time series.
- Diffusion Convolutional Recurrent Neural Network (DCRNN) [9]: DCRNN utilizes a two-way random walk to model spatial dependence and encoder-decoder architecture for temporal dependency.
- Spatio-temporal Graph Convolutional Network (ST-GCN) [39]: ST-GCN introduces a gated linear unit GLU [45] in full convolution layer to capture the time relationship, and uses graph convolution to capture the spatial relationship for mid- and long-term traffic prediction.
- ST-GCN2: ST-GCN predicts one future OD data and iterates the predicted value into the subsequent prediction process following previous long-term prediction methods (like LSTM). Since the LSTM unit is not included in ST-GCN, we apply the CNN with gating mechanisms proposed in this paper to directly extend ST-GCN from short-term prediction to one-step long-term prediction. To distinguish, this variant is represented by ST-GCN2. Alternatively, ST-GCN2 directly modifies the final output channel number of ST-GCN to allow the model to predict 12 future OD requirements simultaneously.
- Contextualized Spatial-Temporal Network (CSTN) [18]: CSTN combines CNN and LSTM to predict OD demand, which is state-of-the-art. When it extracts spatial relationships, the model considers not only spatial relations in a small scale but also global spatial correlations. We use the recommended settings suggested by the author for comparison.

For each baseline method, the hyperparameters are selected by trial-and-error. The parameters, that is, the number of layers of each network and the number of corresponding filters, is tuned for all methods. And different learning rates $(0.1, 0.01, 0.001, 0.0001)$ have also been tested. Additionally, RMSprop [46] or Adam [42] are respectively used as the optimizer to view the results. The best case is selected for each baseline method.

5.2.2 Overall Comparison

The performance of the proposed ST-GDL and the other compared methods are summarized in Table 2. For the NYC-TOD dataset, in the next-step short-term forecast (30 mins), the performance of ST-GDL is only slightly worse than CSTN by 0.92% MAPE. However, concerning the following predictions, ST-GDL outperforms all compared approaches. The improvement significantly increases with the length of the prediction window, which shows the superior long-term OD prediction ability of ST-GDL. Besides, ST-GDL achieves the lowest RMSE among all methods, as shown in Table 3. The error of CSTN increases by $[0.4\%, 1.4\%]$ with each prediction step into the future. In contrast, our method increases by only $[0.07\%, 0.37\%]$, which indicates that the proposed model can effectively alleviate the error accumulation problem of LSTM in long-term OD prediction tasks.

For mid- and long-term OD prediction, the performance of CNN is the worst among the deep learning methods because it only considers the spatial relationship provided by the geographic location, and the data embedding method cannot retain the spatial relationship. Besides, it does not take into account that the OD demand at each moment is more relevant to its closest time slice, which leads to reduced accuracy of long-term prediction. Compared to CNN,

TABLE 2
Comparison of MAPE with the State-of-the-art Approaches on NYC-TOD

MAPE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
CNN	29.51%	32.56%	37.62%	42.42%	47.78%	52.56%	57.56%	61.82%	65.82%	69.38%	72.53%	75.03%
BRNN	29.05%	30.17%	31.37%	31.93%	32.60%	34.15%	36.90%	38.23%	41.40%	45.58%	46.32%	49.45%
ConvLSTM	28.71%	28.96%	29.87%	30.51%	31.58%	33.13%	33.97%	34.72%	35.90%	36.82%	37.77%	38.93%
ST-ResNet	29.00%	30.97%	35.12%	38.67%	41.58%	44.26%	46.29%	48.06%	49.48%	50.70%	51.69%	52.51%
DCRNN	28.49%	28.90%	29.35%	29.98%	30.72%	32.01%	32.79%	34.11%	35.09%	36.28%	36.89%	38.01%
ST-GCN	29.12%	30.44%	32.56%	35.8%	39.72%	43.93%	47.97%	51.77%	55.25%	58.44%	61.16%	63.46%
ST-GCN2	29.69%	30.36%	30.96%	31.76%	32.16%	32.53%	32.73%	32.91%	33.13%	33.48%	33.89%	34.51%
CSTN	**27.22%**	28.61%	29.15%	29.97%	30.90%	31.78%	32.62%	33.34%	34.03%	34.60%	35.13%	35.54%
ST-GDL	28.14%	**28.52%**	**28.89%**	**29.22%**	**29.49%**	**29.65%**	**29.86%**	**30.01%**	**30.10%**	**30.23%**	**30.40%**	**30.48%**

TABLE 3
Comparison of RMSE with the State-of-the-art Approaches on NYC-TOD

RMSE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
CNN	1.1468	1.5249	2.1098	2.1356	2.3437	2.3908	2.4792	2.5387	2.5952	2.6422	2.6879	2.7262
BRNN	1.8159	1.8485	1.8703	1.9006	1.9720	2.3781	2.3921	2.4047	2.5240	2.6297	2.6939	2.9522
ConvLSTM	1.5412	1.5792	1.6157	1.6573	1.6763	1.6827	1.6983	1.7094	1.7240	1.7391	1.7574	1.7762
ST-ResNet	1.1799	1.4742	1.8406	2.0568	2.1176	2.1811	2.2045	3.2308	2.2435	2.2573	2.2659	2.2747
DCRNN	1.1244	1.4578	1.4895	1.5089	1.5279	1.5433	1.5589	1.5701	1.5841	1.6000	1.6105	1.6172
ST-GCN	1.0259	1.0458	1.0760	1.1200	1.1716	1.2253	1.2772	1.3270	1.3737	1.4184	1.4597	1.4980
ST-GCN2	1.0354	1.0452	1.055	1.0659	1.0714	1.0762	1.0788	1.0827	1.0865	1.0922	1.1000	1.1125
CSTN	1.3245	1.3865	1.4487	1.5021	1.5606	1.6134	1.6613	1.702	1.7393	1.7695	1.7992	1.8249
ST-GDL	**1.0120**	**1.0211**	**1.0301**	**1.0385**	**1.0435**	**1.0480**	**1.0513**	**1.0543**	**1.0574**	**1.0603**	**1.0621**	**1.0622**

ST-ResNet introduces a significant accuracy improvement, which is based on residual network and combines traffic data of temporal closeness, period, and trend properties to predict future traffic flow jointly. This shows that the introduction of data with multiple time attributes can benefit OD prediction, especially for the long-term one. Additionally, CNN is limited because it cannot be used directly in non-Euclidean spaces like transportation networks. Usually, the transportation network needs to be converted into a grid map according to geographic coordinates or other rules before processed by CNN. GCN generalizes CNN to graphs and is more suitable for transportation networks. ST-GCN is a hybrid model that applies GCN and CNN to model the spatio-temporal information of traffic data jointly. According to the comparison results of ST-GCN and ST-GCN2, the difference between their MAPE reached a maximum 28.95% when predicting the OD data of the sixth upcoming hour. This is because when directly predicting multiple future demands at once, the model is learned from the ground truth, and the model learns the influence of historical data on long-distance time slices during the training process. While models based on LSTM learns an orderly relationship. As the iteration progresses, the predicted value of the previous segment will be used to calculate the prediction in the later stage, leading the accumulation of errors increases faster. Although ST-GCN uses CNN instead of LSTM, when it is used for long-term prediction, it is still a multi-step prediction that causes error accumulation. Besides, its short-term prediction results are inferior to CSTN. Due to the further accumulation of errors, the result of CSTN is better than ST-GCN for long-term prediction. On the contrary, ST-GCN2 applies one-step prediction instead of multi-step prediction to avoid the error accumulation problem caused by iteration. Although its short-term prediction performance is not as good as CSTN, its accumulated error over time is less than that of CSTN. Additionally, it can be found that CSTN is better than previous methods from the MAPE results, but it is inferior to ST-GCN from the RMSE results. This may happen that in actual applications, if there are individual outliers with very large deviations, even if the number of outliers is very small, it will lead a worse RMSE. The LSTM-based model will rely to a large extent on the information of the most recent time slice, but the OD demand distribution for taxis is unevenly distributed, and the difference between rush hour and non-rush hour is very large, so outliers may appear and cause RMSE to be high. However, the differences are small, and MAPE reduces the impact of individual outliers, so CSTN is still excellent. And because LSTM values the most recent time slice, CSTN has shown its superiority in short-term prediction (30 min).

Additionally, all experiments are conducted on a GPU server with NVIDIA GeForce RTX 2080Ti graphic processing cards for neural network training acceleration. ST-GDL requires 482s to finish the training, and takes approximately 14s to obtain all the prediction results of the next 12 time segments. Considering that the time span is over six hours, the inference time is considered moderate.

To show that the proposed model can converge after a few training iterations, Fig. 3 visualizes the loss (MSELoss) of each epoch for the proposed model during training, which has a stable downward trend. Fig. 4 depicts the results of calculating the relevant RMSE and MAPE metrics for the validation set without min-max using the currently trained parameters after the completion of each epoch. The upper right corner of the figure is an enlarged display of the verification results of the $15^{th} - 50^{th}$ epochs. It can be

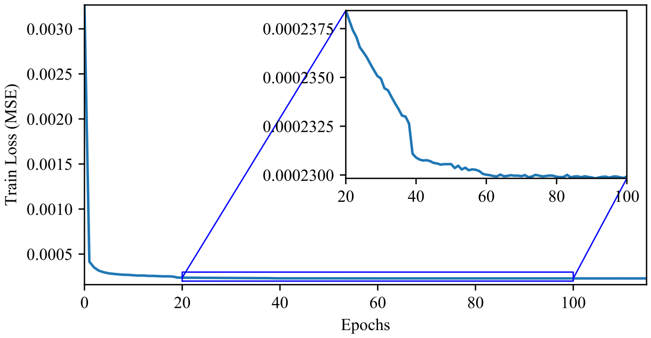

Fig. 3. Visualization of the loss of each epoch in the training process

observed that after about 20 epochs, the model has reached a stable state, and the predicted OD demands of the nearest time slices are closer to the actual situation. With the expansion of the time range, the prediction accuracy gradually decreases, but the cumulative error becomes smaller.

5.3 Sensitivity of Hyperparameters

In this work, a deep model based on graph deep learning is adopted. We empirically select some hyperparameters and parameters when constructing this long-term OD predictor. In this subsection, the sensitivity and influence of these settings to the prediction accuracy are investigated. We tested the different selection of three important hyperparameters in the construction of the model. The parameters are the window length of the selected closest time slices, the predicted time window length, and the number of channels for the convolution network. Specifically, they are set to 3/4/5, 32/75/128, 4/6/8/12, separately. The results are shown in Fig. 5, where the solid line is the MAPE result of short-term prediction (30 min), and the dashed line is the average value of MAPE of all time slices predicted. It can be seen that more historical time slices will improve the short-term prediction results to some extent, but it will have an adverse effect on the long-term prediction. This seems to indicate that long-term traffic changes depend more on short-term history. Also, using historical 3 short-term time slices for prediction, when the number of time slices to be predicted increases (4/8/12), the impact on the short-term results is small, but due to the inevitable accumulation of errors, the average MAPE of all predictions will get worse. In general, however, the results of MAPE/RMSE are still small when scattered to each time slice. However, when one day is predicted, the accuracy will be reduced. Generally, getting the forecast results for the next 12 hours is sufficient for practical applications, and the results are within a tolerable range. Besides, it can be found that more convolution kernels do not mean better performance. Although compared with 32 convolution kernels, 75 convolution kernels bring performance improvements. It is meaningless to continue to increase the number of convolution kernels and will also lead to time-consuming training.

5.4 Ablation Test

To verify the validity of each component of the ST-GDL model for long-term OD prediction and confirm our motivation mentioned in Section 4, we conduct a series of ablation tests. The results are shown in Table 4 and Fig. 6. It can be seen from the figure that the removal of any component has an adversarial effect on short-term OD prediction. However, as time increases, the performance enhancement of MAPE brought by these components becomes more prominent, which shows that they are more useful for long-term OD forecasting than short-term prediction.

5.4.1 With/without Multi-perspective

The ST-GDL model contains two types of multi-view modeling. One is about diverse time attributes, and the other is on the different regional attributes.

5.4.1.1 With/without Multiple Time Attributes: We select OD data of three temporal attributes as input data and send them to the same sub-network module respectively to extract and combine features. To show that the consideration of multiple time attributes of the data can effectively improve the prediction accuracy, we test the prediction results obtained by removing the daily and weekly data segments and the corresponding network branches that process them, which is represented by "-PT" in Table 4. It can be seen from the results that the future OD demand is most related to several recent historical OD data. However, data with temporal attributes of period/trend can bring performance improvements, whether for short-term or long-term forecasts. This is because the OD data has a certain periodicity; it usually has a similar changing trend on weekdays/weekends. The introduction of a multi-perspective of temporal attributes allows the network to learn the related dynamics.

5.4.1.2 With/without Origin/destination Perspective: In ST-GDL, we propose a multi-view modeling of regional attributes. We consider that the areas have different correlations when they are viewed as departure points or arrival ones. To verify this idea, we construct models from either the view of origins or destinations and compare their performance, which is denoted by the "-D perspective" and "-O perspective" in Table 4. It can be seen the OD prediction performance accuracy notably reduced when either perspective is removed from ST-GDL. Our hypothesis has been confirmed: the correlation between regions is different when they are regarded as origins or destinations, and the combination of these two correlations contributes to long-term OD prediction.

5.4.2 With/without Gate Mechanism

As described in Section 4.3, two gating mechanisms are introduced to the ST-Conv module to extract more useful hidden information and eliminate unrelated features. We perform experiments on removing SA-gate and CA-gate separately, with the rest of the model unchanged to observe their own influence. The results indicate that they bring practical improvements to long-term OD prediction performance, of which CA-gate is more effective for long-term predictions.

5.4.3 With/without S-GCN

To show that S-GCN can help obtain more spatial correlation, we remove the S-GCN block from the model and evaluate the new performance, which is denoted by "-S-GCN"

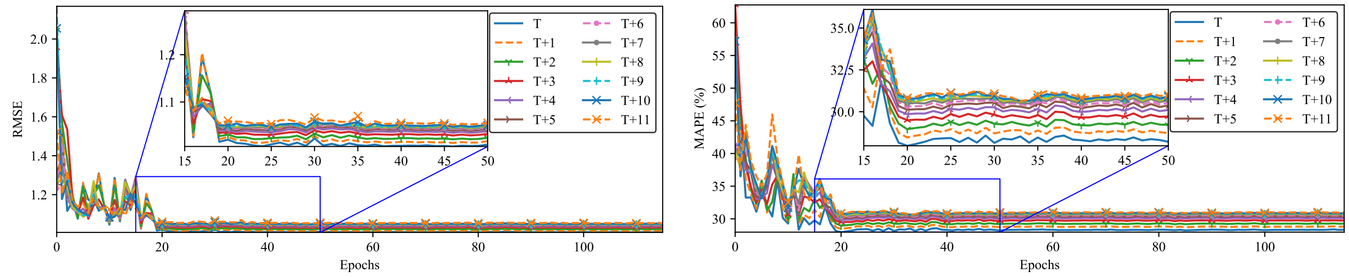

Fig. 4. Visualization of the training process.

TABLE 4
MAPE Comparison of Ablation Tests on NYC-TOD

MAPE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
- PT	28.35%	28.94%	29.49%	30.04%	30.36%	30.77%	31.14%	31.43%	31.62%	31.74%	32.04%	32.22%
- D perspective	28.66%	29.07%	29.51%	29.85%	30.23%	30.46%	30.72%	30.88%	31.06%	31.27%	31.41%	31.55%
- O perspective	28.74%	29.08%	29.5%	29.87%	30.13%	30.35%	30.56%	30.77%	30.95%	31.18%	31.41%	31.67%
- SA gate	28.23%	28.70%	29.05%	29.37%	29.64%	29.9%	30.07%	30.24%	30.42%	30.56%	30.75%	30.86%
- CA gate	28.10%	28.66%	29.20%	29.67%	29.99%	30.24%	30.46%	30.55%	30.63%	30.77%	30.98%	31.10%
- S-GCN	28.28%	28.79%	29.24%	29.68%	30.04%	30.36%	30.62%	30.83%	30.97%	31.10%	31.22%	31.39%
- external	28.20%	28.66%	29.07%	29.45%	29.70%	29.98%	30.24%	30.42%	30.51%	30.59%	30.68%	30.74%
ST-GDL	**28.14%**	**28.52%**	**28.89%**	**29.22%**	**29.49%**	**29.65%**	**29.86%**	**30.01%**	**30.10%**	**30.23%**	**30.40%**	**30.48%**

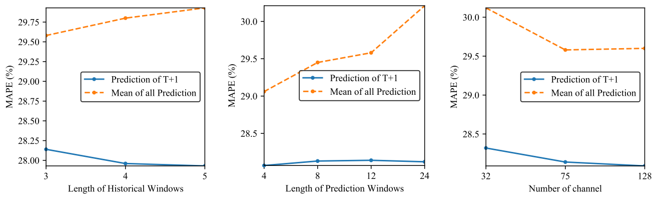

Fig. 5. Prediction with different hyper-parameters.

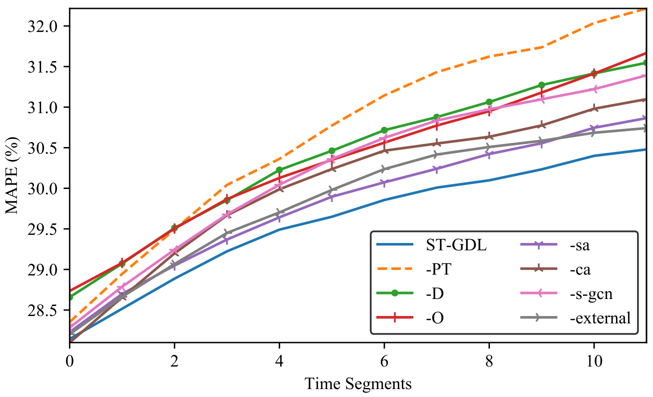

Fig. 6. MAPE metric of Ablation Tests

in Table 4. From the comparison, it can be seen that the addition of graph convolution brings a notable performance improvement. Hence, we can confirm that the regions with the same hidden attributes have similar characteristics that affect the OD demand. Besides, in order to illustrate that the proposed dynamic relationship contributes more to OD prediction than a static relationship, we compare the prediction results with adopting an adjacency matrix calculated by all training data instead of the current input of the network referring to [15]. Functional relationship graphs of all regions at every time slice share the same edge connection. The result comparison is shown in Table 5. It can be seen that introduction of the traditional GCN, that is, the guidance of the static relationship, the prediction performance has not been greatly optimized. This is because the relationship has been equipped with the learned features through the superposition of the previous convolution operation.

5.4.4 With/without External Information

Finally, we compare the performance of the model when the meteorological data is utilized or not. The results confirm a widely-recognized conclusion by the previous literature that introducing external information brings more accurate predictions. Nonetheless, the performance of ST-GDL mainly depends on the spatio-temporal relationship extracted from historical OD data; a minor MAPE improvement is observed from the external information for short-term OD prediction.

5.5 Longer-term OD Prediction Beyond 6h

Our model predicts the future OD demands of 12-time slices (6h) at one time to provide sufficient information for practical applications. However, to further illustrate the superiority of the proposed model in long-term prediction, we perform a more distant long-term forecast and compare the results with CSTN, the previous state-of-the-art. The OD demand of the last three time slices in the predicted data and the model learned are utilized to predict the OD demand of 6h-12h in the future. The results are shown in Fig. 7. Table 6 and Table 7 are the comparison results of MAPE and RMSE, respectively. It can be seen that for the

TABLE 5
MAPE Comparison with Static adjacency matrix on NYC-TOD

MAPE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
S-GCN	**28.14%**	**28.52%**	**28.89%**	**29.22%**	**29.49%**	**29.65%**	**29.86%**	**30.01%**	**30.1%**	**30.23%**	**30.4%**	**30.48%**
TraditionGCN	28.24%	28.71%	29.22%	29.66%	29.94%	30.27%	30.55%	30.77%	30.84%	31.09%	31.19%	31.23%

TABLE 6
MAPE Comparison of Long-term Prediction Beyond 6h on NYC-TOD

MAPE	6h30min	7h	7h30min	8h	8h30min	9h	9h30min	10h	10h30min	11h	11h30	12h
CSTN	35.93%	36.20%	36.42%	36.56%	36.67%	36.77%	36.85%	36.92%	37.01%	37.10%	37.25%	37.47%
ST-GDL	**30.64%**	**30.81%**	**30.96%**	**30.99%**	**31.07%**	**31.10%**	**31.20%**	**31.29%**	**31.30%**	**31.34%**	**31.46%**	**31.57%**

TABLE 7
RMSE Comparison of Long-term Prediction Beyond 6h on NYC-TOD

RMSE	6h30min	7h	7h30min	8h	8h30min	9h	9h30min	10h	10h30min	11h	11h30	12h
CSTN	1.8496	1.8721	1.8941	1.9137	1.9311	1.9473	1.9614	1.9731	1.984	1.994	2.0047	2.0178
ST-GDL	**1.0637**	**1.0694**	**1.0746**	**1.0780**	**1.0799**	**1.0820**	**1.0832**	**1.0838**	**1.0841**	**1.0843**	**1.0852**	**1.0888**

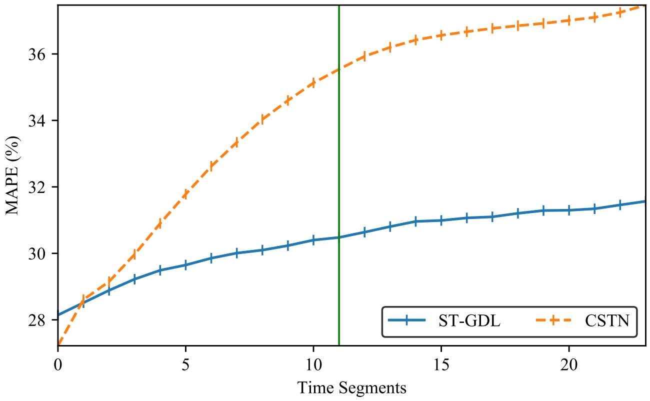

Fig. 7. Comparison of long-term OD prediction beyond 6h

long-term OD prediction over 6h, our method shows a more significant advantage. For further long-term OD prediction, the performance of the proposed model is better than CSTN by $[0.78, 0.93]$ RMSE and $[5.29\%, 5.91\%]$ MAPE. This indicates that the proposed ST-GDL is capable of performing long-tern OD prediction over six hours, and significantly outperform existing approaches.

5.6 Performance on DiDi dataset

Finally, to further confirm the usability and scalability of the proposed model, we also conducted experiments of the DiDi Chuxing dataset and compared the performance with CSTN. The results are shown in Table 8 and Table 9. It can be seen that the RMSE and MAPE results obtained by the proposed method are better than CSTN. However, the result on the DiDi dataset is inferior to that of NYC-TOD. This is because the training samples of the Didi dataset are relatively small, and the coverage area is a certain part of Haikou, and its OD data is more sparse and unevenly distributed.

6 Conclusion

This paper proposes a novel urban long-term OD prediction model based on graph deep learning technology. Compared with the state-of-the-art long-term OD prediction methods based on deep learning, the proposed ST-GDL model integrates CNN and graph convolution to predict both short- and long-term OD demand simultaneously. We classify the historical OD data into closeness, periodicity, and trend to extract spatio-temporal information from various time scales. Specifically, traffic dynamics are captured by CNN with gate mechanisms designed to learn the potential origins/destinations preference on each time slice. Furthermore, a series of graphs with constantly changing edges are constructed according to the underlying functional similarity of regions and processed by graph convolution. The proposed model inherits the advantages of both CNN and graph convolution. Experiments on a real-world dataset show that the proposed ST-GDL develops more accurate long-term OD predictions than baseline approaches with the same volume of training data. Ablation tests are carried out to validate the efficacy of ST-GDL components. Finally, we investigate the performance of ST-GDL for further long-term OD predictions.

In the future, we plan to introduce Graph Attention Network in ST-GDL to develop a more scalable model. Additionally, POI information to generate more reliable regional correlation information will be incorporated.

References

[1] L. Mussone, S. Grant-Muller, and H. Chen, “A neural network approach to motorway od matrix estimation from loop counts,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 10, no. 1, pp. 88 – 98, 2010.

TABLE 8
MAPE Comparison of DiDi dataset

MAPE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
S-GCN	**32.75%**	**33.15%**	**33.94%**	**35.03%**	**35.61%**	**35.93%**	**37.16%**	**37.81%**	**38.47%**	**39.23%**	**39.69%**	**40.20%**
CSTN	33.64%	34.75%	35.08%	35.57%	36.20%	37.00%	37.89%	38.62%	39.83%	40.99%	42.08%	42.70%

TABLE 9
RMSE Comparison of DiDi dataset

MAPE	30min	60min	90min	120min	150min	180min	210min	240min	270min	300min	330min	360min
S-GCN	**0.2609**	**0.2639**	**0.2662**	**0.2696**	**0.2704**	**0.2721**	**0.2735**	**0.2739**	**0.2747**	**0.2751**	**0.2771**	**0.2799**
CSTN	0.2655	0.2779	0.3082	0.3247	0.3439	0.3618	0.3803	0.3966	0.4165	0.4359	0.4537	0.4696

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, “Big data analytics in intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.

[3] X. Fu, H. Yang, C. Liu, J. Wang, and Y. Wang, “A hybrid neural network for large-scale expressway network od prediction based on toll data,” *PLoS ONE*, vol. 14, no. 5, May. 2019.

[4] D. Li, J. Cao, R. Li, and L. Wu, “A spatio-temporal structured lstm model for short-term prediction of origin-destination matrix in rail transit with multisource data,” *IEEE Access*, vol. 8, pp. 84 000–84 019, 2020.

[5] Z. Duan, K. Zhang, Z. Chen, Z. Liu, L. Tang, Y. Yang, and Y. Ni, “Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time,” *IEEE Access*, vol. 7, pp. 127 816–127 832, Aug. 2019.

[6] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow prediction in spatio-temporal networks based on multitask deep learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2020.

[7] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis,” in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, Honolulu, Hawaii, Jul. 2019, pp. 1020–1027.

[8] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, Ithaca, USA, 2017, pp. 1655–1661.

[9] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *Proc. ICLR 2018*, Vancouver, Canada, 2018.

[10] K. Chu, A. Y. S. Lam, and V. O. K. Li, “Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–14, Jul. 2019.

[11] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, “Predicting multi-step citywide passenger demands using attention-based neural networks,” in *Proc. WSDM 2018*, CA, USA, 2018, pp. 736–744.

[12] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, pp. 922–929, Jul. 2019.

[13] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, Nov. 2019.

[14] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, “Multi-range attentive bicomponent graph convolutional network for traffic forecasting.” in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA, 2020, pp. 3529–3536.

[15] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, “Physical-virtual collaboration modeling for intra- and inter-station metro ridership prediction,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[16] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, CA, USA, 2019, pp. 3656–3663.

[17] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du, and L. Lin, “Dynamic spatial-temporal representation learning for traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[18] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, “Contextualized spatial–temporal network for taxi origin-destination demand prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3875–3887, Oct. 2019.

[19] M. Levin and Y.-D. Tsao, “On forecasting freeway occupancies and volumes,” *Transportation Research Record*, vol. 773, pp. 47–49, 01 1980.

[20] B. Williams, “Multivariate vehicular traffic flow prediction: Evaluation of arimax modeling,” *Transportation Research Record*, vol. 1776, pp. 194–200, Jan. 2001.

[21] B. Williams and L. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, pp. 664–672, Nov. 2003.

[22] Y. Shu, M. Yu, O. YANG, J. Liu, and H. Feng, “Wireless traffic modeling and prediction using seasonal arima models,” *IEICE Transactions on Communications*, vol. 88, no. 10, pp. 3992–3999, Jan. 2003.

[23] I. Okutani and Y. Stephanedes, “Dynamic prediction of traffic volume through kalman filtering theory,” *Transportation Research Part B: Methodological*, vol. 18, pp. 1–11, Feb. 1984.

[24] J. Guo, W. Huang, and B. Williams, “Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, Jun. 2014.

[25] A. Azzouni and G. Pujolle, “A long short-term memory recurrent neural network framework for network traffic matrix prediction,” *ArXiv*, vol. abs/1705.05690, 2017.

[26] H. Chang, Y. Lee, B. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences,” *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 292–305, Sep. 2012.

[27] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dongk, “An improved k-nearest neighbor model for short-term traffic flow prediction,” *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 653–662, Nov. 2013.

[28] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, “Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions,” *Expert Systems With Applications*, vol. 36, no. 3, pp. 6164–6173, Jul. 2009.

[29] X. Luo, D. Li, and S. Zhangn, “Traffic flow prediction during the holidays based on dft and svr,” *Journal of Sensors*, vol. 2019, pp. 1–10, Jan. 2019.

[30] Z. He, C. Y. Chow, and J. Zhang, “Stann: A spatio-temporal attentive neural network for traffic prediction,” *IEEE Access*, vol. 7, pp. 4795–4806, Jan. 2019.

[31] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Nevada, USA, 2012, pp. 1097–1105.

[32] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote

microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, May. 2015.

[33] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, Z. Li, J. Ye, and D. Chuxing, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, USA, 2018, pp. 2588–2595.

[34] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, CA, USA, 2019, pp. 5668–5675.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[36] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014. [Online]. Available: https://arxiv.org/abs/1409.2329

[37] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction." in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA, 2020, pp. 1234–1241.

[38] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR 2017*, Toulon, France, 2017, pp. 1–14.

[39] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. IJCAI 2018*, Melbourne, Australia, 2018, pp. 3634–3640.

[40] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, Aug. 2019.

[41] 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 *Proc. 25th ACM SIGKDD 2019*, New York, USA, 2019, pp. 1227–1235.

[42] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR 2015*, San Diego, CA, United states, 2015, pp. 1–15.

[43] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. K. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Cambridge, United States, 2015, pp. 802–810.

[44] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on*, vol. 45, pp. 2673 – 2681, Dec. 1997.

[45] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. Dauphin, "Convolutional sequence to sequence learning," in *Proc. ICML 2017*, vol. 70, Sydney, Austrlia, Aug. 2017, pp. 1243–1252.

[46] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," 2012. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf