Traffic Demand Prediction Based on Dynamic Transition Convolutional Neural Network

Bowen Du, Xiao Hu, Leilei Sun, Junming Liu, Yanan Qiao, and Weifeng Lv

***Abstract*— Precise traffic demand prediction could help government and enterprises make better management and operation decisions by providing them with data-driven insights. However, it is a nontrivial effort to design an effective traffic demand prediction method due to the spatial and temporal characteristics of traffic demand distributions, dynamics of human mobility, and impacts of multiple environmental factors. To handle these problems, a Dynamic Transition Convolutional Neural Network (DTCNN) is proposed for the purpose of precise traffic demand prediction. Particularly, a transition network is first constructed according to the citiwide historical departure and arrival records, where the nodes are virtual stations discovered by a density-peak based clustering algorithm and the edges of two nodes correspond to transition flows of two stations. Then, a dynamic transition convolution unit is designed to model the spatial distributions of the traffic demands, and to capture the evolution of the demand dynamics. Last, a unifying learning framework is provided to incorporate the spatiotemporal states of the traffic demands with environmental factors. Experiments have been conducted on NYC taxi and bike-sharing data, and the results validate the effectiveness of the proposed method.**

***Index Terms*— Traffic demand prediction, spatiotemporal, transition convolution, deep learning.**

,

I. Introduction

TRAFFIC demand prediction is an important problem in transportation management, urban planning, and sharing economy, etc. Due to its importance, a large number of traffic demand prediction approaches have been proposed.

Early traffic demand prediction methods were founded on classical machine learning and empirical statistics. Time series prediction approaches such as Auto-Regressive Integrated Moving Average (ARIMA) and its variants were first studied to predict the traffic demands [1], [2]. K-Nearest Neighbor (KNN) [3] and Time Varying Poisson Model [2] have also been explored for traffic demand prediction. Zheng and Ni proposed a multi-task framework to learn temporal dynamics of road travel costs from historical trajectory data [4]. Wei *et al.* developed a hybrid model to predict the passenger demand [5]. A key disadvantage of these methods is that they mainly explore the temporal evolution of the demand time series separately, spatial correlation of nearby regions has not been taken into account.

In recent years, deep learning methods have achieved great successes in image processing, computer vision, natural language processing, etc. Correspondingly, many scholars studied deep learning based traffic demand prediction methods. It has been realized that Convolutional Neural Network (CNN) is an effective method to model spatial dependencies of traffic demands [6], [7], and Recurrent Neural Network (RNN) have been widely used to model the temporal relations of traffic flows [8], [9]. The most recent research combines CNN and RNN together to construct a joint prediction model to consider both spatial and temporal information simultaneously. For example, Zhang *et al.* [10], [11] encoded the spatial distributions of traffic demands into hidden states by residual neural network, then modeled the evolution of the hidden states by sequential model. Zonoozi *et al.* [12] and Zhou *et al.* [13] separately provided ConvGRU and ConvLSTM to simultaneously extract spatial and temporal relations based on the assumption that closer geographical proximity results in a stronger demand similarity. Additionally, Zonoozi *et al.* [12] provided a convolutional model for prediction of crowd density. Zhou *et al.* [13] predicted the multi-step citiwide passenger demands by combining CNN with attention-based neural network.

Though these methods have achieved higher prediction accuracy than classical methods. A key weakness of these methods is that, they model the spatial distribution of traffic demands by convolutional neural network, but CNN was exclusively designed for image processing, not for spatial demand distribution at all. In image data, the nearby pixels have closer semantics than distant pixels, but for the spatial distribution of traffic demand, distant regions may have stronger correlation than nearby regions. Take the fixed traveling route from residence area to office area for example, the two areas are not geographically adjacency, but strongly interact with each other in terms of traffic demand. For this reason, an opportunity of developing exclusive convolution network for traffic demand prediction has been witnessed by many scholars [14], [15].

However, it is a nontrivial endeavor to design a traffic demand prediction method based on the exclusive spatial processing unit due to the following difficulties: 1) it is unclear how to construct an exclusive spatial network to capture the spatial dependencies of the traffic demands; 2) it is challenging to design a convolution computation unit to

Manuscript received August 29, 2019; revised November 19, 2019; accepted December 17, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 51778033, Grant 51822802, Grant 71901011, and Grant U1811463 and in part by the Science and Technology Major Project of Beijing under Grant Z191100002519012. The Associate Editor for this article was Y. Lv. *(Corresponding author: Leilei Sun.)*

Bowen Du, Xiao Hu, Leilei Sun, Yanan Qiao, and Weifeng Lv are with the State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: dubowen@buaa.edu.cn; xiaohu@buaa.edu.cn; leileisun@buaa.edu.cn; qiaoyanan@buaa.edu.cn; lwf@buaa.edu.cn).

Junming Liu is with the Department of Information Systems, City University of Hong Kong, Hong Kong (e-mail: junmiliu@cityu.edu.hk).

Digital Object Identifier 10.1109/TITS.2020.2966498

handle the constructed spatial network; 3) it is not easy to combine proposed computation unit with dynamic models and external factors. To address these issues, a Dynamic Transition Convolutional Neural Network (DTCNN) is proposed in this paper, which first discovers the aggregating points of pick-ups and drop-offs as hidden virtual stations. Based on these virtual stations, a traffic transition network is then constructed, where nodes are virtual stations and edges are traffic flows between node-pairs. Inspired by the graph convolutional neural network and the diffusion of traffic demand, we design a particular convolution computation unit for DTCNN to encode the spatial distribution of traffic demand into structured hidden states. Gated Recurrent Unit (GRU) is then used to model the temporal association of the hidden states. Finally, DTCNN presents the traffic prediction result by considering present hidden states and environmental factors. This paper has the following contributions:

- A traffic transition network is proposed and constructed, which discovers virtual stations by a density-peak based clustering method, and measures traffic transitions between them by large-scale historical records.
- A dynamic transition convolution unit is provided, which is able to model the complex spatial distributions and temporal dynamics of traffic demands simultaneously.
- A unifying learning framework is designed, which combines environmental factors with hidden states of traffic demands to provide precise traffic demand prediction.

The remaining of this paper is organized as follows: Section II introduces the related research, and Section III provides the preliminaries and definitions. The proposed method is presented in Section IV, and the experiments are conducted in Section V. Finally, Section VI concludes this research.

II. Related Work

A. Traffic Demand Prediction

There are some previous works on the problem of traffic demand prediction. Most of them attend to forecast inflows and outflows of crowds or vehicles for a set of locations during a specific period.

Partitioning and aggregating traffic demand into certain grids is a common way to preprocess the spatial distribution of traffic-related problems, making convenience for CNNs to extract high-level features. Wei *et al.* [5] used two predictors separately model the influence of local and spatial factors, where an artificial neural network is trained to model correlations among demands at different grids. Zhang *et al.* [11] proposed a DeepST, where traffic demand heatmaps of different timestamps are selected and concatenated together to model closeness, period and trend dynamics respectively, and then spatial features are extracted by CNNs. The idea was improved by Zhang *et al.* [10], where spatial correlations are captured by residual networks.

There are two common approaches to jointly capture spatial and temporal relations by combining deep CNNs and RNNs. First, Yao *et al.* [6], [8] employed CNNs for spatial feature extraction and then Long Short-Term Memory (LSTM) was exploited to capture temporal dependencies. Second, Zhou *et al.* [13] and Zonoozi *et al.* [12] utilized ConvLSTM and ConvGRU separately to model spatiotemporal relations at the same time. Zhou *et al.* [13] combined an encoder-decoder model, composed of deep CNN and ConvLSTM, and an attention mechanism for multistep passenger demand prediction. Yao *et al.* [8] provided different LSTM structure for capturing long-term periodic dependency, temporal shifting and short-term temporal dependency. Yao *et al.* [6] proposed a DMVST-Net, consisting of three components for temporal dependency, spatial dependency and global semantic dependency, respectively. Zonoozi *et al.* [12] utilized a pyramidal ConvGRU for spatiotemporal feature extraction and reused hidden features at last cell for periodic representation learning.

B. Spatiotemporal Data Mining

Apart from traffic demand prediction, other traffic-related problems (e.g. predicting traffic speed [7], [9], [16], flow [17], [18], condition [19]), have attracted numerous researchers.

Lv *et al.* [17] simultaneously learned spatial and temporal relationship of transportation by a stacked autoencoder and a linear regression layer was used to predict traffic flow on road segments. Yu *et al.* [9] combined deep CNNs and deep LSTM for speed prediction. Lv *et al.* [7] proposed a look-up convolution layer to embed the topology of road network into convolution to capture more meaning spatial features and then LSTM was employed for temporal feature extraction. Jiang *et al.* [18] proposed a DeepUrbanMomentum model for predicting crowd mobility under big rare events or disasters, when people change their behaviors dramatically. Cui *et al.* [16] considered both forward and backward dependencies of time series and captured both spatial features and bidirectional temporal dependencies by LSTM. Ye *et al.* [20] extracted spatial relationships by an autoencoder composed of CNNs and then utilized a heterogeneous LSTM to capture the dynamics of multiple transportation demands.

However, most of those methods tend to model spatial relations of spatiotemporal data as bitmaps and ignore the topological structure of the transportation network. Some scholars realized the importance of inherent topological structure of spatiotemporal data [7], and their proposed methods show some serious potential and promising improvement on predicting accuracy.

C. Dynamic Graph Convolution Network

For the past few years, some researchers began to pay attention to an emerging field: Graph Convolution Network (GCN) [21]–[26], to generalize the huge success of CNNs to topological structure data. Allowing to model the spatial distribution as topological graphs instead of grids, GCN has an excellent advantage for spatiotemporal data mining once combined with temporal dynamic property.

Dynamic GCN has been successfully applied for action recognition [27] and motion detection [28]. Some scholars began to make use of dynamic GCN for traffic-related problem

TABLE I
MATHEMATICAL NOTATIONS

Notation	Comments
$\lvert\cdot\rvert$	The cardinality of a set
$\lVert\cdot\rVert$	The norm of a vector or a matrix
$\star_{\mathcal{G}}$	Graph convolution operation on $\mathcal{G}$
$\boldsymbol{X}^{(t)}$	Observations at time step t
$\boldsymbol{Y}^{(t)}$	Estimated demand at time step t
$\mathbb{P}_i^{(t)}$, $\mathbb{D}_i^{(t)}$	Pick-ups and drop-offs of station i at time step t
$\mathcal{G}^{(t)}$	Transition network at time slot t
$\boldsymbol{W}^{(t)}$	Adjacent matrix of $\mathcal{G}^{(t)}$
$\boldsymbol{D}_I^{(t)}$, $\boldsymbol{D}_O^{(t)}$	Inflow and outflow matrix at time slot t
$\boldsymbol{I}_N$	Identity matrix of size N
$\boldsymbol{H}^{(t)}$	Hidden state at time step t
θ, $\boldsymbol{w}$	Trainable parameters
S_i	A virtual station, $1 \leq i \leq N$

recently. Liao *et al.* [19] and Yao *et al.* [8] made use of GCN to embed semantic features to improve predicting accuracy. But the developments are still lagging behind what it is capable of. Employing GCNs as spatial relation extractors and then capturing temporal dependencies by LSTM or GRU [29]–[32] remain popular in the early exploration of dynamic GCN. Li *et al.* [14] and Cui *et al.* [15] integrated GCN and RNN to capture both spatial and temporal dependencies of transportation simultaneously. Yu *et al.* [33] combined graph convolutions and gated temporal convolutions to extract spatial and temporal features. Wang *et al.* [34] utilized two convolutions respectively applied to spatial and temporal dimensions.

However, the application of dynamic GCN on spatiotemporal data mining is still in the phase of exploration and attempt. Most of them modeled the complicated traffic network as a static graph, hence they failed to capture its dynamic nature, which limits the predicting performance. To address such limitations, we adopt graph convolution on a dynamic transition network with evolving flows. Moreover, we extract spatial and temporal correlations simultaneously.

III. PROBLEM FORMULATION

In this section, we first introduce some preliminaries used throughout this paper, and then formally define the problem of traffic demand prediction.

A. Preliminaries and Notations

TABLE I lists some notations used in this paper.

Traffic demand describes the trip frequency at certain time periods in an area. According to the demand of starting a trip and the demand of reaching a destination, the traffic demand can be separated as the pick-up demand and drop-off demand.

Definition 1 (Virtual Station): Under station-undesired transportation mode such as taxi or dockless bike-sharing systems, travelers may have arbitrary departure and arrival locations scattered around a certain area. Each departure and arrival record is composed of temporal and spatial features. Traffic demands tend to congregate in certain areas, like a university or a residential quarter. Discovering and identifying areas with more distinct traffic demand changing patterns, we are capable of obtaining more accurate predictions. Pick-up and drop-off demands constitute a traffic space $\mathbb{U}$, which is divided into N disjoint parts. Meanwhile, the studied area is split into N disjoint regions $S_1, \cdots, S_N$, termed as **virtual stations**.

It is worth mentioning that most deep learning based traffic prediction methods acquire virtual stations by dividing a rectangular area into small grids to satisfy the requirement of CNN, which can only process bitmap data. Herein virtual stations can be divided arbitrarily and each of them can be arbitrarily shaped region. There are no constraints on the shape of studied area or partition manner at all. In particular, we split the studied area into virtual stations by a Density Peak Clustering (DPC) [35] based method, since it is flexible and relatively intuitive.

Definition 2 (Station-to-Station Transition): A traffic trajectory is a tuple $((p.t, p.l), (d.t, d.l))$ where p and d represent origin and destination points of the trajectory, while t and l represent the timestamp and location of a point, separately. Locations of all points belong to a physical station or a virtual station. Thus, mapping trajectories into station level, the **station-to-station transition** is redefined as $((p.t, p.s), (d.t, d.s))$, where t indicates the time step and s represents the corresponding station.

Definition 3 (Traffic Demand): We split historical traffic observations into T time steps. Each time step represents a fixed time duration. Let $P_i^{(t)} = \left|\mathbb{P}_i^{(t)}\right|$ and $D_i^{(t)} = \left|\mathbb{D}_i^{(t)}\right|$ separately represent station's pick-up and drop-off demand of S_i at time step t, namely observed or estimated departure and arrival frequencies, which are concatenated into $X_i^{(t)} = [P_i^{(t)}, D_i^{(t)}]^T$ or $Y_i^{(t+1)} = [P_i^{(t+1)}, D_i^{(t+1)}]^T$ to indicate the observed **traffic demand** of station i at time step t or estimated traffic demand at time step $t+1$.

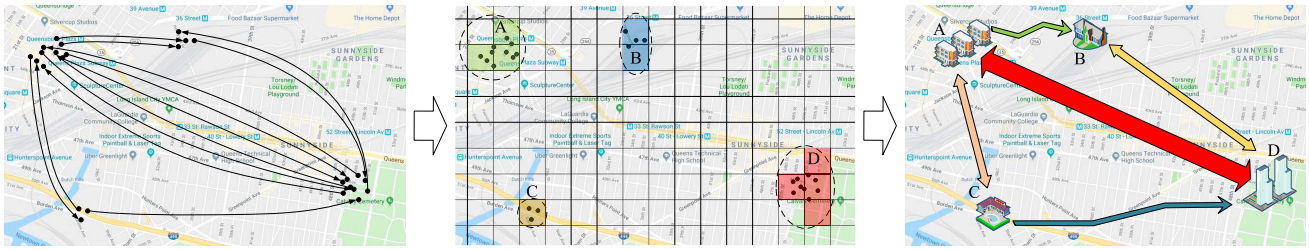
In this paper, we define transition network as a series of graphs and focus on structured traffic time series. Traffic demand $X_i^{(t)}$ is linked by pairwise connection in the transition network and affected by previous traffic observations. For example, pick-ups of one station over time are related to drop-offs around this area recently. Transition flows are proposed to quantize such relations, which are the count of vehicles traveling from one station to another. The larger transition flow between two stations is, the stronger relationship between them is. We then construct a series of directed graphs termed as transition network to capture overall relations.

Definition 4 (Transition Flow): Let tuple $((k, i), (l, j))$ represents a trip from station S_i to S_j, where $1 \leq i, j \leq N$, $1 \leq k, l \leq T$. Transition flow $W_{i,j}^{(t)}$ is defined as the number of trips from S_i to S_j started at time step t or ended at time step t.

$$W_{i,j}^{(t)} = \left|\{((k, i), (l, j)) : k = t \vee l = t\}\right|, \tag{1}$$

where $1 \leq t \leq T$. Transition flows indicate intensities of correlation between two stations at a particular time period.

To better capture temporal periodicity of traffic demand, we ensure the day-long time 24 hours divisible by length of time step (e.g., 30 minutes or one hour), and split day-long time into M time slots, which has the same length as multiple



(a) Trip Trajectories (b) Virtual Stations (c) Transition Network

Fig. 1. A sketch of the constructing process from trip trajectories to transition network: (a) the origins and destinations of trip trajectories gather in four areas; (b) the city is split into grids and virtual stations are discovered; (c) a real world partner of each virtual station exists.

time steps. As a result, time steps at the same time of day belong to the same time slot.

Since it is likely that transition flow at a time step is so sparse that the impact of the accumulated random error is nonnegligible, we redefine **transition flow** at time slot s as the sum of transition flows at time steps belonging to time slot s.

$$W^s_{i,j} = |\{((k,i),(l,j)) : k\%M = t \vee l\%M = t\}|, \quad (2)$$

where $s \in \{1, 2, \ldots, M\}$ and $\%$ is the modulo operation. $W^s_{i,j}$ is the amount of trajectories which departs from station i or arrives to station j at a certain time period of a day.

Definition 5 (Transition Network): We define the transition network as a series of directed graphs, where the graph in time slot t is denoted by $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)}, \boldsymbol{W}^{(t)})$, where $\mathcal{V}$ is the set of nodes $\mathcal{V} = \{S_i : 1 \le i \le N\}$, $|\mathcal{V}| = N$, $\mathcal{E}^{(t)}$ is the set of edges and $\boldsymbol{W}^{(t)} \in R^{N\times N}$ is a weighted adjacency matrix representing the transition flow observed on $\mathcal{G}$ in time slot t, where $1 \le t \le M$.

B. Problem Definition

1) Transition Network Construction: As shown in Fig. 1, given a set of historical departure and arrival records $((p.t, p.l), (d.t, d.l))$, each of which comprises locations and times, virtual stations are recognized by splitting the whole traffic space $\mathbb{U}$ into several parts and the studied area is divided into several regions, namely virtual stations. All trajectories are mapped into station-to-station transitions denoted by $(p.t, p.s, d.t, d.s)$ as shown in Definition 2. A dynamic transition network is constructed with stations as nodes and transitions as flows. By solving the problem of transition network construction, a set of stations and a dynamic transition network are provided for further analysis and predicting.

2) Traffic Demand Prediction: Given a set of traffic stations and the records of historical departures and arrivals at each stations, the objective of traffic demand prediction is to estimate future departure and arrival frequencies. Let $\boldsymbol{X}^{(t)} \in \mathbb{R}^{N\times 2}$ represents graph signals observed at time step t, i.e., the inflow and outflow of each station, traffic demand prediction problem aims to learn a function $h(\cdot)$ that maps T historical graph signals to the future graph signal, given a series of graphs $\mathcal{G}^{(i)}, t-T+1 \le i \le t$ and external features $\boldsymbol{E}^{(t+1)}$ such as meteorological information:

$$[\boldsymbol{X}^{(t-T+1)}, \ldots, \boldsymbol{X}^{(t)}, \boldsymbol{E}^{(t+1)}] \xrightarrow{h(\cdot)} \hat{\boldsymbol{Y}}^{(t+1)}, \quad (3)$$

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

IV. Methodology

This section presents the proposed Dynamic Transition Convolutional Neural Network (DTCNN). We first present the motivation and overview of our method, then introduce the proposed method step by step.

A. Motivation and Overview

Based on the assumption that closer geographical proximity tends to a stronger connection, existing researches mainly model the geographical structure of transition flow as bitmaps and then make use of CNN to capture high-level representation. However, regions with lower geographical proximity are likely to have a larger correlation in term of traffic demands. As Fig. 1 (c) shows, the crowd flow between the living area and the working area (the red arrow) is much bigger than that between other areas (other arrows) even if the latter distances are shorter, which provide intuitive features that help make designing of our prediction method. Therefore, we intuitively model the spatial structure as topological graphs.

Traffic status varies dramatically at different time of the day. For example, travelers flow from residential allotment to business district during the forenoon, vice versa during the afternoon. Therefore, instead of extracting geographical information from one static graph, we exploit graphs with evolving edges.

Fig. 2 shows the architecture of our proposed method which consists of three major components: (a) transition network construction, (b) dynamic transition convolution unit design and (c) embedding of environmental semantics. Each component is explained in detail below.

1) Transition Network Construction: Modes of transportation fall into two categories, namely station-required and station-undesired. As for the former, stations are directly obtained from physical stations. For the latter, we proposed a DPC-based virtual station recognition algorithm to cluster departure and arrival records into virtual stations according to geographical location. After stations discovered, a dynamic transition network is constructed from recognized stations and corresponding trip records, with stations as nodes and transition flows as edges.

2) Dynamic Transition Convolution Unit Design: The transition network is changing over time. To capture such dynamic nature, we provide a Dynamic Graph Convolutional Gated Recurrent Unit (DGCGRU) to extract spatiotemporal dependencies. DGCGRU is a variant of GRU which takes traffic observations $\boldsymbol{X}^{(\cdot)}$ and the corresponding dynamic transition network $\mathcal{G}^{(\cdot)}$ as input and estimates traffic demands in next time step while selectively retaining historical features as hidden states.

3) Embedding of Environmental Semantics: In order to employ auxiliary information and improve the predicting accuracy, hidden states memorized by dynamic graph convolution layer are combined with the embedding of external features, such as meteorological information, to produce the estimation of traffic demands in the next time step $\boldsymbol{Y}^{(t+1)}$.

B. Transition Network Construction

We analyze the underlying distribution of traffic demands and virtual stations are recognized in absence of physical stations. Then the spatiotemporal features of human mobility are modeled by a dynamic transition network. As shown in Fig. 1, the transition network construction consists of two processes: virtual stations discovery and transition matrix construction.

1) Virtual Stations Discovery: The studied area is split into several regions by analyzing historical traffic transition records in 2 steps: 1) Partition the whole area into plenty of lattices; 2) Discover virtual stations from all lattices. A virtual station is a region having a distinct traffic demand pattern.

We first discuss how to partition the studied area into many fine-grained lattices to further constitute virtual stations. To reserve the real shape of a virtual station, the lattices should be small enough. Hence, we partition the area into $I \times J$ lattices L_i based on longitude and latitude, where $1 \le i \le I \times J$ and I and J are large enough to reserve fine-grained information of virtual stations. Each lattice L_i has a local density ρ_i that represents the amount of departures and arrivals within it:

Assuming that a virtual station is composed of a core with larger local density and a relatively sparser halo surrounding it, we develop a DPC-based clustering method to identify cores and label each lattice L_i as the virtual station of its nearest counterpart n_i having higher local density:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}),$$

where d_{ij} denotes the distance between lattice L_i and lattice L_j, and δ_i denotes the distance between i and n_i, $1 \le i, j \le I \times J$.

As shown in Algorithm (1), a k-dimension tree is constructed to fast search the nearest neighbors of lattices having higher local density. Instead of identifying noise caused by inevitable randomness, we assign a cluster for each lattice. Such tricks tremendously reduce the computational price in our proposed virtual station identification method.

In the process of iterating through all lattices from line 6 to 11, new station cores were discovered and each iteration takes $O(\log(I \times J))$ or $O(\sqrt{I \times J})$ time to find the nearest neighbor on average or on the worst and $O(\log(I \times J))$ time to insert a new lattice into the 2-dimension tree. Then, between line 12 and 17, each lattice was assigned to a virtual station to which its nearest neighbor belongs, with time complexity $O(N(I \times J) \log \frac{I \times J}{N})$. Therefore, the overall time complexity is the larger one between $O((I \times J)\sqrt{I \times J})$ and $O(N(I \times J) \log \frac{I \times J}{N})$.

Algorithm 1: Virtual Station Recognition

Input: $\boldsymbol{\rho}$: local density array of each lattice
$\delta_{\min}$: the minimum distance between station cores
$\rho_{\min}$: local density threshold
Output: discovered virtual stations

```
1  let n, δ be two new arrays and 𝕌 be ∅;
2  ρ ← [ρ_{i_1}, ρ_{i_2}, ..., ρ_{i_{I×J}}] ← [ρ_1, ρ_2, ..., ρ_{I×J}]; // let ρ get sorted in descending order;
3  let t be a new k-d tree initialized with i_1, where k = 2;
4  n_{i_1}, δ_{i_1} ← −1, +∞;
5  𝕌 ← 𝕌 ∪ {{i_1}};
6  for j = 2 to I × J do
7      n_{i_j} ← search nearest neighbor of i_j in t;
8      δ_{i_j} ← distance between n_{i_j} and i_j;
9      t.insert(i_j);
10     if δ ≥ δ_min ∧ ρ_{i_j} ≥ ρ_min then
11         𝕌 ← 𝕌 ∪ {{i_j}}
12 while |∪𝕌| < I × J do
13     foreach ρ_{i_j} in ρ do
14         if i_j ∉ ∪𝕌 then
15             foreach 𝕊 in 𝕌 do
16                 if n_{i_j} ∈ 𝕊 then
17                     𝕊 ← 𝕊 ∪ {i_j}
18 return 𝕌
```

2) Transition Matrix Construction: While the geographic distribution of stations is stationary for an extended period, traffic transitions from one station to another are highly time-dependent. We model spatiotemporal features of transportation as a transition network, i.e., a series of directed graphs $\mathcal{G}^{(\cdot)}$ with fixed nodes and evolving weighted edges as elaborated in Definition 5. After virtual stations recognized, we calculate the transition flows as Definition 4 shows and the transition flow matrices at each time slot in Formula (2) is represented by $\hat{\boldsymbol{W}}^{(\cdot)}$, where $\hat{W}^{(\cdot)}_{i,j}$ is the number of transitions from station S_i to station S_j. Then we normalize original adjacent matrices $\hat{\boldsymbol{W}}^{(\cdot)}$ as shown below:

$$W^{(\cdot)}_{i,j} = \frac{\hat{W}^{(\cdot)}_{i,j}}{\left\| \hat{\boldsymbol{W}}^{(\cdot)}_{i,:} \right\|} + 1_{i=j},$$

where $\| \cdot \|$ is the L_2-norm of a vector and

$$1_p = \begin{cases} 1, & \text{if } p \text{ is true,} \\ 0, & \text{if } p \text{ is false.} \end{cases}$$

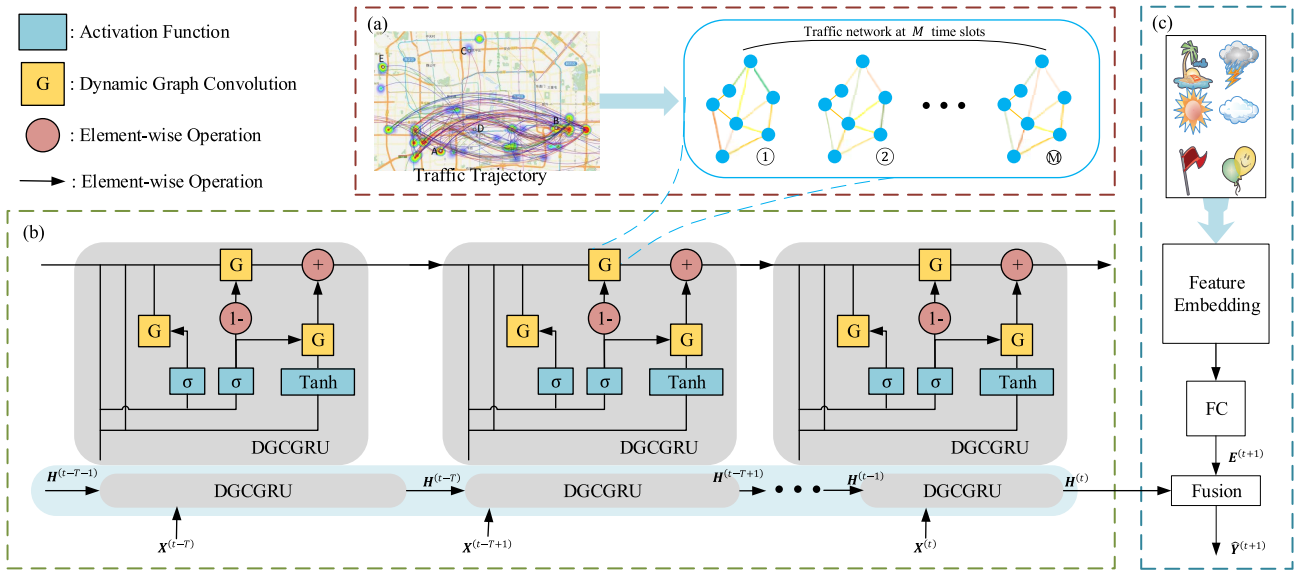


Fig. 2. Overview of Dynamic Transition Convolution Neural Network (DTCNN), consists of three components: (a) transition network construction, which discovers virtual stations and builds the dynamic transition network from raw crowd trajectory data; (b) dynamic transition convolution unit design, which simultaneously extracts temporal and spatial features by a Dynamic Graph Convolution Gated Recurrent Unit (DGCGRU). (c) Embedding of environmental semantics, which incorporates auxiliary information to obtain more accurate predictions.

The first term is to normalize the values of $\hat{\boldsymbol{W}}^{(\cdot)}$ between 0 and 1. Since the existence of self-dependence and periodicity in stations, the normalized transition flow matrices are added with an identity matrix, which is equivalent to adding self-loops to the transition network.

C. Dynamic Transition Convolution Unit Design

In consideration of that transition flows in urban road network are streaming from one station to another, the outflow and inflow of one station are correlated with that of its adjacent stations in previous time steps. For example, the larger outflows of its adjacent stations are, the greater its inflow is. Therefore, traffic demand can be calculated in closed form:

$$g_{\boldsymbol{\theta}} \star_{\mathcal{G}} \boldsymbol{x} = \sum_{k=1}^{K} (\theta_{k,0}(\boldsymbol{D}_I^{-1}\boldsymbol{W}^T)^k + \theta_{k,1}(\boldsymbol{D}_O^{-1}\boldsymbol{W})^k)\boldsymbol{x}, \quad (4)$$

where k is the look back step to convolve k^{th}-order neighborhood of a station and $\star_{\mathcal{G}}$ represents the convolution operation on graph $\mathcal{G}$. As Formula (4) shows, $\boldsymbol{W}$ is the normalized transition matrix and $(\boldsymbol{D}_I^{-1}\boldsymbol{W}^T)$ and $(\boldsymbol{D}_O^{-1}\boldsymbol{W})$ represent the upstream and downstream relations, while $\boldsymbol{D}_I$ and $\boldsymbol{D}_O$ are respectively the inflow matrix and outflow matrix, where $\boldsymbol{D}_I = \text{diag}(\boldsymbol{W}^T\boldsymbol{1})$ and $\boldsymbol{D}_O = \text{diag}(\boldsymbol{W}\boldsymbol{1})$, and $\boldsymbol{1} \in \mathbb{R}^N$ denotes the all one vector. We assign trainable weights $\boldsymbol{\theta} \in \mathbb{R}^{K\times 2}$ to inflow and outflow matrix at each stride respectively. Such an operation is capable of capturing the influence from both upstream and downstream traffic.

In general, the computational price of the operation is expensive. Therefore, we endeavor to improve calculative efficiency. Similar to [24] and [22], after limiting the convolution step to $K = 1$, flexibility and capacity can still be obtained by deep structure or recurrent structure while reducing the computational overhead.

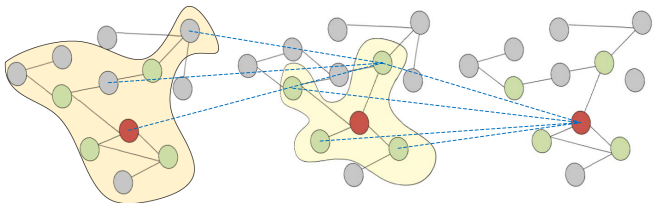


Fig. 3. Illustration of graph convolution on dynamic transition network.

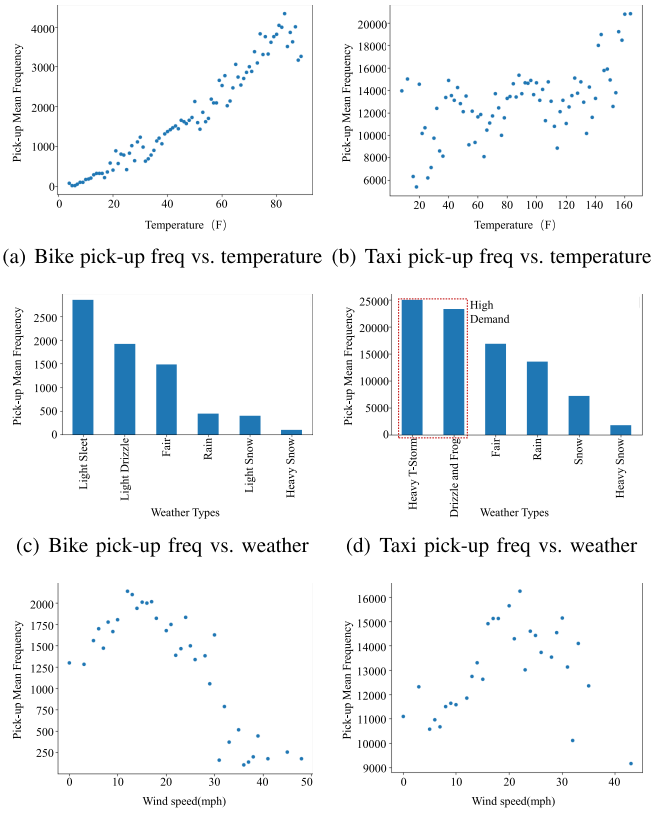
$$g_{\boldsymbol{\theta}} \star_{\mathcal{G}} \boldsymbol{x} \approx (\theta_0(\boldsymbol{D}_I^{-1}\boldsymbol{W}^T) + \theta_1(\boldsymbol{D}_O^{-1}\boldsymbol{W}))\boldsymbol{x}. \quad (5)$$

On account of that a transition flow and the corresponding reversed flow cancel each other out in the long run, let $\theta = \theta_0 = -\theta_1$, we further approximate graph convolution and accelerate computation by reducing the number of trainable parameters to 1:

$$g_\theta \star_{\mathcal{G}} \boldsymbol{x} \approx \theta\left(\boldsymbol{D}_I^{-1}\boldsymbol{W}^T - \boldsymbol{D}_O^{-1}\boldsymbol{W}\right)\boldsymbol{x}. \quad (6)$$

As shown in Fig. 3, information from neighbors is aggregated and used to update nodes' hidden states step by step, where the information disseminates through the dynamic transition network. With the stacked transition convolution, infromation from a larger neighborhood is captured by nodes.

To extract temporal and spatial features simultaneously, inspired by [14], [36], we replace matrix multiplications in GRU with our proposed graph convolution operation between



(a) Bike pick-up freq vs. temperature (b) Taxi pick-up freq vs. temperature

(c) Bike pick-up freq vs. weather (d) Taxi pick-up freq vs. weather

(e) Bike pick-up freq vs. wind speed (f) Taxi pick-up freq vs. wind speed

Fig. 4. Traffic demands significantly vary over and above natural variabilities, such as temperature, weather type and wind speed.

features and filters f_θ by graph $\mathcal{G}^{(\cdot)}$.

$$\begin{aligned}
\boldsymbol{r}^{(t)} &= \sigma(f_{\theta_r} \star_{\mathcal{G}^{(t)}} [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b}_r), \\
\boldsymbol{u}^{(t)} &= \sigma(f_{\theta_u} \star_{\mathcal{G}^{(t)}} [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b}_u), \\
\boldsymbol{C}^{(t)} &= \tanh(f_{\theta_C} \star_{\mathcal{G}^{(t)}} [\boldsymbol{X}^{(t)}, (\boldsymbol{r}^{(t)} \odot \boldsymbol{H}^{(t-1)})] + \boldsymbol{b}_C), \\
\boldsymbol{H}^{(t)} &= \boldsymbol{u}^{(t)} \odot \boldsymbol{H}^{(t-1)} + (1 - \boldsymbol{u}^{(t)}) \odot \boldsymbol{C}^{(t)},
\end{aligned}$$

where $\boldsymbol{X}^{(t)}$ and $\boldsymbol{H}^{(t)}$ denote input features and hidden states at time step t, $\mathcal{G}^{(t)}$ denotes the corresponding graph of the time slot to which time step t belongs, $\boldsymbol{b}_r$, $\boldsymbol{b}_u$ and $\boldsymbol{b}_C$ are three biases, σ and tanh are activation functions and $\odot$ is the element-wise multiplication.

D. Embedding of Environmental Semantics

Transition flows are affected by many complex external factors, such as meteorological information. Fig. 4 shows that transition flows vary significantly over different weather patterns. Let $\boldsymbol{E}^{(t+1)}$ be the embedding vector representing these external factors at the predicted time step $t+1$.

In our experiment, we take these auxiliary information into consideration. The features with respect to environmental context are not explicitly associated in terms of space or time, and such factors impact traffic demand in a complex and non-linear manner. Therefore, fully connected networks are suitable for auxiliary information extraction. The concatenation of hidden states $\boldsymbol{H}^{(t+1)}$ from dynamic transition convolution unit and external feature embedding $\boldsymbol{E}^{(t+1)}$ is linear transformed to obtain the prediction result $\hat{\boldsymbol{Y}}^{(t+1)}$.

$$\hat{\boldsymbol{Y}}^{(t+1)} = \sigma(\boldsymbol{W}_O \cdot [\boldsymbol{H}^{(t+1)}; \boldsymbol{E}^{(t+1)}] + \boldsymbol{b}_O), \tag{7}$$

where $\boldsymbol{W}_O$ and $\boldsymbol{b}$ are weights and biases, and σ is the $ReLU$ activation function.

E. Loss Function

The loss function used for training our proposed model is defined as Formula (8):

$$\mathcal{L}(\boldsymbol{w}) = \left\| \hat{\boldsymbol{Y}}^{(t+1)} - \boldsymbol{Y}^{(t+1)} \right\|, \tag{8}$$

where $\|\cdot\|$ is the L_2-norm of a vector, $\boldsymbol{w}$ are all trainable parameters in our proposed model and $\boldsymbol{Y}^{(t+1)}$ is the ground truth We use Adam for optimization and TensorFlow to implement our proposed model.

V. EXPERIMENT

This section evaluates the effectiveness of our proposed method by experiments conducted on real-world datasets. To get a comprehensive evaluation, five baselines and three metrics are employed.

A. Data Set

We conduct experiments on two real-world traffic datasets, namely Bike-NYC and Taxi-NYC, and a weather dataset is used to supplement additional information:

1) **Bike-NYC**[1] includes fields capturing dates, times and station IDs of pick-up and drop-off points. We use this dataset to verify traffic demand prediction component.
2) **Taxi-NYC**[2] includes fields capturing times and locations (longitude and latitude) of pick-up and drop-off points. We use this dataset to verify virtual station identification component and traffic demand prediction component.
3) **Weather-NYC**[3] is collected from Kennedy International Station (40.6413°N, 73.7781°W) hourly. In both of aforementioned datasets, external weather data are combined to acquire more accurate prediction results.

All datasets and their detailed statistics are presented in TABLE II.

TABLE II
DETAILS OF DATASETS

Data Source	Bike-NYC	Taxi-NYC
Time from to	2018/1/1 2018/6/30	2016/1/1 2016/6/30
#Weekdays (#Weekends)	130 51	130 52
#Holidays	6	7
#(Virtual) Stations	816	1192
#Records	7624136	78424550

[1]https://www.citibikenyc.com/system-data
[2]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[3]https://www.wunderground.com/history/daily/us/ny/new-york-jfk

B. Baselines and Metrics

We compare our model with following methods:

- **Historical Average (HA)**: simply taking the average value of historical inflow and outflow at the same time intervals in previous days as the prediction.
- **Vector Auto-Regression (VAR)**: the multi-variate extension of auto-regressive model which is a classic machine learning method for time series prediction.
- **XGBoost** [37]: a powerful boosting tree-based method, widely used in many regression tasks.
- **RNN**, **LSTM** [38] and **GRU** [39]: popular deep learning models for temporal data, which are widely used in time series prediction. We compare all of them with our proposed model.
- **Diffusion Convolutional Recurrent Neural Network (DCRNN)**: a graph convolution-based method for spatiotemporal traffic forecasting proposed in [14]. They replace the matrix multiplications in GRU with their proposed diffusion convolution. We apply such a variant of GRU to predict future traffic demand.

Root Mean Squared Error ($RMSE$), Pearson's Correlation Coefficient (PCC) and Mean Absolute Error (MAE) are adopted to measure and evaluate the performance of different methods as follows:

- $RMSE$ is widely used in measuring the error of regression methods, which is defined as:

$$RMSE = \sqrt{\frac{1}{\xi}\sum_{i=1}^{\xi}(\hat{y}_i^{(t+1)} - y_i^{(t+1)})^2}, \tag{9}$$

  where $\hat{y}_i^{(t+1)}$ and $y_i^{(t+1)}$ mean the prediction value and ground truth of station i in time interval $t+1$, and ξ is the total number of samples.
- PCC is used to measure the linear correlation between two variables. It has a value between -1 and 1, where -1 indicates total negative linear correlation, 1 indicates total positive linear correlation and 0 indicates that there is no linear correlation.

$$PCC = \frac{\sum_{i=1}^{\xi}(\hat{y}_i^{(t+1)} - \bar{\hat{y}}^{(t+1)})(y_i^{(t+1)} - \bar{y}^{(t+1)})}{\sqrt{\sum_{i=1}^{\xi}\left\|\hat{y}_i^{(t+1)} - \bar{\hat{y}}^{(t+1)}\right\|}\sqrt{\sum_{i=1}^{\xi}\left\|y_i^{(t+1)} - \bar{y}^{(t+1)}\right\|}},$$

  where $\bar{\hat{y}}^{(t+1)}$ and $\bar{y}^{(t+1)}$ are the average of $\hat{y}^{(t+1)}$ and $y^{(t+1)}$, respectively.
- MAE is an average of the absolute errors, which also reflects the error between prediction and ground truth.

$$MAE = \frac{1}{\xi}\sum_{i=1}^{\xi}|y_i^{(t+1)} - \bar{y}^{(t+1)}|,$$

  where $|\cdot|$ represents the absolute value.

C. Experimental Settings

The length of a time step and a time slot are both set as an hour identically. For RNN-based methods including Basic RNN, LSTM, GRU, DCRNN and our proposed methods, we set the sequence length $T = 40$. All data are divided into a training set, a validation set and a test set. The first 60% of the samples are selected for training each model, the following 20% are the validation set for hyperparameter tuning and the remaining 20% are the test set for model evaluation. We also used early-stop in all the experiments. The batch size in our experiments is set as 5.

1) Settings of Virtual Stations Identification: The region we discuss is a rectangle covering the whole New York, i.e., 40;28;38.6363°N to 40;55;3.2772°N and 73;42;0.9792°W to 74;15;32.7240°W. We divide the region into a 1000×1000 grid map. The size of each lattice is about $47m \times 49m$. With such a fine-grained grid map, we are capable of retaining the real shape of virtual stations to help us conduct further predicting. Setting the minimum distance between station centers as 50 lattice, i.e., about 2400 meters, and the local density threshold is set as 30, which means that the local densities of all station cores are greater than 30. Then 1192 virtual stations are identified. The virtual station recognition result is shown in Fig. 5(a).

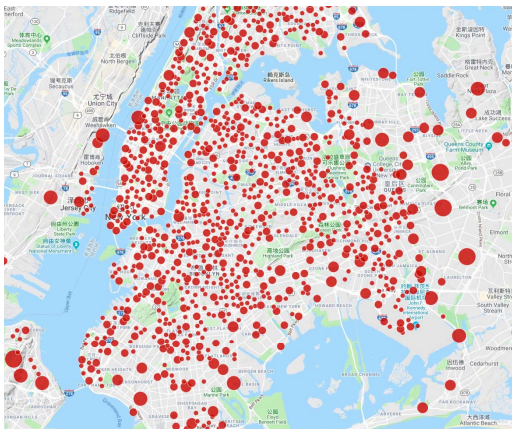
2) Settings of Environmental Semantics Embedding: The environmental semantics is composed of weather data and date metadata. Weather data was recorded by Kennedy International Station per hour. The original data contains 26 features except for timestamp. The missing values are estimated by the last valid observation before it. We supplement semantic data with the day of the week as extra information to improve forecasting accuracy. Choosing temperature, dew-point, humidity, wind direction, wind speed, air pressure, climate and sensible temperature as weather features, we encode categorical features (wind direction and climate) as an one-hot numeric array. The output dimension of semantics embedding is set as 10. We use $ReLU$ function as the activation function for semantics embedding and features fusion.

D. Experimental Results

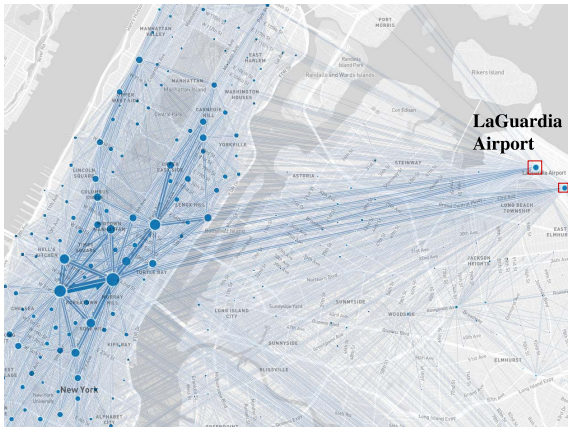
TABLE IV and TABLE III show the comparison with baselines, where DTCNN* is a DTCNN without environmental semantics combined. We conduct experiments on Bike-NYC and Taxi-NYC demands. For most methods, $RMSE$ and MAE are smaller and PCC is bigger in bike-NYC than in Taxi-NYC. This is because Taxi-NYC has a larger demand than Bike-NYC. The model we propose achieves the best performance and has the lowest $RMSE$ and MAE, and has the highest PCC on all comparison experiments.

On Bike-NYC demand prediction, VAR has the largest $RMSE$ and MAE, and the lowest PCC, for bike requirements are sparse in the wee hours and bike usage is far from a stationary random process. All methods except VAR have similar performance on $RMSE$ and MAE. HA and XGBoost tend to predict the average values and deep learning methods are able to capture the variance of bike demand, which have higher PCC compared with classical methods.

On Taxi-NYC demand prediction, classical machine learning methods show a relatively poor performance than deep learning methods except for DCRNN. HA has the largest



(a) Virtual Stations Identification Result: It is identified using taxi trajectory data. Bigger red filled circle denotes virtual stations covering a larger area.



(b) Taxi Transition Network: Bigger blue filled circle denotes virtual stations having a larger inflow and outflow. Transition flows with amount of trajectories below 30 are filtered out.

Fig. 5. Taxi virtual station identification result (a) and transition network construction result (b).

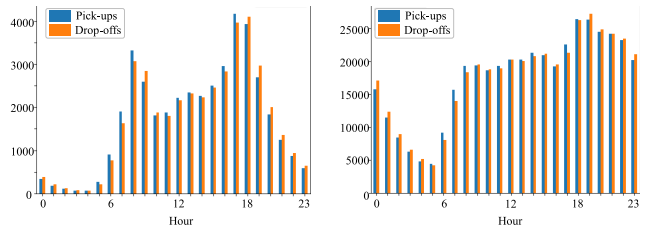
TABLE III
RESULTS COMPARISON ON BIKE-NYC

Method	*RMSE*	*PCC*	*MAE*
HA	3.906493	0.541540	1.991092
VAR	20.118131	0.329602	7.208243
XGBoost	3.512882	0.561269	2.007128
Basic RNN	3.864767	0.779739	1.872700
LSTM	3.897652	0.788945	1.850340
GRU	4.295366	0.719893	2.162775
DCRNN	3.550321	0.815017	1.739865
DTCNN*	3.444087	0.820051	1.740992
DTCNN	**3.436808**	**0.820185**	**1.703449**

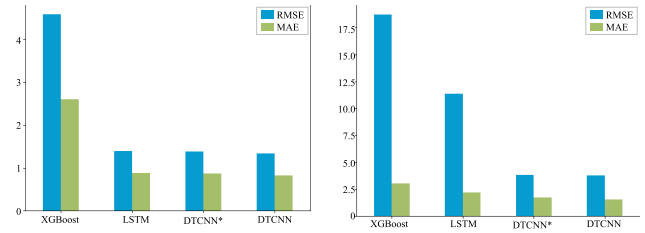
RMSE and *MAE* value and the lowest *PCC*. Although HA and AR extract temporal features to some extent, they fail to employ spatial dependencies, which may explain the worse performance. In deep learning methods, three RNN models show similar performance. Basic RNN represents different performance compared with LSTM and GRU. Because LSTM and GRU capture both long and short term correlations to benefit the predicting accuracy. Basic RNN only takes short-term dependencies into consideration since its lack of memory mechanism. The worse results of DCRNN is because of its high computational requirements and our limited resources.

TABLE IV
RESULTS COMPARISON ON TAXI-NYC

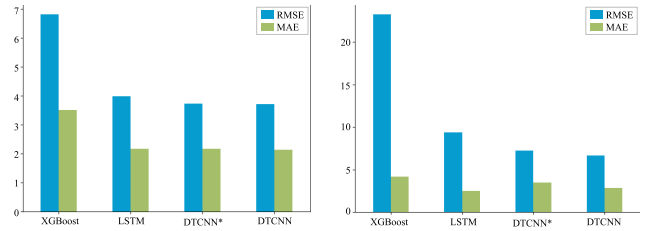
Method	*RMSE*	*PCC*	*MAE*
HA	38.663564	0.673600	9.529034
VAR	16.209607	0.954641	3.083265
XGBoost	16.973139	0.925507	2.955025
Basic RNN	15.390889	0.951026	3.749574
LSTM	13.077499	0.965589	2.930318
GRU	13.429137	0.963306	3.289154
DCRNN	19.918039	0.842173	4.801315
DTCNN*	8.023104	0.987517	3.084475
DTCNN	**7.778874**	**0.988342**	**2.899619**



(a) Bike demands change over time. (b) Taxi demands change over time.



(c) Bike, 7:00 a.m. to 9:00 a.m. (d) Taxi, 12:00 a.m. to 2:00 p.m.



(e) Bike, 5:00 p.m. to 7:00 p.m. (f) Taxi, 6:00 p.m. to 8:00 p.m.

Fig. 6. Comparison of various methods during different time periods.

The traffic demand changes dramatically over time. Bike and taxi have different usage patterns. For instance, bikes are barely used during midnight but the time period where taxi has the fewest demand is early in the morning. To further investigate the practicality of our methods, the experiment results comparison is zoomed in time periods having high traffic demands. We compare DTCNN and DTCNN* with XGBoost and LSTM, which separately have the best experiment performance in classic machine learning methods and deep learning methods. As Fig. 6 shows, our two proposed methods have the best performance.

VI. CONCLUSION AND DISCUSSION

For the purpose of precise traffic demand prediction, a novel deep neural network is proposed in this paper, namely,

Dynamic Transition Convolutional Neural Network (DTCNN). It consists of three modules: 1) transition network construction, which encodes the discovered virtual stations as nodes, and transition flows between them as edges; 2) a dynamic transition convolution unit, which captures the spatial distributions and temporal dynamics of traffic demands simultaneously; 3) a fusion module, which integrates the hidden states of historical traffic demands with environmental factors to predict the next-period traffic demands. Experiments have been conducted on NYC taxi and bike-sharing data, the results demonstrate the superiority of DTCNN over both classical and the state-of-the-art prediction methods.

Graph Convolutional Network (GCN) is a newly emerging field in artificial intelligence, this paper contributes a novel dynamic transition GCN method to this field, and also provides an effective method for traffic demand prediction. In future, we plan to further investigate the following problems: 1) modeling the spatial and temporal dependencies by transition networks with dynamic nodes; 2) increasing the depth of the model and further decreasing the computational complexity; 3) extending this model to multi-step forecasting.

REFERENCES

[1] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2024, no. 1, pp. 116–125, Jan. 2007.

[2] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

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

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

[5] H. Wei, Y. Wang, T. Wo, Y. Liu, and J. Xu, "ZEST: A hybrid model on predicting passenger demand for chauffeured car service," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2016, pp. 2203–2208.

[6] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 2588–2595.

[7] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 3470–3476.

[8] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. 33nd AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 5668–5675.

[9] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[10] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1655–1661.

[11] 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. Geograph. Inf. Syst. (GIS)*, 2016, pp. 92:1–92:4.

[12] A. Zonoozi, J.-J. Kim, X.-L. Li, and G. Cong, "Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3732–3738.

[13] 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 (WSDM)*, Feb. 2018, pp. 736–744.

[14] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2018, pp. 1–16.

[15] 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 Trans. Intell. Transp. Syst.*, to be published.

[16] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," in *Proc. 6th Int. Workshop Urban Comput. (UrbComp)*, Aug. 2017, pp. 1–9.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[18] R. Jiang *et al.*, "Deepurbanmomentum: An online deep-learning system for short-term urban mobility prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 784–791.

[19] B. Liao *et al.*, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2018, pp. 537–546.

[20] 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 (KDD)*, Aug. 2019, pp. 305–313.

[21] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[22] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 2014–2023.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 3837–3845.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017, pp. 1–14.

[25] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *CoRR*, vol. abs/1812.08434, 2018.

[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *CoRR*, vol. abs/1901.00596, 2019.

[27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 33nd AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 7444–7452.

[28] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.

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

[30] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A graph convolutional sequence-to-sequence learning approach with attention mechanism," *CoRR*, vol. abs/1810.10237, 2018.

[31] J. Hu, C. Guo, B. Yang, C. S. Jensen, and L. Chen, "Recurrent multi-graph neural networks for travel cost prediction," *CoRR*, vol. abs/1811.05157, 2018.

[32] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[33] 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.

[34] M. Wang, B. Lai, Z. Jin, X. Gong, J. Huang, and X. Hua, "Dynamic spatio-temporal graph-based cnns for traffic prediction," *CoRR*, vol. abs/1812.02019, 2018.

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

[36] 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. 29th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2015, pp. 802–810.

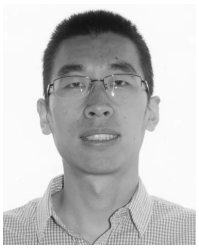
[37] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2016, pp. 785–794.

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

[39] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

Bowen Du received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently a Professor with the State Key Laboratory of Software Development Environment and the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), School of Computer Science and Engineering, Beihang University. His research interests include smart city technology, multisource data fusion, and traffic data mining.

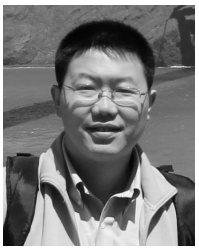
Xiao Hu received the B.S. degree from the College of Software, Beihang University, Beijing, China, in 2018, where he is currently pursuing the M.S. degree in computer science and engineering. His research interests include intelligent transportation, deep learning, and smart city technology.



Leilei Sun received the B.S. and M.S. degrees from the School of Control Theory and Control Engineering, Dalian University of Technology, in 2009 and 2012, respectively, and the Ph.D. degree from the Institute of Systems Engineering, Dalian University of Technology, in 2017. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment (SKLSDE) and the Big Data Brain Computing Lab (BDBC Lab), Beihang University, Beijing, China. He was a Post-Doctoral Research Fellow with the School of Economics and Management, Tsinghua University, from 2017 to 2019. His research interests include machine learning and data mining. He has published many articles on the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), *Knowledge and Information Systems* (KAIS), and *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD).

Junming Liu received the B.S. degree from the University of Science and Technology of China and the Ph.D. degree from the Rutgers Business School, Rutgers University. He is currently an Assistant Professor with the Department of Information Systems, City University of Hong Kong. His general areas of research are data mining and supply chain analytics, with a focus on developing data mining techniques for emerging big data and supply chain applications.

Yanan Qiao received the B.S. degree from the College of Computer Science, Sichuan University, China, in 2018. She is currently pursuing the M.S. degree in computer science and engineering with Beihang University, Beijing, China. Her research interests include intelligent transportation, deep learning, and sequential prediction.



Weifeng Lv received the B.S. degree from Shandong University, Jinan, China, in 1992, and the Ph.D. degree from Beihang University, Beijing, China, in 1998, all in computer science and engineering. He is currently a Professor with the State Key Laboratory of Software Development Environment and the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), School of Computer Science and Engineering, Beihang University. His research interests include smart city technology and mass data processing.