This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Adaptive Spatiotemporal Dependence Learning for Multi-Mode Transportation Demand Prediction

Haihui Xu, Tao Zou, Mingzhe Liu, *Graduate Student Member, IEEE*, Yanan Qiao, Jingjing Wang, and Xucheng Li

***Abstract*— Due to the increasing diversification of urban transportation modes, many urban areas have the problem of unbalanced traffic demand, which makes accurate prediction of traffic demand very important. However, most of the existing studies focus on improving the prediction accuracy of traffic demand on the single spatial relationship of a single traffic mode, ignoring the diversity of spatial relationships and the heterogeneity of transportation stations in the traffic network. In this paper, we propose a Co-Modal Graph Attention neTwork(CMGAT) framework to uncover the impact of different spatial relationships and traffic mode interactions on traffic demand. Specifically, we first utilize a feature embedding block to capture the semantic information from several features. Then, a multiple traffic graphs-based spatial attention mechanism and a multiple time periods-based temporal attention mechanism are proposed to capture spatial and temporal dependencies in multi-mode traffic demands. Moreover, an output layer is provided to incorporate the hidden states and raw time sequences to predict future traffic demand. Finally, we conduct experiments on two real-world datasets, NYC Bike and NYC Taxi, and the results not only demonstrate the superiority of our model, but also indicate the necessity of considering multiple spatial relationships and traffic modes.**

***Index Terms*— Traffic demand prediction, spatial-temporal attention mechanism, deep learning.**

I. Introduction

IN RECENT years, with the development of public transportation, a variety of traffic modes have appeared and the problem of unbalanced traffic demand has been more and more serious. Therefore, accurate traffic demand prediction is one of the most necessary issues in the urban area. It can help taxi drivers avoid staying in the low demand region for a long time, which reduces energy consumption and passengers waiting time. Simultaneously, passengers may be more easy to find a ride on rush-hour in their area. To relief the imbalance between demand and supply, and help build the intelligent transportation system, several efforts have been made to solve the problem of traffic demand prediction in recent years.

In early studies of the existing methods, traffic demand forecasting can fall into two categories. On the one hand, traditional time series methods such as auto-regressive (AR), moving average (MA), and autoregressive moving average model (ARIMA) [1] mainly adopted the time series which are stationary or become stationary through differencing. On the other hand, machine learning methods like k-nearest neighbor model (KNN) [2], support vector machines (SVM) [3], Kalman Filter [4] were proposed to get more accurate forecasting results and predict more complex series. Zheng and Ni [5] used a multi-task framework in learning the temporal dynamics of traffic travel costs. Li *et al.* [6] predicted rents and returns in bike-sharing system by gradient boosting decision tree model (GBRT). However, these methods just focused on temporal dependencies and ignored other aspects such as spatial relationship.

Recently, deep learning models have been more mature which can deal with complex and nonlinear relationship better. Yu *et al.* [7], Laptev *et al.* [8] and Cui *et al.* [9] used recurrent neural network (RNN) to model temporal dependency. Another part of these methods use convolution neural network (CNN) to extract spatial features [10], [11]. Besides, some methods combined CNNs with RNNs in capturing spatial-temporal dependencies simultaneously [12]–[15]. Nevertheless, the structure of studied region should be regarded as a topological graph, which contains some non-Euclidean attributes and CNN fails to analyze them. For example, there is a similar traffic pattern between non-adjacent traffic stations while CNN just studies the information around the stations.

With the proposal of graph convolution network (GCN) [16], it is useful to handle non-Euclidean structure. In the traffic domain, much more effective traffic information on the studied area can be exacted and various work based on spatial-temporal graph neural networks have been proposed [17]–[19] recently. However, a model trained in the specific structure can not deal with the other graphs in different structures. Therefore, attention mechanism [20] is applied into graph for modeling spatial dependencies with variable size input [21]–[23]. But most attention-based methods ignore the multiple spatial dependencies and

Manuscript received June 15, 2021; revised December 26, 2021; accepted February 9, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 51822802, Grant 51991395, Grant 71901011, and Grant U1811463; and in part by the Guangxi Innovation-Driven Development Special Fund Project under Grant AA18118053. The Associate Editor for this article was G. Guo. *(Corresponding author: Xucheng Li.)*

Haihui Xu and Jingjing Wang are with the Beijing Municipal Transportation Operations Coordination Center (TOCC), Beijing 100028, China (e-mail: xuhaihui@jtw.beijing.gov.cn; wangjingjing@jtw.beijing.gov.cn).

Tao Zou, Mingzhe Liu, and Yanan Qiao are with the State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zoutao@buaa.edu.cn; mzliu1997@buaa.edu.cn; qiaoyanan@buaa.edu.cn).

Xucheng Li is with Shenzhen Urban Transport Planning Center Company Ltd., Shenzhen 518101, China (e-mail: xucheng.li@sutpc.com).

Digital Object Identifier 10.1109/TITS.2022.3155753

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

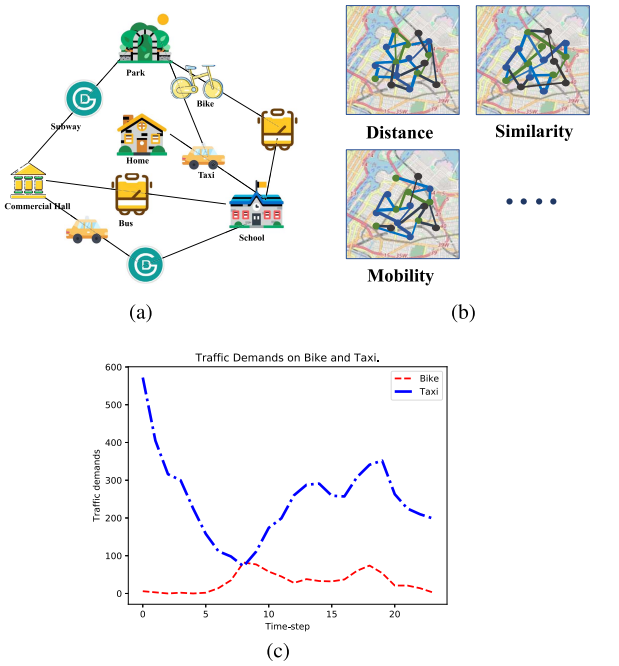


Fig. 1. (a) represents the multiple modes people can transfer when traveling in real life. (b) describes some kind of traffic network among stations in heterogeneous graph, where the same color dots show the same mode of stations. Different graph contains certain relationship. (c) is the traffic demand chart in one day on taxi and bike from the same region, which reflects some correlation between these two traffic modes.

the internal information among different traffic modes. Chai *et al.* [24] combined multiple semantic traffic relations when capturing spatial relations. However, it ignored the co-relations among different traffic modes. Ye *et al.* [25] proposed a co-prediction framework to predict multiple transportation demand while it failed to capture the multiple spatial relationships.

Although the mentioned methods have made progress in traffic demand prediction, there are some difficulties unsolved comprehensively yet: 1) it is not easy to incorporate multiple traffic modes into traffic demand prediction. As shown in Fig. 1, there is a correlation between the demand among different traffic modes. Combined urban road condition, travel time, price and other factors into consideration, people may transfer among different traffic modes when commuting. For example, people ride a bike to company after taking the subway. 2) Recent work seldom capture the spatial connection from multi-view. Most models extract spatial features from single relationship such as distance, which cannot reflect the other semantic information among nodes like transfer relationship. 3) It is challenging to model dynamic temporal dependencies since the temporal patterns of traffic demand at a station varies at different times of the day or week. Most methods neglect the mentioned complex problems and fail to capture the dynamic temporal dependencies.

To address these issues, a ***Co-Modal Graph Attention neTworks***(CMGAT) is proposed in this paper, which supplements auxiliary information in advance and combines multiple graphs information to predict traffic demand more accurately. As demonstrated by Fig. 2, our model contains the following core components. Specifically, first, more spatial and temporal features are added at the embedding layer, including the node's latitude and longitude, the day of the week, the hour of a day and the information on holidays. Inspired by Graph Attention Networks (GAT) [26], we design a spatial attention layer to learn relationship among stations dynamically based on multiple graphs. Similarly, a multi-head attention layer is then used to associate all heads of hidden states. Finally, the decoder part predicts traffic demand result through aggregating information from stacked spatial-temporal layers. In summary, this paper has the following contributions:

- We integrate different graphs viewed as different spatial relationships into the spatial learning module, which captures various spatial dependencies among nodes in a certain period.
- We propose a multi-mode traffic prediction framework based on attention mechanism, in which all pair-wise node spatial relations and both long-term and short-term temporal patterns are captured effectively.
- We conduct experiments on two real-world datasets. The experimental results indicate that the superiority of our model and the effect of multi-mode traffic and different graphs in spatial learning module.

II. Related Work

In this section, we review the literature related to our work from two perspectives: traffic prediction and attention neural network.

A. Traffic Prediction

In order to make more accurate prediction for traffic condition, researchers have proposed various methods and made continuous improvement in traffic flow, speed, demand and density of roads in traffic networks. Recently, deep learning has achieved successfully results in some scenarios such as image classification [27], speech recognition [28]. In traffic forecasting, some recent work convert the traffic region into 2-D grid and utilized deep models to capture the relationships in time series. For example, Zhang *et al.* [11] employed residual network in CNN and added external feature vectors such as weather and holiday events for embedding. To model the complex temporal dependency, Yu *et al.* [7] combined traffic flow patterns and accident-specific features based on LSTM neural network. Cui *et al.* [9] designed a bidirectional network to correlate forward and backward temporal dependencies with a data imputation mechanism for filling in missing values. Nevertheless, these methods considered spatial or temporal aspect with external features, none of them combined these two factors into model construction together.

Yao *et al.* [12] proposed DMVST-Net, which handled spatial dependencies via local CNN and modeled temporal relations via LSTM firstly. Moreover, they captured the latent semantic in similarity of demand patterns. For improvement, attention mechanism was put forward to address long-term periodic temporal shifting [13]. Shi *et al.* [29] built ConvLSTM, incorporating CNN and LSTM for precipitation nowcasting which is a typical spatial-temporal sequence prediction problem. Zhou *et al.* [14] utilized an encoder-decoder

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

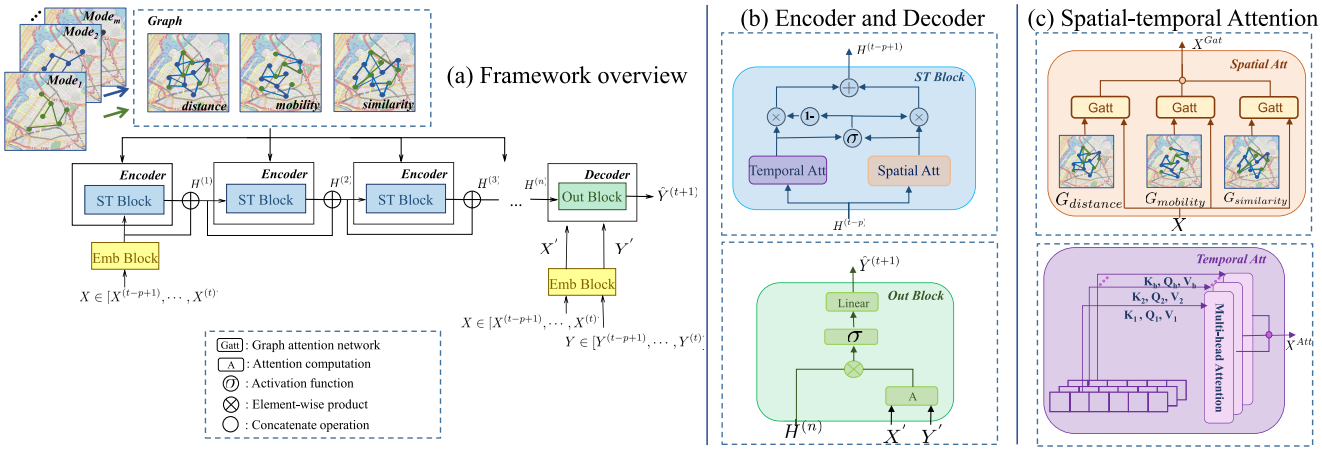


Fig. 2. Overview of **C**o-**M**odal **G**raph **A**ttention ne**T**works(CMGAT). Three colors are described in our model, which show the main parts of our model. Extra features are embedded with raw time series to extract the inner spatial-temporal dependencies. With encoder-decoder architecture, we encode the traffic demand into a hidden state and apply several encoded layers to capture their correlation. Additionally, three heterogeneous graphs are used in graph attention network, which results are concatenated as a part of hidden state. Residual network are applied in our model to avoid the over-fitting of model and improve the generalization ability of model. Besides, an attention mechanism is added to combine historical sequences with future sequences to predict the next step traffic demand.

framework based on CNN and ConvLSTM as the encoder unit with an attention model as the decoder part. Guo and Zhang [30] extended ConvLSTM and proposed a kind of data representation based on dynamic request vector for improvement. However, the grid-based structure fails to gain spatial information effectively. For example, there are no stations in many grids when the grid granularity is too small, otherwise, several stations will be divided into the same grid.

Station-based traffic prediction is more reasonable, effective but challenging. GCN [16] has been proposed for helping deal with spatial dependencies in network. In station-based traffic prediction, the traffic region is regarded as a topological graph. Some recent work combines GCN and RNN into exacting spatial-temporal dependencies. STGCN [31] combined GCN in spectral domain and gate convolution to predict traffic demand. DCRNN [32] learned spatial dependency by diffusion graph convolution and replaced the matrix multiplications in GRU with the diffusion convolution. However, the above work ignored the co-relations among different traffic modes or the diversity of traffic network. Ye *et al.* [25] combined different traffic modes into prediction. However, it failed to capture multiple spatial relationships. Chai et al. [24] fused different traffic graphs into spatial relationships learning, but it ignored the different periodic patterns in temporal view.

B. Attention Neural Network

Attention mechanism first applied in machine translation. As it concentrates on some certain parts in input data, various domains gain effective promotion, and surveys on attention mechanism have been proposed in specific areas such as computer vision [33], graph [34] and natural language process(NLP) [35]. For assigning different importance to one node's neighbors, attention mechanism can dynamically model the spatial-temporal relationship. Zhang *et al.* [22] used a convolutional sub-network to control different attention head's importance based on traditional multi-head attention mechanism [20], and constructed CGRU module to predict traffic speed. Applying attention mechanism in spatial-temporal correlations, Guo *et al.* [23] also employed graph convolutions to capture the spatial dependency with CNNs in modeling temporal patterns. Zhang and Guo [36] captured diverse traffic patterns among different stations based on LSTM embeded with attention mechanism.

Considering multiple graphs in traffic network, we construct three graphs based on distance, similarity and mobility. Different from computing each pair of station in the region, we apply graph attention network to reduce the computational cost and capture more proper relationship among stations. Furthermore, we co-predict different traffic modes such as taxi and bike, with the purpose of mining the correlation between these traffic modes and promoting the accuracy.

III. PRELIMINARIES

In this section, we first present some notions and definitions used to formalize our traffic demand prediction problem and then define the problem formally.

A. Notations and Definitions

TABLE I lists some notations used in this paper.

Definition 1 (Traffic Station): According to whether there are stationary stations on the traffic modes, the existing traffic modes can be divided into two categories. On the one hand, people can take a bus or ride a bike in the fixed stations. Intuitively, these stations can be regarded as nodes in a graph and the traffic data in each station can be used for the node's features directly. On the other hand, for some traffic modes such as taxis, it is hard to model them into graph-based structure due to random pick-up and drop-off locations for passengers. In order to analyze the traffic patterns on traffic modes with non-stationary stations, we construct some virtual fixed stations from their scattered traffic demand in the studied region, namely transportation virtual For example, some area

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
MATHEMATICAL NOTATIONS

Notation	Comments
$\|\cdot\|$	The total number of a set
$\|\|\cdot\|\|$	The norm of a vector or a matrix
$\boldsymbol{X}^{(t)}$	Observations at time step t
$\boldsymbol{Y}^{(t)}$	Estimated demand at time step t
$p_i^{(t)}$, $d_i^{(t)}$	Pick-ups and drop-offs of station i at time step t
$\mathcal{G}$	Traffic network in the studied region
$\boldsymbol{I}_N$	Identity matrix of size N
$\boldsymbol{H}^{(t)}$	Hidden state at time step t
θ, $\boldsymbol{w}$	Trainable parameters
S_i	A virtual station, $1 \leq i \leq N$
$\boldsymbol{X}_{M_i}^{(t)}$	Observations at time step t of traffic mode i

with dense traffic demand such as universities and commercial mansions are likely to be the potential transportation stations.

Generally, recent researches partition the area into grids to fit the requirement of CNN, which produces redundant information and weakens local characteristics. To avoid the above problems, we employ a Density Peak Clustering (DPC) [37] based method to discover virtual stations $S_1, \cdots, S_N$, which will be introduced in detail in Section IV-A.1.

Definition 2 (Traffic Demand): The traffic demand of single traffic mode in our work can be defined as the quantities of getting in or leaving the region at a time interval. Specifically, we define $p_i^{(t)}$ and $d_i^{(t)}$, respectively records historical observation for pick-up and drop-off demand of S_i station at time step t.

In order to study the correlation between pick-up and drop-off behavior, we concatenate these two historical time series simultaneously and represent as $x_i^{(t)} = [p_i^{(t)}, d_i^{(t)}]$. For a certain traffic mode M_i in the studied region, the previous observations at time t can be defined as $\boldsymbol{X}_{M_i}^{(t)} = [x_{M_{i_1}}^{(t)}, \cdots, x_{M_{i_n}}^{(t)}]$ and M_{i_n} denotes the number of sensors of traffic mode M_i. To learn the mutual relations among different traffic modes, we incorporate their traffic modes' demand to train our model and the input can be defined as $\boldsymbol{X}_0^{(t)} = [\boldsymbol{X}_{M_1}^{(t)}, \boldsymbol{X}_{M_2}^{(t)}, \cdots, \boldsymbol{X}_{M_m}^{(t)}]$.

Definition 3 (Graph, Neighborhood and Adjacency matrix): To describe the relationships among stations in the studied region, traffic graphs with different semantics are constructed. So some graph-related definitions are given below.

Graph: Given a graph $G = (V, E)$, we denote V as the set of nodes and E as the set of edges.

Neighborhood: The neighborhood of node v is defined as $N(v) = \{u \in V | \langle u, v \rangle \in E\}$, where v ($v \in V$) denotes a node and $e = \langle v, u \rangle \in E$ means an directed edge pointing from v to u.

Adjacency Matrix: Given a graph $G = (V, E)$, we define $\boldsymbol{A} \in R^{N \times N}$ as the adjacency matrix, which is shown as

$$\boldsymbol{A}_{ij} = \begin{cases} a_{ij}, & \text{if } \langle v_i, v_j \rangle \in E, \\ 0, & \text{if } \langle v_i, v_j \rangle \notin E. \end{cases} \tag{1}$$

where $a_{ij} > 0$ represents the degree of association.

Definition 4 (External Features): In essence, there is some semantic information hidden among time series. We can extract such content and use as auxiliary features $\boldsymbol{Z}_i^{(t)} \in \mathbb{R}^F$ to help capture the spatial-temporal dependencies for S_i station. More precisely, we select some temporal features such as day of the week, hour of the day, the holiday information and some spatial features such as the latitude and longitude of each station. Therefore, we design the inputs as $\boldsymbol{X}^{(t)} = [\boldsymbol{X}_0^{(t)}, \boldsymbol{Z}^{(t)}] \in \mathbb{R}^{N \times F'}$.

B. Problem Definition

Given the traffic network of the studied region and multiple traffic modes' historical demand of departures and arrivals with additional features, the target of traffic demand prediction is to predict future departures and arrivals demand of corresponding traffic modes at the next step time interval. We aim to construct a mapping function $h(\cdot)$ from $\boldsymbol{X}$ to $\boldsymbol{Y}$ with several traffic graphs $\mathcal{G}$ over a period of T-th time intervals:

$$[\boldsymbol{X}^{(t-T+1)}, \cdots, \boldsymbol{X}^{(t)}, \mathcal{G}] \xrightarrow{h(\cdot)} \hat{\boldsymbol{Y}}^{(t+1)}, \tag{2}$$

where $\hat{\boldsymbol{Y}}^{(t+1)}$ is the estimated traffic demand at next time step.

IV. METHODOLOGY

In this section, We first introduce the general framework of the model. CMGAT adopts an encoder-decoder architecture with feature embedding on input data. Spatial learning modules and temporal attention modules are collaborated by gated mechanism in encoder architecture. To avoid the problem of gradient vanishing, residual connections are added. The details of each module are described as follows.

A. Traffic Network Construction

1) Virtual Station: As passengers can pick up or drop off at arbitrary position by those traffic modes without stations such as taxi, we have to generate some virtual stations $\mathbb{S} = \{S_1, S_2, \cdots, S_N\}$ to aggregate traffic demand for these traffic modes. Inspired by DPC cluster algorithm, we regard the cluster centers built in our algorithm as virtual stations and the amount of clustering as the virtual stations number. In DPC algorithm, there are two basic hypotheses when selecting cluster centers. 1) The cluster center's local density ρ_i is over its neighbors' density. 2) Distance between different cluster centers is rather far away.

Considering multiple demand of a traffic mode in the urban area, we partition the studied area into $I \times J$ grids and construct two sets $R = \{\rho_1, \rho_2, \cdots, \rho_{I \times J}\}$, $U = \{\delta_1, \delta_2, \cdots, \delta_{I \times J}\}$ in advance. For the i-th grid, we count all the historic pick-up and drop-off demand as ρ_i, and compute distance δ_i: $\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij})$, where d_{ij} denotes the distance between grid i and grid j. For the maximum ρ_i, $\delta_i = \max_j(d_{ij})$. Then we employ a DPC-based algorithm to generate virtual stations.

Based on DPC algorithm, we generate virtual stations from those cluster centers above thresholds in distance and density. The process is split into two stages: virtual stations generation and clustering construction. To accelerate computation,

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

we construct a kd-tree which reduces the complexity of search and insert procedure. In the second stage, we divide all grids into the corresponding clustering, based on the theory that the clustering where grid n_j belongs to is the same as its nearest neighbor belongs to.

2) Traffic Graph: We construct three traffic graphs based on distance, mobility and similarity. The distance between stations represents geographical proximity. Closer the two stations are, more similar their traffic demand are. The mobility graph learns the traffic transition among stations, which helps capture the correlation between pick-up and drop-off traffic demand. The similarity graph studies the association among stations as the traffic demand change over time. Specific definitions are as follows.

First, we introduce the distance graph. For station s_i, s_j, the distance adjacent matrix $\boldsymbol{E}^{dist}$ is shown as follows,

$$w_{i,j}^{dist} = \begin{cases} \dfrac{1}{dist(s_i, s_j)}, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$
$$\boldsymbol{E}_{i,j}^{dist} = \begin{cases} w_{i,j}^{dist}, & \text{if } w_{i,j}^{dist} \in topk(w_{i,:}^{dist}), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where function $dist(\cdot)$ is an Euclidean distance calculation formula. For virtual stations, we build a grid-based network and use each grid center's longitude and latitude as their geographical locations. Besides, considering the minimal effect on remote stations for station s_i, we select top-k stations to keep the graph sparse.

Second, we construct the mobility graph $\boldsymbol{E}^{mob}$ to demonstrate transition relationship among stations, $w_{i,j}^{mob}$ represent the amount of travel records from station s_i to station s_j over a period of time. In mobility graph $\boldsymbol{E}^{mob}$, we apply data normalization operation to accelerate our model training.

$$\boldsymbol{E}_{i,j}^{mob} = \frac{w_{i,j}^{mob}}{||w_{i,:}^{mob}||}. \tag{4}$$

Last, different regions show different traffic patterns in real world. For example, commercial mansion has a noticeable morning and evening peak while school sections meet larger traffic flow during school hours. The stations with similar time changing trends will be beneficial for extracting spatial dependencies. In order to simulate the similar patterns among stations, we experiment some classic similarity algorithms to acquire their relationship and choose the most effective algorithm in our work to build similarity graph $\boldsymbol{E}^{sim}$.

$$w_{i,j}^{sim} = \frac{D_{0\sim t}^{s_i} \cdot D_{0\sim t}^{s_j}}{||D_{0\sim t}^{s_i}|| \cdot ||D_{0\sim t}^{s_j}||},$$
$$\boldsymbol{E}_{i,j}^{sim} = \begin{cases} w_{i,j}^{sim}, & \text{if } w_{i,j}^{sim} \in topk(w_{i,:}^{sim}), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $D_{0\sim t}^{s_i}$ is the historical traffic demand sequence with t months at traffic station s_i and we select those top-k stations with high similarity.

B. Encoder Architecture

The encoder architecture consists of three parts: spatial attention module, temporal attention module and gated mechanism. Given the historical traffic demand $\boldsymbol{X} = [\boldsymbol{X}^{t-p}, \boldsymbol{X}^{t-p+1}, \cdots, \boldsymbol{X}^{t}]$, hidden state $\boldsymbol{H} = [\boldsymbol{H}^{t-p}, \boldsymbol{H}^{t-p+1}, \cdots, \boldsymbol{H}^{t}]$ is produced. Specifically, spatial attention module extracts dynamic spatial dependencies among stations on the certain period of time. Temporal attention module models the temporal dependencies on each single station. When fusing their information, a gated mechanism is proposed and captures the spatial-temporal correlations as shown in Fig. 2. Moreover, residual network is added to avoid vanishing phenomenon happening as the number of encoder layer increasing. The hidden state from the last encoder layer will be used as the input for decoder module.

Before spatial attention and temporal attention module, we propose an embedding unit to correlate the semantic information among nodes, which helps attention mechanism utilize the proximity information. To learn the interactions between different features, we design embedding unit based on DeepFM [38]. Combined FM layer and deep layer, the feature embeddings produced are applied for spatial-temporal attention mechanism. In FM layer, low-order computation extracts individual feature and high-order capture correlations between features. By embedding different features simultaneously, we can capture much more latent information among stations. With several neural network stacked, we can learn high-order feature interactions deeply. Besides, batch normalization is added to promote the generalization performance.

C. Spatial Learning Module

Since researchers have taken traffic network as graph structure, now most deep learning methods apply GCN [31], [32] for spatial feature extraction. Meanwhile, due to assign different weight for neighbor nodes, attention mechanism is wide-spread in graph operation. GAT [21] is proposed for graph-constructed data and has been successfully applied in many tasks such as node classification. It does not require costly matrix operation and can compute among nodes in parallelization. Therefore, we apply graph attention mechanism to extract highly meaningful patterns in the space domain based on multiple relationships in the traffic network.

First, we calculate the relationship $\boldsymbol{R}$ among nodes in our graph with input $\boldsymbol{H}^{(t)} \in \mathbb{R}^{N \times F}$.

$$\boldsymbol{H}_1^{(t)} = \boldsymbol{W}\boldsymbol{H}^{(t)},$$
$$\boldsymbol{R} = \boldsymbol{H}_1^{(t)}\boldsymbol{H}_1^{(t)T}. \tag{6}$$

where $\boldsymbol{R}_{i,j}$ represent the relationship from node i to node j, and $\boldsymbol{W} \in \mathbb{R}^{F' \times F}$ is the learning parameter. Then we calculate the weight matrix $\boldsymbol{A}$ based on the relationship matrix $\boldsymbol{R}$ and raw traffic graphs. In order to focus on those nodes with higher importance,we use a masked matrix $\boldsymbol{W}_{mk}$ in the graph which ensures the computation just between those nodes and their neighbors instead of all the pair-wise nodes. The masked matrix $\boldsymbol{W}_{mk}$ is based on adjacent matrix E.

$$\boldsymbol{R}_{mk}^{(t)} = \boldsymbol{W}_{mk}\boldsymbol{R}^{(t)},$$
$$\boldsymbol{A} = s(g(\boldsymbol{R}_{mk}^{(t)})). \tag{7}$$

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

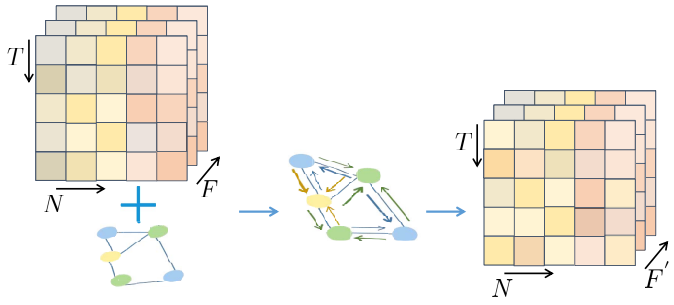


Fig. 3. A framework of spatial learning layer: We use $\boldsymbol{X}_{spa} \in \mathbb{R}^{N\times T\times F}$ as input, and output $\boldsymbol{X}'_{spa} \in \mathbb{R}^{N\times T\times F'}$.

where $g(\cdot)$ is the activation function to speed up the computation process and avoid gradient vanishing, $s(\cdot)$ is the softmax function to make the data normalized.

To make the learning process more stable, multi-head attention mechanism is applied. We use K independent attention mechanisms and average their results together.

$$\boldsymbol{H}_o^{(t)} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{A}^{(k)}\boldsymbol{W}^{(k)}\boldsymbol{H}^{(t)}\right). \tag{8}$$

In spatial learning module, the representation changed from input $\boldsymbol{X}$ to $\boldsymbol{X}^G$ can be described as,

$$\boldsymbol{X}^G = Gatt(\boldsymbol{X}, G), \tag{9}$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times F}$, N represents the number of nodes, F is the feature dimension, $G \in \mathbb{R}^{N\times N}$ is the graph network built from our studied region and the output $\boldsymbol{X}^G \in \mathbb{R}^{N\times F'}$. Fig. 3 shows the process.

Three traffic graphs have been constructed in our work based on mobility, distance and similarity mentioned in Section IV-A.2. For the distance graph, it is generally assumed that the nearby station traffic demand will be similar and we set a threshold to ensure the effect of the distance matrix. The mobility graph can model the relationship between pick-up and drop-off demand. The similarity graph is defined to discover the change between those stations with similar traffic patterns as time going. Based on graph attention mechanism $Gatt$ separately, three graphs are executed with input $\boldsymbol{X}$.

$$\begin{aligned}\boldsymbol{X}^G_{mob} &= Gatt(\boldsymbol{X}, G_{mobility}),\\ \boldsymbol{X}^G_{dis} &= Gatt(\boldsymbol{X}, G_{distance}),\\ \boldsymbol{X}^G_{sim} &= Gatt(\boldsymbol{X}, G_{similarity}).\end{aligned} \tag{10}$$

The results from graph attention mechanism based on different graphs are concatenated from Equation (10) and the representations are as followed,

$$\boldsymbol{X}^G = \left\{\boldsymbol{X}^G_{mob} \| \boldsymbol{X}^G_{dis} \| \boldsymbol{X}^G_{sim}\right\}. \tag{11}$$

Besides, attention embedding $\boldsymbol{X}^{Embed}$ is employed and residual connection $\boldsymbol{X}^{Res}$ is added to avoid gradient vanishing and make propagation more successful.

Combined similarity computation, we present the attention mechanism,

$$\boldsymbol{X}^{Embed} = softmax\left(\frac{\boldsymbol{X}\boldsymbol{X}^T}{\sqrt{F}}\right)\boldsymbol{X}, \tag{12}$$

in which $\boldsymbol{X}, \boldsymbol{X}^{Embed} \in \mathbb{R}^{N\times T\times F}$. Hence, this module is represented as,

$$\boldsymbol{X}^{Spa}_O = \boldsymbol{X}^G \times sigmoid(\boldsymbol{X}^{Embed}) + \boldsymbol{X}^{Res}. \tag{13}$$

D. Temporal Attention Module

Transformer [20] has been successfully applied in machine translation, which improves the effect on capturing long-term dependencies and increases the speed of training. Relying entirely on self-attention to compute representations of its input and output, transformer allows for significantly more parallelization and gains more accuracy in prediction. For capturing temporal relationship in time sequence, we apply the multi-head self-attention mechanism and position-wise fully connected feed-forward network in our temporal learning module. Furthermore, different periods are provided in attention mechanism which helps correlate previous long-term and short-term time series.

Generally, the input of attention mechanism contains a query $\boldsymbol{Q}$, and a set of key-value pairs $\boldsymbol{K}$, $\boldsymbol{V}$, then the output is computed as a weighted sum of the values. Through computing the similarity among the q_i and $k_j (i, j \in T)$ for each pair of elements, we get the result $w_{ij} (i, j \in T)$. For period i, the output is a weighted sum functioned on $v_j (j \in T)$, with $w_{ij} (j \in T)$ as the weighted coefficients respectively. The calculation formula is shown below,

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{14}$$

where $\boldsymbol{Q} \in \mathbb{R}^{T\times d_k}$, $\boldsymbol{K} \in \mathbb{R}^{T\times d_k}$, $\boldsymbol{V} \in \mathbb{R}^{T\times d_v}$. As for the selection for $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$, we choose input $\boldsymbol{X}$ as $\boldsymbol{Q}$ and $\boldsymbol{K}$ to directly compute the similarity among time series, and the embedding of input as $\boldsymbol{V}$.

In essence, there are two attention functions used commonly: additive attention and dot-product attention. Considering the advantages of dot-product attention: faster training speed and more space-efficient, we employ dot-product and scale the dot product by $\frac{1}{\sqrt{d_v}}$ to counteract extremely small gradients phenomenon. Compared to single-head attention, performing multi-head attention function with different learned queries, keys and values is effective. The process is defined as,

$$\begin{aligned}MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= \|[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_h]\boldsymbol{W}^O,\\ \boldsymbol{h}_i &= Attention(\boldsymbol{Q}\boldsymbol{W}^Q_i, \boldsymbol{K}\boldsymbol{W}^K_i, \boldsymbol{V}\boldsymbol{W}^V_i),\end{aligned} \tag{15}$$

in which $\boldsymbol{W}^Q_i, \boldsymbol{W}^K_i \in \mathbb{R}^{d_{model}\times d_k}$, $\boldsymbol{W}^V_i \in \mathbb{R}^{d_{model}\times d_v}$, $\boldsymbol{W}^O \in \mathbb{R}^{hd_v\times d_{model}}$, d_{model} is the dimension of input sequence features.

Besides, a fully connected feed-forward network (FFN) is applied to each position identically, which can capture

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

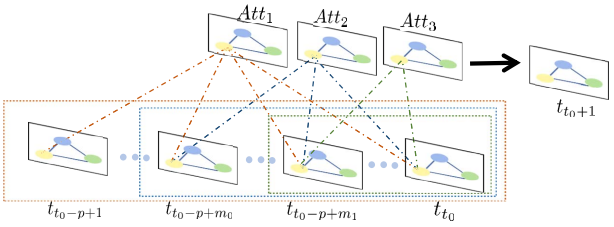


Fig. 4. The long-short term attention mechanism to model temporal dependency. We select three periods when using attention mechanism and then concatenate these results.

the nearby station dependency simultaneously. The network consists of two linear transformations with a ReLU activation in between.

$$FFN(x) = max(0, x\boldsymbol{W}1 + b_1)\boldsymbol{W}_2 + b_2. \tag{16}$$

Considering long-term attention mechanism fails to capture short periods time dependency effectively, we select several number of period to capture the temporal dependencies of time series and concatenate them together as shown in Fig. 4. The output $\boldsymbol{X}^{Att}$ based on attention mechanism is as follows,

$$\boldsymbol{X}^{Att} = \Big|\Big|_{i=1}^{\beta} FFN(MultiHead(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i)), \tag{17}$$

where $\boldsymbol{Q}_i \in \mathbb{R}^{T_i \times d_k}$, $\boldsymbol{K}_i \in \mathbb{R}^{T_i \times d_k}$, $\boldsymbol{V}_i \in \mathbb{R}^{T_i \times d_v}$. With experiments on different number of period, we select $\beta = 3$, $T = \{3, 6, 12\}$ to gain better prediction ultimately.

Similar to spatial attention module, we add similarity computation process $\boldsymbol{X}^{Embed}$ and residual connection $\boldsymbol{X}^{Res}$ in temporal attention module and the equation of this module can be described as,

$$\boldsymbol{X}_O^{Tem} = \boldsymbol{X}^{Att} \times sigmoid\left(\boldsymbol{X}^{Embed}\right) + \boldsymbol{X}^{Res}. \tag{18}$$

E. Decoder Architecture

Different from encoder architecture, we propose an attention mechanism to predict future traffic demand. traffic demand sequence can be regarded as a kind of typical time series, historical time series have great great influence on future predictions. Moreover, each historical time step makes different importance on the future demand and the importance may change with time. We should capture the different correlations among previous observations and the dynamic temporal dependencies as time goes. First, we embed historical time series and future time sequences based on the embedding unit mentioned in Section IV-B. Then an attention mechanism is applied for computing the importance score among historical time series. We define the similarity between predicted time-step t_j and historical time-step $t_i \in T$,

$$\boldsymbol{\lambda}_{t_j,t_i} = \boldsymbol{e}_{t_j}\boldsymbol{e}_{t_i}, \tag{19}$$

where $\boldsymbol{e}_{t_i}, \boldsymbol{e}_{t_j} \in \mathbb{R}^F$ represent the t_i-step and t_j-step information, $\boldsymbol{\lambda}_{t_j,t_i} \in \mathbb{R}^F$.

These coefficients are normalized through a softmax function to simplify computation,

$$\boldsymbol{\alpha}_{t_j,t_i} = softmax_{t_j}(\boldsymbol{\lambda}_{t_j,t_i}) = \frac{exp(\boldsymbol{\lambda}_{t_j,t_i})}{\sum_{t_k \in T} exp(\boldsymbol{\lambda}_{t_j,t_k})} \tag{20}$$

TABLE II
DETAILS IN DATASETS

Dataset	Threshold	Raw number	Filtered number
NYC Citi Bike	5	578	185
NYC Taxi	2.5	490	221

The normalized attention coefficients are used for a weighted combination and then we produce the representation of predicted time-step t_j. With a linear transformation, we predict the future traffic demand as follows.

$$\boldsymbol{F}_{O_j} = \sigma\left(\sum_{t_i \in T} \boldsymbol{\alpha}_{t_j,t_i}\boldsymbol{e}_{t_i}\right),$$
$$\boldsymbol{Y}' = \boldsymbol{W}_O\boldsymbol{F}_O + b_O. \tag{21}$$

where $\boldsymbol{F}_O \in \mathbb{R}^{T \times 2}$, $W_O \in \mathbb{R}^{T' \times T}$, $\boldsymbol{Y}' \in \mathbb{R}^{T' \times 2}$. We apply L2 loss to measure the performance of our model. Considering the different effect on these traffic modes, we use a hyper-parameter set $\Gamma = \left[\gamma_{M_1}, \gamma_{M_2}, \cdots, \gamma_{M_m}\right]$ to control the specific weight as the error parts. The loss function is shown as follow:

$$Loss(\boldsymbol{X}, \hat{\boldsymbol{Y}}) = \sum_{M_i \in M} \gamma_{M_i} \left\| \boldsymbol{Y}_{M_i} - \hat{\boldsymbol{Y}}_{M_i} \right\|^2 \tag{22}$$

V. EXPERIMENT

A. Data Set

Experiments are conducted on two real-world datasets collected from NYC OpenData. Each dataset contains order records and geographic information of taxi and bike in NYC.

- *NYC Citi Bike*[1]: This dataset includes the NYC Citi bike travel records from April $1st$, 2016 to June $30th$, 2016 (91 days). Relative information is shown as follows: station IDs about bike drop-off and pick-up, specific time about drop-off and pick-up, and trip duration.
- *NYC Taxi*[2]: This dataset contains 35 million taxicab trip records in New York from April $1st$, 2016 to June $30th$, 2016. Relative information is shown as follows: specific time, geographic location (longitude and latitude) and trip distance of pick-up and drop-off.

We filter some stations which always provide little amount in demand through a threshold. If the station's average traffic demand are lower, it means that it has rare traffic demand most of the time. Therefore, it is meaningless to predict these stations. Table II shows the datasets and their thresholds.

B. Baseline Methods for Comparision

We compare our model with following methods:

- *Historical Average (HA)*: We calculate the average of historical values at the previous quarter as history average.
- *Vector Auto-Regression (VAR)*: a multivariate forecasting method which is used for capture the bi-direction relationship among time series.

[1]https://www.citibikenyc.com/system-data
[2]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
EVALUATIONS OF CMGAT AND BASELINES ON PICK-UP DEMAND IN NYC CITI BIKE AND NYC TAXI

Method	NYC Citi Bike			NYC Taxi		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	5.4797	3.1200	0.6533	59.1394	22.3933	0.5326
VAR	4.9545	3.1877	0.6768	30.6486	13.0893	0.4175
XGBoost	4.4377	2.7326	0.7319	25.2156	10.7850	0.3656
FC-LSTM	5.0179	2.9266	0.5371	42.3823	16.5370	0.4214
DCRNN	4.5333	2.6566	**0.5230**	24.6196	11.4612	0.4635
STGCN	5.0508	2.9202	0.5525	33.2700	14.4713	0.4847
Graph WaveNet	4.6103	2.7771	0.5820	29.6283	13.4844	0.4904
MTGNN	4.5366	2.9132	0.4935	22.8915	10.1402	**0.3218**
CMGAT	**4.3197**	**2.5392**	0.5236	**22.8825**	**9.8839**	0.3274

TABLE IV
EVALUATIONS OF CMGAT AND BASELINES ON DROP-OFF DEMAND IN NYC CITI BIKE AND NYC TAXI

Method	NYC Citi Bike			NYC Taxi		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	5.0946	3.0013	0.6246	51.8465	20.8810	0.4731
VAR	4.3425	2.9106	0.6143	26.4684	11.9688	0.3321
XGBoost	3.9063	2.5931	0.7241	21.3235	9.7809	0.2901
FC-LSTM	4.5500	2.7850	0.5190	34.6010	14.6142	0.3477
DCRNN	3.9220	2.4927	**0.4174**	20.7640	10.6751	0.3509
STGCN	4.1222	2.5972	0.4184	20.4772	10.0474	0.3234
Graph WaveNet	4.1095	2.5933	0.5064	26.5136	12.5827	0.3911
MTGNN	3.9681	2.6786	0.4572	**18.7186**	**9.0560**	0.2897
CMGAT	**3.7613**	**2.3897**	0.4963	19.3968	9.1099	**0.2729**

- *XGBoost*: XGBoost is a widely used method based on gradient boosting tree.
- *FC-LSTM*: LSTM incorporates with fully connected network.
- *DCRNN* [32]: Diffusion convolution recurrent neural network combines diffusion graph convolution with GRU in an encoder-decoder manner.
- *STGCN* [31]: Spatial-temporal graph network combines graph convolution with casual convolution.
- *Graph WaveNet* [21]: Graph WaveNet conducts graph convolution with adaptive adjacency matrix.
- *MTGNN* [39]: Mix-hop propagation layer and dilated inception layer are combined in an end-to-end network

We apply three evaluation metrics to evaluate performance of different methods: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error(MAPE).

C. Experiment Settings

The length of a time step and the time interval are set as an hour. Each dataset is split into three parts: train set, valid set and test set. Their proportion is 60%, 20% and 20% in sequence respectively. For deep learning methods (e.g. STGCN, DCRNN, MTGNN), mobility graph is applied as the fundamental graph. As for FC-LSTM model, we set 200 as the feature dimension. For DCRNN, the number of convolution layers is 3, and the dimension for hidden unit is 128. In STGCN, we set 3 as the size of kernel. In our method CMGAT, we initialize feature embeddings by a uniform distribution with a size of 45. The optimizer is Adam with an initial learning rate of 0.0015 in our method and all of the deep learning methods. We use 16 as the batch size and 48 as the raw sequence length.

D. Experimental Results

Table III and Table IV compare the performance in pick-up and drop-off demand based on baselines and CMGAT. For most methods, evaluations in NYC Citi Bike are rather lower than NYC Taxi. Because there is less demand in bike than taxi in the city, the relative error is smaller in NYC Citi Bike then prediction is more accurate. Except for MAPE, the model we propose performs the best result and has the lowest RMSE and MAE.

On NYC Citi Bike prediction: The deep learning methods show better performance than temporal methods due to capture the variance of bike demand. HA shows the worse result for ignorance of non-linearity and uncertainty in time series. Based on integrated learning method, XGBoost has a better generalization ability. In our experiment, XGBoost performs even better than half of deep learning methods shown in Table III and Table IV, while it does not require much compute resources. In deep learning modules, STGCN performs a little worse. For STGCN, the model uses a pre-defined graph in spatial module then it fails to capture diversity of spatial dependencies. Compared to the TCN-based model Graph WaveNet, CMGAT performs better for applying different time periods in temporal attention mechanism. Although MTGNN takes adaptive spatial relationship into consideration, CMGAT performs a little better for adopting multiple graphs based on spatial attention mechanism.

On NYC Taxi prediction: CMGAT outperforms the temporal methods like HA and VAR. For more demand for taxi and more stations constructed from taxi demand, HA and VAR fail to capture the spatial features in dataset. In deep learning modules, DCRNN performs well due to its bidirectional graph random walk to learn spatial features and RNN-based structure in capturing temporal features. Although Graph WaveNet adopts an adaptive graph to capture the latent information among stations from training process, the temporal convolution block may fail to capture the dynamic and periodic temporal dependencies. Compared to these deep learning methods, we capture spatial dependencies from several traffic spatial relationships while take different temporal patterns into consideration. At the same time, we can capture the correlations among different traffic modes, which help for the improvement.

E. Ablation Study

1) Effect of Multi-Modal Traffic: To evaluate the effectiveness of multi-mode in our model, the experiment is carried out from two aspects: single mode and multiple modes. The results in Table V show that co-prediction is effective and there are some correlations among different traffic modes such as taxi and bike. For future studies, we can add more traffic modes into research and analyze the specific influence among various traffic modes.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V

EVALUATIONS OF CO-PREDICTION(CMGAT) AND SINLE MODE(SINGLE-TAXI AND SINGLE-BIKE) ON TRAFFIC DEMAND IN NYC CITI BIKE AND NYC TAXI

Method	Pick-up Evaluation			Drop-off Evaluation		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Single-taxi	23.6338	10.4417	0.3445	22.7699	10.4909	0.30057
CMGAT(taxi)	22.8825	9.8839	0.3274	19.3968	9.1099	0.2729
Single-bike	4.6234	2.8073	0.5865	4.1217	2.6539	0.5888
CMGAT(bike)	4.3197	2.5392	0.5236	3.7613	2.3897	0.4963

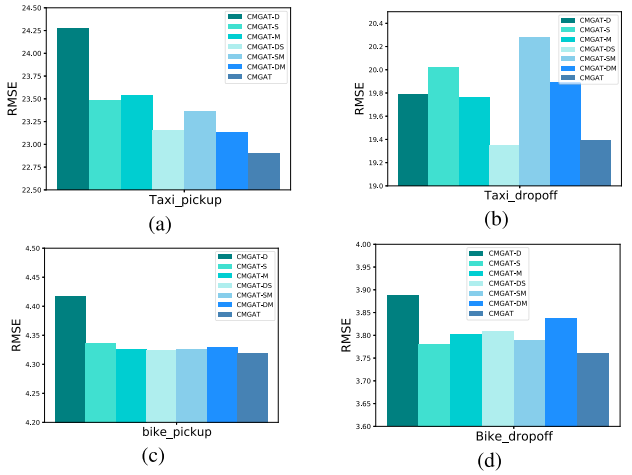


Fig. 5. Performance comparison among CMGAT and CMGAT variants. The results are divided into four parts, each of them contains the validation error from all the models. Generally, CMGAT obtains the lower validation error.

2) Effect of Multi-View Graph: To investigate the effect of multi-view graphs on spatial learning module, we compare with the following variants: (1) CMGAT-D, CMGAT-S, CMGAT-M. These models use only one graph in spatial learning module based on distance, similarity and mobility respectively, which means that the spatial dependency is captured by a single graph in Equation (10). (2) CMGAT-DS, CMGAT-SM, CMGAT-DM, which combines two graphs in spatial learning module; Fig. 5 shows the performance of these models with almost the same parameters.

From the result, we can find that using distance adjacent matrix among these prediction have worse performance, which demonstrates the spatial dependencies among the nearby stations are not stronger than the remote stations. Generally, with mobility and similarity adjacent matrices together, spatial dependencies among traffic data can be captured well and validation error is smaller. When combining the three graphs together, we get great improvement in the pick-up prediction on taxi data and drop-off prediction on bike data. From these experiment, it indicates that if we combine different graphs properly, spatial dependencies can be captured more comprehensively.

F. Case Study

To validate the necessity of the proposed multiple graphs used in our graph attention network, we design a case study to investigate the different graph adjacency matrix. Specifically, we visualize some stations and their neighbors based on different graphs. In Fig. 6 (a), we chart the raw time series of

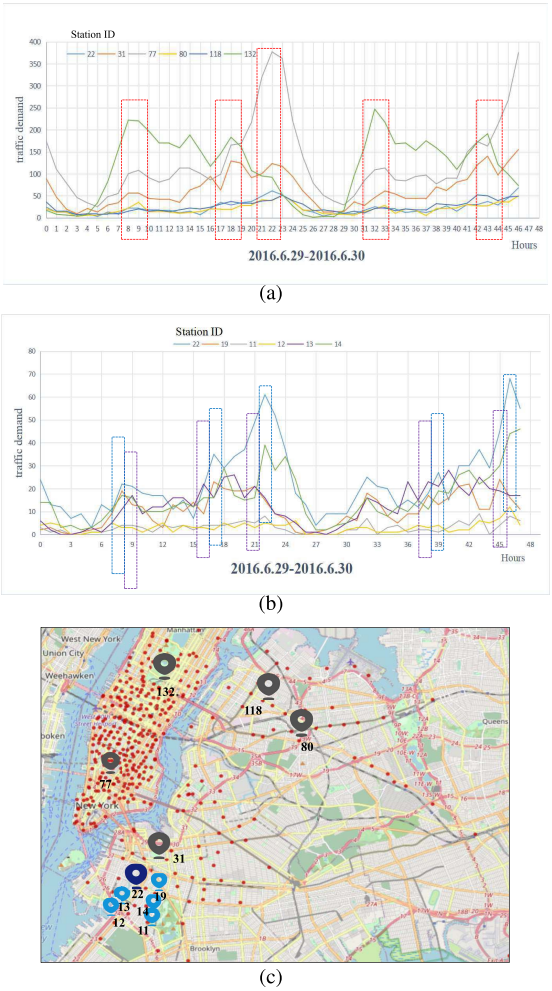


Fig. 6. (a) shows the time series of node 22 and its top-5 neighbors on similarity graph. (b) shows the time series of node 22 and its top-5 neighbors on distance graph.(c) depicts the traffic stations built in the studied region on Google Maps and emphasizes the positions about nodes 22 and its top-5 neighbors mentioned on similarity graph and distance graph. The different colors marked on the map show the neighbors on respective graph.

node 22 and its defined top-5 neighbors from similarity graph. In Fig. 6 (b), we plot the raw time series of node 22 and its defined top-5 neighbors from distance graph. Fig. 6 (c) shows the distribution of all stations on the Google map and the mentioned stations are highlighted. We observe that although the central node 22 and its top-5 neighbors show more similar time trends on time series, their geometric positions are not much closer on the map. Besides, according to Fig. 6 (b), the time series of top-5 neighbors based on distance graph may experience the peak time in advance or later which provide auxiliary information for the central node. From these phenomena, we find that utilize distance adjacency matrix to capture spatial dependencies may ignore the similar tendency on time series among those remote stations. When combined other graphs such as similarity graph, it is effective to expand the representation and decrease the model bias.

VI. CONCLUSION AND DISCUSSION

In this paper, a novel co-prediction model CMGAT based on spatial-temporal attention mechanism is proposed and successfully applied in traffic demand prediction. CMGAT applies

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

multiple graphs in graph attention mechanism to capture spatial dependencies and different time periods attention mechanism to capture dynamic temporal dependencies. Experiments on two real-world datasets show that CMGAT outperforms other state-of-the-art methods. In the future work, we will from more perspective to design traffic graph and optimize the model structure. Besides, environment aspects will be taken into consideration for promoting prediction accuracy such as weather information. As traffic demand prediction belongs to spatial-temporal forecasting problem, we can apply the proposed model to other spatial-temporal forecasting scenarios.

REFERENCES

[1] B. Williams and L. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov./Dec. 2003.

[2] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Proc.-Social Behav. Sci.*, vol. 96, pp. 653–662, Nov. 2013.

[3] Y.-N. Yang and H.-P. Lu, "Short-term traffic flow combined forecasting model based on SVM," in *Proc. Int. Conf. Comput. Inf. Sci.*, Dec. 2010, pp. 262–265.

[4] J. W. Yu and J. E. Jang, "A Kalman filter ramp traffic forecasting model for real-time traffic control and information provision," in *Proc. Eastern Asia Soc. Transp. Stud.*, vol. 8, 2011, p. 101.

[5] J. Zheng and L. M. Ni, "Time-dependent trajectory regression on road networks via multi-task learning," in *Proc. 27th AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, 2013, pp. 1048–1055.

[6] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2015, pp. 1–10.

[7] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proc. SIAM Int. Conf. Data Mining*. Houston, TX, USA: SIAM, Apr. 2017, pp. 777–785.

[8] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Proc. Int. Conf. Mach. Learn.*, vol. 34, 2017, pp. 1–5.

[9] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. C, Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102674.

[10] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 1–4.

[11] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, S. P. Singh and S. Markovitch, Eds. San Francisco, CA, USA: AAAI Press, Feb. 2017, pp. 1655–1661.

[12] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2588–2595.

[13] 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 Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5668–5675.

[14] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 736–744.

[15] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *J. Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[16] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014.

[17] L. Yu, B. Du, X. Hu, L. Sun, L. Han, and W. Lv, "Deep spatio-temporal graph convolutional network for traffic accident prediction," *Neurocomputing*, vol. 423, pp. 135–147, Jan. 2021.

[18] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Singapore, Aug. 2021, pp. 547–555.

[19] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," in *Proc. 35th AAAI Conf. Artif. Intell., AAAI, 33rd Conf. Innov. Appl. Artif. Intell., IAAI, 11th Symp. Educ. Adv. Artif. Intell.*, 2021, pp. 4617–4625.

[20] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 6000–6010.

[21] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 1907–1913.

[22] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," in *Proc. 34th Conf. Uncertainty Artif. Intell.*, 2018, pp. 339–349.

[23] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[24] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 397–400.

[25] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 305–313.

[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[28] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[29] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Montreal, QC, Canada, 2015, pp. 802–810.

[30] G. Guo and T. Zhang, "A residual spatio-temporal architecture for travel demand forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102639.

[31] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[32] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[33] F. Wang and D. M. J. Tax, "Survey on the attention based RNN model and its applications in computer vision," 2016, *arXiv:1601.06823*.

[34] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 6, pp. 1–25, 2019.

[35] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.

[36] T. Zhang and G. Guo, "Graph attention LSTM: A spatio-temperal approach for traffic flow forecasting," *IEEE Intell. Transp. Syst. Mag.*, early access, Jun. 2020, doi: 10.1109/MITS.2020.2990165.

[37] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

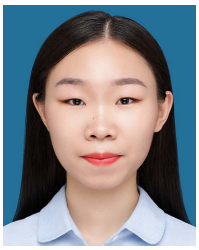
[38] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1725–1731.

[39] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

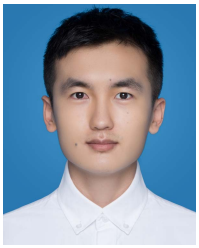
Haihui Xu is a Business Expert in comprehensive traffic command and dispatch and major event traffic security. He has lead the organization and implementation of three provincial and ministerial projects, and won five provincial and ministerial awards and personal honors. His research interests include comprehensive traffic command and intelligent transportation system.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Tao Zou received the B.S. degree from the School of Computer Science and Engineering, Beihang University, China, in 2021, where she is currently pursuing the M.S. degree. Her research interests include intelligent transportation, smart city technology, and traffic data mining.



Mingzhe Liu (Graduate Student Member, IEEE) received the B.S. degree from the College of Computer Science and Technology, Shandong University, China, in 2020. He is currently pursuing the M.S. degree in computer science and engineering with Beihang University, Beijing, China. His research interests include intelligent transportation, deep learning, and spatio-temporal data mining.

Yanan Qiao received the M.S. degree in computer science and engineering from Beihang University, Beijing, China. Her research interests include intelligent transportation, deep learning, and sequential prediction.

Jingjing Wang is a Senior Engineer and the Vice Director of the Beijing Municipal Transportation Operations Coordination Center. She has been engaged in the field of intelligent transportation for more than ten years, with rich experience in intelligent transportation planning, scientific research and engineering project construction, and comprehensive traffic operation monitoring services. She was responsible for and also involved in a number of national research projects, major projects, and research topics at the provincial and ministerial levels, won four ministerial and provincial-level awards, published more than ten papers, applied for three invention patents, and obtained more than ten software copyrights as well. Her research interests include comprehensive traffic command and dispatch, major event traffic guarantee, and intelligent transportation systems.

Xucheng Li received the B.Eng. degree in civil engineering and the Ph.D. degree in transportation from the University of Southampton, U.K. His research interests include deep learning, demand modeling, and intelligent mobility.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on July 30,2022 at 02:28:51 UTC from IEEE Xplore. Restrictions apply.