Complex & Intelligent Systems (2024) 10:3305–3317
https://doi.org/10.1007/s40747-023-01324-9

ORIGINAL ARTICLE

A Multi-Scale Residual Graph Convolution Network with hierarchical attention for predicting traffic flow in urban mobility

Jiahao Ling[1] · Yuanchun Lan[1] · Xiaohui Huang[1] · Xiaofei Yang[2]

Received: 12 April 2023 / Accepted: 13 December 2023 / Published online: 29 January 2024

Abstract

Accurate prediction of traffic flow is essential for optimizing transportation resource allocation and enhancing urban mobility efficiency. However, traffic data generated daily are vast and complex, involving dynamic and intricate changes in the traffic road network and traffic flow. Therefore, real-time and accurate prediction of traffic flow is a challenging task that requires modeling the intricate spatial–temporal dynamics of traffic data. In this paper, we propose a novel approach for traffic flow prediction, based on a Multi-Scale Residual Graph Convolution Network with hierarchical attention. First, we design a novel encoder–decoder with multi-independent channels to capture traffic flow information from different time scales and diverse temporal dependencies. Second, we employ a coupled graph convolution network with residual graph attention to dynamically learn the varying spatial features among and within traffic stations. Third, we utilize channel attention to fuse the multi-scale spatial–temporal dependencies and accurately predict traffic flow. We evaluate the proposed approach on multiple benchmark datasets, and the experimental results demonstrate its superior performance compared to state-of-the-art approaches in terms of various metrics.

Keywords Spatial–temporal · Traffic forecasting · Periodicity · Multivariate time series

Introduction

Accurate traffic flow prediction is a significant research topic in the field of urban computing [1], as it can optimize transportation resource allocation [2] and improve the efficiency of intelligent transportation systems [3]. The immense amount of traffic data generated daily holds crucial insights into the long-term evolution of traffic dynamics, which are integral to future traffic management and planning. Despite the potential benefits of utilizing traffic flow data, forecasting traffic flow accurately in real-time remains a complex and multifaceted task [4, 5].

Given the paramount importance of accurate traffic flow prediction in optimizing intelligent transportation systems, a substantial research effort has been devoted to this field in recent years [1, 6]. Early research endeavors were focused on predicting traffic flow at the city-level grid-based map [7–9] utilizing convolutional neural networks (CNN) [10] or recurrent neural networks (RNN) [11] to capture the intricate spatial–temporal dependencies inherent in traffic flow data. However, grid-based partitioning methods are limited in their ability to capture the intricate spatial dependencies of non-Euclidean data [12], such as traffic flow. With the emergence of new and advanced techniques, recent research has witnessed a shift towards utilizing graph construction-based methods [13], such as Graph Convolutional Networks (GCN) or Graph Neural Networks (GNN), to capture the spatial–temporal dependence of urban traffic flow [2, 3, 14]. This shift has been driven by the ability of these methods to handle physical and semantic information on road networks, providing a more comprehensive understanding of traffic dynamics. Moreover, a recent trend has emerged wherein

✉ Xiaohui Huang
hxh016@gmail.com

Jiahao Ling
2021068081200015@ecjtu.edu.cn

Yuanchun Lan
lyc28688@163.com

Xiaofei Yang
xiaofeiyang@um.edu.mo

[1] Department of Information Engineering, East China Jiaotong University, No. 808, Shuanggang East Street, Nanchang 330000, Jiangxi, China

[2] Faculty of Science and Technology, University of Macau, Macao 519000, China

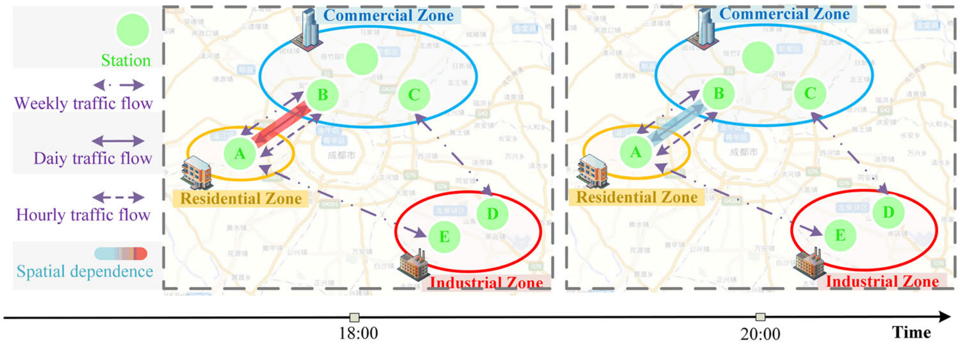

Fig. 1 A simple example that illustrates the dynamic multi-periodic spatial–temporal dependence of traffic flow in areas segmented into residential, commercial, and industrial zones. The spatial–temporal dependence of different zones may vary over time (e.g., station A and station B) and traffic between zones may exhibit regularity in hourly, daily, and weekly patterns

multivariate time series forecasting is employed to analyze multi-periodic temporal patterns of traffic data [7, 14]. Despite these advancements, the intensive spatial–temporal dynamics of traffic data and the variable multi-periodic temporal patterns found in cities pose significant challenges for these methods, making it difficult to dynamically learn from multiple perspectives of traffic flow. As a result, accurate traffic flow prediction faces two major technical challenges, which are:

- *Dynamic spatial dependence* One of the key technical challenges in predicting traffic flow is to capture the dynamic spatial dependencies present in traffic data. The traffic flow at a particular station may vary dynamically over time, as shown in Fig. 1. For instance, during the peak travel period of 18:00, residential zones may have a high spatial dependency with commercial areas (i.e., station A and station B), but this dependency may greatly decrease at 20:00. Accurately modeling such complex spatial dependencies is critical for developing precise traffic flow prediction models. Despite the efforts to address these challenges through the use of advanced machine learning techniques [15], there is still room for improvement and further research to fully understand and effectively model the complex spatial dynamics of traffic data.
- *Multi-periodic temporal patterns* Another key technical challenge in predicting traffic flow is extracting the mutiperiodic temporal patterns of traffic flow data [16]. The ability to account for the dynamic interplay of historical traffic data over different periods of time, which are characterized by varying temporal scales, such as hourly, daily, and weekly patterns, is critical for effective modeling of traffic flow. Specifically, as depicted in Fig. 1, closely spaced zones (e.g., residential and commercial zones), may demonstrate strong periodicity in three-time scales, while more distant zone (e.g., industrial zone) may exhibit primarily weekly patterns. Such complexities require the development of sophisticated techniques capable of capturing the intricate temporal dependencies and patterns that underlie traffic flow data. However, many existing methods do not explicitly account for and integrate the features of different periods of traffic flow.

To address the above challenges, we propose a Multi-Scale Residual Graph Convolution Network with hierarchical attention to analyze traffic flow data comprehensively. Firstly, we partition the traffic flow data into three temporal scale based on hourly, daily, and weekly intervals, respectively. Subsequently, we introduce a novel encoder–decoder architecture to capture the intricate spatial–temporal dependencies of the data. Specifically, we design three independent channel integrate coupled graph convolution and residual graph attention to establish a relationship matrix that captures the dynamic spatial dependencies among the stations. Furthermore, our method employs a residual channel attention mechanism to fuse the spatial–temporal dependencies across different temporal scales. Our main contributions are as follows:

- We propose MSRGCN, a novel framework that can effectively model the spatial, temporal, and semantic correlations among roads in a traffic network by analyzing historical traffic flow data across multi-periodic temporal patterns.
- We design a residual graph attention with coupled graph convolution network that capture edge weight and node weight at each time interval to better reflect the dynamic traffic scenarios in the real world.
- We propose a residual channel attention that can integrate and fuse the traffic features extracted at multiple scales and calculate the final prediction result.
- We conduct extensive experiments with real-world data from multiple cities and provide evidence of the superiority of our proposed MSRGCN over state-of-the-art approaches.

The remainder of this paper is organized as follows. We review related work in “Related work” section. We describe the definition used in this work in “Preliminaries” section.

The MSRGCN design is elaborated and evaluated in “Multi-Scale Residual Graph Convolution Network, Experiments” sections, respectively. Finally, “Conclusion and prospect” concludes this paper.

Related work

In this section, we review the literature related to our work from the perspectives of traffic flow prediction and multivariate time series forecasting.

Traffic flow prediction

Early traffic prediction models, such as grid-based methods, utilize a grid map to perform the task. For instance, Zhang et al. [8] proposed the two-phase DeepSTD deep learning framework to uncover spatial–temporal disturbances and predict citywide traffic flow. Zhang et al. [9] designed a multi-core-based clustering method to delineate traffic map sub-regions and proposed the mutual-transition-aware co-prediction framework to capture spatial–temporal transformation patterns of traffic demand. However, these models compromise the natural topological properties of the traffic network, making them unsuitable for dynamic and volatile real-life traffic prediction scenarios.

Recent research has demonstrated that graph-based data structures are more effective in representing non-Euclidean distance data, such as traffic road networks. Consequently, graph neural networks have become a popular approach in traffic flow prediction. For instance, Peng et al. [17] propose a long-term traffic flow prediction method based on dynamic graphs and use reinforcement learning to generate dynamic graphs that enable stable and effective long-term predictions of traffic flow. Lv et al. [18] present a Temporal Multi-Graph Convolutional Network (T-MGCN) that jointly models the spatial, temporal, semantic correlations, and various global features in the road network. Ali et al. [14] propose a unified dynamic deep spatial–temporal neural network model, based on graph convolutional neural networks and long short-term memory, named as (DHSTNet), that can simultaneously predict crowd flows in every region of a city. Ye et al. [2] develop a layer-wise coupling mechanism and self-learning adjacency matrices to capture multi-level spatial dependence and temporal dynamics simultaneously.

These methods have improved the accuracy and interpretability of predictions to some extent, but they may not fully consider the analysis of the periodicity of time series. To address this limitation and ensure the effective capture of short-term and long-term dependencies, as well as adaptability to dynamic traffic patterns, our proposed approach involves the extraction of specific time intervals from hourly, daily, and weekly scales. This refined data organization strategy enables our proposed method to comprehensively learn the temporal influences on traffic flow data, encompassing both fine-grained short-term variations and broader long-term trends. By carefully selecting relevant time intervals from different temporal scales, our proposed model gains a more comprehensive understanding of the temporal dynamics in traffic flow data.

Multivariate time series forecasting

Multivariate Time Series (MTS) data is a prime example of spatial–temporal data [19], comprising various interrelated time series with different scales. The precise and efficient forecasting of MTS is of great significance in diverse fields, including transportation, energy, and economics [7, 20, 21]. Therefore, this has been an ongoing area of research. To address this challenge, several innovative techniques have been proposed in recent years. For instance, Cao et al. [22] have proposed a Spectral Temporal Graph Neural Network (StemGNN) that captures intra-series temporal correlations and inter-series correlations simultaneously. Du et al. [23] have introduced Bi-directional Long Short-Term Memory networks (Bi-LSTM) to learn long-term dependency and hidden correlation features of multivariate temporal data adaptively. Wu et al. [24] have presented a general graph neural network framework that automatically extracts the uni-directed relations among variables through a graph learning module. In the context of traffic flow prediction, Yang et al. [25] have proposed Spatial–Temporal information and Traffic Pattern Similarity information (STTPS), which considers the impact of temporal factors on traffic from the perspective of seasonal and super-recent factors. Additionally, Zhu et al. [26] have utilized an LSTM-based variational autoencoder to capture the multi-scale dependence of time series.

However, most of the existing methods that capture spatial patterns employ static relationship matrices, which neglect the dynamic nature of the interactions among and within stations over time, limiting their ability to capture deeper spatial–temporal features of traffic flow.

We present a pioneering approach that effectively combines multi-independent channels with residual graph attention mechanisms, thereby capturing dynamic spatial dependencies at each time interval. Through the integration of information from multiple time steps, our proposed model achieves a comprehensive understanding of traffic dynamics, leading to more resilient predictions that encompass both short-term fluctuations and long-term trends. Furthermore, we leverage the benefits of residual channel attention to enhance and refine spatial–temporal features extracted from these multi-independent channels, further improving the model’s predictive capabilities.

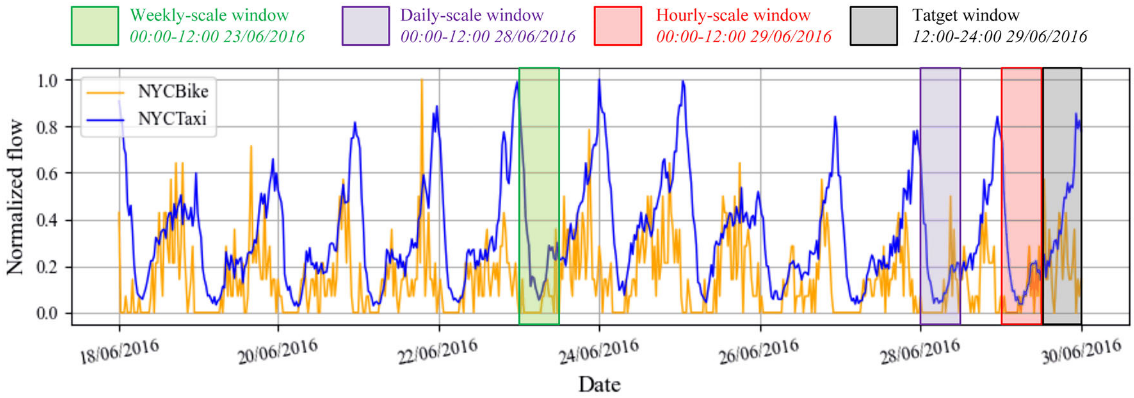

Fig. 2 Normalized traffic flow time series on the NYCBike and NYCTaxi datasets from June 18–30, 2016. The windows of predicted traffic and three different temporal periodic patterns (hourly, daily and weekly) are marked by colored rectangles, respectively

Preliminaries

In this section, we introduce some important notations and definitions and formalize the traffic flow prediction problem as below.

Definition 1 (*Traffic network graph*) The traffic network is modeled as a directed graph $G(V, E)$, where the nodes V correspond to the stations and the edges E indicate the traffic flow between two stations. The feature vector of each node consists of the historical pick-up and drop-off flow at that station.

Definition 2 (*Traffic flow data*) The traffic flow data (e.g., volume) on the traffic network graph at time t is denoted as $X^h_t \in \mathbb{R}^{N\times c}$, $X^d_t \in \mathbb{R}^{N\times c}$ and $X^w_t \in \mathbb{R}^{N\times c}$, which represent hourly, daily and weekly scales of traffic flow data, where N is the number of nodes in the graph and c is the number of features. A traffic flow data instance consists of an input part and an output part, which are defined as $X_t = [X^h_{t-L:t}, X^d_{t-D-L:t-D}, X^w_{t-W-L:t-W}]$, $X_t \in \mathbb{R}^{3L\times N\times c}$ and $Y_t = X_{t:t+M}$, $Y_t \in \mathbb{R}^{M\times N\times c}$ respectively. The input part is a sequence of L historical traffic flow data for each scale and D and W represent the number of time intervals per day and per week respectively. The output part is a sequence of M predicted traffic flow data.

Different from traditional methods that denote the traffic flow data (e.g., volume) on the traffic network graph at time t as $x_t \in \mathbb{R}^{N\times c}$, we split the input traffic flow into three cycle types. This is because, as Fig. 2 shows, the traffic data in three different period windows (hourly, daily and weekly) related to the target window have a certain similarity when we study the traffic characteristics of the target window.

Definition 3 (*Relationship matrix*) The relationship matrices under different temporal scales are initialized by using the traffic flow data and calculating the similarity of the historical traffic flow among the stations [2] as the weight of the edges in the graph. For a given τ time intervals, starting from the initial time t_0, a function f_A maps traffic status signals from three periodic scales to three different relationship matrices, which can be expressed as:

$$[A^h_{(0)}, A^d_{(0)}, A^w_{(0)}] = f_A(X^h_{t_0+W:t_0+W+\tau}, X^d_{t_0+W-D:t_0+W-D+\tau}, X^w_{t_0:t_0+\tau}), \tag{1}$$

where $A^h_{(0)} \in \mathbb{R}^{N\times N}$, $A^d_{(0)} \in \mathbb{R}^{N\times N}$, and $A^w_{(0)} \in \mathbb{R}^{N\times N}$ can be used to perform graph convolution operations on graph G to learn spatial dependencies at hourly, daily, and weekly scales.

We describe the specific operation process of function f_A as follows. We first apply singular value decomposition (SVD) to traffic data $X^h_{t_0+W:t_0+W+\tau}$, $X^d_{t_0+W-D:t_0+W-D+\tau}$, $X^w_{t_0:t_0+\tau}$ to obtain multiple low-rank submatrices [27]. Taking the weekly scale as an example, we can formulate this as:

$$X^w_{t_0:t_0+\tau} = U_{X^w}\Sigma_{X^w}V^{\mathbb{T}}_{X^w}, \tag{2}$$

where U_{X^w}, Σ_{X^w} and V_{X^w} is the low-rank submatrices from $X^w_{t_0:t_0+\tau}$, and U_{X^w}, V_{X^w} are represents the temporal-based and spatial-based submatrices, respectively. To reduce the dimensionality and describe the relationships among stations as accurately as possible, we filter out redundant information from the spatial-based submatrix V_{X^w}. We use a method based on Gaussian kernel to calculate the similarity of row i and row j of V_{X^w} as their edge weight values in the adjacency matrix, which can be formulated as:

$$\hat{A}^w(i, j) = exp\left(\frac{\|V(i, :) - V(j, :)\|^2}{\epsilon}\right), \tag{3}$$

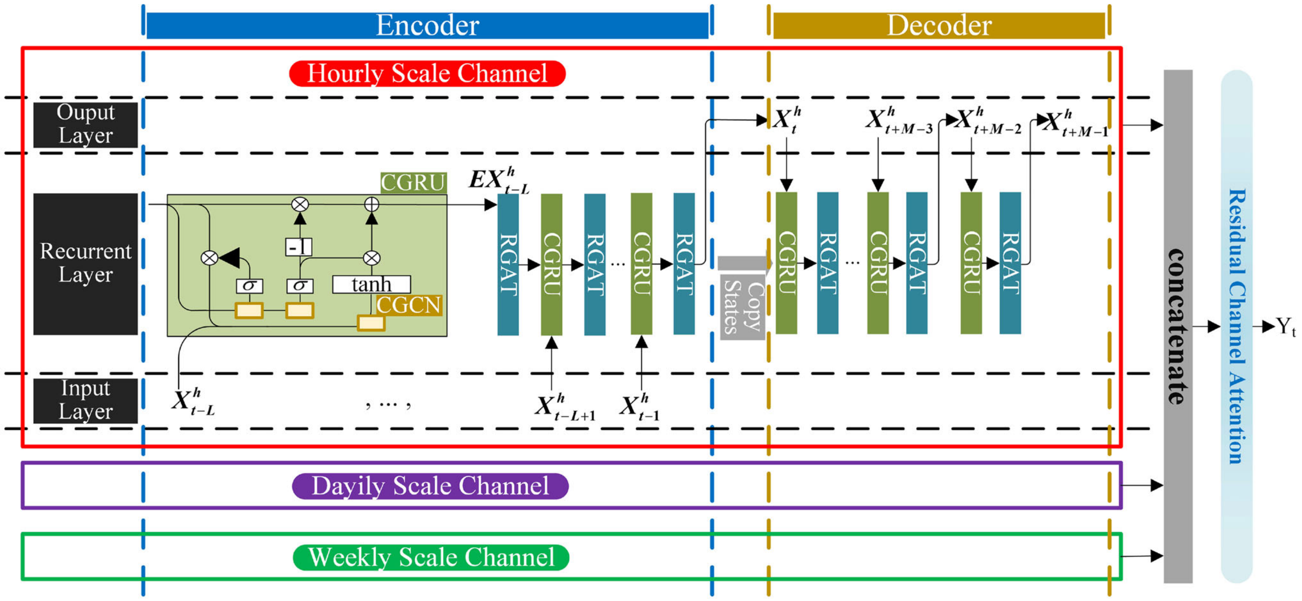

Fig. 3 Overall architecture of MSRGCN

where ϵ is the standard deviation. In practical operation, $\hat{A}^w \in \mathbb{R}^{N\times N}$has a large number of nodes may reduce the system efficiency. In contrast to the traditional method [28], which retains all the elements in the relationship matrix irrespective of their values, we suggest to initialize the relationship matrix at the initial time t_0 by discarding the elements that are negligible and do not influence the node connections. This way, we can maintain the sparsity of the relationship matrix and decrease the computational cost. This procedure can be formulated as:

$$A^w_{(0)} = Max(0, D^{-1}\hat{A}^w), \tag{4}$$

where D is a diagonal matrix such that $D(i,i) = \Sigma_j \hat{A}^w(i,j)$. The initialization process of $A^h_{(0)}$ and $A^d_{(0)}$ are similar to $A^w_{(0)}$.

Definition 4 (*Traffic flow prediction problem*) The traffic prediction problem is formulated as learning a function f_P from a large number of traffic flow data. The function f_P maps $3L$ historical traffic status signals from three periodic scales of the current time t to future traffic status signals from time t to $t + M - 1$, which can be expressed as:

$$Y_t = f_P(X_t, G). \tag{5}$$

Multi-Scale Residual Graph Convolution Network

This section outlines the design of MSRGCN for precise traffic flow prediction. It covers the overall architecture, the Coupled Graph Convolution Network (CGCN), the Residual Graph Attention (RGAT) in the recurrent layer, and the Residual Channel Attention (RCAT) for fusing temporal features and generating the final prediction.

The framework of MSRGCN

MSRGCN is a novel framework for traffic flow prediction that leverages three independent channels to capture the spatial–temporal patterns of traffic flow at different periodic: hourly scale channel, daily scale channel, and weekly scale channel. Each channel employs the same network structure, but differs in the input graph structures and time series that it utilizes. For instance, as illustrated in Fig. 3, the hourly scale channel has three layers: input layer, recurrent layer, and output layer. The input layer takes as input the traffic flow of L consecutive time series $X^h_{t-L:t} = [X^h_{t-L}, X^h_{t-L+1}, ..., X^h_{t-1}]$and predicts as output the traffic flow of M consecutive time series $X^h_{t:t+M} = [X^h_t, X^h_{t+1}, ..., X^h_{t+M-1}]$. The recurrent layer comprises two novel components: CGCN and RGAT. These components are designed to dynamically update the relationship matrix and learn node embedding at each time interval. As illustrated in Fig. 3, the recurrent layer first employs coupled graph recurrent unit (CGRU) to obtain the aggregated edge-weighted traffic flow state EX^h_{t-L}, and then feeds it into RGAT to dynamically learn node weights. After obtaining the predictions of three channels, $X^h_{t:t+M}$, $X^d_{t:t+M}$ and $X^w_{t:t+M}$, RCAT is applied to integrate and fuse the traffic features extracted at three scales and calculate the final prediction result Y_t.

The recurrent layer

Recurrent operations can effectively learn semantic associations across time sequences and capture temporal correlations [29]. Convolution operations can effectively learn local dependency and maintain shift invariance and capture spatial correlations [30]. Therefore, we use recurrent units with GCN to capture the spatial–temporal features of traffic flow. However, most existing recurrent units based on GCN use fixed relationship matrices in graph convolution, which may overlook the dynamic variation of dependencies among and within stations in actual traffic scenarios. Moreover, a fixed relationship matrix may not be able to adapt to the spatial changes of traffic flow at different temporal periodic scales. To address this issue, we design a coupled graph convolutional network and residual graph attention in the recurrent layer, which can dynamically computes the edge weight and node weight for each time interval.

Coupled graph convolution network One of the difficulties in traffic prediction is to capture the temporal variations of the spatial correlations among stations that may occur in traffic data at different time intervals [31], such as morning, afternoon, evening, night, etc. To tackle this difficulty, we first introduce a graph convolution method that employs a coupling mapping mechanism to learn the relationship matrix among the stations [2]. The relationship matrix reflects the similarity and influence of the traffic flow patterns among the stations. Taking the hourly scale channel as an example, Fig.4 illustrates the structure of CGCN. We incorporate the traffic flow characteristics $X^h_t = Z^h_{(0)}$ within each time interval and the initial relationship matrix $A^h_{(0)}$ as the input for graph convolution. Each layer of graph convolution can extract the corresponding station feature $Z^h_{(i)}$ and relationship matrix $A^h_{(i)}$ under the hourly scale. This can be expressed as:

$$Z^h_{(i)} = \sum_{k=0}^{K} \left(A^h_{(i-1)}\right)^k Z^h_{(i-1)} \theta^k_{(i-1)}, i = 1, 2, \ldots, l \tag{6}$$

where l represents the total number of convolution layers, and A^h_{i-1} is relationship matrix representing of $i-1$ layer, which can be formulated as:

$$A^h_{(i-1)} = w_{(i-2)} \left(U^h_{(i-2)} \left(V^h_{(i-2)}\right)^{\mathbb{T}}\right) + b_{(i-2)}. \tag{7}$$

$U^h_{(i-2)}$ and $V^h_{(i-2)}$ are low-rank submatrices [27] obtained by $A^h_{(i-2)}$ through SVD, and $w_{(i-2)}, b_{(i-2)}$ are learnable parameters in the fully connected layer. To aggregate $Z^h_{(1:l)}$ across different layers, we need to assess the attention scores $\beta^h_{(i)}$ of the relationship matrix for each layer, which can be expressed as follows:

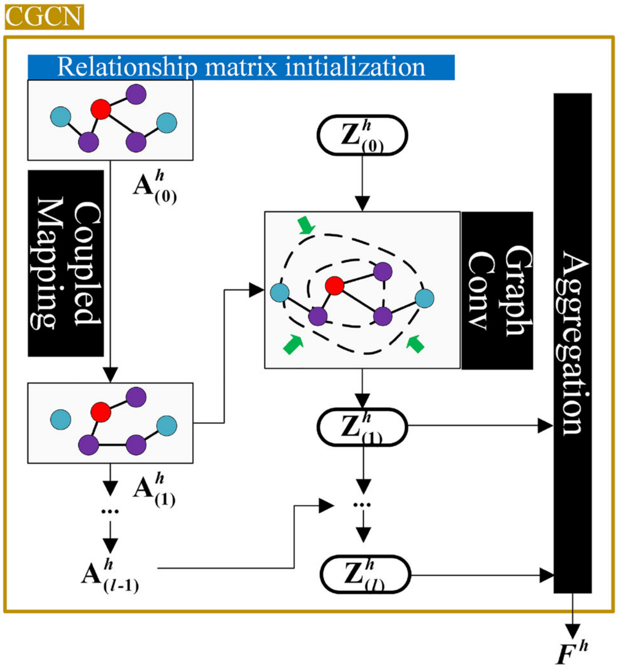

Fig. 4 The framework of CGCN within the hourly scale channel

$$F^h = \sum_{i=1}^{l} \beta^h_{(i)} Z^h_{(i)}, \tag{8}$$

$$\beta^h_{(i)} = \frac{exp(Z^h_{(i)} w_\beta + b_\beta)}{\sum_{i=1}^{l} exp(Z^h_{(i)} w_\beta + b_\beta)}, \tag{9}$$

where w_β and b_β are learnable parameters in the fully connected layer, F^h is the aggregated feature expression of convolution, as output of CGCN.

Residual graph attention Traffic flow prediction also faces the challenge of handling the diverse transportation modes that may vary across different time intervals [32], such as peak hours, off-peak hours, weekends, holidays, etc. To address this challenge, we adopt RGAT to learn adaptive node-specific weights and embeddings that capture the dynamic traffic flow patterns of each node. As shown in Fig. 5, for the RGAT in the hourly channel, we input the edge-weighted traffic flow state EX^h_t that aggregates the traffic information from different edges into RGAT, and calculate the attention scores α^h by using the scaled dot product method, which can be formulated as:

$$\alpha^h_{i,j} = \frac{\left[w^h_q \left(EX^h_{t,i} \| e^h_i\right)\right] \otimes \left[W^h_k \left(EX^h_{t,j} \| e^h_j\right)\right]}{\sqrt{d_a}}, \tag{10}$$

where $\alpha^h_{i,j}$ denotes the attention score between station i and station j on the hourly scale; e^h_i is the randomly initialized node embedding of station i on the hourly scale; $EX^h_{t,i}$ is the edge-weighted traffic flow feature of station i at time interval

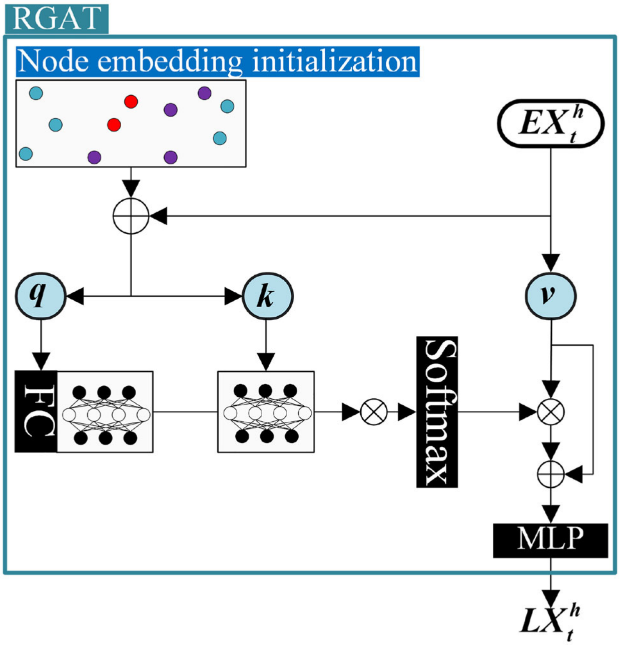

Fig. 5 The framework of RGAT within the hourly scale channel

t that aggregates the traffic information from different edges; $\parallel$ and $\otimes$ represent the concatenation operation and the inner product operation respectively; w_q^h and w_k^h are the learnable parameters of query and key; d_a is the dimension of query and key. After obtaining the attention score, we compute a weighted sum of the correlations of all stations to obtain the latent state LX_t^h, which incorporates both edge-weighted and node-weighted information. The formula is as follows:

$$LX^h_{t,j} = \sum soft\max\left(\alpha^h_{i,j}\right) EX^h_{t,j} + EX^h_{t,j} \tag{11}$$

In the hourly scale channel, the recurrent layer in the encoder and decoder uses the output latent states LX^h of RGAT to capture the time series features. Moreover, the output layer in the decoder consists of the latent states $X^h_{t:t+M} = LX^h_{t:t+M}$. The daily scale channel and the weekly scale channel follow the same operation logic as the hourly scale channel.

Residual channel attention

Traffic flow is a complex spatial–temporal phenomenon that exhibits different periodic patterns at different time scales, such as hourly, daily, and weekly. These patterns reflect the influence of various factors, such as traffic demand, road network structure, weather conditions, and special events. Therefore, to obtain accurate and reliable traffic flow predictions, it is necessary to fuse the traffic features extracted from these time scales in an effective and efficient way. To achieve this, we propose a novel P-layers dynamic residual channel attention mechanism that can adaptively assign different weights to the traffic features from each time scale based on their relevance and importance for the prediction task. The dynamic residual channel attention mechanism can also enhance the feature representation by adding residual connections between the input and output channels, which can facilitate the information flow and alleviate the gradient vanishing problem. This can be formulated as:

$$\hat{Y}_t^{(p+1)} = sigmoid(w^{(p)} f_{ap}(\hat{Y}_t^{(p)})) \otimes \hat{Y}_t^{(p)} + \hat{Y}_t^{(p)} \tag{12}$$

where $\hat{Y}_t^{(0)} = X^h_{t:t+M} \parallel X^d_{t:t+M} \parallel X^w_{t:t+M}$, $\hat{Y}_t^{(0)} \in \mathbb{R}^{3\times M\times N\times D}$ denotes the concatenation of the outputs from the hourly, daily, and weekly scale channels; f_{ap} represents the average pooling operation and $w^{(p)}$ denotes the learnable parameters of the fully connected network of the p-th layer. After applying adaptive weighting P times to the results of the three scale channels, we obtain the final output by combining the results of the three scale channels, which can be formulated as:

$$Y_t = \sum_{j=1}^{3} \hat{Y}^{(P)}_{j,t}, \tag{13}$$

where $Y_t \in \mathbb{R}^{M\times N\times D}$ represents the final prediction and $\hat{Y}^{(P)}_{j,t}$ denotes the result of the jth scale channel after applying adaptive weighting P times.

Table 1 Summary of the datasets used in the experiments

Dataset	Sensors	Sampling rate (min)	Time steps
NYCTaxi	266	30	4368
NYCBike	250	30	4368
PeMS04	307	5	16,992
PeMS08	170	5	17,856

Experiments

Datasets

The experiments are conducted on four real traffic flow data sets in urban mobility: NYCTaxi,[1] NYCBike,[2] PeMS04, and PeMS08.[3] Table 1 presents a comprehensive summary of the datasets utilized in the experimental analysis. Specifically, the datasets comprise NYCTaxi and NYCBike, with 30-min sampling rates and 4368 time steps each. Additionally, the PeMS04 dataset encompasses 307 sensors, a 5-min sampling

[1] https://www1.nyc.gov/site/tlc/about/data.page.

[2] https://www.citiNYCBike.com/system-data.

[3] http://pems.dot.ca.gov/.

rate, and 16,992 time steps, while the PeMS08 dataset contains 170 sensors, a 5-min sampling rate, and 17,856 time steps. These datasets play a pivotal role in evaluating the proposed method's efficacy in traffic flow prediction tasks. Notably, the prediction targets involve forecasting the next twelve steps of traffic flow, leveraging the information from the preceding twelve steps of traffic signals and the traffic graph.

Preprocessing steps for traffic flow datasets

Prior to conducting our experiments, we preprocessed the datasets to convert them into suitable graph data for model input. The NYCBike dataset is station-based, with each bicycle parking spot serving as a station. In contrast, the NYCTaxi dataset is generated by a station-free system, necessitating the identification of potential stations to effectively capture traffic flow characteristics. To address this issue, we utilized the density peak clustering (DPC) algorithm [33] to identify virtual stations within the NYCTaxi dataset.

Both datasets were segmented into time intervals of 30 min, with traffic data standardized using the Z-score standardization technique prior to training. The feature dimension D of each station was set to 2, representing the number of pick-ups and drop-offs, respectively. Historical time steps and predicted time steps were both set to 12.

Evaluating metrics

To evaluate the proposed method's effectiveness, we use three common metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Pearson Correlation Coefficient (PCC). These metrics are widely accepted in traffic flow prediction and offer a comprehensive evaluation of our approach. RMSE and MAE assess prediction accuracy compared to the ground truth, while PCC measures the correlation between predicted and ground truth values. Multiple metrics ensure a robust and reliable assessment of our results. The formulas for these metrics are as follows:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2} \\ MAE &= \frac{1}{M}\sum_{i=1}^{M}\left|\hat{y}_i - y_i\right| \\ PCC &= \frac{\rho_{y_i \hat{y}_i}}{\sigma_{y_i}\sigma_{\hat{y}_i}}, \end{aligned} \tag{14}$$

where y_i and $\hat{y}_i$ respectively represent the real traffic value and predicted traffic value of the stations, M represents the number of all predicted values, $\rho_{y_i \hat{y}_i}$ represent the covariance of y_i and $\hat{y}_i$, σ_{y_i} and $\sigma_{\hat{y}_i}$ to represent the standard deviations of y_i and $\hat{y}_i$ respectively.

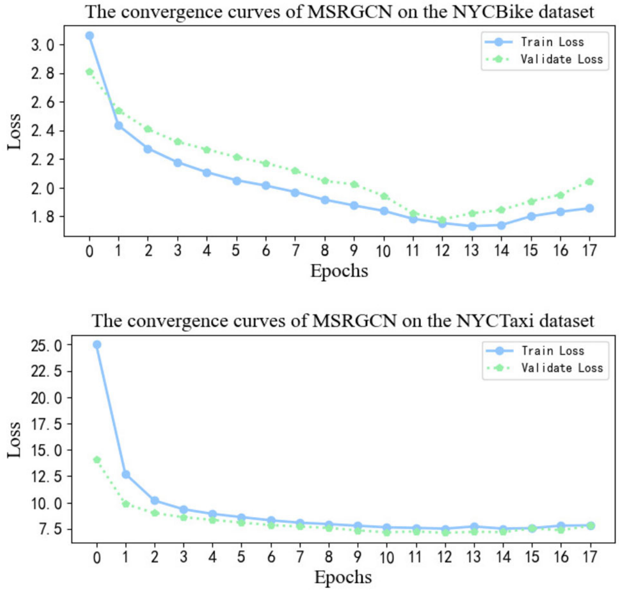

Fig. 6 The convergence curves on the NYCBike dataset and NYCTaxi dataset

Convergence analysis

The convergence behavior of our proposed model on the NYCTaxi and NYCBike datasets is illustrated in the Fig. 6. As can be observed, the training and validation loss values for both datasets reached a plateau after about the 13th epoch, which indicates that the model reached its final convergence state at this point. In order to prevent potential overfitting or underfitting, an early stopping mechanism [34] was employed with a patience of 5. From the figure, it can be seen that the validation loss did not exhibit a significant improvement after the 13th epoch. Therefore, we stopped the training of both models at the 18th epoch to avoid any potential overfitting. It is worth noting that this training strategy is standard in deep learning, and it is essential to ensure that the model can generalize well on unseen data.

Comparison methods

We evaluate our model against several baseline methods that belong to two categories: classic methods and graph-based methods. The classic methods are traditional approaches for traffic prediction, while the graph-based methods are recent advances that leverage graph structures to capture spatial dependencies. We briefly introduce the methods that we use for comparison as follows:

Table 2 Performance comparison of different methods on NYCBike and NYCTaxi datasets

Method	NYCBike			NYCTaxi		
	RMSE	MAE	PCC (%)	RMSE	MAE	PCC (%)
HA	5.2003	3.4617	16.69	29.7803	16.1509	63.39
XGBoost [35]	4.049	2.469	48.6	21.199	11.680	80.77
FC-LSTM [36]	2.813	2.302	56.75	18.071	10.220	86.45
DCRNN [37]	3.205	1.895	72.27	14.792	8.427	91.22
STGCN [38]	3.604	2.761	73.16	22.648	18.455	91.56
STG2Seq [3]	3.984	2.497	51.52	18.045	9.941	86.50
GWNet [27]	3.294	1.991	70.03	13.072	8.103	93.22
STSGCN [39]	2.884	1.753	71.26	10.962	5.829	82.42
MTGNN [40]	2.779	1.659	–	10.947	5.919	–
GTS [41]	2.925	1.779	–	12.751	7.209	–
AST-GCN [42]	–	1.881	–	–	–	–
CCRNN [2]	2.838	1.740	79.34	9.563	5.497	96.48
ESG [43]	2.672	1.612	–	8.975	5.034	–
GMSDR [44]	2.721	1.676	81.07	8.653	4.983	97.11
GraphTS [45]	2.776	1.738	81.52	9.550	5.452	97.21
MSRGCN	**2.101**	**1.258**	**88.86**	**7.932**	**4.428**	**97.64**

bold formatting is used to highlight the top-performing metrics for each method and dataset, making it easier for readers to identify the most effective methods based on the specified evaluation criteria

- *HA* Historical Averaging (HA) is a traditional method based on the mean of past observations.
- *XGBoost* [35] XGBoost is a traditional regression tree-based method.
- *FC-LSTM* [36] FC-LSTM is a traditional method applies RNN with fully connected layer.
- *DCRNN* [37] DCRNN is a classical spatial–temporal method that uses GCN and LSTM for spatial dependence.
- *STGCN* [38] STGCN uses graph and 1-D convolutions to model traffic flow's spatial feature.
- *STG2Seq* [3] STG2seq creates traffic diagram sequences to model long-short term dependence.
- *GWNet* [27] GWNet combines GNN and CNN with diffusion and dilated convolutions.
- *STSGCN* [39] STSGCN utilizes a spatial–temporal synchronous mechanism to capture localized correlations.
- *MTGNN* [40] MTGNN is a framework that extracts uni-directed variable relations and integrates external attributes for multivariate time series forecasting.
- *GTS* [41] GTS learns a probabilistic graph model with GNNs and optimizes mean performance.
- *AST-GCN* [42] AST-GCN is a method that uses GCN to relate graphical nodes in space and time.
- *CCRNN* [2] CCRNN uses a coupling mechanism to update the relationship matrix across layers for multi-step traffic flow prediction.
- *ESG* [43] ESG utilize hierarchical graphs with dilated convolutions for scale-specific correlations.
- *GMSDR* [44] GMSDR is a GRU variant that uses multiple historical inputs per time unit.
- *GraphTS* [45] GraphTS combines GRU and graph attention with multi-graph fusion to fuse sptial-temproal information.

Experimental results and analysis

In this subsection, we conducted a comprehensive comparison of the proposed MSRGCN model with 15 other methods, including classic and state-of-the-art algorithms. The classic methods involved widely adopted models in traffic time series prediction, such as Historical averaging (HA), XGBoost, FC-LSTM, and DCRNN. On the other hand, the state-of-the-art methods we compared MSRGCN with include ESG, GMSDR, GraphTS, and others. To evaluate the performance of these methods, we used four publicly available datasets, NYCTaxi, NYCBike, PeMS04 and PeMS08, and assessed their performance based on evaluation criteria such as RMSE, MAE, PCC.

!b

The results of the experiments are presented in Tables 2 and 3, where it can be observed that MSRGCN outperforms all compared methods for all datasets in most of cases, indicating its superior ability to capture the spatial–temporal dependencies and multi-scale features of traffic data. Table 2 presents a comprehensive evaluation of our proposed method, MSRGCN, against state-of-the-art algorithms, GraphTS and GMSDR, on the NYCBike and NYC-

Table 3 Performance comparison of different methods on PeMS04 and PeMS08 datasets

Method	PeMS04		PeMS08	
	RMSE	MAE	RMSE	MAE
FC-LSTM [36]	40.28	26.76	43.96	30.69
DCRNN [37]	38.12	25.70	27.83	17.86
STGCN [38]	35.55	22.70	27.83	18.02
STG2Seq [3]	38.48	25.20	30.71	20.17
GWNet [27]	39.70	25.45	31.05	19.13
AST-GCN [42]	35.22	22.93	28.16	18.61
STSGCN [39]	33.65	21.19	26.80	17.13
MTGNN [40]	34.66	**20.96**	26.96	16.80
MSRGCN	**32.97**	21.46	**26.14**	**16.57**

bold formatting is used to highlight the top-performing metrics for each method and dataset, making it easier for readers to identify the most effective methods based on the specified evaluation criteria

Taxi datasets. The results indicate a significant improvement of 24.3% in RMSE and 28.9% in MAE compared to GraphTS for NYCBike, and 16.9% in RMSE and 18.7% in MAE compared to GMSDR for NYCTaxi. Moreover, MSRGCN exhibits the highest PCC values of 88.86% and 97.64% on NYCBike and NYCTaxi datasets, respectively, affirming its superior ability to capture the correlation between predicted and actual values when compared to other methods. Table 3 further underscores the effectiveness of MSRGCN as it outperforms all other methods in terms of RMSE on both PeMS04 and PeMS08 datasets. Specifically, on PeMS04, MSRGCN achieves the lowest RMSE of 32.97, surpassing all other approaches. Similarly, on PeMS08, MSRGCN achieves the lowest RMSE of 26.14, further highlighting its accuracy in traffic flow prediction for both datasets. Additionally, while MSRGCN excels in terms of RMSE, it also competes favorably in terms of MAE. Across both datasets, MSRGCN attains some of the lowest MAE values, attesting to its effectiveness in providing accurate traffic flow predictions.

Among the baselines, GMSDR, MTGNN and GraphTS are the closest competitors to our method, but they still lag behind by a large margin. The classic methods such as HA, XGBoost, and FC-LSTM perform poorly compared to the graph-based methods, which demonstrates the importance of modeling the graph structure of traffic networks. The results also show that some methods such as DCRNN and STGCN have high PCC but relatively high RMSE and MAE, which suggests that they can capture the overall trend of traffic demand but fail to predict the exact values accurately.

Ablation study

To assess the impact of each module in MSRGCN on system performance, we conducted ablation studies. The variants of MSRGCN are as follows:

- *Hourly-scale* Analysis based on the hourly scale channel only, excluding daily and weekly scales.
- *Daily-scale* Analysis based on the daily scale channel only, excluding hourly and weekly scales.
- *Weekly-scale* Analysis based on the weekly scale channel only, excluding hourly and daily scales.
- *Seq2seq* Data from all time scales concatenated into a sequence as input for the model.
- *No-RGAT* Dynamic aggregation of the relationship matrix at each time interval, without RGAT modules for adaptive node embedding learning.
- *No-RCAT* Direct summation of flow features from the three channels, without the RCAT module's adaptive weighting."

As shown in Table 4, the results demonstrate that MSRGCN outperforms all other variants in terms of RMSE, MAE, and PCC on both datasets, achieving a reduction in RMSE by 9.1% and 6.1% and an improvement in PCC by 0.78% and 0.53% compared to the best-performing variants on NYCBike and NYCTaxi, respectively. This suggests that using the REAT module to assign different weights to the flow

Table 4 Performance comparison of different variants of MVDGCN on NYCBike and NYCTaxi datasets

Method	NYCBike			NYCTaxi		
	RMSE	MAE	PCC (%)	RMSE	MAE	PCC (%)
Hourly-scale	2.3145	1.3754	84.94	8.4621	4.8356	96.85
Daily-scale	2.3894	1.3952	83.69	8.4975	4.8962	96.80
Weekli-scale	2.2912	1.3446	85.71	8.4196	4.8011	97.13
Seq2seq	2.7346	1.6999	80.81	8.9315	5.3156	93.34
No-RGAT	2.6256	1.6746	82.10	8.8700	5.1491	96.91
No-REAT	2.1551	1.2487	88.51	8.0770	4.4825	97.46
MSRGCN	**2.1009**	**1.2582**	**88.86**	**7.9320**	**4.4284**	**97.64**

bold formatting is used to highlight the top-performing metrics for each method and dataset, making it easier for readers to identify the most effective methods based on the specified evaluation criteria

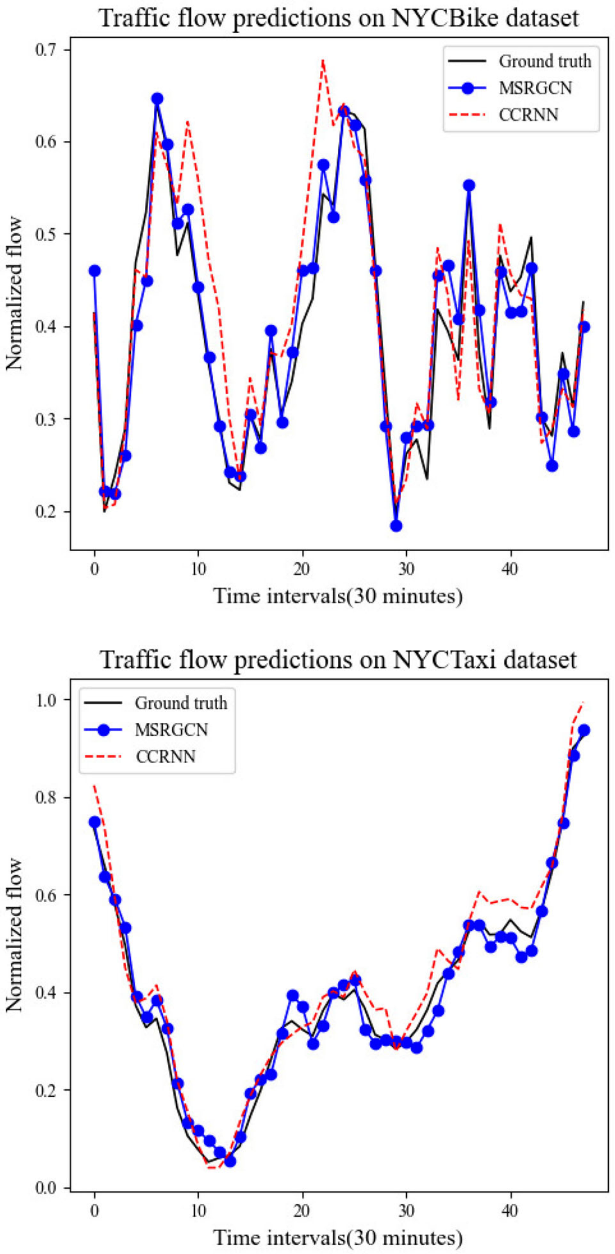

Fig. 7 A case study: MSRGCN and CCRNN predicting next-day traffic flow at a random station on the NYCTaxi and NYCBike datasets

features from different scales is more effective than directly summing them or concatenating them into a sequence. It also suggests that using the RGAT module to learn node embeddings adaptively for each time interval is more effective than dynamically aggregating the relationship matrix only at each time interval. Moreover, it suggests that learning through independent channels for different time scales is more effective than learning through a single channel for a time series sequence.

Case study

To comprehensively assess MSRGCN's performance, we conducted a comparative study involving the advanced method CCRNN, which shares a similar structural framework with MSRGCN. CCRNN, akin to MSRGCN, employs graph convolution techniques based on the underlying graph structure and employs a coupling mechanism for iterative relationship matrix updates during the convolution process. Under these settings, we randomly selected a station from both the NYCBike and NYCTaxi datasets and predicted its traffic flow for the ensuing day. As depicted in the Fig. 7, MSRGCN's predictions demonstrate a superior fit to the ground truth compared to CCRNN, particularly in scenarios characterized by significant traffic fluctuations.

Quantitatively, we computed the RMSE values for both MSRGCN and CCRNN in these instances. For the NYCBike dataset, the RMSE value between MSRGCN's predictions and the ground truth is 2.16, the RMSE value between CCRNN's predictions and the ground truth is 3.34. For the NYCTaxi dataset, these values are 7.56 for MSRGCN and 8.79 for CCRNN. The superior performance of MSRGCN, as evidenced by lower RMSE values, substantiates its effectiveness in this predictive task.

Conclusion and prospect

In conclusion, the proposed Multi-Scale Residual Graph Convolution Network (MSRGCN) with hierarchical attention is a novel approach for accurate traffic flow prediction. It addresses the challenges of modeling the intricate spatial–temporal dynamics of traffic data by employing a multi-channel encoder–decoder, coupled graph convolution network with residual graph attention, and channel attention. The experimental results on multiple datasets demonstrate the superior performance of the MSRGCN compared to existing state-of-the-art approaches in terms of prediction accuracy.

For future work, we plan to incorporate additional data sources such as weather, event schedules, and public transportation data to further improve the accuracy of traffic flow prediction. Furthermore, we plan to apply our approach to other domains that involve complex spatial–temporal data, such as social media analysis and recommender systems, to explore more application possibilities.

Acknowledgements This research was supported by the National Natural Science Foundation of China under No. 62062033, the Natural Science Foundation of Jiangxi Province under Grant No. 20232BAB202018 and the Jiangxi Province Graduate Innovation Special Fund Project YC2022-s495.

Data availability The data that support the findings of this study are openly available at https://www1.nyc.gov/site/tlc/about/data.page https://www.citiNYCBike.com/system-data and http://pems.dot.ca.gov/.

References

1. Xie P, Li T, Liu J, Du S, Yang X, Zhang J (2020) Urban flow prediction from spatiotemporal data using machine learning: a survey. Inf Fusion 59:1–12
2. Ye J, Sun L, Du B, Fu Y, Xiong H (2021) Coupled layer-wise graph convolution for transportation demand prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, no 5, pp 4617–4625
3. Bai L, Yao L, Kanhere SS, Wang X, Sheng QZ (2019) Stg2seq: spatial–temporal graph to sequence model for multi-step passenger demand forecasting. In: Proceedings of the 28th international joint conference on artificial intelligence, pp 1981–1987
4. Krishna BR, Reddy MH, Vaishnavi PS, Reddy SV (2022) Traffic flow forecast using time series analysis based on machine learning. In: Proceedings of the 6th international conference on computing methodologies and communication (ICCMC). IEEE, pp 943–947
5. Han L, Du B, Sun L, Fu Y, Lv Y, Xiong H (2021) Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. Association for Computing Machinery, New York, NY, USA, pp 547–555
6. Jiang R, Yin D, Wang Z, Wang Y, Deng J, Liu H, Cai Z, Deng J, Song X, Shibasaki R (2021) Dl-traff: survey and benchmark of deep learning models for urban traffic prediction. In: Proceedings of the 30th ACM international conference on information and knowledge management, pp 4515–4525
7. Deng J, Chen X, Jiang R, Song X, Tsang IW (2022) A multi-view multi-task learning framework for multi-variate time series forecasting. IEEE Trans Knowl Data Eng 35:7665–7680
8. Zheng C, Fan X, Wen C, Chen L, Wang C, Li J (2020) Deepstd: mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction. IEEE Trans Intell Transp Syst 21(9):3744–3755
9. Zhang Y, Wang B, Shan Z, Zhou Z, Wang Y (2022) Cmt-net: a mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction. In: Proceedings of the fifteenth ACM international conference on web search and data mining. Association for Computing Machinery, New York, NY, USA, pp 1406–1414
10. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
11. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-C (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv Neural Inf Process 28(9):802–810
12. Wang P, Zhang T, Zheng Y, Hu T (2022) A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation. Int J Geogr Inf Sci 36(6):1231–1257
13. James J (2022) Graph construction for traffic prediction: a data-driven approach. IEEE Trans Intell Transp Syst 23(9):15015–15027
14. Ali A, Zhu Y, Zakarya M (2022) Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural Netw 145:233–247
15. Kashyap AA, Raviraj S, Devarakonda A, Nayak K SR, Santhosh KV Bhat SJ (2022) Traffic flow prediction models—a review of deep learning techniques. Cogent Eng 9(1):2010510
16. Fang S, Prinet V, Chang J, Werman M, Zhang C, Xiang S, Pan C (2021) Ms-net: multi-source spatio-temporal network for traffic flow prediction. IEEE Trans Intell Transp Syst 23(7):7142–7155
17. Peng H, Du B, Liu M, Liu M, Ji S, Wang S, Zhang X, He L (2021) Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. Inf Sci 578:401–416
18. Lv M, Hong Z, Chen L, Chen T, Zhu T, Ji S (2021) Temporal multi-graph convolutional network for traffic flow prediction. IEEE Trans Intell Transp Syst 22(6):3337–3348
19. Shao Z, Zhang Z, Wang F, Wei W, Xu Y (2022) Spatial-temporal identity: a simple yet effective baseline for multivariate time series forecasting. In: Proceedings of the 31st ACM international conference on information & knowledge management. Association for Computing Machinery, New York, NY, USA, pp 4454–4458
20. Silva RRC, Caminhas WM, Silva PC, Guimarães FG (2021) A c4.5 fuzzy decision tree method for multivariate time series forecasting. In: Proceedings of the IEEE international conference on fuzzy systems, pp 1–6
21. Zerveas G, Jayaraman S, Patel D, Bhamidipaty A, Eickhoff C (2021) A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. Association for Computing Machinery, New York, NY, USA, pp 2114–2124
22. Cao D, Wang Y, Duan J, Zhang C, Zhu X, Huang C, Tong Y, Xu B, Bai J, Tong J et al (2020) Spectral temporal graph neural network for multivariate time-series forecasting. Adv Neural Inf Process Syst 33:17766–17778
23. Du S, Li T, Yang Y, Horng S-J (2020) Multivariate time series forecasting via attention-based encoder–decoder framework. Neurocomputing 388:269–279
24. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C (2020) Connecting the dots: multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. Association for Computing Machinery, New York, NY, USA, pp 753–763
25. Yang L, Zhang Y, Zuo J (2021) An attention-based spatial-temporal traffic flow prediction method with pattern similarity analysis. In: Proceedings of the 24th IEEE international intelligent transportation systems conference (ITSC), pp 3710–3717
26. Zhu J, Deng F, Zhao J, Ye Z, Chen J (2022) Gaussian mixture variational autoencoder with whitening score for multimodal time series anomaly detection. In: Proceedings of the 17th IEEE international conference on control & automation (ICCA), pp 480–485
27. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the 28th international joint conference on artificial intelligence, pp 1907–1913
28. Guo K, Hu Y, Sun Y, Qian S, Gao J, Yin B (2021) Hierarchical graph convolution network for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 151–159
29. Jing P, Su Y, Jin X, Zhang C (2018) High-order temporal correlation model learning for time-series prediction. IEEE Trans Cybern 49(6):2385–2397

30. Liu Y, Gong C, Yang L, Chen Y (2020) Dstp-rnn: a dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. Expert Syst Appl 143:113082
31. Yuan H, Li G (2021) A survey of traffic prediction: from spatio-temporal data to intelligent transportation. Data Sci Eng 6:63–85
32. Yin X, Wu G, Wei J, Shen Y, Qi H, Yin B (2021) Deep learning on traffic prediction: methods, analysis, and future directions. IEEE Trans Intell Transp Syst 23(6):4927–4943
33. Gao S, Zhou X, Shuai LI (2017) Clustering by fast search and find of density peaks based on density-raito. Comput Eng Appl 208:210–217
34. Lu H, Ge Z, Song Y, Jiang D, Zhou T, Qin J (2021) A temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting. Neurocomputing 427:169–178
35. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, New York, NY, USA, pp 785–794
36. Zhu X, Sobihani P, Guo H (2015) Long short-term memory over recursive structures. In: International conference on machine learning. PMLR, pp 1604–1612
37. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: International conference on learning representations, pp 1–14
38. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 3634–3640
39. Song C, Lin Y, Guo S, Wan H (2020) Spatial–temporal synchronous graph convolutional networks: a new framework for spatial–temporal network data forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 914–921
40. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C (2020) Connecting the dots: multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 753–763
41. Shang C, Chen J (2021) Discrete graph structure learning for forecasting multiple time series. In: Proceedings of international conference on learning representations
42. Chen Z, Wu H, O'Connor NE, Liu M (2021) A comparative study of using spatial–temporal graph convolutional networks for predicting availability in bike sharing schemes. In: 2021 IEEE international intelligent transportation systems conference (ITSC). IEEE, pp 1299–1305
43. Ye J, Liu Z, Du B, Sun L, Li W, Fu Y, Xiong H (2022) Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, pp 2296–2306
44. Liu D, Wang J, Shang S, Han P (2022) Msdr: multi-step dependency relation networks for spatial temporal forecasting. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, pp 1042–1050
45. Liu G, Jiang Y, Zhong K, Yang Y, Wang Y (2023) A time series model adapted to multiple environments for recirculating aquaculture systems. Aquaculture 567:739284

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.