World Wide Web (2021) 24:805–823
https://doi.org/10.1007/s11280-021-00877-4

On prediction of traffic flows in smart cities: a multitask deep learning based approach

Fucheng Wang[1] · Jiajie Xu[1] · Chengfei Liu[2] · Rui Zhou[2] · Pengpeng Zhao[1]

Received: 29 May 2020 / Revised: 2 December 2020 / Accepted: 15 March 2021 /
Published online: 1 April 2021

Abstract
With the rapid development of transportation systems, traffic data have been largely produced in daily lives. Finding the insights of all these complex data is of great significance to vehicle dispatching and public safety. In this work, we propose a multitask deep learning model called *Multitask Recurrent Graph Convolutional Network* (MRGCN) for accurately predicting traffic flows in the city. Specifically, we design a multitask framework consisting of four components: a region-flow encoder for modeling region-flow dynamics, a transition-flow encoder for exploring transition-flow correlations, a context modeling component for contextualized fusion of two types of traffic flows and a task-specific decoder for predicting traffic flows. Particularly, we introduce *Dual-attention Graph Convolutional Gated Recurrent Units* (DGCGRU) to simultaneously capture spatial and temporal dependencies, which integrate graph convolution and recurrent model as a whole. Extensive experiments are carried out on two real-world datasets and the results demonstrate that our proposed method outperforms several existing approaches.

Keywords Traffic flow · Deep learning · Graph convolutional networks · Multitask learning

1 Introduction

In recent years, the concept of smart city has been proposed for delivering a better quality of life for residents living in the city. Meanwhile, citywide traffic flow prediction plays an important role in the construction of smart cities [1–3]. As Figure 1 shows, it aims to predict traffic flows of each region at given time intervals based on historical observations.

✉ Fucheng Wang
wfcandni@163.com

Jiajie Xu
xujj@suda.edu.cn

[1] Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Soochow, China

[2] Swinburne University of Technology, Swinburne, Australia

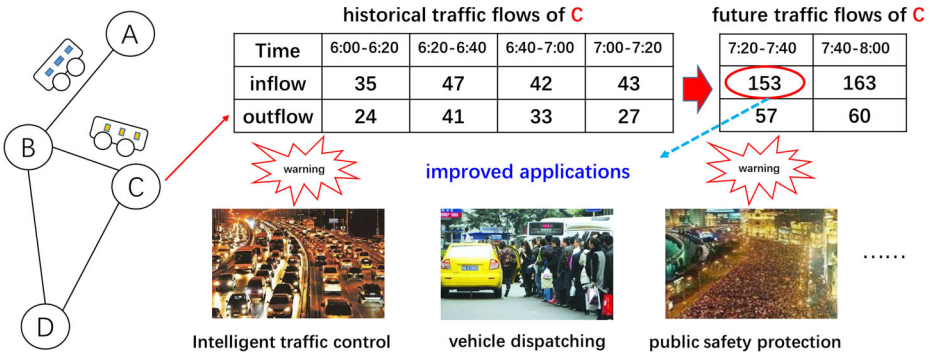

Figure 1 An instance of traffic flow prediction

If we can accurately predict the traffic flows, it will be greatly meaningful to intelligent transporation, vehicle dispatching and public safety protection [4].

However, accurately predicting traffic flows in a citywide range remains to be challenging due to complicated spatial-temporal dependencies and various context information (e.g., holiday, weekend). Fortunately, deep learning [5] has been introduced as a powerful tool for analyzing these complex traffic data. Recently, by dividing the whole city into grid-like regions and employing convolutional neural networks (CNN) for spatial modeling, several works have achieved impressive progress [6–9]. However, the regions in a city are actually separated by road networks, which naturally segment urban areas into sub-regions with v arying sizes and shapes [10–12]. Thus, the graph-based structure is better to depict real-world conditions and reveal hidden semantic meanings. As a result, CNN based models which require input to be grid-like data are not perfectly applicable in this scenario. To tackle this problem, graph convolutional networks (GCN) have been proposed by researchers [13, 14].

Although graph convolutional networks have shown powerful capability of modeling spatial dependencies in graph-based data, there still exist two main limitations for current GCN models: (1) First, the spatial dependencies among regions are highly dynamic under different circumstances. For instance, the traffic flows tend to transfer from residential areas to working areas in the morning while the opposite in the evening. The adjacency matrix pre-defined in GCN is fixed and thus cannot well respond to temporal dynamics. (2) Second, existing GCN models mainly utilize the results learned from the largest sub-graph structure (e.g., k-hop neighbors), ignoring the knowledge obtained in the early-stage, which results in a significant loss of information.

Furthermore, we find that the traffic flow of a region is strongly correlated to the transition-flow between different regions. Specifically, the region flow is the sum of its transition-flows with other regions, while this transition-flow describes the correlation between individual regions. Most studies merely focus on the region-flow other than the transition-flow [6–8], which results in incomprehensive prediciton. In addition, although there exist so many combinations of transitions, some of them may not occur in reality and some combinations do not happen for many times, which makes the transition-flow highly sparse and poses a challenge for training deep learning models.

To address the above issues, we propose a multitask deep learning model called MRGCN for traffic flow prediction. Overall, our idea is specified as the following process. First, we introduce a novel dual-attention mechanism to improve the performance of original GCN on spatial modeling. Then, we incorporate the dual-attention graph convolution into the RNN

structure and thus propose DGCGRU to capture temporal dynamics. Next, we design a multitask architecture to predict two types of traffic flows (region-flow and transition-flow) at the same time. Particularly, the recent and periodical traffic data will be extracted for predicting future traffic flows in each task. At last, we aggregate the information learned from each task and make task-specific predictions. We summarize our contributions as follows:

- We propose dual-attention graph convolution for spatial modeling, which consists of neighbor-aware attention and hop-aware attention. We further incorporate it into RNN structure to capture temporal dynamics.
- We propose a multitask framework called MRGCN for spatial-temporal modeling, which integrates a region-flow encoder, a transition-flow encoder, context modeling and a task-specific decoder as a whole.
- We conduct extensive experiments on two real-world datasets from Beijing and Chengdu. The results demonstrate that our proposed method achieves better performance compared with several existing approaches.

As an extension of our previous work [15], we further polish our original solution by improving the components of multitask framework and enriching the experiments. Specifically, our new contributions are as follows:

- We propose a new RNN structure called *Dual-attention Graph Convolutional Gated Recurrent Units* (DGCGRU), which combines the recurrent model with dual-attention graph convolution as an integration.
- We design a context attention mechanism which takes various context factors (e.g., holiday and weekend) into consideration, by which our model can adaptively assign different weights to different branches of traffic flows.
- We add more experiments including baseline comparisons and ablation study for our newly proposed approach. The results show the effectiveness of our method. In addition, we give a more detailed description of our framework so that readers can better understand and follow our work.

The remaining parts of this paper are organized as follows: First, we review the related work about traffic flow prediction in Section 2. Then, we give several definitions and formulate the research problem in Section 3. In Section 4, the detailed information about our proposed multitask framework are discussed. Finally, we show the extensive experimental results on two real-world datasets in Section 5 and give a brief conclusion in Section 6.

2 Related work

2.1 Traffic flow prediction

Being an important problem in the smart city, traffic flow prediction has been studied for many years. Classic methods viewed it as a time series forecasting problem, using ARIMA and H-ARIMA [16] to predict future traffic flows. Some supervised learning models, such as LR [17], VAR [18] and SVR [19] considered it to be a regression problem. However, they either required the time series to satisfy the stationarity assumption or ignored the spatial dependencies in the surrounding areas, thus making inaccurate predictions.

Recently, employing deep neural networks has achieved great advances in traffic flow prediction. Since LSTM has shown great capability for capturing temporal dynamics in

sequential modeling, it is widely uesd to improve the performance of traffic state prediction [20]. Later, noticing that not only temporal dependencies but also spatial correlations are critical, the information of surrounding areas are also taken into account in traffic flow prediction [21, 22]. From then on, researchers started to use CNN as the major tool for modeling spatial dependencies. Specifically, ST-ResNet [6] proposed a residual framework to model long-range spatial dependencies throughout a city. Besides, it also considered external factors such as weather, holiday and events for a more accurate prediction. DeepSTN+ [8] proposed a ConvPlus operation in place of ordinary convolution to directly capture long-range spatial dependencies. In addition, some researchers considered utilizing transition flow for capturing dynamic spatial-temporal correlations between regions [23, 24]. To be specific, [25] proposed a multitask learning framework for jointly predicting node flow and edge flow throught a city. Apart from above CNN based models, the combination of recurrent models with CNN such as ConvLSTM [26], LC-RNN [11] and Periodic-CRN [27] were also proposed, but the training of recurrent models always consumed a lot of time.

Nevertheless, the above methods partition the whole city into regular regions and adopt CNN based models to process grid-like traffic data. When input becomes to be graph-structured data, they may not be suitable any more. Specifically, regions in a city are actually irregular and are of varying shapes and sizes. The above models ignore the road network distribution in the city and are probably not perfect to be applied to real-world scenarios.

2.2 Graph-based traffic flow prediction

Different from the above regular-regions based traffic flow research, some works were based on graph-structured data (e.g, road network data and highway sensors), which has been extensively studied as well. Recently, by employing graph convolution [28–30], several works achieved impressive results in graph based traffic flow prediction [31–33]. Specifically, STGCN [14] combined gated CNN with graph convolution to respectively model temporal dependencies and spatial correlations. ASTGCN [13] further proposed spatial attention and temporal attention to help capture traffic dynamics in these spatial-temporal data. Graph WaveNet [34] developed a novel self-adaptive adjacency matrix, which enables the original GCN to discover more hidden spatial dependencies that were not revealed by the pre-defined adjacency matrix. In addition, it adopted stacked dilated casual convolution to capture long-term temporal dependencies. DCRNN [35] modeled the dynamics of traffic flow as a diffusion process and proposed a sequence to sequence architecture for multi-step traffic flow prediction. Furthermore, MRes-RGNN [36] introduced a novel residual recurrent graph neural nwtwork, which integrated gated recurrent units with graph convolution in a gated residual learning manner.

Based on our observations, most of the existing GCN-based methods only utilize the results learned from a given neighborhood range (e.g., k-hop), yet ignoring the information obtained in the early-stage, causing an incomplete utilization of the valuable gained knowledge. Furthermore, we find that the traffic flow of a region is strongly correlated to the transition-flow between pair-wise regions. Most existing methods overlooked this important relation, which could potentially increase the prediction accuracy.

3 Preliminaries

In this section, we first give several definitions related to traffic flow prediction and then formulate the problem. Details are as follows.

Definition 1 (Urban graph) An urban graph is denoted as $G = (V, E, A)$, where V represents the set of regions and E represents the edges between regions (if two regions are geographically connected, there will be an edge between them). A is a binary unweighted adjacency matrix. Figure 2a shows an example of urban graph, where V={R_1, R_2, R_3, R_4}, E consists of five edges and the adjacency matrix A can be obtained by V and E.

Definition 2 (Inflow/outflow) In this work, we mainly study two types of region-flows: inflow and outflow. At a given time interval, inflow is the total number of traffic flows entering a region while outflow is the total number of traffic flows leaving a region for other places. At time interval t, we use $X_t \in R^{N \times C}$ to denote the region-flow of N nodes. In this paper, C equals to 2. $X_t[i, 0]$ and $X_t[i, 1]$ respectively denote inflow and outflow of node i at time interval t. Then we can obtain a flow matrix like Figure 2c from b which shows the traffic flow transitions between different regions. Taking R_1 as an example, the transition from R_1 to R_2 and that from R_1 to R_4 constitute the outflow of R_1, while the transition from R_2 to R_1 and that from R_3 to R_1 comprise the inflow of R_1.

Definition 3 (Transition-flow) We mainly study two types of transition-flows: in-transition and out-transition. At time interval t, we use $Z_t \in R^{N \times 2N}$ to denote the transition-flow of N nodes. Specifically, $Z_t[i, k]$ ($k = 0, 1, ..., N-1$) denotes the in-transitions from all N nodes to node i and $Z_t[i, k]$ ($k = N, N+1, ..., 2N-1$) denotes the out-transitions to other nodes from node i. Then we can obtain a transition matrix like Figure 2d from b. Taking R_1 as an example, it has up to 8 possibilities of transition-flows. The in-transition of R_1 includes the transitions from R_2 and R_3 while the out-transition is composed of the transitions from R_1 to R_2 and R_1 to R_4.

3.1 Problem Formulation

Given a graph $G = (V,E,A)$ and historical observations of region-flows and transition-flows $\{X_t, Z_t | \text{ t} = 1,...,\text{T}\}$, we aim to predict region-flow $X = \{X_j | \text{ j=T+1,...,T+}z\}$ and transition-flow $Z = \{Z_j | \text{ j=T+1,...,T+}z\}$, where z is the number of time intervals to be predicted.

4 Methodology

4.1 Overview

Traditional traffic flow prediction task only predicts the total number of traffic flows entering or leaving a region (i.e., inflow and ouflow), while we argue that the transition-flow prediction is equally important based on the following explanations: (1) The transiton-flow depicts the pair-wise transition patterns among regions, which can be further used to analyze the correlations between regions. Thus, it is meaningful for both urban development and regional governance to study the transition-flow between regions. (2) We find that the sum of transition-flow equals to the region-flow for each region. Hence, these two types of traffic flows are quite correlated as they can be seen as each other's auxiliary information. (3) Region-flow and transition-flow are different representations of each region's traffic attributes, thus they share the knowledge about hidden traffic patterns. Since multi-task learning [37, 38] aims to leverage useful information contained in multiple tasks to help improve the generalization performance of those tasks, we consider using multitask learning to enhance the accuracy of both region-flow and transition-flow prediction.

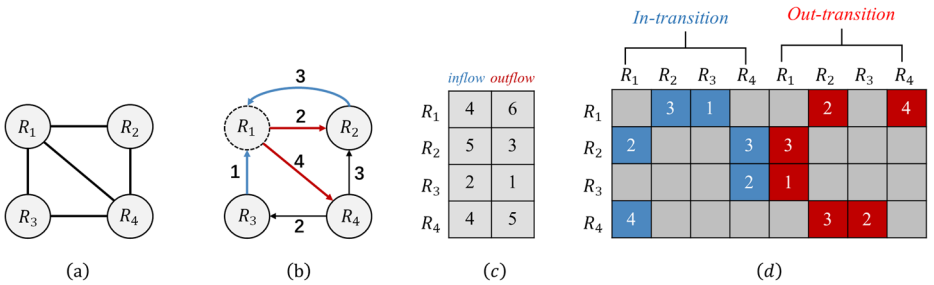

Figure 2 Region-flow and transition-flow in different regions

Following the idea of multi-task leanring, we propose our model and Figure 3 presents the architecture of our proposed MRGCN. It mainly consists of four components: a region-flow encoder, a transition-flow encoder, context modeling and a task-specific decoder. Overall, it simultaneously carries out two prediction tasks: region-flow prediction and transition-flow prediction. These two tasks have similar network structures and the only difference is that the model of transition-flow prediction has an extra embedding layer to alleviate the data sparsity problem. In each task, the recent and periodical traffic flows are extracted for spatial-temporal modeling, followed by a context attention module to adaptively assign different weights to different branches of traffic data. Finally, we aggregate the information learned from each task and make task-specific predictions (i.e., region-flow and transition-flow).

4.2 Region-flow encoder

In our region-flow encoder, we select both recent and periodic time steps for spatial-temporal modeling by DGCGRU and fully-connected networks.

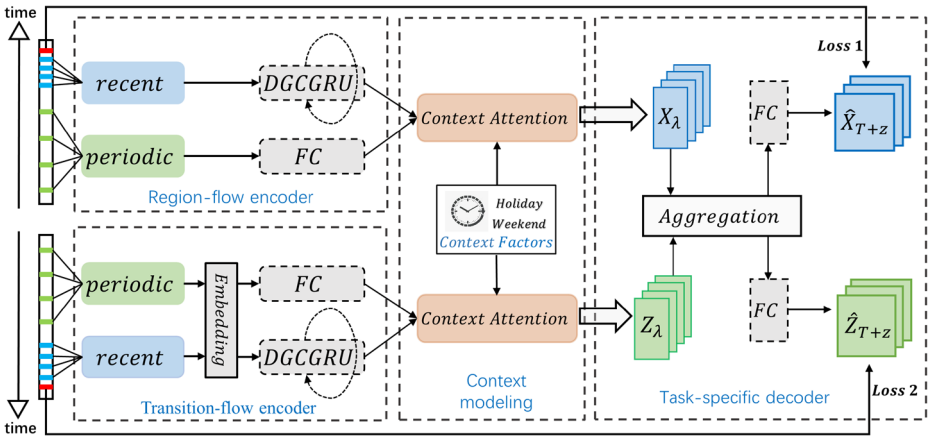

Figure 3 **Architecture of MRGCN**. DGCGRU: *Dual-attention Graph Convolutional Gated Recurrent Units*; FC: *Fully-Connected layer*

4.2.1 Dual-attention Graph Convolutional GRU

Intuitively, the traffic flow of a region is influenced by its recent time intervals. For instance, once a congestion takes place, it will inevitably affect the traffic flows in the near future. Thus, we select traffic flows from the recent P time intervals with regard to the current time step T. Specifically, we denote them as $\chi_{rec} = [X_{T-P+1}, X_{T-P}, ..., X_T]$. Then, χ_{rec} will be fed into DGCGRU which consists of dual-attention graph convolution and gated recurrent units.

Due to the geographical connection among urban areas, the traffic flow of a region is naturally affected by that of nearby regions. Furthermore, owing to the rapid development of urban transportation systems, people can reach distant areas just within a short time. Thus, we need to consider both near and distant correlations in spatial modeling. In this work, we propose using GCN to model spatial dependencies in graph-based data.

Graph convolution Most recent works employ graph convolutional networks to model spatial dependencies on irregular regions. According to [29], the graph convolution operation in l-th layer is defined as follows:

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

where $H^{(l+1)}$ and $H^{(l)}$ are respectively output and input of the l-th hidden layer. $\hat{A} = A + I$, $A \in R^{N\times N}$ is the adjacency matrix of graph G and I is an identity matrix. $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$ and $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$. $W^{(l)}$ is the weight matrix and $\sigma(\cdot)$ is a non-linear activation function like ReLU. We can find that one graph convolution layer is able to aggregate information from 1-hop neighbors. By stacking k such layers, we can expand the receptive field and gain information from k-hop neighbors.

Hop-aware attention During the repetion of Eq. 1, we can find that the result in every step can only be used to generate the next convolution result. During this process, a large amount of early-stage information will be lost and only the largest sub-graph structure (e.g., k-hop) can be captured by the graph convolution layer. Thus, we intend to make full use of information from each hop of neighbors by designing a hop-aware attention mechaism.

As is shown in Figure 4, assuming that we stack k graph convolution layers and obtain node representations in different neighborhood hops, which can be denoted as $\psi = [\psi_1, \psi_2, ..., \psi_k]$, $\psi \in R^{k\times N\times F}$. N is the number of nodes and F is the dimension of each node representation. $\psi_i^j \in R^F$ denotes the j-th node's representation in the i-th layer. $c \in R^d$ is a neighborhood hop embedding vector which is randomly initialized and learned in the training process. We impose a transformation matrix $W_\psi \in R^{F\times d}$ to transform the

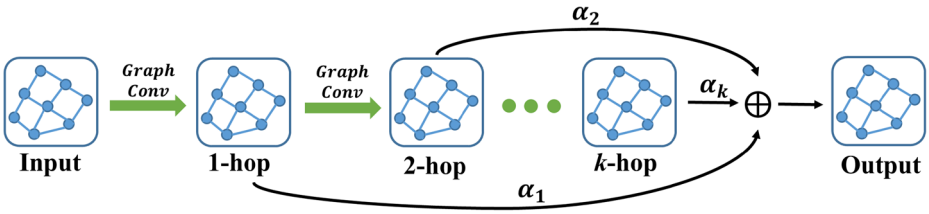

Figure 4 Hop-attention based graph convolution

dimension of ψ_i^j: $F \rightarrow d$. Then, the attention coefficients can be computed as follows:

$$s_i^j = a(W_\psi \psi_i^j, c) \tag{2}$$

$$\gamma_i^j = softmax_i(s_i^j) = \frac{exp(s_i^j)}{\sum_{i=1}^{k} exp(s_i^j)} \tag{3}$$

After we obtain the attention coefficients, for j-th node, a linear combination of node representations from each layer will be applied as follows:

$$U_j = \sum_{i=1}^{k} \gamma_i^j \psi_i^j \tag{4}$$

Neighbor-aware attention Traffic conditions change over time, so do the spatial dependencies among different regions. For instance, the traffic flows tend to transfer from residential areas to working areas in the morning while the opposite in the evening. However, the adjacency matrix used in graph convolution assigns fixed weights to different neighbours , which can not respond to temporal dynamics. Therefore, we design a neighbor-aware attention mechanism to adaptively adjust the spatial correlations between regions.

Supposing that the l-th graph convolution layer transforms input F features into high-level F' features. Here, we take node i and node j as an example, $h_i \in R^F$ and $h_j \in R^F$ are corresponding feature vectors, then the operations in the attention layer can be written as follows:

$$e_{i,j} = a(Wh_i, Wh_j) \tag{5}$$

$$\alpha_{i,j} = softmax_j(e_{i,j}) = \frac{exp(e_{i,j})}{\sum_{j=1}^{N} exp(e_{i,j})} \tag{6}$$

where $W \in R^{F' \times F}$, a is a single-layer feedforward network which is parameterized by a $2F'$-dimension weight vector. $e_{i,j}$ denotes the correlation strength between region i and region j. To make coefficients easily comparable across different nodes, we normalize them across all j using the softmax function.

After obtaining the attention matrix, we combine it with initial normalized adjacency matrix to dynamically adjust correlation strength between regions. Specifically, $\alpha \odot \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$ will replace initial $\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$ when performing graph convolution, where $\odot$ denotes the element-wise multiplication.

DGCGRU Following [35], we combine the proposed dual-attention graph cnvolution with RNN structure to capture temporal correlations. Specifically, we adopt Gated Recurrent Units, which is a simply yet powerful variant of RNN. We replace the MLP layer with our dual-attention graph convolution and introduce a new RNN structure called DGCGRU, which is formulated as:

$$z_t = \sigma(f([X_t, H_{t-1}], \Theta_z)) \tag{7}$$

$$r_t = \sigma(f([X_t, H_{t-1}], \Theta_r)) \tag{8}$$

$$\hat{h}_t = tanh(f([X_t, r_t \odot h_{t-1}], \Theta_{\hat{h}})) \tag{9}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t \tag{10}$$

where $f(,)$ denotes dual-attention graph convolution and Θ is the learnable parameter. σ is sigmoid function and $\odot$ denotes element-wise multiplication. In particular, our proposed DGCGRU receives traffic data from recent P time steps as its input and outputs Y_r.

4.2.2 Periodicity learning

Based on our observations, traffic flows of a region usually follow periodical patterns as daily routines repeat during continuous days and consecutive weeks. In this paper, we mainly study two types of periodical patterns: daily periodicity and weekly periodicity. It means the traffic flow at current timestep is correlated with the same timestep of the previous day or the previous week in the absence of special condition. Therefore, we adopt a two-layer fully-connected networks to extract periodicity information. We select traffic flow data from the same timestep of the previous day and previous week as inputs and output Y_P. Specifically, we select d daily periods $X^{(d)} \in R^{d\times N\times C}$ and w weekly periods $X^{(w)} \in R^{w\times N\times C}$. The periodicity learning is formulated as:

$$Y_p = W_{x'}(W_x(concat(X^{(d)}, X^{(w)})) + b_x) + b_{x'} \tag{11}$$

where $W_x, b_x, W_{x'}, b_{x'}$ are all learnable parameters.

4.3 Transition-flow encoder

The region-flow encoder and the transition-flow encoder share the similar network structure and the only difference is that the latter has an extra embedding layer. According to our observations, we find that the transition-flow that really occurs between regions only accounts for a small portion. Hence, the transition-flow of each region becomes highly sparse. Unfortunately, this data sparsity problem poses a challenge for training deep learning models. Inspired by the embedding strategy in natural language processing [39], we design a spatial embedding method to alleviate this problem.

Transition Embedding The embedding operation is aimed to learn a fucntion that maps a high-dimensional vector ($2N$, depending on the nodes of the urban graph) to a relatively low-dimensional vector. Specifically, the embedding process can be written as follows:

$$\varphi_t = \sigma(Z_t W_z + b_z) \tag{12}$$

where $W_z \in R^{2N\times k}$ and $b_m \in R^k$ are learnable parameters, $Z_t \in R^{N\times 2N}$ is the transition-flow matrix at time interval t, $\sigma(\cdot)$ is a non-linear function (e.g., ReLU). In our experiment, we adopt a two-layer fully-connected network to implement embedding operation. After embedding, we obtain a k-dimension feature vector of transition-flow.

4.4 Context modeling

So far, we obtain two outputs respectively from DGCGRU and periodicity learning. Intuitively, the degree of the influence of each module is different under different context factors (e.g., temporal-meta, weekend and holiday). For instance, the traffic congestion is more related to recent traffic flows while daily routines are more likely to follow periodical patterns. Moreover, traffic flows in the holiday are quite different from those in the normal day since holidays trigger more crowd flows, increasing the amount of traveling vehicles on road networks. Hence, we propose a context attention mechaism to study the impact of context factors on traffic flow changes.

Context attention After DGCGRU and periodicity learning, we obtain two tensors $Y_r \in R^{N\times f\times P}$ and $Y_p \in R^{N\times f\times(d+w)}$, where N is the number of nodes, f is the feature dimension, P, d, w are respectively length of recent, daily and weekly time intervals. Then,

we concatenate Y_r and Y_p on the third dimension and obtain $Y \in R^{N \times f \times (P+d+w)}$. We denote $Y_i^t \in R^f$ as the node i's representation at time interval t. To achieve a contextualized fusion of different time intervals, we first apply a shared linear transformation $W_a \in R^{f \times f'}$ to every node at each time step. Then, the attention coefficients of each time interval are measured by computing the similarity between $Y_i^t W_a$ and u, where $u \in R^{f'}$ is a one-hot encoding context vector. Finally, the softmax operator is applied to normalize the coefficients. Specifically, the context attention is formulated as follows:

$$e_i^t = (Y_i^t W_a)u \tag{13}$$

$$a_i^t = softmax_t(e_i^t) = \frac{exp(e_i^t)}{\sum_{t=1}^{P+d+w} exp(e_i^t)} \tag{14}$$

After we gain the normalized attention coefficients, we calculate a linear combination of representations at each time interval for every node as:

$$\lambda_i = \sum_{t=1}^{P+d=w} a_i^t Y_i^t \tag{15}$$

Then, we concatenate the obtained node representation and get X_λ and Z_λ respectively for region-flow prediction and transition-flow prediction.

4.5 Task-specific decoder

After region-flow encoder and transition-flow encoder, we obtain two tensors $X_\lambda \in R^{N \times f}$ and $Z_\lambda \in R^{N \times f}$, which respectively denote the representations learned from region-flow modeling and transition-flow modeling. Specifically, we intend to aggregate information from each task and make task-specific predictions. First, we introduce several aggregation strategies:

Sum Aggregation The sum aggregation mechanism directly sums up X_λ and Z_λ in an element-wise manner, the output is as follows:

$$\tilde{\lambda} = X_\lambda + Z_\lambda \tag{16}$$

Concat Aggregation The concatenation aggregation has no constraint that two inputs must have the same shape. To be specific, it concatenates two inputs in the feature dimension. The process can be written as:

$$\tilde{\lambda}[r, : f] = X_\lambda[r, : f] \quad r = 0, 1, ..., N-1 \tag{17}$$

$$\tilde{\lambda}[r, f : 2f] = Z_\lambda[r, : f] \quad r = 0, 1, ..., N-1 \tag{18}$$

It is obvious that the concatenation aggregation extends the dimension of above feature vectors from original f to $2f$.

Gating Aggregation We also design a gating aggregation method which adaptively controls the influence of inputs. It can be written as follows:

$$\mu = \sigma(X_\lambda W_x + Z_\lambda W_z + b_\lambda) \tag{19}$$

$$\tilde{\lambda} = \mu \odot X_\lambda + (1 - \mu) \odot Z_\lambda \tag{20}$$

where W_x, W_z and b_λ are learnable parameters, $\odot$ represents the element-wise multiplication, $\sigma(\cdot)$ denotes the sigmoid activation function.

After aggregation, we adopt fully-connected networks to adapt the feature dimension of the shared representation $\tilde{\lambda}$ to each prediction task:

$$\hat{X}_{T+z} = FC(\tilde{\lambda}; \Theta_x) \tag{21}$$

$$\hat{Z}_{T+z} = FC(\tilde{\lambda}; \Theta_z) \tag{22}$$

where Θ_x and Θ_z are learnable parameters in fully-connected networks.

4.6 Multi-task Loss

In this paper, we aim to predict region-flow and transition-flow of each region at the same time. Let $\hat{X}_{T+z}$ and $\hat{Z}_{T+z}$ be the predicted z results of region-flow and transition-flow, X_{T+z} and Z_{T+z} are corresponding true values. We use mean squared error as our loss function, which is defined as follows:

$$L(\Theta) = \lambda_{region} \| X_{T+z} - \hat{X}_{T+z} \|_2^2 + \lambda_{trans} \| Z_{T+z} - \hat{Z}_{T+z} \|_2^2 \tag{23}$$

where Θ are all learnable parameters in our MTGCN, λ_{region} and λ_{trans} are weight coefficients of two tasks.

5 Experiments

In this section, we mainly conduct experiments based on taxi trajectories from Beijing and Chengdu to evaluate the effectiveness of MRGCN.

5.1 Datasets

We use two real-world datasets from Beijing and Chengdu[1] to evaluate our model. The details are as follows:

- **Beijing:** The trajectory data was collected from 1st June to 31st July in 2016. There are about 2 million trajectories covering the road network every day. In our experiment, we mainly extracted the area within the fourth ring road of Beijing and got 256 regions. About 3 important holidays and 16 weekends exist in the dataset. 80% of the data are used to train our model and the remaining 20% are the test set.
- **Chengdu:** It is the trajectory data collected from 1st Nov to 30th Nov in 2016, which is published by Didi. There are more than 20 thousand trajectoris obtained every day and we got 64 regions within the central area of whole city. About 8 weekends exist in the dataset. We use the last 6 days as test set and the others are used to train our model.

[1] https://gaia.didichuxing.com.

5.2 Pre-processing and settings

Specifically, the traffic data from two real-world datasets are aggregated into every 10-minute interval from raw data. As for region partition and graph construction, we adopt the same procedure described in [31]. Besides, for context factors, we employ one-hot encoding to transform temporal-meta (i.e., the day of the week, the time of the day), holidays and weekends information into binary vectors. In addition, Z-score normalization is applied to input data.

We implement our MRGCN model based on PyTorch framework. We train our network with the following hyper-parameter settings: mini-batch size is 64 and the initial learning rate is 0.01 with a decay rate of 0.9 after every 15 epochs. The number of DGCGRU layer is set as 2 with 64 hidden units. The maximum hop k of the dual-attention graph convolution is set to 3. We use the previous 12 intervals and corresponding intervals of previous 3 days and 2 weeks to predict future 6 time steps. The multitask coefficients are set as one of {0.01, 0.1, 1} and the optimal combination is obtained by grid search which is $\lambda_{region} = \lambda_{trans} = 1$. Adam optimizer is adopted to optimize the parameters in our network and the number of epochs is set as 150.

5.3 Baselines

We compare our model with the following baseline approaches:

- **HA:** Historical Average, which uses the average values of previous timesteps to predict the future values. We simply use the average values of previous 12 timesteps to predict future traffic flows.
- **ARIMA:** Auto-Regressive Integrated Moving Average model, which is implemented by *statsmodel* python package and the orders are set as (3,0,1).
- **SVR** [19]**:** Support Vector Regression is a great regression method of powerful generalization ability. We use previous 6 timesteps as input.
- **LSTM** [40]**:** As a variant of RNN, LSTM can effectively capture long-range temporal dependencies among traffic flow data. We use previous 12 timesteps as inputs. The number of hidden units is set as 64 and the learning rate is set as one of {0.1,0.01,0.001}.
- **STGCN** [14]**:** Spatial-Temporal Graph Convolutional Networks, which combines graph convolution with 1D convolution for spatial-temporal modeling. We set the filter number of CNN and GCN as 64 and the size of temporal kernel is set as 2. The learning rate is 0.0002.
- **DCRNN** [35]**:** Diffusion Convolutional Recurrent Neural Network, which models the traffic flow as a diffusion process and predicts traffic flows in a sequence-to-sequence framework. We set the hidden units as 64, the maximum steps of random walks as 3 and the learning rate as 0.0001.
- **MVGCN** [31]**:** Multi-View Graph Convolutional Networks. We design a simplified version of MVGCN which is composed of 3 graph convolution layers. The inputs are previous 12 timesteps and learning rate is 0.0001.
- **ASTGCN** [13]**:** A Spatio-Temporal Graph Convolutional Network which incorporates both spatial and temporal attention mechanism into traffic flow prediction. We design the ASTGCN model which has 2 ST blocks. The number of terms of Chebyshev polynomial K is 3 and we use previous 12 timesteps as inputs. The learning rate is set as 0.0001.

- **MTGCN** [15]**:** A model that integrates transition-flow multitask learning and graph convolutional networks to support traffic flow prediction on irregular regions. We select previous 12 timesteps as inputs. The number of graph convolution layer is set to be 3 and the learning rate is 0.0001.

5.4 Evaluation metrics

In this paper, we measure the performance of different approaches by Root Mean-Squared Error (RMSE) and Mean Absolute Error (MAE). As most existing methods only predict region-flow, we merely compare the performance of region-flow prediction between different approaches. RMSE and MAE are computed as follows:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i}(x_i - \hat{x}_i)^2} \tag{24}$$

$$MAE = \frac{1}{M}\sum_{i}\left|x_i - \hat{x}_i\right| \tag{25}$$

where M is the total number of all predicted values, x_i is the true value and $\hat{x}_i$ is the predicted value.

5.5 Performance comparison

Table 1 gives a comprehensive comparison between MRGCN and several baseline models. The result differences between two datasets are owing to different total traffic flow volumes. Specifically, the average values of traffic flow volume are 270.58 (Beijing) and 23.88 (Chengdu) respectively. Overall, we can observe that our MRGCN achieves the best performance on two datasets. Meanwhile, we can find that the performance of HA method is not bad on two datasets, which proves that periodical patterns in traffic flows really exist. Traditional time series methods such as SVR and ARIMA are not greatly better than HA, this is because the above two methods assume that the change of traffic flow is stable and linear, which is totally different in real-world conditions. The well-known recurrent model LSTM is able to effectively capture long-range temporal dependencies, but it ignores the spatial correlation between regions, thus showing worse performance compared with GCN based models (Table 1).

Among these GCN based models, the simplified MVGCN and STGCN both take spatio-temporal dependencies into account, yet ignoring the dynamics in spatial correlation and the transition-flow which is correlated with region-flow, thus performing worst among them. ASTGCN proposes a spatio-temporal attention mechanism to model dynamic spatial dependencies, yet neglecting the correlated task of transition-flow prediction and ignoring the context information. DCRNN shows relatively outstanding performance on two datasets since it replaces original graph convolution with diffusion convolution, which is more powerful for modeling spatial dependencies in graph-based traffic data. Compared with our previous model MTGCN, our newly proposed MRGCN polishes the multitask component by adding context modeling and dual-attention graph convolution, showing better performance than MTGCN.

Table 1 Performance comparison between different methods

Dataset	*Beijing*		*Chengdu*	
Metrics	RMSE	MAE	RMSE	MAE
HA	38.83	27.19	14.96	9.47
SVR	35.70	25.85	12.17	7.66
ARIMA	33.20	23.15	10.12	7.97
LSTM	31.58	21.27	10.98	8.73
STGCN	28.76	19.33	5.41	3.72
DCRNN	25.11	16.54	5.03	3.47
MVGCN	29.37	19.79	5.68	3.91
ASTGCN	28.51	19.06	5.34	3.63
MTGCN	25.38	16.91	4.95	3.35
MRGCN (ours)	**24.17**	**15.82**	**4.59**	**3.08**

5.6 Influence of dual-attention graph convolution

Specifically, our dual-attention graph convolution consists of two types of attention mechanisms: neighbor-aware attention and hop-aware attention. To validate the effectiveness of proposed dual-attention mechanism, we design another three GCN models which are: MRGCN (V) which removes both neighbor-aware attention and hop-attention, MRGCN (N) which removes only hop-aware attention and MRGCN (H) which removes the neighbor-aware attention. We retrain these models and the results are shown in Table 2.

Generally, we can observe that the model removing the dual-attention component performs worst among the above four models. This proves that our proposed dual-attention meachnism is effective to capture spatial dependencies and thus enhance the generalization of our proposed model. Furthremore, we find MRGCN (N) performs better than MRGCN (H) in Chengdu dataset but worse in Beijing dataset. This is mainly due to the difference of dataset size and the traffic evolution diversity in different cities. Thus, we can conclude that the degree of improvement that neighbor-aware attention and hop-aware attention can bring are quite depending on the specific dataset.

5.7 Impact of context information

To evaluate the impact of context information, we design a variant of MRGCN called MRGCN (PM), which replaces the context modeling with parametric-matrix based fusion. The comparative results are shown in Table 3.

Overall, the MRGCN with context modeling outperforms MRGCN (PM) on two datasets. The PM fusin method only assigns weights to different branches of traffic flows by randomly initialized matrices, yet ignoring the effect which context information imposes on

Table 2 RMSE comparison between different GCN models

Dataset	MRGCN (V)	MRGCN (N)	MRGCN (H)	MRGCN
Beijing	28.63	27.62	26.89	**24.17**
Chengdu	5.47	5.28	5.34	**4.59**

Table 3 MAE comparison of context modeling component

Dataset	MRGCN (PM)	MRGCN
Beijing	16.73	**15.82**
Chengdu	3.21	**3.08**

traffic flows, thus showing worse performance than MRGCN. Specifically, we consider various context factors in our proposed multitask framework, inluding time of the day, day of the week, weekend and holiday information. In our experiments, Beijing dataset has 3 important holidays and 16 weekends, while Chengdu dataset only has 8 weekends and has no holidays. As a result, the improvement that context modeling component brings in Beijing dataset is more obvious than that in Chengdu, which is up to 5.44% in Beijing dataset (4.05% in Chengdu) in terms of MAE metric.

5.8 Evaluation on multi-task learning

In this subsection, we mainly evaluate the influence of multi-task learning on our proposed model. Our MRGCN simultaneously predicts region-flow and transition-flow in a multi-task manner. Here, we remove the submodel of transition-flow prediction and design a new single-task model called RGCN. The comparative results of these two models are shown in Table 4.

As is shown in Table 4, our multitask model MRGCN outperforms the single-task model RGCN on two datasets by a large margin. This indicates that the prediction of transition-flow can really help improve the accuracy of region-flow prediction. Moreover, we find that the improvement of multitask is more significant on Chengdu dataset than that in Beijing dataset, which is about 10.17% (7.85% in Beijing) in terms of RMSE metric. This is mainly because of the difference on region size and region amount of two datasets. Since the transition-flow depicts the pair-wise transition correlation among regions, a higher region size and region amount tend to cause a relatively large combination of inter-region transition, even the major of them may not occur in the real-world scenario. As a result, the transition-flow matrix becomes much sparse and makes it more difficult to train and optimize networks.

5.9 Effect of aggregation strategy

In this work, we propose three methods for aggregating knowledge learned from each task: sum aggregation, concat aggregation and gating aggregation. The comparative results are shown below in Table 5. Clearly, we can find that sum aggregation achieves the best performance on two datasets. Besides, gating aggregation performs worst among these three

Table 4 Comparison between single-task and multi-task

Dataset	*Beijing*		*Chengdu*	
Metrics	RMSE	MAE	RMSE	MAE
RGCN	26.23	17.61	5.11	3.56
MRGCN	**24.17**	**15.82**	**4.59**	**3.08**

Table 5 Comparison between different aggregation methods

Dataset	*Beijing*		*Chengdu*	
Metrics	RMSE	MAE	RMSE	MAE
MRGCN (concat)	24.95	16.52	4.87	3.39
MRGCN (gating)	26.29	17.73	5.28	3.61
MRGCN (sum)	**24.17**	**15.82**	**4.59**	**3.08**

models. This is mainly caused by extra parameters and sigmoid function in gating strategy, which makes the training of networks more challenging and increases the potential of overfitting. Moreover, we find that the performance of concat aggregation and sum aggregation are really close. Since these two methods directly operate the learned representaions from each task and avoids extra parameters, these two methods can be seen as a kind of knowledge sharing.

5.10 Results on multi-step prediction

In this subsection, we compare our proposed MRGCN with other GCN-based methods on multi-step prediction. Particularly, the initial time interval is set as 10-minute, we aim to predict traffic flows in the future 40 minutes.

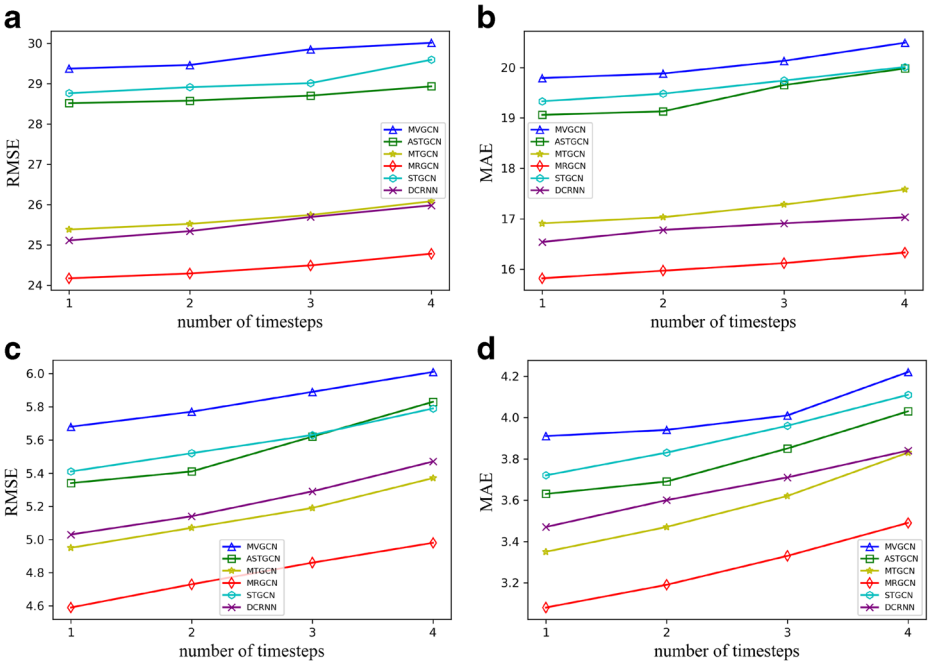

Figure 5 Comparison of multi-step prediction

As is shown in Figure 5, with the increase of predicting timesteps, the performance of all models become worse. Obviously, this is mainly caused by the decrease of correlation between predicting time intervals and current moment. Nevertheless, our proposed MRGCN always achieves the best performance compared with other methods on Beijing and Chengdu datasets and shows highly stability regardless of increase of the predicting timesteps. In addition, since the data of Beijing dataset are more sufficient, the hidden spatial-temporal dependencies can be more fully explored with the help of large amount of learnable parameters in these models. Thus, most of the GCN-based methods show better stability compared with that in Chengdu dataset.

6 Conclusion

In this work, we study the problem of traffic flow prediction in a citywide range, which is sigficantly beneficial for the construction of smart cities. Specifically, we propose a multitask framework called MRGCN which can simultaneously predict region-flow and transition-flow for each region. Overall, it consists of four components: a region-flow encoder and a transition-flow encoder, which adopt DGCGRU and fully-connected networks for spatial-temporal feature learning; a context modeling module which considers context factors for the contextualized fusion of previous traffic data; a task-specific decoder for predicting traffic flows. We evaluate our model on two real-world datasets and the results demonstrate the effectiveness of our proposed method.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant Nos. 61872258, 61772356, 61876117, and 61802273, the Australian Research Council discovery projects under Grant Nos. DP170104747, DP180100212, A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant No. KYCX20_2714.

References

1. Zhang, J., Wang, F., Wang, K., Lin, W., Xu, X., Chen, C.: Data-driven intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst. **12**(4), 1624–1639 (2011)
2. Dai, J., Liu, C., Xu, J., Ding, Z.: On personalized and sequenced route planning. World Wide Web **19**(4), 679–705 (2016)
3. Xu, J., Chen, J., Zhou, R., Fang, J., Liu, C.: On workflow aware location-based service composition for personal trip planning. Fut. Gener. Comp. Syst. **98**, 274–285 (2019)
4. Bai, L., Yao, L., Kanhere, S.S., Wang, X., Sheng, Q.Z.: Stg2seq: Spatial-Temporal Graph to Sequence Model for Multi-Step Passenger Demand Forecasting. In: IJCAI, pp. 1981–1987 (2019)
5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
6. Zhang, J., Zheng, Y., Qi, D.: Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In: AAAI, pp. 1655–1661 (2017)
7. Peng, S., Shen, Y., Zhu, Y., Chen, Y.: A Frequency-Aware Spatio-Temporal Network for Traffic Flow Prediction. In: DASFAA, pp. 697–712 (2019)
8. Lin, Z., Feng, J., Lu, Z., Li, Y., Jin, D.: Deepstn+: Context-Aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. In: AAAI, pp. 1020–1027 (2019)
9. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y.: Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
10. Yuan, J., Zheng, Y., Xie, X.: Discovering Regions of Different Functions in a City Using Human Mobility and Pois. In: SIGKDD, pp. 186–194 (2012)

11. Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X.: LC-RNN: A Deep Learning Model for Traffic Speed Prediction. In: IJCAI, pp. 3470–3476 (2018)
12. Zhao, J., Xu, J., Zhou, R., Zhao, P., Liu, C., Zhu, F.: On Prediction of User Destination by Sub-Trajectory Understanding: A Deep Learning Based Approach. In: CIKM, pp. 1413–1422 (2018)
13. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In: AAAI, pp. 922–929 (2019)
14. Yu, B., Yin, H., Zhu, Z.: Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In: IJCAI, pp. 3634–3640 (2018)
15. Wang, F., Xu, J., Liu, C., Zhou, R., Zhao, P.: MTGCN: A Multitask Deep Learning Model for Traffic Flow Prediction. In: DASFAA, vol. 12112, pp. 435–451 (2020)
16. Pan, B., Demiryurek, U., Shahabi, C.: Utilizing Real-World Transportation Data for Accurate Traffic Prediction. In: ICDM. IEEE Computer Society, pp. 595–604 (2002)
17. Ristanoski, G., Liu, W., Bailey, J.: Time Series Forecasting Using Distribution Enhanced Linear Regression. In: PAKDD, pp. 484–495 (2013)
18. Chandra, S.R., Al-Deek, H.: Predictions of freeway traffic speeds and volumes using vector autoregressive models. J. Intell Transp. Syst. **13**(2), 53–72 (2009)
19. Wu, C., Ho, J., Lee, D.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
20. Cui, Z., Ke, R., Wang, Y.: Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. CoRR, arXiv:1801.02143 (2018)
21. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-Based Prediction Model for Spatio-Temporal Data. In: SIGSPATIAL. ACM, pp. 92:1–92:4 (2016)
22. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial-temporal 3d convolutional neural networks for traffic data forecasting. IEEE, Trans. Intell. Transp. Syst. **20**(10), 3913–3926 (2019)
23. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. In: AAAI, pp. 5668–5675 (2019)
24. Feng, J., Lin, Z., Xia, T., Sun, F., Guo, D., Li, Y.: A Sequential Convolution Network for Population Flow Prediction with Explicitly Correlation Modelling. In: IJCAI, pp. 1331–1337 (2020)
25. Zhang, J., Zheng, Y., Sun, J., Qi, D.: Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Trans. Knowl. Data Eng. **32**(3), 468–478 (2020)
26. Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In: NIPS, pp. 802–810 (2015)
27. Zonoozi, A., Kim, J., Li, X., Cong, G.: Periodic-Crn: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns. In: IJCAI, pp. 3732–3738 (2018)
28. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: ICLR (2014)
29. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)
30. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In: NIPS, pp. 3837–3845 (2016)
31. Sun, J., Zhang, J., Li, Q., Yi, X., Liang, Y., Zheng, Y.: Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. IEEE Transactions on Knowledge and Data Engineering (2020)
32. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In: AAAI, pp. 914–921 (2020)
33. Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., Feng, X.: Multi-Range Attentive Bicomponent Graph Convolutional Network for Traffic Forecasting. In: AAAI, pp. 3529–3536 (2020)
34. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph Wavenet for Deep Spatial-Temporal Graph Modeling. In: IJCAI, pp. 1907–1913 (2019)
35. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In: ICLR (2018)
36. Chen, C., Li, K., Teo, S.G., Zou, X., Wang, K., Wang, J., Zeng, Z.: Gated Residual Recurrent Graph Neural Networks for Traffic Prediction. In: AAAI, pp. 485–492 (2019)
37. Zhang, Y., Wei, Y., Yang, Q.: Learning to multitask. In: NeurIPS, pp. 5776–5787 (2018)
38. Zhao, L., Wang, J., Guo, X.: Distant-Supervision of Heterogeneous Multitask Learning for Social Event Forecasting with Multilingual Indicators. In: AAAI, pp. 4498–4505 (2018)

39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. In: NIPS, pp. 3111–3119 (2013)
40. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C: Emerg. Technol. **54**, 187–197 (2015)