

Transportmetrica B: Transport Dynamics

ISSN: (Print) (Online) Journal homepage: <https://www.tandfonline.com/loi/ttrb20>

Forecasting citywide short-term turning traffic flow at intersections using an attention-based spatiotemporal deep learning model

Tao Jia & Chenxi Cai

To cite this article: Tao Jia & Chenxi Cai (2023) Forecasting citywide short-term turning traffic flow at intersections using an attention-based spatiotemporal deep learning model, *Transportmetrica B: Transport Dynamics*, 11:1, 683-705, DOI: [10.1080/21680566.2022.2116125](https://doi.org/10.1080/21680566.2022.2116125)

To link to this article: <https://doi.org/10.1080/21680566.2022.2116125>



Published online: 30 Aug 2022.



Submit your article to this journal 



Article views: 215



View related articles 



View Crossmark data 



Citing articles: 4 View citing articles 



Forecasting citywide short-term turning traffic flow at intersections using an attention-based spatiotemporal deep learning model

Tao Jia and Chenxi Cai

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, People's Republic of China

ABSTRACT

Prediction of short-term traffic flow has been examined recently, but little attention has been paid to the prediction of citywide turning traffic flow at intersections. Based on an in-depth analysis of turning traffic flow patterns, we propose a novel attention-based spatiotemporal deep learning model to predict citywide short-term turning traffic flow at road intersections with high accuracy. First, we examine the spatiotemporal patterns of turning traffic flow. Then, an end-to-end deep learning structure with four components is designed to model turning traffic flow. In our model, graph convolutional network is revised to learn spatial dependencies and sparseness, and gate recurrent unit network with an attention mechanism is developed to learn temporal dependencies and fluctuations. Experiments were conducted in Wuhan, China, where taxicab trajectory data were used to train and validate our model. The results suggest that our model outperforms current state-of-the-art models with higher accuracy on estimating turning traffic flow.

ARTICLE HISTORY

Received 20 March 2022

Accepted 18 August 2022

KEYWORDS

Deep learning;
attention-based
spatiotemporal neural
networks; turning traffic
flow; trajectory data

1. Introduction

It has been a long-standing topic to predict traffic flow in transportation studies, but very few studies have focused on estimating citywide turning traffic flow at intersections (Shoup et al. 2013; Ghanim and Shaaban 2019; Mahmoud et al. 2021). Turning traffic flow prediction intends to estimate vehicle turning movement counts at road intersections (Chen et al. 2012; Miglani and Kumar 2019; Ghanim and Shaaban 2019; Mahmoud et al. 2021). It is indispensable for intelligent transportation systems (ITS), which needs frequently updated and reliable information in future time and is of fundamental importance for transportation management and planning, signal optimisation, and even traffic safety (Smith, Williams, and Oswald 2002; Shoup et al. 2013; Abadi, Rajabioun, and Ioannou 2015; Ghanim and Shaaban 2019; Jia and Yan 2021; Mahmoud et al. 2021). At a road intersection, prediction of turning traffic flow can help in obtaining accurate vehicle turning movements in a future time, and these data can reflect the complicated traffic status from one road segment to another. This could be used to facilitate geometric design at intersection and optimise traffic signals in traffic engineering applications, so as to improve signal and intersections' efficiency (Chen et al. 2012; Ghanim and Shaaban 2019; Mahmoud et al. 2021), alleviate traffic congestion (Chen and Cheng 2010), reduce pollutant emissions (Jia et al. 2018; Jia et al. 2022), and manage urban traffic efficiently (Zhang et al. 2011). Additionally,

CONTACT Tao Jia  tao.jia@whu.edu.cn  School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, People's Republic of China

prediction of turning traffic flow can be valuable for the general public, allowing them to make travel plans smartly to save time and avoid traffic congestion (Zhu et al. 2016). Thus, it is practical significance to predict citywide short-term turning traffic flow at intersections.

Conventionally, traffic data can be collected manually by investigators at toll stations or automatically by fixed instruments like in-ground loop detectors or traffic surveillance camera systems. However, two limitations may be resulted from the conventional way. First, traffic data cannot be acquired in a large area, because it is expensive to deploy the instruments to cover the entire area; second, traffic flow is mainly collected along a road or in a fixed area, and it is difficult to obtain turning movements. Recent advancements in the technologies of communication and positioning have allowed the availability of massive vehicle trajectory data (Chen et al. 2019; Li et al. 2020; Jia and Yan 2012), which can be easily used to extract turning traffic flow at road intersections and thereby boosts a renew interest to develop data-driven approaches for traffic prediction at freeways and intersections (Zhang et al. 2013; Lv et al. 2014; Kumar & Vanajakshi, 2015; Polson and Sokolov 2017; Zhao et al. 2019; Zhang, Cheng, and Ren 2019; Li et al. 2020; Mahmoud et al. 2021).

Basically, two different types of machine learning methods are used to predict traffic flow, including parametric methods (Williams and Hoel 2003; Min et al. 2009; Ghosh, Basu, and Mahony 2009; Guo, Huang, and Williams 2014; Li et al. 2017; Zhan et al. 2018; Cai et al. 2019) and nonparametric approaches (Wang and Shi 2013; Ma, Zhou, and Abdulhai 2015; Cai et al. 2016; Xia et al. 2016; Guo, Polak, and Krishnan 2018; Zhu et al. 2018). The parametric methods tend to make a priori about the variables and construct the model explicitly using the parameters, but they cannot capture the nonlinear property of traffic flow. One of the widely used parametric models is the time series-based methods, which includes autoregressive integrated moving average (ARIMA) and its variants, such as seasonal ARIMA (Williams and Hoel 2003), dynamical space-time ARIMA (Min et al. 2009), and structural time series model (Ghosh, Basu, and Mahony 2009). Another popular parametric method is Kalman filter model and its variants, such as noise-immune Kalman filter (Cai et al. 2019) and adaptive Kalman filter (Guo, Huang, and Williams 2014). The nonparametric approaches do not require strong assumptions on model structure and learn it automatically with data. There are many nonparametric approaches for traffic flow forecasting, such as support vector regression (Wang and Shi 2013), artificial neural networks (Ma, Zhou, and Abdulhai 2015), k-nearest neighbour (KNN) method (Cai et al. 2016) and its variants (Xia et al. 2016; Guo, Polak, and Krishnan 2018). It is reported that KNN techniques are much more efficient and scalable than parametric models (Yoon and Chang 2014; Xia et al. 2016; Guo, Polak, and Krishnan 2018). However, these nonparametric methods cannot reveal the hidden spatiotemporal dependencies of traffic flow owing to their shallow structures (Lv et al. 2014).

Recently, many types of deep learning models have been proposed (LeCun, Bengio, and Hinton 2015; Zhang et al., 2021a) and applied to predict traffic flow (Lv et al. 2014; Jiang and Luo 2021). Convolutional neural networks (CNNs) were proposed to capture spatial dependencies (Gu et al. 2018; Yao, Zhang, and Long 2022), while recurrent neural networks (RNNs) were used to learn temporal dependencies (Fu, Zhang, and Li 2016; Yang et al. 2019). Specifically, a combination of CNNs and RNNs was adopted to understand spatiotemporal dependencies of traffic flow (Wu et al., 2018a; Li and Ban 2019; Zheng et al. 2019; Peng et al. 2020; Peng et al. 2021), which were found to have better results than the machine learning methods. For instance, Li and Ban (2019) predicted short-term lane-based traffic flow at intersections, and Zheng et al. (2019) developed a deep and embedding learning approach to estimate high accurate traffic flow in freeways. These studies were mostly focused on very few roads or intersections, while citywide traffic flow forecasting remains challenging. There are three strands of studies to tackle this problem. First, hybrid networks were adopted by combining CNNs and RNNs for grid-based traffic prediction, where the area of interest is divided into grids and

prediction is made at the grid level (Polson and Sokolov 2017; Zhao et al. 2017; Zhang et al. 2018; Du et al. 2019). These studies can determine spatial relationships between adjacent grids, but they cannot model complicated topological relationship of roads. Second, hybrid networks were used for traffic prediction at the road level, where the road network is mapped onto an image with each pixel corresponding to a single road (Jia and Yan 2021). These studies overcome the deficiency of prediction at the grid level and attempt to preserve the topological relationship to a large extent. Third, hybrid networks of graph convolutional networks (GCNs) and RNNs were adopted for traffic prediction at the road level, where the topological relationships can be fully captured by GCNs in a non-Euclidean space (Wu et al., 2018b; Zhao et al. 2019; Zhang, Cheng, and Ren 2019; Zhang et al., 2021b).

However, it remains a question that needs to be studied urgently to predict citywide short-term turning traffic flow at intersections with high accuracy. First, most previous studies were focused on traffic prediction at the road or grid level, and only very few studies have examined the turning traffic flow forecasting at intersections. For instance, the Path Flow Estimator was used to predict vehicle turning movement counts at intersections of major roads (Chen et al. 2012); Ghanim and Shaaban (2019) developed an artificial neural network to estimate vehicle turning movement counts using approach volumes under peak hours; Li et al. (2020) adopted the partial least square model to predict 15-minute vehicle turning movements using trajectory data at very few intersections; Mahmoud et al. (2021) proposed a machine learning model to forecast the cycle-level through and left-turn movements at 16 intersections. Nonetheless, a citywide turning traffic flow forecasting at intersections is rarely reported in the literature, due to the following challenges: (1) The number of road turns is much larger than the number of road segments, and it will undoubtedly bring difficulties to the training of the model, owing to the large increase in the number of model parameters; (2) The spatial distribution of turning traffic flow is much sparser than that of road segment traffic flow. Although GCNs can be used to handle this issue, it has a limited effect on the sparseness of turning traffic flow. Second, conventional deep learning models have rarely considered the topology of directed road networks (Zhang, Cheng, and Ren 2019), which can improve the accuracy of turning traffic flow prediction. Third, temporal dependencies of traffic flow are not fully leveraged. For instance, it is well known that current traffic flow can be affected by those in the previous times (Jia and Yan 2021), but the influences may fluctuate across time.

Therefore, to answer this question, this study aims to contribute an attention-based spatiotemporal deep learning model to predict citywide turning traffic flow at intersections in a short-term time interval of 10 minutes with high accuracy. The contributions of this study are as follows. (1) The road network is represented as a directed weighted topological graph, where nodes are roads, edges are connections of roads, and edge weights are turning traffic flow extracted from trajectory data. This representation can fully capture the topological relationship of turning traffic flow at intersections and can improve prediction accuracy. (2) Based on an in-depth analysis of turning traffic flow at intersections, we propose an end-to-end deep learning structure, which are hybrid neural networks composed of four different components accounting for modelling the spatiotemporal dependencies, spatial sparseness, and temporal fluctuations. Specifically, gated recurrent unit networks (GRUs) are revised by integrating an attention mechanism to handle temporal dependencies and fluctuations, while GCNs are revised by adding self-loops and edge weights to capture spatial dependencies and sparseness. (3) Turning traffic flow is predicted at intersections for the entire city with high accuracy in a fine temporal scale of 10 minutes, which is quite challenging and rarely reported in the literature. Experiments were conducted in Wuhan, China, where taxicab trajectory data were used to verify our model. The experimental results suggest that our model achieves higher accuracy than current state-of-the-art models in turning traffic flow prediction tasks, which may have practical implications in ITS.

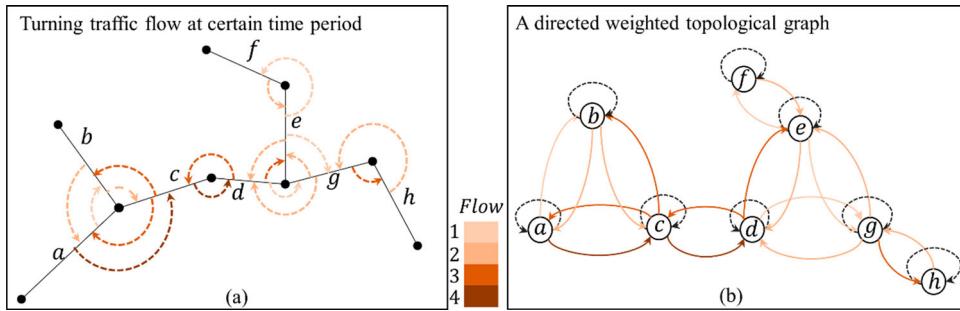


Figure 1. An illustration of (a) turning traffic flow in road network and (b) its representation.

2. Preliminary

2.1. Problem statement

The research problem is to predict turning traffic flow RTX_t during time period t given historical observations of turning traffic flow $RTX_1, RTX_2, RTX_3, \dots, RTX_{t-1}$ and road traffic flow $RX_1, RX_2, RX_3, \dots, RX_{t-1}$. To solve this problem, conceptual definitions are briefly elaborated as follows.

Road traffic flow: it is conventionally measured as vehicle movement counts on the road segment at a certain time period. In this context, road traffic flow at time period t is denoted as $RX_t = \{rx_1^t, rx_2^t, rx_3^t, \dots, rx_{nr}^t\}$, where rx_i^t means the vehicle movement counts on road segment i at time period t and nr is the number of road segment in the road network.

Turning traffic flow: it is defined as vehicle turning movement counts from one road segment to its adjacent neighbours, which can be used to investigate the complicated traffic status at road intersections. In this context, turning traffic flow at time period t is denoted as $RTX_t = \{rtx_1^t, rtx_2^t, rtx_3^t, \dots, rtx_{nrt}^t\}$, where rtx_i^t means the counts of vehicle movement passing through road turn i at time period t and nrt is the number of road turns in the road network.

As shown in Figure 1(a), we display turning traffic flow at a certain time period in a synthetic road network, where it is visualised as arcs with different colours connecting two road segments.

2.2. Representation of road network as a directed weighted topological graph

To fully model the spatial dependencies of turning traffic flow, as shown in Figure 1(b), the road network is represented as a directed weighted topological graph. It is denoted as $G = (V, E, W)$, where V is the set of nodes, E is the set of arcs, and W is the weight adjacency matrix. In this study, graph nodes represent road segments, graph arcs indicate directional connections of two road segments, and weight adjacency matrix stores the value of turning traffic flow associated with each arc. It should be noted that two adjacent road segments are modelled as two arcs with opposite traffic directions, which may inevitably increase the volume and sparseness of traffic flow. Additionally, a self-loop is added to each graph node, because current traffic flow of a road can be affected by its historical traffic flow.

To obtain weight values in adjacency matrix, we calculated turning traffic flow from trajectory trips. A trajectory trip can reflect the mobility of a taxicab in the road network, and it is composed of GPS points in chronological order. First, given a trip, each GPS point is spatially matched to the nearest road segment, and eventually it converts the entire trip into a sequence of road segments. Second, it counts the number of vehicle movement in each road segment at a certain time period, which generates the road traffic flow RX_t . Third, for each trip, it gradually judges whether two consecutive GPS points are matched into the same road segment. If they are not, then it continues to check whether the two road segments are spatially adjacent. If they are adjacent, then it increases the value of turning traffic flow by 1 between the two road segments; otherwise, it calculates the shortest path between the two road

segments, and the value of turning traffic flow between two consecutive road segments along the path is increased by 1 in turn. This can generate the turning traffic flow at a certain time period RTX_t .

3. An attention-based spatiotemporal deep learning model

3.1. Spatiotemporal patterns of turning traffic flow

To reveal the underlying spatiotemporal patterns of turning traffic flow, we examine their temporal dependencies, spatial dependencies, and spatial sparseness, aiming to provide insights for the development of our model. Temporal dependencies of turning traffic flow intend to explore its diverse periodicities and how the current turning traffic flow can be affected by the previous one (Zhang et al. 2018; Zhang, Cheng, and Ren 2019; Jia and Yan 2021). As shown in Figure 2(a), we plot the total turning traffic flow in 10 minutes with time for two weeks, which indicates clearly the daily periodicities. This pattern can be also observed for a single road turn as shown in Figure 2(b). Specifically, periodic temporal patterns can be reported for each day with peak times around 08:00–09:00 in the morning, 12:00–13:00 in the afternoon, and 22:00–23:00 in the evening. As for the peak around 22:00–23:00, it comes to the maximum on Saturday but reaches to the minimum on Sunday, which accurately reflects well the rhythm of human activities. Aggregately, as shown in Figure 2(c), we plot the auto-correlation of the total turning traffic flow in different time lags, which presents a similar daily periodic pattern with a continuous fading trend. Specifically, we can observe strong correlations in short time lags and weak correlations in long time lags. For instance, the total turning traffic flow at 00:00 is still strongly correlated with that at 01:40, but it is weakly correlated with that in one week later. Individually, as shown in Figure 2(d), we plot the correlation of turning traffic flow in the current time period with those in previous time periods, which suggests that it is not strictly decreasing but fluctuates over time. These findings indicate that turning traffic flow can be affected by those in near time, one hour later, or one day later, but the influences may fluctuate over time, which should be considered by our model.

Turning traffic flow is encoded as each arc weight in topological graph, and thus spatial dependencies intend to examine how turning traffic flow is correlated with that spatially adjacent turning traffic flow, because its dynamic relationship can be conventionally modelled as a Markov chain (Zhang, Cheng, and Ren 2019). In this context, we explore the correlation between turning traffic flow and the average of neighbouring turning traffic flow, which resembles assortative mixing in network science. As shown in Figure 3(a), a remarkable linear relationship with R-squared value of 0.86 can be observed, which means that roads with high values of traffic flow tend to diffuse large amounts of traffic flow to adjacent roads. In other words, nodes with high values tend to be connected to arcs with high weight values. As shown in Figure 3(b), we visualise the distribution of observed traffic flow in space, where turning traffic flow is represented as an arc from one road to another with different colours. Spatially, roads with high values of traffic flow are more likely to be clustered with arcs that have high values of turning traffic flow, and they are mostly distributed along the urban ring lines or major roads; roads with low values of traffic flow tend to be clustered with arcs that have low values of turning traffic flow, and they are mainly located within urban blocks. Additionally, we find that turning traffic flow is sparsely distributed in space and varies across time. As shown in Figure 3(c,d), 61.8% of turning traffic flow can be observed at 09:00 in the morning, but the percentage decreases to 37.4% at 04:00 in the early morning. These findings indicate that turning traffic flow is highly correlated with road traffic flow and is sparsely distributed in space, which should be captured and handled by our model.

3.2. Structure of the model

Based on an in-depth analysis of spatiotemporal patterns of turning traffic flow, we propose an attention-based spatiotemporal deep learning model with the ability to learn their spatiotemporal dependencies and spatial sparseness. As shown in Figure 4(a), the structure of our model constitutes four major components, namely recent part, hourly part, daily part, and weekly part. The four

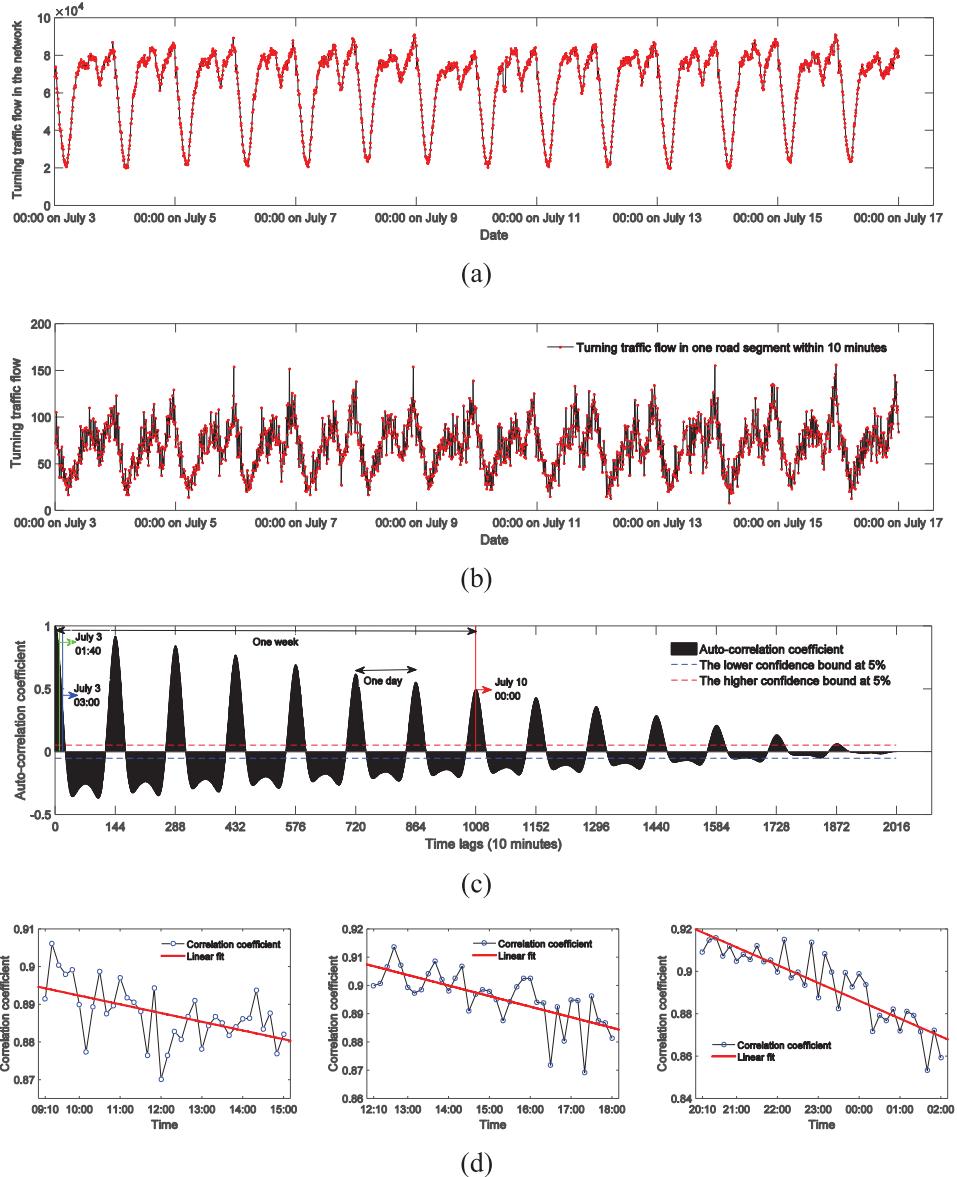


Figure 2. Temporal dependencies of turning traffic flow: (a) change of the total turning traffic flow with time; (b) change of a specified turning traffic flow with time; (c) auto-correlations of the total turning traffic flow in different time lags; and (d) correlations of turning traffic flow in the current time period with those in the previous time periods.

parts have different inputs and similar structures. Recent part receives two input sequences of graphs in a few previous steps, denoted as $\{G_{t-n_r l_r}, \dots, G_{t-2l_r}, G_{t-l_r}\}$ and $\{G_{t-m_r l_r}, \dots, G_{t-2l_r}, G_{t-l_r}\}$, where l_r is set as 10 minutes, n_r and m_r are the lengths of recent long-term and short-term inputs, respectively. Hourly part takes two sequences of graphs with time interval of 1 hour as inputs, denoted as $\{G_{t-n_h l_h}, \dots, G_{t-2l_h}, G_{t-l_h}\}$ and $\{G_{t-m_h l_h}, \dots, G_{t-2l_h}, G_{t-l_h}\}$, where l_h is set as 1 hour, n_h and m_h are the lengths of hourly long-term and short-term inputs, respectively. Daily part receives two input sequences of graphs with time interval of 1 day, denoted as $\{G_{t-n_d l_d}, \dots, G_{t-2l_d}, G_{t-l_d}\}$ and $\{G_{t-m_d l_d}, \dots, G_{t-2l_d}, G_{t-l_d}\}$, where l_d is set as 1 day, n_d and m_d are the lengths of daily long-term and short-term inputs, respectively. Weekly part receives two input sequences of graphs with time interval

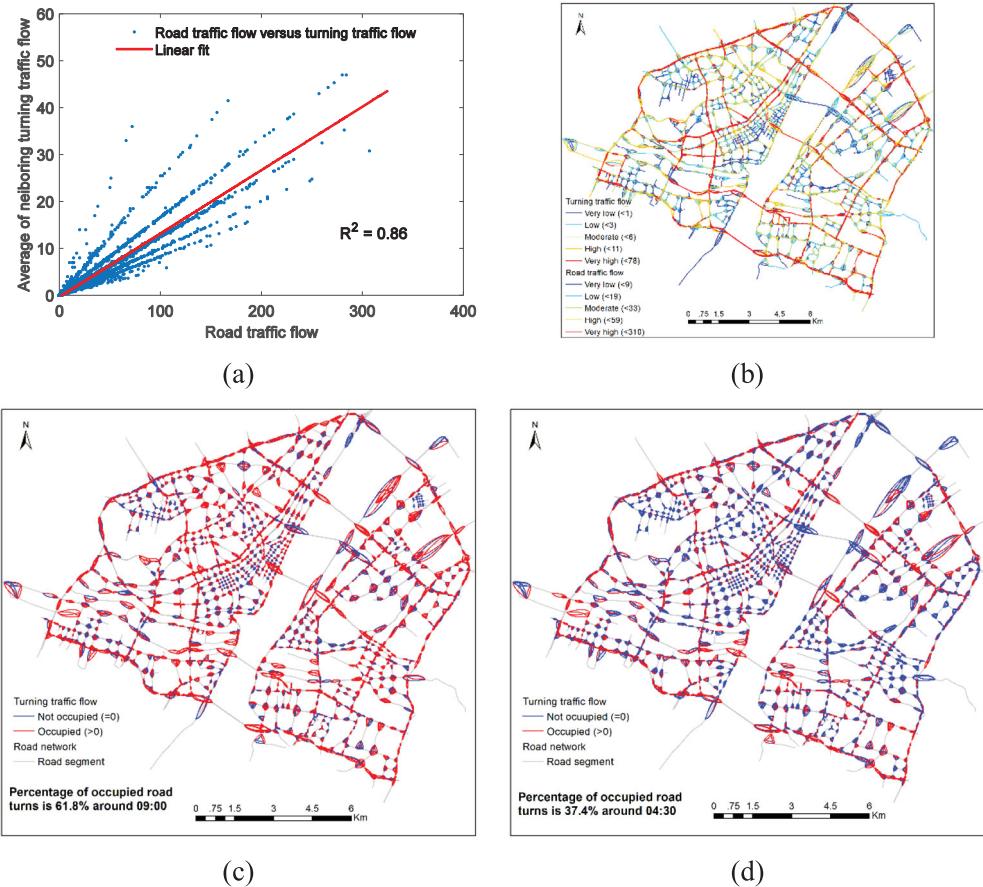


Figure 3. Spatial dependencies of turning traffic flow: (a) a linear relationship between road traffic flow and the average of neighbouring turning traffic flow; (b) spatial visualisation of road traffic flow and turning traffic flow; and spatial sparseness of turning traffic flow around (c) 09:00 and (d) 04:30.

of 1 week, denoted as $\{G_{t-n_w l_w}, \dots, G_{t-2l_w}, G_{t-l_w}\}$ and $\{G_{t-m_w l_w}, \dots, G_{t-2l_w}, G_{t-l_w}\}$, where l_w is set as 1 week, n_w and m_w are the lengths of weekly long-term and short-term inputs, respectively. The four parts adopt the same hybrid networks of GCNs and GRUs with an attention mechanism, aiming to capture the recent, hourly, daily, and weekly spatiotemporal dependencies, and each of them is further fused with the external input in terms of weather and Workday/Weekend. GCNs are revised to extract spatial dependencies of traffic flow by fully considering the topological structure, GRUs are revised by incorporating an attention mechanism to learn complicated temporal dependencies (Bahdanau, Cho, and Bengio 2014), and the external input is used to extract external features using the one-hot encoding technique. The outputs from the four parts are merged together using different weights, containing the deep features of traffic flow in each road, and thereafter, they are used to predict the final turning traffic flow RTX_t .

- (1) *The revised GCNs:* The revised GCNs are presented to learn spatial dependencies of traffic flow at a certain time period. It is formally defined as (1) for each node at time period t , where $x_i^l(t)$ denotes the feature vector of node i for the l th network layer, $f_{ij}(t)$ is the turning traffic flow from road segment i to j , $\varphi(i)$ is the neighbouring set of node i , $W^l(t)$ is the learnable parameter matrix for the l th network layer, $\sigma(\cdot)$ is a non-linear activation function, and $b^l(t)$ means an interception item. Actually, it improves the model from two aspects (Kipf and Welling 2017). First, it enforces

self-loops in the network, which means that feature vector of current node i ($x_i^l(t)$) is simply added. Second, it normalises the adjacency matrix using turning traffic flow, which produces a weighted average traffic flow of neighbouring road segments. This is because turning traffic flow is spatially correlated with road traffic flow, where a large value of turning traffic flow tends to have a high influence on the aggregated feature. Intuitively, from the perspective of traffic engineering, the revised GCNs can model spatial dependencies by aggregating traffic flow of directly connected neighbours according to turning traffic flow and extracting hidden features using convolution. It should be noted that $X^{l+1}(t) = \{x_1^{l+1}(t), x_2^{l+1}(t), \dots, x_{nr}^{l+1}(t)\}$ is the output of GCNs at time period t , which is then fed into GRUs elaborated below.

$$x_i^{l+1}(t) = \sigma \left(b^l(t) + \left(x_i^l(t) + \sum_{j \in \varphi(i)} \frac{f_{ij}(t)}{\sum_{j \in \varphi(i)} f_{ij}(t) + \delta} x_j^l(t) \right) W^l(t) \right). \quad (1)$$

- (2) *The revised GRUs:* GRUs can overcome the common issues of vanishing and exploding gradients and can learn long short-term dependencies through the gate mechanism (Cho et al. 2014; Fu, Zhang, and Li 2016). To effectively capture the fluctuations in temporal dependencies, we revised the GRUs by introducing an attention mechanism (Bahdanau, Cho, and Bengio 2014), which has achieved great success in natural language processing tasks and is used to give high weights to certain graphs in the input sequence. This is implemented in an encoding-decoding process using two GRUs with different lengths (GRUs1 and GRUs2). During the encoding process, GRUs1 extracts all the hidden states of a long-term input sequence, which will be participated in the calculation of the attention weight (importance) of each point in the long-term sequence for a given point in the short-term sequence. During the decoding process, it firstly calculates all the attention weights for the current point in the short-term sequence using (4), and these attention weights are used to derive a context vector by merging the hidden states of the long-term input sequence, capturing the patterns of temporal fluctuations. Thereafter, GRUs2 extracts all the hidden states of the short-term sequence by incorporating the context vector using (3).

As shown in Figure 4(b), the inputs to the revised GRUs are long-term sequence $\{X(t-n), X(t-n+1), \dots, X(t-1)\}$ and short-term sequence $\{X(t-m), X(t-m+1), \dots, X(t-1)\}$ with respect to GRUs1 and GRUs2, and the output h'_{t-1} can be derived using (2–5), where $GRU(\cdot)$ denotes inner operation in the cell, h_0 is a randomly initialised vector, $\{h_{t-n}, h_{t-n+1}, \dots, h_{t-1}\}$ denotes the hidden states of the long-term input sequence, $\{h'_{t-m}, h'_{t-m+1}, \dots, h'_{t-1}\}$ denotes the hidden states of the short-term input sequence, $Con[\cdot]$ means concatenation of elements, α_{ij} is the attention weight of the j^{th} element on the i^{th} element using the softmax function, v and W are matrices of learnable parameters, $\tanh(\cdot)$ denotes the activation function, c_i is the context vector calculated by element-wise multiplication of each attention weight with the corresponding hidden state of the long-term sequence for the current point i in the short-term sequence.

$$\begin{cases} h_{t-n} = GRU(X(t-n), h_0) \\ h_{t-n+1} = GRU(X(t-n+1), h_{t-n}) \\ \dots \\ h_{t-1} = GRU(X(t-1), h_{t-2}) \end{cases}, \quad (2)$$

$$\begin{cases} h'_{t-m} = GRU(h'_0, Con[X(t-m), c_0]) \\ h'_{t-m+1} = GRU(h'_{t-m}, Con[X(t-m+1), c_{t-m}]), \\ \dots \\ h'_{t-1} = GRU(h'_{t-2}, Con[X(t-1), c_{t-2}]) \end{cases}, \quad (3)$$

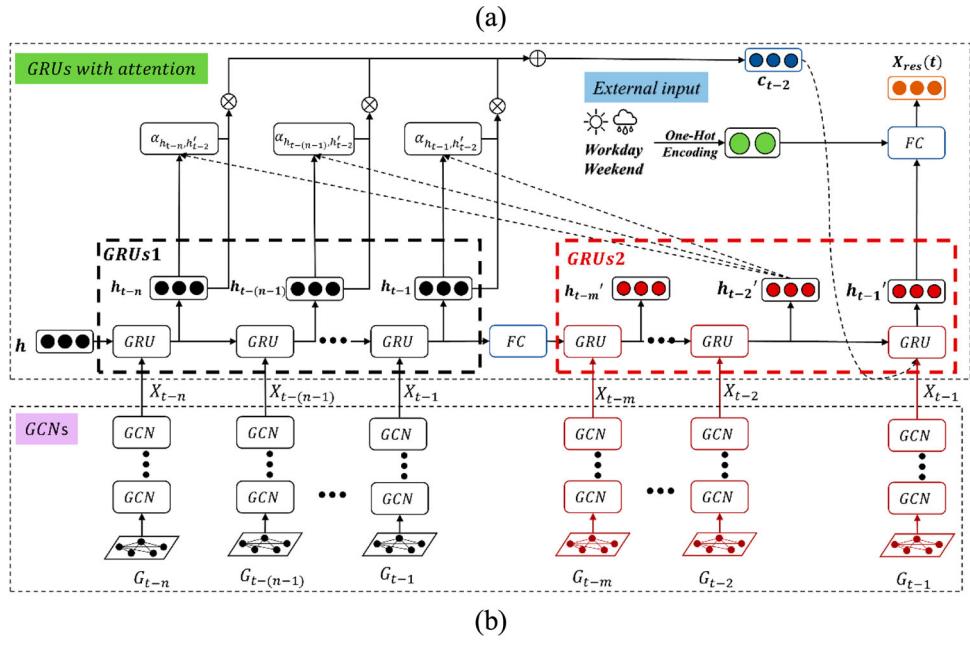
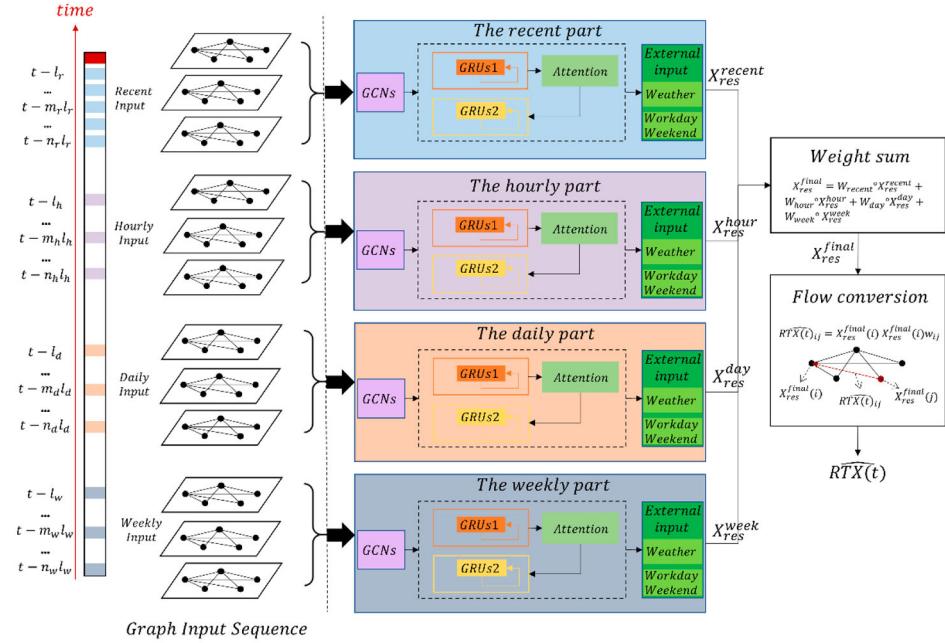


Figure 4. An illustration of (a) our model structure and (b) its components.

$$\alpha_{ij} = \frac{\exp(v^\top \tanh(W[h'_i, h_j]))}{\sum_{s=t-n}^{t-1} \exp(v^\top \tanh(W[h'_i, h_s]))}, \quad (4)$$

$$c_i = \sum_{j=t-n}^{t-1} \alpha_{ij} h_j. \quad (5)$$

- (3) *Fusion and Conversion:* Fusion includes two steps in our model. The first step is to fuse the outputs of the revised GCNs and GRUs with the external input using a fully-connected layer. As shown in (6), $\text{FC}(\cdot)$ denotes the operation of fully connection, X_{ext} represents the external input of weather and Workday/Weekend, and $X_{\text{res}}(t)$ is the output corresponding to the recent, hourly, daily, or weekly part. Specifically, a one-hot sixteen-bit representation is used to denote X_{ext} , where the first 15 codes suggest the weather condition ({'Partly Cloudy', 'Cloudy', 'Heavy Rain Shower', 'Heavy Rain Shower/Windy', 'Haze', 'Light Rain with Thunder', 'Light Rain Shower', 'Rain', 'Light Rain', 'Mostly Cloudy', 'Fair / Windy', 'Thunder', 'Rain Shower', 'Fair', 'Fog'}, available at www.wunderground.com) and the last code indicates information of Workday/Weekend, for example, {0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1} indicates cloudy weather in workday and {0 0 0 0 0 0 0 0 0 0 0 0 1 0 0} indicates fair weather in weekend. The second step is to fuse the outputs of the recent, hourly, daily, and weekly parts using a parametric-matrix-based method. As shown in (7), $X_{\text{res}}^{\text{recent}}$, $X_{\text{res}}^{\text{hour}}$, $X_{\text{res}}^{\text{day}}$, and $X_{\text{res}}^{\text{week}}$ are outputs from the four parts, W_{recent} , W_{hour} , W_{day} , and W_{week} are learnable parameters that can give different weights to the four parts. In this way, we can produce $X_{\text{res}}^{\text{final}}(t)$, a vector representation of the deep features of traffic flow for all the roads at time period t .

Conversion intends to estimate the turning traffic flow using the deep features of traffic flow of two adjacent roads. Specifically, to forecast the turning traffic flow, a transition matrix is used to model the traffic flow from one road segment to another. As shown in (8), W is a parameter matrix that can be learned in our model, (\circ) indicates the element-wise multiplication of two matrices, and $\widehat{\text{RTX}}(t)$ is the predicted turning traffic flow between two adjacent road segments.

$$X_{\text{res}}(t) = \text{FC}(\text{Con}[X_{t-1}, h'_{t-1}, c_{t-2}, X_{\text{ext}}]), \quad (6)$$

$$X_{\text{res}}^{\text{final}}(t) = W_{\text{recent}} \circ X_{\text{res}}^{\text{recent}} + W_{\text{hour}} \circ X_{\text{res}}^{\text{hour}} + W_{\text{day}} \circ X_{\text{res}}^{\text{day}} + W_{\text{week}} \circ X_{\text{res}}^{\text{week}} \in \mathbb{R}^{nr \times 1}, \quad (7)$$

$$\widehat{\text{RTX}}(t) = ((X_{\text{res}}^{\text{final}}(t) * X_{\text{res}}^{\text{final}}(t)^T) \circ A) \circ W \in \mathbb{R}^{nr \times nr}. \quad (8)$$

3.3. Training and evaluation of the model

- (1) *Network training:* To train our model, we firstly prepare the directed weighted topological graphs at the time interval of 10 minutes for the four input sequences from historical trajectory trips. Secondly, the input graphs are gradually used to train our model using forward and backward propagation and Adam method, aiming to minimise the total mean squared error and to handle the sparse distribution of turning traffic flow. As shown in (9), \widehat{rtx}_i^t and rtx_i^t are the predicted and observed turning traffic flow i at time period t , respectively, nrt is the number of road turns, $\varphi(rtx_i^t)$ is an indicator function with the value of 1 if rtx_i^t is equal to 0 and with the value of β ($\beta > 1$) if rtx_i^t is greater than 0. Thirdly, we obtain the well-trained model Λ when the total mean squared error $L(\emptyset)$ is significantly small.

$$L(\emptyset) = \sum_{i=1}^{nrt} (\widehat{rtx}_i^t - rtx_i^t) \varphi(rtx_i^t)^2. \quad (9)$$

- (2) *Network evaluation:* Once our model Λ is trained, its accuracy is evaluated using two metrics: the mean absolute error MAE and the root mean squared error $RMSE$. As shown in (10) and (11), the two metrics can avoid the ‘infinite error’ issue, meaning that the occurrences of infinite or undefined values owing to zero or close-to-zero observed values.

$$MAE = \frac{1}{nrt} \sum_{i=1}^{nrt} |\widehat{rtx}_i^t - rtx_i^t|, \quad (10)$$

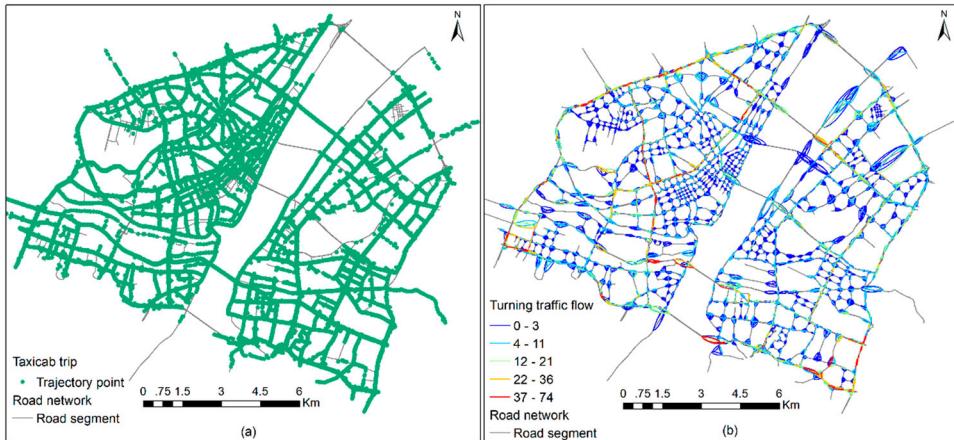


Figure 5. Data used in our study: (a) taxicab trajectory trips overlaid on the road network; and (b) visualisation of turning traffic flow with each arc in different colours.

$$\text{RMSE} = \sqrt{\frac{1}{nrt} \sum_{i=1}^{nrt} (\widehat{rtx}_i^t - rtx_i^t)^2}. \quad (11)$$

4. Experimental results and analysis

4.1. Data

In this study, taxicab GPS trajectory trips are used to derive turning traffic flow within the second ring road in Wuhan, China. Although the data have been criticised for providing incomplete information, they are ubiquitous and used intensively in traffic flow prediction tasks (Zhang et al. 2018; Jia and Yan 2021). Each trip constitutes a series of trajectory points in chronological order and is associated with a specific travel purpose, and thus the data accurately reflect the human activity patterns in the city (Jia and Jiang 2012). As shown in Figure 5(a), the data are composed of 4,720,624 trips containing a number of 236,004,391 trajectory points, which are contributed by 7035 taxicabs operating from July 1 to 28 in 2015. Using our method elaborated in Section 2.2, the road network is firstly converted into a directed topological weighted graph, which is composed of 4135 nodes and 18,734 arcs. Then, trajectory points of each trip are spatially matched to the nearest road segments, which produces road traffic flow and turning traffic flow at the time interval of 10 min. Specifically, road traffic flow is assigned to each node as its initial feature, while turning traffic flow is assigned to each arc as the observed value. As shown in Figure 5(b), we visualise turning traffic flow with each arc in different colours.

4.2. Model settings

Parameters of our model mainly include learning rate, batch size, training epoch, and those related to network structures. In the experiment, we manually set learning rate, batch size, and training epoch to 0.00025, 64, and 500, respectively. Network structures refer to parameters of GCNs, GRUs1, GRUs2, FC1, and FC2. As shown in Table 1, GCNs use two layers of propagations to extract spatial dependencies of neighbouring nodes and outputs a spatial feature vector with size 4135×1 for each graph, and it works for the recent part, hourly part, daily part, and weekly part; GRUs1 adopt 9 units, 3 units, 7 units, and 2 units in one layer to learn temporal dependencies of long-term sequence for the four parts, respectively, and it outputs a hidden state vector of 4135×1 ; FC1 converts the hidden state vector into a feature vector of 2000×1 and fed it into GRUs2, and it works for the four parts; GRUs2 uses 3 units,

Table 1. Parameter settings of our model.

Layers	Output	Recent part	Hourly part	Daily part	Weekly part
GCNs	4135×1	2 layers	2 layers	2 layers	2 layers
GRUs1	4135×1	1 layer, 9 units	1 layer, 3 units	1 layer, 7 units	1 layer, 2 units
FC1	2000×1	4135×2000	4135×2000	4135×2000	4135×2000
Tanh					
GRUs2	2000×1	1 layer, 3 units	1 layer, 2 units	1 layer, 3 units	1 layer, 1 units
FC2	4135×1	10286×4135	10286×4135	10286×4135	10286×4135
Weighted sum layer					
Conversion layer					
ReLU					

2 units, 3 units, and 1 unit in one layer to learn the attention-based temporal dependencies of short-term sequence for the four parts, respectively, and it outputs a temporal feature vector of 2000×1 ; FC2 converts the concatenation of spatial feature vector, temporal feature vector, attention feature vector, and external feature vector into an output feature vector of 4135×1 , and it works for the four parts.

It should be noted that parameter settings on network input are related to the structures of GRUs1 and GRUs2. Specifically, n_r and m_r are lengths of two input sequences for recent part and are set as 9 and 3, n_h and m_h are set as 3 and 2 for hourly part, n_d and m_d are set as 7 and 3 for daily part, n_w and m_w are set as 2 and 1 for weekly part. In addition, the observed graphs are divided into two sets: the training set and the testing set. We used 57% of graphs for training and 43% for testing. Among the training set, we used 90% of graphs for model training and 10% for cross validation.

4.3. Results of the predicted turning traffic flow

As shown in Figure 6(a), we display spatial distribution of the predicted turning traffic flow around 07:00 from 06:50 on 26 July 2015, which resembles very well with that observed shown in Figure 6(b). Our results suggest that turning traffic flow is mostly distributed at very few major road intersections in the early morning, such as the intersection between the Wuhan Avenue and the Peace Avenue, the intersection between the Xiongchu Avenue and the Shipailing Road. Specifically, given one road intersection, we can also observe clearly how traffic flow is transferred from one road segment to another. For instance, a large volume of traffic flow is transferred from the Wuhan avenue to the second Yangtze River Bridge, which is reasonable because it is one of the bridges carrying most traffic connecting Hankou and Wuchang. However, turning traffic flow is deviated slightly from that observed at the road intersections along the second ring-road. This can be attributed to the edge effects in ecology, which means that road segments along the boundary may be also connected to these outside the boundary and turning traffic flow outside may affect that along the boundary. Nonetheless, as shown in Figure 6(e), we can see that a majority of predicted turning traffic flow (83.86%) has a very small value of absolute error (≤ 2) and only a minority (1.04%) has a very large value (≥ 8).

As shown in Figure 6(c), we present spatial distribution of the predicted turning traffic flow around 19:10 from 19:00 on 26 July 2015, which is much heavier than that observed at 07:00 owing to the commuting behaviour in the late afternoon. For instance, road intersections along the Wuhan Avenue or the second ring-road display much heavier traffic flow. Although turning traffic flow is overestimated at a few road intersections, such as B or C in Figure 6(c), it has been predicted well with high accuracy at most road intersections, which is generally consistent with that observed at 19:10. Specifically, as shown in Figure 6(f), 79.91% of predicted turning traffic flow is reported to have a small value of absolute error (≤ 2), while only 1.43% has a very large value (≥ 8). This finding further agrees roughly with that reported in Figure 6(e). Therefore, it suggests that our model can work very well in both spare time and rush hour.

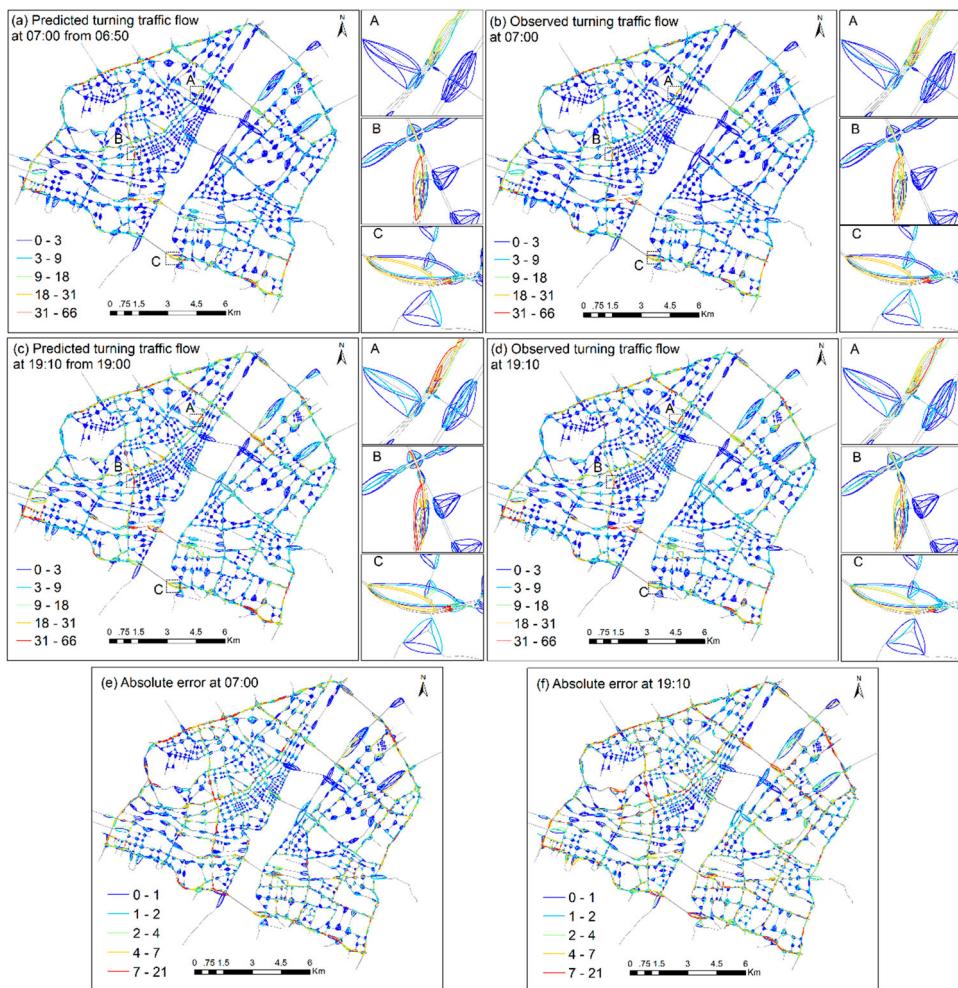


Figure 6. Comparisons of predicted turning traffic flow with that observed on 26 July 2015: (a) predicted at 07:00; (b) observed at 07:00; (c) predicted at 19:10; (d) observed at 19:10; (e) absolute error at 07:00; and (f) absolute error at 19:10.

To verify the reliability of our model, turning traffic flow is predicted every 10 minutes in one day. As shown in Figure 7(a), two facts can be clearly observed, strongly supporting our findings in Figure 6. On one hand, it displays a rough stable pattern for the percentage of predicted turning traffic flow with a small value of absolute error (≤ 2), although on average a low percentage of 77.10% can be observed in rush hour and a high percentage of 93.39% can be observed in spare time. On the other hand, the similar stable pattern can be seen for the percentage of predicted turning traffic flow with a large value of absolute error (≥ 8), which fluctuates around 1.11% on average. Overall, our results suggest the reliability of our model for application in ITS. In addition, it is unknown how our predicted turning traffic flow can be influenced by the recent, hourly, daily, and weekly parts. As shown in Figure 7(b), it is interesting to find that the influence of the recent part is much stronger than the other three parts, which is reasonable since it is much more correlated with the current turning traffic flow. Moreover, the influence of the daily part is stronger than hourly and weekly parts, which can be mainly explained by the daily periodic pattern of turning traffic flow.

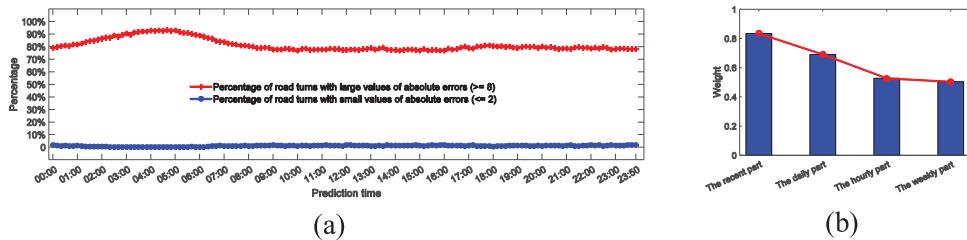


Figure 7. Plots of (a) the percentage of road turns with large/small values of absolute errors every 10 minutes in one day and (b) the influence of the four components on prediction.

4.4. Evaluation of the results

To evaluate our model, we compare the results with those from current state-of-the-art predictive methods, including Autoregressive (AR), Auto Regressive Integrated Moving Average (ARIMA), LSTM, GRU, ST-ResNet, T-GCN, A3T-GCN, and a simplified version of our model (Sim-model).

AR is a multiple linear regression in which the value of the series at the current time is a linear function of the values at the previous times. In this study, arc weights of the graph were converted into a 1D vector as input.

ARIMA is a statistical model for analysing and forecasting time series data, which is based on the statistical concept of serial correlation that previous data points influence future data points. In this study, arc weights of the graph were converted into a 1D vector as input.

LSTM is a special kind of RNNs that can learn long-term dependencies and is widely used for time series forecasting. In this study, arc weights of each graph were converted into a 1D vector, and the length of the input sequence was set as 6 and 12, which gave LSTM-6 and LSTM-12.

GRU is a new kind of RNNs that was proposed recently and is particularly suitable for forecasting in time series data (Fu, Zhang, and Li 2016). Compared with LSTM, it can offer competitive performance and faster computation. In this study, we adopted GRU-6 and GRU-12, which have the same settings as LSTM.

ST-ResNet is a recent proposed model for capturing spatiotemporal dependencies of image sequence data, but it is mainly used for predicting traffic flow at the region level (Zhang et al. 2018). In this study, the topological graph was firstly converted into a 2D image, where each pixel corresponds to a road turn. Then, these 2D images were taken as inputs, and other settings were the same as in the previous study (Zhang et al. 2018).

T-GCN is a novel temporal graph convolutional network model (Zhao et al. 2019), which is simply a combination of GCNs and GRUs and is only used to predict road traffic flow. The major difference from our model is that it does not implement an attention mechanism to handle the fluctuations of temporal dependencies of traffic flow. In this study, a transition matrix was added to the model for converting road traffic flow into turning traffic flow.

A3T-GCN is a novel attention-based temporal graph convolutional network for traffic flow prediction (Bai et al. 2021), which is based on T-GCN and adds the attention mechanism to adjust the importance of different time points. The major difference from our model is that the attention mechanism is implemented in a different way, for instance, it uses a very basic soft attention method, while our model uses an encoder-decoder-based soft attention method. In this study, a transition matrix was added to the model for converting road traffic flow into turning traffic flow.

Sim-model is designed by removing the attention mechanism from our model, which aims to check the influence of the attention mechanism on our model. In this study, it only took the long-term graph sequence of our model as input, and other settings were the same as our model.

As shown in Figure 8, we present the comparative results, where the values of MAE and RMSE are displayed for evaluating different models. Generally, the values of MAE are in consistence with those

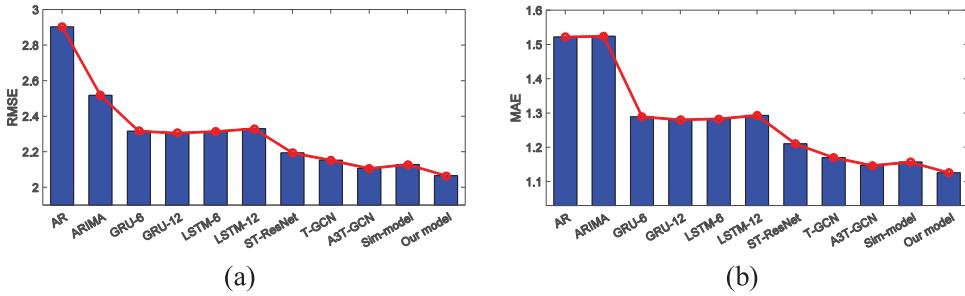


Figure 8. Accuracy comparisons of our model with current state-of-the-art models in short-term turning traffic flow prediction using: (a) RMSE and (b) MAE.

of RMSE with respect to different models. Specifically, our model achieves the highest accuracy in terms of both MAE and RMSE, which has a better performance than the other models. The results of AR and ARIMA have the lowest accuracy, which can be mainly attributed to their shallow structures. Although the prediction accuracy can be improved using LSTM-6, LSTM-12, GRU-6, and GRU-12, its accuracy remains 12.05%, 12.80%, 12.13%, and 11.61% lower than our model in terms of RMSE and 13.95%, 14.86%, 14.50%, and 13.75% lower than our model in terms of MAE. This is because spatial dependencies of turning traffic flow cannot be captured very well by LSTM and GRU variants. Meantime, the prediction accuracy can be improved largely using ST-ResNet, but the values of RMSE and MAE of our model are still 6.22% and 7.49% higher than those of ST-ResNet. This is because ST-ResNet can capture partial spatial topological dependencies and cannot handle spatial sparseness. The novel models of T-GCN and A3T-GCN can achieve higher accuracy than the other models, but their accuracy remain 4.18% and 1.95% lower than our model in terms of RMSE and 3.93% and 1.88% lower than our model in terms of MAE. Additionally, we evaluated our model by removing the attention mechanism, and the result suggests increases of the values of RMSE by 3.04% and MAE by 2.77%. Therefore, these results demonstrate the significance of spatiotemporal dependencies and the attentions mechanism on turning traffic flow prediction, and they suggest the effectiveness and reliability of our model.

5. Discussions

5.1. Limitation of taxicab trajectories

It is true that turning traffic flow derived from taxicab trajectories is biased and cannot represent the real turning traffic flow of all vehicles, because taxicab is only one type of vehicle in the urban road network. On one hand, it is very difficult to obtain the real turning traffic flow of all vehicles in the roads of a large area, because it is very expensive to deploy and maintain the traffic surveillance camera systems to cover the roads of a large area. On the other hand, the main purpose of this study is to contribute an attention-based spatiotemporal deep learning model, which can solve the challenges faced by the prediction of turning traffic flow in a large area. In this respect, it is our conjecture that our model can predict well the real turning traffic flow if it can be trained with the trajectory data of all vehicles. Nonetheless, future work is needed to verify our model using traffic flow of all vehicles or to revise our model to integrate the taxicab trajectory data with the sampled field survey data for the prediction of real turning traffic flow.

5.2. Analysis of the influence of road traffic flow features on turning traffic flow prediction

It is interesting to discuss how the deep features of road traffic flow affect the prediction of turning traffic flow. To answer this question, we need to examine the characteristics of the weight values in

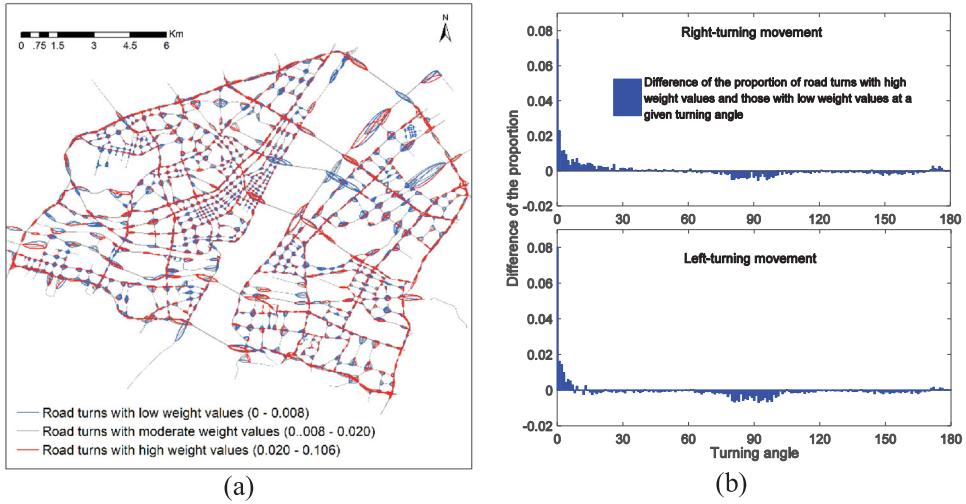


Figure 9. (a) Visualisation of the weight values of all road turns and (b) their relationship with turning angles.

parameter matrix W shown in (8), which quantitatively measures the influences of the deep features of road traffic flow on the turning traffic flow between two adjacent road segments. In this respect, the larger the weight value, the much more influence of the deep features of road traffic flow on the turning traffic flow. For the convenience of discussion, we demarcate the road turns into three classes according to the weight value using the natural break method. As shown in Figure 9(a), we display the spatial distribution of the weight values. Road turns with high values of weights tend to be located at the intersections of main roads, while those with low values are more likely to be located at the intersections of secondary or tertiary roads. Additionally, we calculate the proportion of road turns with a given turning angle for each class, and then we obtain the difference of the proportion of road turns with high weight values and those with low weight values for each turning angle. As shown in Figure 9(b), the results suggest that road turns with high weight values tend to be straightforward with very small turning angles, while those with low weight values are more likely to have right angles irrespective of right-turning or left-turning.

5.3. Impact of network input

The predictive results can be impacted by the length of input graph sequence. In this study, eight parameters, including $n_r, m_r, n_h, m_h, n_d, m_d, n_w$, and m_w should be specified to determine the length of input sequences for the recent, hourly, daily, and weekly parts. As for n_r and m_r shown in Figure 10(a,b), our model performs the best when they are set as 9 and 3 for the lengths of long-term and short-term input graph sequences in the recent part. As for n_h and m_h shown in Figure 10(c,d), our model performs the best when they are set as 3 and 2 for the lengths of long-term and short-term input graph sequences in the hourly part. As for n_d and m_d shown in Figure 10(e,f), our model achieves the highest accuracy when they are set as 7 and 3 for the lengths of long-term and short-term input graph sequences in the daily part. As for n_w and m_w shown in Figure 10(g,h), our model achieves the highest accuracy when they are set as 2 and 1 for the lengths of long-term and short-term input graph sequences in the weekly part. These findings suggest that the length of input sequence is not the longer the better and that the impacts of the hourly and weekly parts are weaker than the other two parts with short length of input sequence. In this respect, the length of input sequences should be carefully treated.

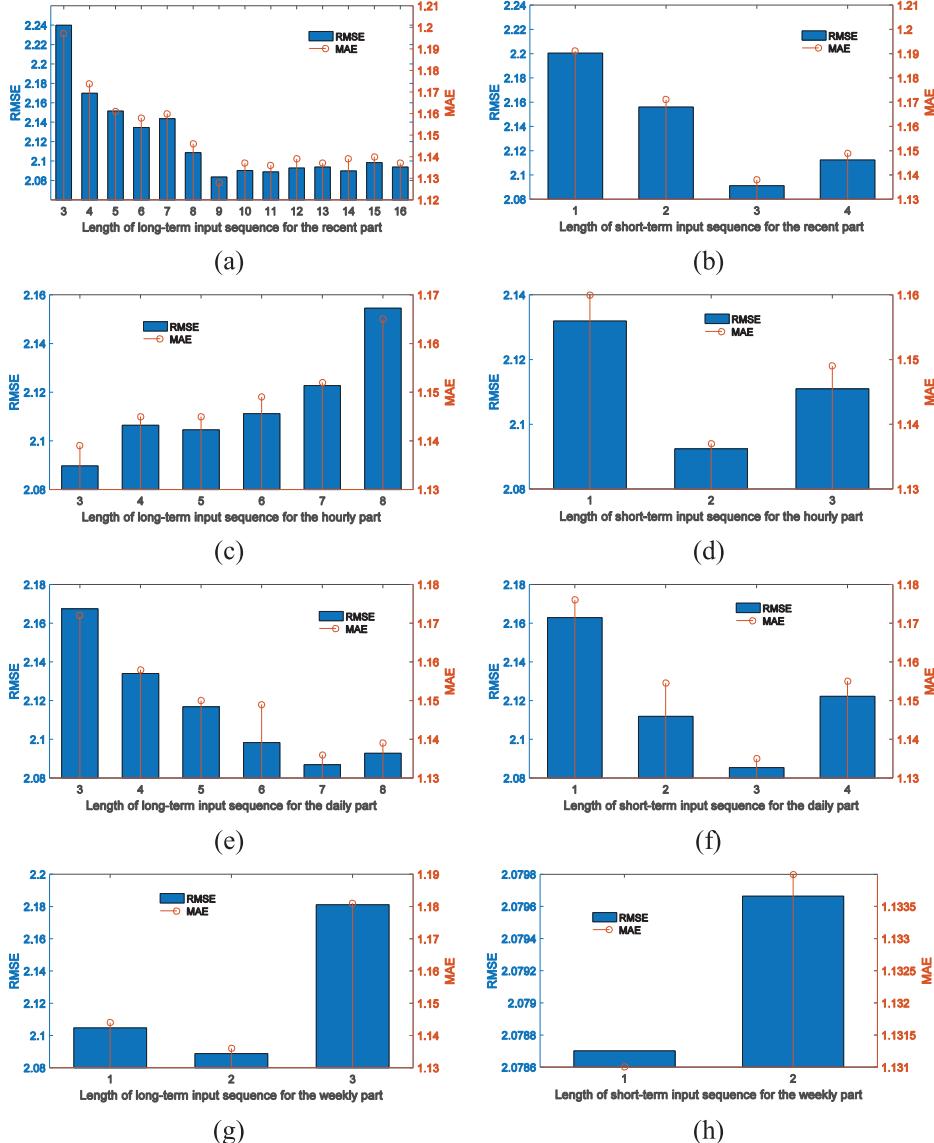


Figure 10. Impact of length of network input sequence: (a) n_r ; (b) m_r ; (c) n_h ; (d) m_h ; (e) n_d ; (f) m_d ; (g) n_w ; and (h) m_w .

5.4. Impact of network structure

(1) Structure of GCNs

GCNs are used in all the four parts, and they are related to only one parameter l , namely the number of layers. However, it varies in different parts of our model, and thus, l_r , l_h , l_d , and l_w should be determined to give different values for the recent, hourly, daily, and weekly parts, respectively. As for l_r shown in Figure 11(a), we present the change of layer number in the recent part with accuracy in terms of both RMSE and MAE. At the beginning, RMSE or MAE decreases with the increment of the number of layers, which is reasonable because much more information of nodes in a few topological distances away can be aggregated. However, it increases with more layers being included, which can be attributed to the

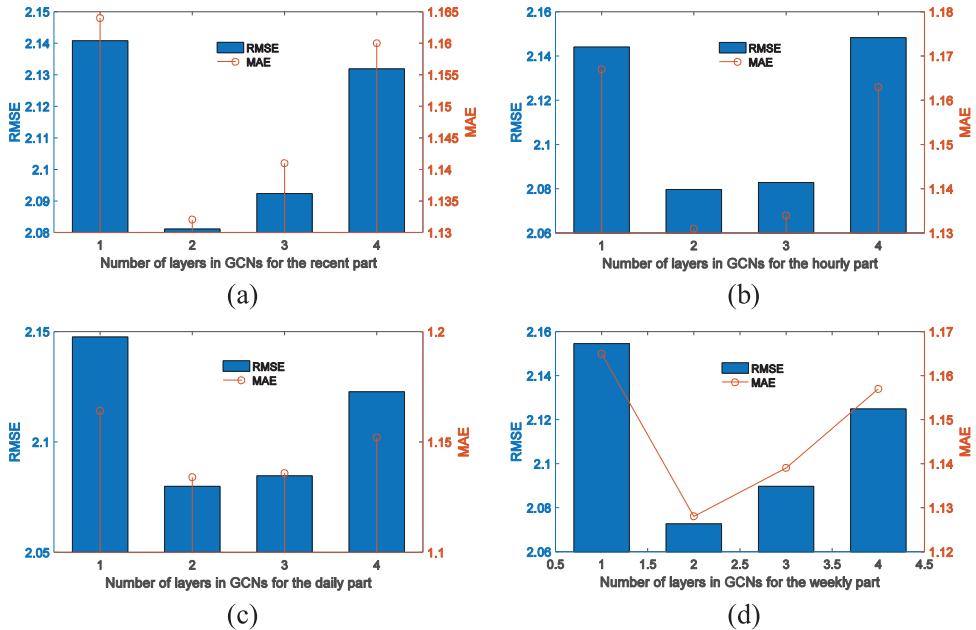


Figure 11. Impact of network structure: (a) I_r ; (b) I_h ; (c) I_d ; and (d) I_w .

redundant information increasing the burden of model learning. Thus, the value of I_r is set as 2, giving a relatively high accuracy. Additionally, we can observe the similar pattern in Figure 11(b–d), where the values of I_h , I_d , and I_w are all set as 2 to give small values of RMSE and MAE.

(2) Structure of GRUs

GRUs are also used in all the four parts, and they are mainly relevant to the number of units. Each GRU unit can handle one item of the input graph sequence, and in this sense, settings of this parameter are the same as those related to network input (see Figure 9). In our model, for each part, GRU1 is designed to model the long-term input sequence, and GRU2 is developed to model the short-term input sequence. Thus, the number of units in GRU1 and GRU2 are 9 and 3 for the recent part, 3 and 2 for the hourly part, 7 and 3 for the daily part, and 2 and 1 for the weekly part.

(3) Parameter of loss function

In the loss function, a parameter β is used to control the penalty for the observed turning traffic flow with non-zero values. This is because the topological graph tends to be sparse, where the number of turning traffic flow with non-zero values is far less than those with zero values. In this respect, a large value of β can avoid the occurrence of the situation that almost all turning traffic flow are predicted with zero values, which was inappropriately considered as high prediction accuracy. As shown in Figure 12, we display the change of β value with accuracy in terms of RMSE and MAE. It can be seen clearly that RMSE or MAE decreases with the increment of β values at the beginning, which is reasonable because most of the observed turning traffic flow with non-zero values can be predicted with very small errors. However, the accuracy begins to decrease as the value of β continues to increase, which can be mainly attributed to the large errors from the prediction on turning traffic flow with observed zero values. Thus, the value of β is set as 1.5 to give a relatively high accuracy.

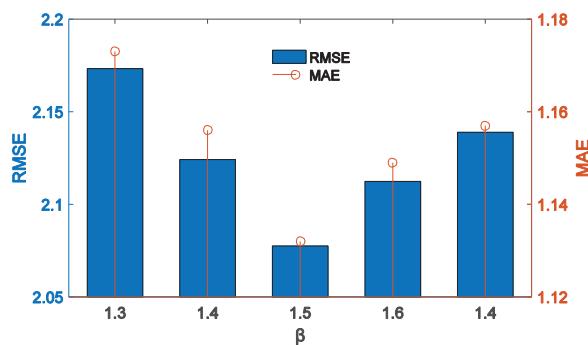


Figure 12. Impact of the parameter β in the loss function.

5.5. Impact of external factors

External factors of Workday/Weekend and weather are considered in our model, and experimental results suggest that the prediction accuracy in terms of RMSE can be improved by 0.41% when including these external factors. However, factors of holidays, urban accidents, social events, temperature, and land use patterns are not considered in our model due to the following reasons. First, our dataset lacks the samples collected on holidays, which poses a limitation of our model incapable of modelling turning traffic flow on holidays. Although Jia and Yan (2021) suggested that factor of holidays has a limited influence on prediction, it will be considered in our future work. Second, it is true that factors of urban accidents, social events, and land use patterns can play a nontrivial influence on the prediction accuracy, but they are not considered in our model owing to the unavailability of these data. In this respect, it is a limitation of our study and will be considered by our model in future. Third, it was reported that temperature may affect the prediction of taxi demand (Liu et al. 2020a), but it may have limited influence on our predictive results, owing to the fact that it rarely changes in a short time period of 10 minutes. In this sense, we assume that it should be considered in long-term prediction tasks, and we will examine its influence in future work. Additionally, our model did not consider the influence from other public transportation modes, such as the placement of a subway station or a bus station. This influence could be non-negligible if the task is to predict taxi demand (Liu et al. 2020b; Liu and Chen 2022), but it might be very limited in our study. Nevertheless, future work should collect relevant data to evaluate its influence on our predictive results.

5.6. Impact of network components and prediction interval

It is interesting to discuss the influence of network components and prediction interval on the accuracy of our results. (1) We have conducted experiment to assess the influence of the hourly part, the daily part, and the weekly part. For instance, the accuracy of our predictive results in terms of RMSE can be decreased by 2.04% when the hourly part, the daily part, and the weekly part are removed; it can be decreased by 1.19% when the daily part and the weekly part are removed; and it can be decreased by 0.34% when only the weekly part is removed. In this respect, we conclude that the other three parts have slight influences on the predictive results. (2) To examine the influence of the prediction interval, we used our model to conduct multi-steps predictions, meaning that the predictive results at the current time point are taken as inputs to the model to predict the results at the next time point. As for the prediction intervals of 20 minutes, 30 minutes, 40 minutes, 50 minutes, and 1 hour, we have conducted two-steps, three-steps, four-steps, five-steps, and six-steps predictions. The results suggest that the accuracy of our results has decreased with the increment of the prediction interval. For instance, the RSME values of the predictive results are 2.08, 2.13, 2.17, 2.23, and 2.32 for the prediction intervals of 20 minutes, 30 minutes, 40 minutes, 50 minutes, and 1 hour, respectively.

6. Conclusions

This study developed a novel attention-based spatiotemporal deep learning model to predict citywide turning traffic flow in a fine temporal scale of 10 minutes with high accuracy. We firstly converted the road network into a directed weighted topological graph and derived the observed turning traffic flow as the weight of each arc from trajectory trips. Then, based on an in-depth analysis of spatiotemporal patterns of turning traffic flow, we proposed the structure of our network including the recent, hourly, daily, and weekly parts. The four parts share the same network structure but have different inputs, and specifically, they are responsible for capturing spatiotemporal dependencies of sparse graphs in different temporal granularities. Finally, outputs from the four parts were fused together with different weights learned in the training process, which were used to predict citywide traffic flow from one road segment to another. It should be noted that parameters of our model are carefully tuned for making predictions with high accuracy.

To verify our model, experiments were conducted in Wuhan, China. Taxicab trajectory trips were adopted to derive turning traffic flow, which was used as the weight of each arc to construct a sequence of directed weighted topological graphs. Based on these graphs, our model was trained and the network parameters were empirically determined. Using our model, turning traffic flow was forecasted in 10 minutes across the entire city, and it is highly consistent with that observed through statistical analysis. Specifically, it was much more affected by the recent and daily parts than the hourly and weekly parts. Importantly, comparative results suggest that our model outperforms current state-of-the-art models in short-term turning traffic flow prediction tasks, indicating its effectiveness and reliability for citywide prediction with high accuracy.

Acknowledgements

We thank the editor and anonymous referees for their valuable comments for improving the manuscript, and we thank Jonathan Miller for polishing our language.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by National Natural Science Foundation of China [grant number 41971332].

Data and codes availability statement

The data and codes that support the findings of this study are available at the link: http://ggssc.whu.edu.cn/ggsscAssets/download/AttentionModel/code_and_data.zip.

References

- Abadi, A., T. Rajabioun, and P. A. Ioannou. 2015. "Traffic Flow Prediction for Road Transportation Networks with Limited Traffic Data." *IEEE Transactions on Intelligent Transportation Systems* 16 (2): 653–662. doi:[10.1109/TITS.2014.2337238](https://doi.org/10.1109/TITS.2014.2337238). Author(s), blinded for peer review.
- Bahdanau, D., K. Cho, and Y. Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv, preprint*, arXiv:1409.0473. <https://arxiv.org/abs/1409.0473>.
- Bai, J., J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li. 2021. "A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting." *ISPRS International Journal of Geo-Information* 10 (7): 485.
- Cai, P., Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun. 2016. "A Spatiotemporal Correlative K-Nearest Neighbor Model for Short-Term Traffic Multistep Forecasting." *Transportation Research Part C: Emerging Technologies* 62: 21–34. doi:[10.1016/j.trc.2015.11.002](https://doi.org/10.1016/j.trc.2015.11.002).
- Cai, L., Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin. 2019. "A Noise-Immune Kalman Filter for Short-Term Traffic Flow Forecasting." *Physica A: Statistical Mechanics and its Applications* 536: 122601. doi:[10.1016/j.physa.2019.122601](https://doi.org/10.1016/j.physa.2019.122601).
- Chen, B., and H. H. Cheng. 2010. "A Review of the Applications of Agent Technology in Traffic and Transportation Systems." *IEEE Transactions on Intelligent Transportation Systems* 11 (2): 485–497. doi:[10.1109/TITS.2010.2048313](https://doi.org/10.1109/TITS.2010.2048313).

- Chen, A., P. Chootinan, S. Ryu, M. Lee, and W. Recker. 2012. "An Intersection Turning Movement Estimation Procedure Based on Path Flow Estimator." *Journal of Advanced Transportation* 46: 161–176.
- Chen, Y., M. Guizani, Y. Zhang, L. Wang, N. Crespi, G. M. Lee, and T. Wu. 2019. "When Traffic Flow Prediction and Wireless Big Data Analytics Meet." *IEEE Network* 33 (3): 161–167. doi:10.1109/MNET.2018.1800134.
- Cho, K., B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning Phrase Representations using RNN Encoder Decoder for Statistical Machine Translation. In: 2014 Conference on Empirical Methods in Natural Language Processing, 1724–1734. doi:10.3115/v1/D14-1179.
- Du, B., H. Peng, S. Wang, M. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li. 2019. "Deep Irregular Convolutional Residual LSTM for Urban Traffic Passenger Flows Prediction." *IEEE Transactions on Intelligent Transportation Systems* 21 (3): 972–985.
- Fu, R., Z. Zhang, and L. Li. 2016. Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction. In: 31st Youth Academic Annual Conference of Chinese Association of Automation, 324–328. doi:10.1109/YAC.2016.7804912.
- Ghanim, M. S., and K. Shaaban. 2019. "Estimating Turning Movements at Signalized Intersections Using Artificial Neural Networks." *IEEE Transactions on Intelligent Transportation Systems* 20 (5): 1828–1836.
- Ghosh, B., B. Basu, and M. O. Mahony. 2009. "Multivariate Short-Term Traffic Flow Forecasting Using Time-Series Analysis." *IEEE Transactions on Intelligent Transportation Systems* 10 (2): 246–254. doi:10.1109/TITS.2009.2021448.
- Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, et al. 2018. "Recent Advances in Convolutional Neural Networks." *Pattern Recognition* 77: 354–377. doi:10.1016/j.patcog.2017.10.013.
- Guo, J., W. Huang, and B. M. Williams. 2014. "Adaptive Kalman Filter Approach for Stochastic Short-Term Traffic Flow Rate Prediction and Uncertainty Quantification." *Transportation Research Part C: Emerging Technologies* 43: 50–64. doi:10.1016/j.trc.2014.02.006.
- Guo, F., J. W. Polak, and R. Krishnan. 2018. "Predictor Fusion for Short-Term Traffic Forecasting." *Transportation Research Part C: Emerging Technologies* 92: 90–100.
- Jia, T., and B. Jiang. 2012. "Exploring human activity patterns using taxicab static points." *ISPRS International Journal of Geo-Information* 1 (1): 89–107. doi:10.3390/ijgi1010089.
- Jia, T., Q. Li, and W. Shi. 2018. "Estimation and analysis of emissions from on-road vehicles in Mainland China for the period 2011–2015." *ISPRS International Journal of Geo-Information* 191: 500–512. doi:10.1016/j.atmosenv.2018.08.037.
- Jia, T., and P. Yan. 2021. "Predicting citywide road traffic flow using deep spatiotemporal neural networks." *ISPRS International Journal of Geo-Information* 22 (5): 3100–3111. doi:10.1109/TITS.2020.2979634.
- Jia, T., P. Zhang. 2022. "A microscopic model of vehicle CO₂ emissions based on deep learning-A spatiotemporal analysis of taxicabs in Wuhan, China." *ISPRS International Journal of Geo.* doi:10.1109/TITS.2022.3151655.
- Jiang, W., and J. Luo. 2021. "Graph Neural Network for Traffic Forecasting: A Survey." arXiv:2101.11174v4.
- Kipf, T. N., and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks, In: *International Conference on Learning Representations*, 1–14.
- Kumar, S. V., and L. Vanajakshi. 2015. "Short-term Traffic Flow Prediction Using Seasonal ARIMA Model with Limited Input Data." *European Transport Research Review* 7 (3): 1–9. <https://doi.org/10.1007/s12544-015-0170-8>.
- LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444. doi:10.1038/nature14539.
- Li, W., and X. Ban. 2019. A Deep Learning Approach for Lane-Based Short-Term Traffic Volume Prediction at Signalized Intersections. Transportation Research Board 98th Annual Meeting.
- Li, Z., S. Jiang, L. Li, and Y. Li. 2017. "Building Sparse Models for Traffic Flow Prediction: An Empirical Comparison Between Statistical Heuristics and Geometric Heuristics for Bayesian Network Approaches." *Transportmetrica B: Transport Dynamics* 7 (1): 107–203. doi:10.1080/21680566.2017.1354737.
- Li, W., J. Wang, R. Fan, Y. Zhang, Q. Guo, C. Siddique, and X. Jeff. 2020. "Short-term Traffic State Prediction from Latent Structures: Accuracy vs. Efficiency." *Transportation Research Part C: Emerging Technologies* 111: 72–90.
- Liu, Z., and H. Chen. 2022. "Short-term Online Taxi-Hailing Demand Prediction Based on the Multi-Mode Traffic Data in Metro Station Areas." *Journal of Transportation Engineering, Part A: Systems* 148 (6): 1–14.
- Liu, Z., H. Chen, Y. Li, and Q. Zhang. 2020a. "Taxi Demand Prediction Based on a Combination Forecasting Model in Hotspots." *Journal of Advanced Transportation* 1302586: 1–13.
- Liu, Z., H. Chen, X. Sun, and H. Chen. 2020b. "Data-driven Real-Time Online Taxi-Hailing Demand Forecasting Based on Machine Learning Method." *Applied Sciences* 10 (6681): 1–18.
- Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Wang. 2014. "Traffic Flow Prediction with Big Data: A Deep Learning Approach." *IEEE Transactions on Intelligent Transportation Systems* 16 (2): 865–873. doi:10.1109/TITS.2014.2345663.
- Ma, T., Z. Zhou, and B. Abdulhai. 2015. "Nonlinear Multivariate Time Space Threshold Vector Error Correction Model for Short Term Traffic State Prediction." *Transportation Research Part B: Methodological* 76: 27–47. doi:10.1016/j.trb.2015.02.008.
- Mahmoud, N., M. Abdel-Aty, Q. Cai, and J. Yuan. 2021. "Predicting Cycle-Level Traffic Movements at Signalized Intersections Using Machine Learning Models." *Transportation Research Part C: Emerging Technologies* 124: 102930.
- Miglani, A., and N. Kumar. 2019. "Deep Learning Models for Traffic Flow Prediction in Autonomous Vehicles: A Review, Solutions, and Challenges." *Vehicular Communications* 20: 100–184. doi:10.1016/j.vehcom.2019.100184.
- Min, X., J. Hu, Q. Chen, T. Zhang, and Y. Zhang. 2009. "Short-Term Traffic Flow Forecasting of Urban Network Based on Dynamic STARIMA Model." In: *12th International IEEE Conference on Intelligent Transportation Systems*, 1–6. doi:10.1109/ITSC.2009.5309741.

- Peng, H., B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, and L. He. 2021. "Dynamic Graph Convolutional Network for Long-Term Traffic Flow Prediction with Reinforcement Learning." *Information Science* 578: 401–416.
- Peng, H., H. Wang, B. Du, M. Bhuiyan, H. Ma, J. Liu, L. Wang, et al. 2020. "Spatial Temporal Incidence Dynamic Graph Neural Networks for Traffic Flow Forecasting." *Information Science* 521: 277–290.
- Polson, N. G., and V. O. Sokolov. 2017. "Deep Learning for Short-Term Traffic Flow Prediction." *Transportation Research Part C: Emerging Technologies* 79: 1–17. doi:[10.1016/j.trc.2017.02.024](https://doi.org/10.1016/j.trc.2017.02.024).
- Shoup, G., S. M. Remias, A. M. Hainen, G. Grimmer, and A. D. Davis. 2013. "Characterizing Reliability of Manual Intersection Turning Movement Counts Using Modern Data Collection Technology." *JTRP Other Publ. Reports* 6: 1–24.
- Smith, B. L., B. M. Williams, and R. K. Oswald. 2002. "Comparison of Parametric and Nonparametric Models for Traffic Flow Forecasting." *Transportation Research Part C: Emerging Technologies* 10 (4): 303–321. doi:[10.1016/S0968-090X\(02\)0009-8](https://doi.org/10.1016/S0968-090X(02)0009-8).
- Wang, J., and Q. Shi. 2013. "Short-term Traffic Speed Forecasting Hybrid Model Based on Chaos–Wavelet Analysis-Support Vector Machine Theory." *Transportation Research Part C: Emerging Technologies* 27: 219–232. doi:[10.1016/j.trc.2012.08.004](https://doi.org/10.1016/j.trc.2012.08.004).
- Williams, B. M., and L. A. Hoel. 2003. "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results." *Journal of Transportation Engineering* 129 (6): 664–672. doi:[10.1061/\(ASCE\)0733-947X\(2003\)129:6\(664\)](https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664).
- Wu, T., F. Chen, and Y. Wan. 2018a. "Graph Attention LSTM Network: A New Model for Traffic Flow Forecasting." In: *5th International Conference on Information Science and Control Engineering*, 241–245. doi:[10.1109/ICISCE.2018.00058](https://doi.org/10.1109/ICISCE.2018.00058).
- Wu, Y., H. Tan, L. Qin, B. Ran, and Z. Jiang. 2018b. "A Hybrid Deep Learning Based Traffic Flow Prediction Method and its Understanding." *Transportation Research Part C: Emerging Technologies* 90: 166–180. doi:[10.1016/j.trc.2018.03.001](https://doi.org/10.1016/j.trc.2018.03.001).
- Xia, D., B. Wang, H. Li, Y. Li, and Z. Zhang. 2016. "A Distributed Spatial-Temporal Weighted Model on MapReduce for Short-Term Traffic Flow Forecasting." *Neurocomputing* 179: 246–263.
- Yang, B., S. Sun, J. Li, X. Lin, and Y. Tian. 2019. "Traffic Flow Prediction Using LSTM with Feature Enhancement." *Neurocomputing* 332: 320–327.
- Yao, R., W. Zhang, and M. Long. 2022. "DLW-Net Model for Traffic Flow Prediction Under Adverse Weather." *Transportmetrica B: Transport Dynamics* 10 (1): 499–524. doi:[10.1080/21680566.2021.2008280](https://doi.org/10.1080/21680566.2021.2008280).
- Yoon, B., and H. Chang. 2014. "Potentialities of Data-Driven Nonparametric Regression in Urban Signalized Traffic Flow Forecasting." *Journal of Transportation Engineering* 140: 548–556.
- Zhan, H., G. Gomes, X. S. Li, K. Madduri, A. Sim, and K. Wu. 2018. "Consensus Ensemble System for Traffic Flow Prediction." *IEEE Transactions on Intelligent Transportation Systems* 19 (12): 1–12.
- Zhang, Y., T. Cheng, and Y. Ren. 2019. "A Graph Deep Learning Method for Short-Term Traffic Forecasting on Large Road Networks." *Computer-Aided Civil and Infrastructure Engineering* 34: 877–896. doi:[10.1111/mice.12450](https://doi.org/10.1111/mice.12450).
- Zhang, K., F. He, Z. Zhang, X. Lin, and M. Li. 2021a. "Graph Attention Temporal Convolutional Network for Traffic Speed Forecasting on Road Networks." *Transportmetrica B: Transport Dynamics* 9 (1): 153–171. doi:[10.1080/21680566.2020.1822765](https://doi.org/10.1080/21680566.2020.1822765).
- Zhang, L., Q. Liu, W. Yang, N. Wei, and D. Dong. 2013. "An Improved K-Nearest Neighbor Model for Short-Term Traffic Flow Prediction." *Procedia Social and Behavioral Sciences* 96: 653–662. doi:[10.1016/j.sbspro.2013.08.076](https://doi.org/10.1016/j.sbspro.2013.08.076).
- Zhang, J., F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen. 2011. "Data-Driven Intelligent Transportation Systems: A Survey." *IEEE Transactions on Intelligent Transportation Systems* 12 (4): 1624–1639. doi:[10.1109/TITS.2011.2158001](https://doi.org/10.1109/TITS.2011.2158001).
- Zhang, L., J. Wu, J. Shen, M. Chen, R. Wang, X. Zhou, and Q. Wu. 2021b. "SATP-GAN: Self-Attention Based Generative Adversarial Network for Traffic Flow Prediction." *Transportmetrica B: Transport Dynamics* 9 (1): 552–568. doi:[10.1080/21680566.2021.1916646](https://doi.org/10.1080/21680566.2021.1916646).
- Zhang, J., Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li. 2018. "Predicting Citywide Crowd Flows Using Deep Spatio-Temporal Residual Networks." *Artificial Intelligence* 259: 147–166. doi:[10.1016/j.artint.2018.03.002](https://doi.org/10.1016/j.artint.2018.03.002).
- Zhao, Z., W. Chen, X. Wu, P. Chen, and J. Liu. 2017. "LSTM Network: A Deep Learning Approach for Short-Term Traffic Forecast." *IET Intelligent Transport Systems* 11 (2): 68–75. doi:[10.1049/iet-its.2016.0208](https://doi.org/10.1049/iet-its.2016.0208).
- Zhao, L., Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. 2019. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction." *IEEE Transactions on Intelligent Transportation Systems* 21 (9): 3848–3858. doi:[10.1109/TITS.2019.2935152](https://doi.org/10.1109/TITS.2019.2935152).
- Zheng, Z., Y. Yang, J. Liu, H. N. Dai, and Y. Zhang. 2019. "Deep and Embedded Learning Approach for Traffic Flow Prediction in Urban Informatics." *IEEE Transactions on Intelligent Transportation Systems* 20: 3927–3939.
- Zhu, Z., B. Peng, C. Xiong, and L. Zhang. 2016. "Short-term Traffic Flow Prediction with Linear Conditional Gaussian Bayesian Network." *Journal of Advanced Transportation* 50: 1111–1123.
- Zhu, Z., L. Tang, C. Xiong, X. M. Chen, and L. Zhang. 2018. "The Conditional Probability of Travel Speed and its Application to Short-Term Prediction." *Transportmetrica B: Transport Dynamics* 7 (1): 684–705. doi:[10.1080/21680566.2018.1467289](https://doi.org/10.1080/21680566.2018.1467289).

Appendix

Algorithm 1 Training our model and predict

1. Input:
 - (1) Historical directed weighted topological graphs $\{G_1, G_2, \dots, G_n\}$;
 - (2) Recent time lags l_r , hourly time lags l_h , daily time lags l_d , and weekly time lags l_w ;
 - (3) Length of long-term input sequences n_r, n_h, n_d, n_w ;
 - (4) Length of short-term input sequences m_r, m_h, m_d, m_w ;
- //Step 1: Feature organisation
2. For $t = n$ to 1
3. $G_r^1 = [G_{t-n_r}, \dots, G_{t-2l_r}, G_{t-l_r}]$; $G_r^2 = [G_{t-m_r}, \dots, G_{t-2l_r}, G_{t-l_r}]$;
4. $G_h^1 = [G_{t-n_h}, \dots, G_{t-2l_h}, G_{t-l_h}]$; $G_h^2 = [G_{t-m_h}, \dots, G_{t-2l_h}, G_{t-l_h}]$;
5. $G_d^1 = [G_{t-n_d}, \dots, G_{t-2l_d}, G_{t-l_d}]$; $G_d^2 = [G_{t-m_d}, \dots, G_{t-2l_d}, G_{t-l_d}]$;
6. $G_w^1 = [G_{t-n_w}, \dots, G_{t-2l_w}, G_{t-l_w}]$; $G_w^2 = [G_{t-m_w}, \dots, G_{t-2l_w}, G_{t-l_w}]$;
7. $G = [G_r^1, G_r^2, G_h^1, G_h^2, G_d^1, G_d^2, G_w^1, G_w^2]$;
8. Insert (G, G_t) into \mathcal{L} ;
9. End
- //Step 2: Model Training and optimisation
10. Initialise model parameters randomly as \emptyset and total loss as $Loss = 0$;
11. Repeat
12. For each (G, G_t) in \mathcal{L}
13. Obtain $RTX(t)$ from G_t ;
14. Perform forward propagation to compute $\widehat{RTX}(t)$;
15. Compute error $L(\emptyset) = loss(RTX(t), \widehat{RTX}(t))$;
16. $Loss += L(\emptyset)$;
17. Perform backward propagation to compute $\Delta\emptyset$;
18. $\emptyset = \emptyset + \Delta\emptyset$;
19. End
20. Until $Loss < \sigma$;
21. Output: Trained Model Λ

Algorithm 2 Prediction at Time t Using Our Model

1. Input:
 - (1) Trained Model Λ ;
 - (2) $G_r^1 = [G_{t-n_r}, \dots, G_{t-2l_r}, G_{t-l_r}]$; $G_r^2 = [G_{t-m_r}, \dots, G_{t-2l_r}, G_{t-l_r}]$;
 - (3) $G_h^1 = [G_{t-n_h}, \dots, G_{t-2l_h}, G_{t-l_h}]$; $G_h^2 = [G_{t-m_h}, \dots, G_{t-2l_h}, G_{t-l_h}]$;
 - (4) $G_d^1 = [G_{t-n_d}, \dots, G_{t-2l_d}, G_{t-l_d}]$; $G_d^2 = [G_{t-m_d}, \dots, G_{t-2l_d}, G_{t-l_d}]$;
 - (5) $G_w^1 = [G_{t-n_w}, \dots, G_{t-2l_w}, G_{t-l_w}]$; $G_w^2 = [G_{t-m_w}, \dots, G_{t-2l_w}, G_{t-l_w}]$;
2. $G = [G_r^1, G_r^2, G_h^1, G_h^2, G_d^1, G_d^2, G_w^1, G_w^2]$;
3. Predict $\widehat{RTX}(t) \leftarrow \Lambda(G)$;
4. Output: $\widehat{RTX}(t)$
