



RT-GCN: Gaussian-based spatiotemporal graph convolutional network for robust traffic prediction

Yutian Liu ^a, Soora Rasouli ^a, Melvin Wong ^a, Tao Feng ^{a,c}, Tianjin Huang ^{b,*}

^a Department of Built Environment, Eindhoven University of Technology, Eindhoven, Netherlands

^b Department of Computer science and Mathematics, Eindhoven University of Technology, Eindhoven, Netherlands

^c Urban and Data Science Lab, Graduate School of Advanced Science and Engineering, Hiroshima University, Japan

ARTICLE INFO

Keywords:

Traffic prediction
Robustness
Missing data
Graph convolutional network
Gaussian distribution

ABSTRACT

Traffic forecasting plays a critical role in intelligent transportation systems (ITS) in smart cities. Travelers as well as urban managers rely on reliable traffic information to make their decisions for route choice and traffic management. However, noisy or missing traffic data poses a problem for accurate and robust traffic forecasting. While data-driven models such as deep neural networks can achieve high prediction accuracy with complete datasets, sensor malfunctions, and environmental effects degrade the performance of such models, as these models rely heavily on precise traffic measurements for model training and estimation. Consequently, incomplete traffic data poses a challenge for robust model design that can make accurate traffic forecasts with noisy/missing data. This research proposes the Robust Spatiotemporal Graph Convolutional Network (RT-GCN), a traffic prediction model that handles noise perturbations and missing data using a Gaussian distributed node representation and a variance based attention mechanism. Through experiments conducted on four real-world traffic datasets using diverse noisy and missing scenarios, the proposed RT-GCN model has demonstrated its ability to handle noise perturbations and missing values and provide high accuracy prediction.

1. Introduction

Traffic forecasting is a crucial aspect of intelligent transportation systems (ITS) as it enables both travelers and network managers to make informed decisions and manage traffic effectively. Over the past decades, the development of big data and computational techniques has led to a shift in traffic prediction models from statistical theory-driven methods such as the Kalman filter method [1] and autoregressive integrated moving average model (ARIMA) [2] to deep learning-based methods, those have emerged as the most widely used approaches in this category due to their strong predictive capabilities [3,4]. More recently, graph convolutional networks (GCN) have gained popularity in traffic prediction as they can capture the topological structure of transportation networks, which leads to higher prediction accuracy [5–7]. Recent studies have also used temporal convolution networks (TCN) [8,9] or combinations of GCNs with time series neural networks such as recurrent neural networks (RNN) [10], long-short term memory neural networks (LSTM) [11], and gated recurrent units (GRU) [12] to extract spatiotemporal features from traffic data.

Although deep learning models have shown remarkable performance in traffic prediction, they can be susceptible to inefficiencies due to over/under-fitting, noise, and data shortages. For example, when the

data is insufficient or corrupted, the model's performance can significantly deteriorate. Noisy and missing traffic information are inevitable due to various factors, such as sensor malfunction and environmental influences such as damage to loop detectors, bad weather, and data management platform failure [13]. Recent studies have shown that neural networks are vulnerable [14,15], even slight data perturbations would significantly degrade the performance of GCN [16]. Any model whose results are sensitive to slight changes in the dataset has limited value. Implementing deep learning methods for traffic prediction in the real world requires models that are not only accurate on clean datasets but also stable when dealing with naturally occurring corruptions. A robust traffic prediction model is still a challenge, and previous works have shown that there is a trade-off between accuracy and robustness [17].

Robust prediction refers to the ability to resist noise and perturbations in the input data while predicting future traffic states [18]. In light of this, many researchers preprocess the input data (i.e. noisy and missing data handling) separately from the model training process. For instance, smoothing methods are employed to clean anomalies in traffic flow data [19], filling methods based on trend-historical data are used to reconstruct missing data [20,21], and data infer algorithms [22,

* Corresponding author.

E-mail address: t.huang@tue.nl (T. Huang).

[23] are utilized to impute missing data. Some other researchers have proposed tailoring deep learning models to incorporate the imputation mechanism into prediction tasks [24–29]. However, data imputation methods have the potential to disrupt crucial latent features, thereby compounding the problem of error accumulation. The generative adversarial network (GAN) based data augmentation method has also been applied in traffic prediction fields, providing an alternative to the data shortage problem [30,31]. Implementing these intelligent methods often requires large computing resources to train the model, leading to suboptimal analysis and forecasting.

In recent years, Bayesian neural networks (BNN) have been employed as promising approaches to address the challenges arising from noisy and sparse traffic data. These studies have demonstrated their value in mitigating these issues and improving the quality of analysis and predictions in the context of traffic analysis [27,32–34]. Nevertheless, when applying BNN for traffic flow prediction with missing data, certain limitations arise. Firstly, BNNs do not inherently deal with missing data, necessitating additional preprocessing steps like imputation techniques or explicit handling of missing values, which introduce complexity and potential biases. Moreover, structural learning in BNN in the presence of missing data can be problematic and can significantly impact the constructed network and thus prediction accuracy. Therefore, developing an inherently robust traffic prediction model and avoiding complex imputation or preprocessing in the presence of abnormal or missing traffic data remains a highly challenging yet significant research problem.

To this end, the authors propose a novel model named Robust Spatiotemporal Graph Convolutional Network (RT-GCN) to predict traffic flow. The RT-GCN is a combination of a variant GCN and GRU models, which can capture the spatiotemporal dependence of traffic data and improve predictive robustness while maintaining accuracy and efficiency. Specifically, rather than representing the hidden states by plain vectors, Gaussian distributions are used to represent the hidden states in the GCN and GRU layers. The motivation for this work is that the Gaussian distribution for hidden representation in the network could potentially negate the effects of noise and data corruption. Additionally, a variance-based attention mechanism is introduced to reduce the propagation of information from perturbed data. Specifically, large weights are assigned to the hidden state with small variance and small weights are assigned to the hidden state with large variance to prevent the impact of abnormal or missing data from propagating to other nodes and time stamps. Furthermore, an additional training strategy called *Batch Random Noise*, inspired by the notion of “data augmentation” [35,36], is incorporated to further enhance the robustness of our model. This strategy introduces on-the-fly random noise to each batch of input data, resulting in an augmented dataset that represents a more comprehensive set of possible perturbed data. To evaluate the performance of the model, extensive experiments are conducted utilizing four real-world datasets, consisting of two raw and two trained clean datasets. Acknowledging the complexity of missing patterns in real-world transportation systems, such as time-consecutive or spatially correlated patterns, distinct types of missing patterns with varying ratios of corrupted data are also considered during the evaluation process. The results show that RT-GCN maintains stable prediction performance especially when the amount of abnormal and missing data is high.

This paper is organized as follows. Section 2 reviews the related literature. Section 3 introduces the preliminaries of the baseline method GCN and GRU. The details of our model RT-GCN are described in Section 4. The experiment results, the analyses and discussion of findings are reported in Section 5. Finally, Section 6 concludes the research and presents the future research directions.

2. Related work

2.1. Traffic state prediction

Traffic state prediction models, such as ARIMA [37], historical average model (HAM) [38], Support vector regression (SVR) [39], and Kalman filtering [40], have been extensively studied. Although these models are effective in modeling complex traffic states, they have limitations when dealing with large and high-resolution traffic data, attributed to complex feature correlations, spatiotemporal dependencies, and nonlinear relationships.

Recent studies have focused on neural networks due to their potential to capture complex, non-linear dependencies of traffic patterns on urban road networks. For instance, researchers have applied convolutional neural Networks (CNN) to capture the time and space relations of spatiotemporal traffic dynamics, by converting the data to images [41]. The CNN models are suitable for Euclidean space, whereas traffic data is typically graph-based [42]. Thus, CNN have limited utility in modeling the topology of urban transport networks. To overcome this limitation, graph convolutional network models have been developed that incorporate the topology of transportation networks into deep neural networks. The principle is treating the road network as a graph, representing the connectivity of roads through the adjacency matrix, and applying the convolution operation on the graph structure. Yu et al. [43] introduced a spatiotemporal graph convolutional network (STGCN) to address the time series prediction challenge of traffic data. By formulating the problem in the context of a graph and leveraging a comprehensive convolutional structure to build the model, they have achieved notable enhancements in training speed and performance. Wu et al. [44] developed a graph neural network architecture Graph WaveNet which captures the hidden spatiotemporal dependency in traffic data through an innovative adaptive matrix and node embeddings. Zhao et al. [45] proposed a temporal graph convolutional network (T-GCN) for traffic prediction, which captures the spatiotemporal features of traffic data. Yu et al. [5] incorporated the prior domain knowledge into the GCNs to simulate traffic flow based on the adjacency matrix.

For modeling the time dependencies of traffic data, RNN, LSTM and GRU have been used. RNN models are well suited for learning time sequence patterns with long time span [46]. LSTM models are designed to address the exploding and vanishing gradient issues of traditional RNN by introducing gated networks [47]. In order to speed up training, the GRU models were proposed with a simpler structure than LSTM [48,49]. With the aim of capturing the spatiotemporal relationships of traffic data, composite architectures such as combining CNN-RNN [50], CNN-LSTM [51], GCN-LSTM [52], GCN-GRU [53] have been proposed and demonstrated superior performance compared to “prime” models [54]. Zheng et al. [55] introduced a graph multi-attention network (GMAN) to forecast traffic conditions for future time steps at different locations. In a one-hour ahead prediction, GMAN demonstrated a substantial improvement in MAE measurement compared to state-of-the-art methods. Song et al. [56] proposed the spatiotemporal synchronous graph convolutional networks (STSGCN) to capture the complex localized spatiotemporal features through a well-designed synchronous modeling mechanism incorporating multiple modules. Li et al. [57] introduced the STAGCN model to predict urban car-hailing destinations. By representing the traffic network as a graph by grid region dividing and utilizing attention mechanisms, the model effectively captures spatiotemporal dependencies in traffic data, outperforming traditional baseline models in destination prediction on real urban datasets. Liu et al. [58] presented an innovative method named ASTGCRN, seamlessly integrating graph convolutional recurrent modules with a global attention component to dynamically acquire graph structures, capture spatial dependencies, model local temporal relationships, and introducing a unique method for extracting global temporal dependencies, thereby demonstrating its promise in

advancing spatiotemporal modeling. Jiang et al. [59] developed a prediction model called PDFormer, which incorporates a spatial self-attention module to capture dynamic spatial dependencies and utilizes graph masking matrices to emphasize short- and long-range spatial dependencies. Although neural network models have demonstrated great capability in predicting traffic states, little attention has been paid to their robustness.

2.2. Vulnerability of graph neural networks

Spatiotemporal graph neural networks have found extensive applications in the field of traffic forecasting [60]. However, in real work, traffic information is often noisy or incomplete, due to sensor malfunction or environmental influences. Several studies have looked at the vulnerability of GNN models when training or predicting on these imperfect data. Szegedy et al. [61] verified that a certain imperceptible perturbation on the test set would cause a high prediction error. Sun et al. [62] argued that GNN models lack interpretability and robustness, and are vulnerable to adversarial attacks. In comparison to non-graph data, the consequences of such attacks on graph data are more severe, due to spatial dependencies and propagation of error throughout the network. Wang et al. [63] pointed out that the effectiveness of GCN relies heavily on the graph convolution operation. Ensuring the robustness of GCN can be challenging due to the fact that perturbations on the edges or features of a small subset of nodes can have ripple effects on nearby nodes, and even propagate through the graph topology structure, thereby compromising the stability of the model. Zügner et al. [64] developed algorithms to target the features of nodes and the graph structure to demonstrate the vulnerability of GCN. Their experiments demonstrated that even when the attack was limited to a small subset of nodes, the performance of GCN continued to degrade. Jin et al. [65] highlighted that the GNN are highly vulnerable to carefully crafted perturbations, their internal structure can be easily manipulated and the prediction can be misguided. The susceptibility of GNN to such attacks has raised concerns about their use in security-sensitive situations.

2.3. Abnormal and missing traffic data

Traffic information abnormalities caused by irregular sampling or detector failures are unavoidable. These abnormalities can have a significant impact on the accuracy and effectiveness of model training and traffic prediction. Smith et al. [13] argued that missing recordings in loop detectors are inevitable to some extent, owing to factors such as construction, communication network issues, and failures in data management systems. Qu et al. [66] pointed out that the missing traffic data in Beijing, China is made by malfunction of sensors and other reasons constituting 10% of the data on average, while some sensors even have higher missing ratios, of up to 25%.

Several methods have been proposed to improve the accuracy of traffic prediction models under abnormal or missing data, which can be broadly classified into two categories.

(1) Preprocessing the input data using imputation algorithms before training the model. Qu et al. [66] demonstrated a probabilistic principal component analysis based filling method for overcoming traffic volume data incompleteness. Li et al. [67] proposed a multistep traffic prediction model with dynamic graph convolution and used an adaptive smoothing method (ASM) to fill in missing sensor data. However, Liu et al. [68] argued that using imputation methods may reduce prediction efficiency. Ge et al. [69] leveraged a tensor decomposition method to impute the missing values existed in the traffic data then fused the external factors (road structure, POI, and social factors) with a spatiotemporal forecasting model to predict the traffic speed. Qu et al. [66] pointed out that while imputation methods are capable of recovering missing data by using mathematical interpolation algorithms that consider neighboring known data, they neglect

the stochastic variation of traffic flow. This can result in suboptimal predictions and reduce prediction efficiency. Cui et al. [22] inferred missing values by a data imputation mechanism within the LSTM structure (LSTM-I) and predicted traffic states by bidirectional LSTM (BDLSM) model. Liu et al. [23] utilized a data recovery algorithm to impute missing data and proposed the use of the GraphSAGE model for forecasting spatially heterogeneous traffic speed within the road network. Cini et al. [21] presented a novel GNN architecture called GRIN to reconstruct missing data within the various channels of a multivariate time series by leveraging message-passing techniques and learning spatiotemporal representations.

(2) Integrating imputation and prediction. Cui et al. [70] introduced a novel neural network architecture called the graph Markov network (GMN). They treated the traffic network as a graph and employed a graph Markov process to model the transition between consecutive time steps, enabling the inference of missing traffic states and the inclusion of spatiotemporal relationships among roadway links. Zhang et al. [26] proposed an approach named customized graph convolutional bidirectional recurrent neural network (GCBRNN), which integrated network-scale data imputation and traffic forecasting into a single integrated task. Liang et al. [28] introduced memory-augmented dynamic graph convolution networks (MDGCN), which were explicitly designed for imputing missing traffic data and forecasting simultaneously. Wang et al. [29] regarded imputation and prediction as two concurrent tasks and trained them sequentially to mitigate the adverse effects of imputation on raw traffic data for prediction while expediting the model training process. Zuo et al. [71] presented GCN-M to address the intricate missing values within the spatiotemporal context. They combined the missing value processing and traffic forecasting tasks, taking into account both local spatiotemporal features and global historical patterns through an attention-based memory network.

Some researchers enhance the generalization and robustness abilities of intelligent models by “creating” additional sample inputs. In the field of traffic prediction, generative adversarial network (GAN) based data generation methods have been applied, providing a way to alleviate the problem of data shortage. Xu et al. [30] proposed a novel deep learning framework called graph embedding generative adversarial network (GE-GAN). The framework utilized graph embedding to generate road traffic states using data from adjacent links and applied GAN to generate real-time road traffic state information based on the spatiotemporal information of selected road segments. Wang et al. [31] mentioned data scarcity and privacy as major obstacles in developing an accurate prediction model. They proposed a data-augmentation-based cellular traffic prediction model (ctGAN-S2S) and showed that the data augmentation submodel was effective in improving the prediction performance. However, the implementation of these methods requires significant computing resources for training the model.

Other recent works adopted Bayesian neural networks (BNNs) to address noisy or sparse traffic data issues. Sun et al. [32] modeled traffic flow between adjacent road segments using Bayesian networks. Comprehensive experiments demonstrated that Bayesian networks are a promising and valid method for traffic prediction, regardless of whether the data is complete or incomplete. Wu and Yu [33] integrated a Bayesian model within a universal traffic forecasting framework, which incorporated uncertainty quantification for prediction. The model outperformed traditional point estimation networks, providing more reliable predictions and demonstrating the ability to extract meaningful representations even in the presence of limited traffic data. Chen and Sun [27] proposed a Bayesian temporal factorization (BTF) model that can effectively handle incomplete and large-scale/high-dimensional spatiotemporal data. The model utilized a proper vector autoregressive (VAR) to capture temporal factor dependencies and reduce the effects of data corruption. Chen and Sun [27] utilized Bayesian inference in conjunction with matrix factorization to capture temporal dependency, resulting in exceptional performance when compared to alternative imputation techniques relying on tensor decomposition. Xia et al. [34]

introduced a multi-view Bayesian spatiotemporal graph neural network (MVB-STNet), which utilized the structural traffic graph and the semantic graph to capture spatial correlations. Extensive evaluations demonstrated its effectiveness under sparse and noisy traffic data. While BNNs and other Bayesian approaches offer advantages for traffic flow prediction, they also have drawbacks. One limitation is their inability to handle missing data inherently, requiring preprocessing steps such as imputation techniques or explicit handling of missing values. These steps can introduce complexity and potential biases. Moreover, BNNs incorporate uncertainty in predictions through Bayesian inference, but estimating uncertainty becomes challenging with missing data, leading to increased uncertainty. Additionally, BNNs rely on prior assumptions, and selecting appropriate priors can be challenging, especially in the presence of missing data, as it can impact prediction accuracy.

Improving the stability of prediction models continues to be a vital and formidable area of focus. This paper aims to introduce a novel and robust traffic prediction model that inherently enhances resilience and accuracy when handling noisy and missing data, eliminating the requirement for imputation techniques or explicit handling of missing values.

3. Preliminaries

Traffic network. In this research, the road network is defined as $\mathcal{G} = (V, E)$, where $V = \{v_1, \dots, v_N\}$ is a set of nodes, and each loop detector is treated as a node, N is the numbers of nodes. E is the set of edges and $e_{i,j}$ is the connection between two nodes. $X \in \mathbb{R}^{N \times T}$ represents the historical traffic information of the N nodes with historical length T . Assuming that the current time is t , the traffic prediction problem aims to learn the mapping function f that is able to predict the traffic data in the future S steps given the past T historical data points:

$$[\hat{X}_{t+1}, \dots, \hat{X}_{t+S}] = f(\mathcal{G}; [X_{t-T+1}, \dots, X_t]), \quad (1)$$

where $[\hat{X}_{t+1}, \dots, \hat{X}_{t+S}] \in \mathbb{R}^{N \times S}$ and $[X_{t-T+1}, \dots, X_t] \in \mathbb{R}^{N \times T}$. \hat{X}_i represents the prediction value and X_i is the actual value of traffic data at node i .

GCN. The GCN model is a powerful tool that operates on graph-structured data. It can represent the connectivity of roads by adjacency matrix, so as to capture the topology features of road networks. The l th layer of the GCN model is $H^{(l)} = [h_1^{(l)}, \dots, h_N^{(l)}] \in \mathbb{R}^{N \times Q^{(l)}}$ and $i \in N$, where $h_i^{(l)}$ is the representation of node i with length $Q^{(l)}$. $Q^{(l)}$ is the dimension of l th hidden layer, if L is the last layer, $Q^{(L)} = S$. For convenience, we denote X as $H^{(0)}$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ contains value 0 or 1, where 0 indicates no link between nodes and 1 denotes the existence of a link. $D_{i,i} = \sum_j A_{i,j} \in \mathbb{R}^{N \times N}$ represents the diagonal degree matrix of A . As defined by Kipf and Welling [72], the $(l+1)$ th convolutional layer is:

$$h_i^{(l+1)} = \theta \left(\sum_{j \in \text{nv}(i)} \frac{1}{\sqrt{\tilde{D}_{i,i} \tilde{D}_{j,j}}} h_j^{(l)} W^{(l)} \right), \quad (2)$$

where $\text{nv}(i)$ is the node v_i and its neighbors.

Or the equivalent matrix form is as follows:

$$H^{(l+1)} = \theta \left(\tilde{A}^{-\frac{1}{2}} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right), \quad (3)$$

where $\tilde{A} = A + I_N$, $\tilde{D} = D + I_N$, I is the identity matrix, $H^{(l+1)} \in \mathbb{R}^{N \times Q^{(l+1)}}$ denotes the output of the $l+1$ layer, $W^{(l)} \in \mathbb{R}^{Q^{(l)} \times Q^{(l+1)}}$ is the weight parameters and θ is the nonlinear activation function.

Gaussian distribution theorem. When the independent random variables follow a Gaussian distribution, their weighted sum still follows a Gaussian distribution:

Theorem 1. If $X_i \sim \mathcal{N}(\mu_i, \text{diag}(\sigma_i))$, $i = 1, \dots, N$ which are independent, for the weights w_i , we have:

$$\sum_{i=1}^N w_i x_i \sim \mathcal{N} \left(\sum_{i=1}^N w_i \mu_i, \text{diag} \left(\sum_{i=1}^N w_i^2 \sigma_i^2 \right) \right). \quad (4)$$

Based on [Theorem 1](#), assuming all the hidden representations of N nodes are independent, the node neighbors aggregation also follows the Gaussian distribution:

$$h_{Nv(i)}^{(l)} = \sum_{j \in \text{nv}(i)} \frac{1}{\sqrt{\tilde{D}_{i,i} \tilde{D}_{j,j}}} h_j^{(l)} \\ \sim \mathcal{N} \left(\sum_{j \in \text{nv}(i)} \frac{1}{\sqrt{\tilde{D}_{i,i} \tilde{D}_{j,j}}} \mu_j^{(l)}, \text{diag} \left(\sum_{j \in \text{nv}(i)} \frac{1}{\sqrt{\tilde{D}_{i,i} \tilde{D}_{j,j}}} \sigma_j^{(l)} \right) \right). \quad (5)$$

GCN–GRU. The GRU model can handle long-term memory in back-propagation. There are two gates inside the hidden unit of GRU, r_t is the reset gate, z_t is the update gate, c_t denotes the memory content reserved at time t , and y_t represents the output. The previous memory and the input at the current time step are taken together as inputs. By inputting data into these units in a sequence, the potential temporal dependence of data can be captured through the operation and learning of these units.

The combined model GCN–GRU can acquire the spatiotemporal dependencies of the traffic data simultaneously. The principle of GCN–GRU is as follows: (1) The graph convolutional network is used to aggregate the spatial information of neighbors; (2) Followed by a gated recurrent unit network to capture the temporal correlations; (3) The prediction results are obtained through the fully connected layer. The specific process of GCN–GRU is as follows:

$$z_t = \theta(W_z [H^{(L)}, y_{t-1}] + b_z) \quad (6a)$$

$$r_t = \theta(W_r [H^{(L)}, y_{t-1}] + b_r) \quad (6b)$$

$$c_t = \theta(W_c [H^{(L)}, (r_t \odot y_{t-1})] + b_c) \quad (6c)$$

$$y_t = z_t \odot y_{t-1} + (1 - z_t) \odot c_t, \quad (6d)$$

where y_{t-1} is the output at time $t-1$, L is the final layer of GCN. W_z, W_r, W_c and b_z, b_r, b_c represent the weights and bias of the model in the training process. The dot-circle symbol denotes the element-wise product.

4. Robust spatiotemporal graph convolutional network (RT-GCN)

In this section, a robust spatiotemporal graph convolutional network (RT-GCN) is proposed to predict traffic speed. The framework of the model is shown in [Fig. 1](#). Different from existing deep learning predictive methods, RT-GCN takes latent representations of each node in convolutional layers as Gaussian distribution:

$$H_t^{(l)} \sim \mathcal{N}(M_t^{(l)}, \Sigma_t^{(l)}), \quad (7)$$

where $M_t^{(l)} = [\mu_{1(t)}^{(l)}, \dots, \mu_{N(t)}^{(l)}] \in \mathbb{R}^{N \times Q^{(l)}}$ and $\Sigma_t^{(l)} = [\sigma_{1(t)}^{(l)}, \dots, \sigma_{N(t)}^{(l)}] \in \mathbb{R}^{N \times Q^{(l)}}$ represent the matrix of means and variance¹ for all the nodes representations in the layer l at time t .

To accommodate the fact that the input features $X = [X_{t-T+1}, \dots, X_t]$ are vectors instead of Gaussian distributions, a fully connected layer is used to learn the parameters $M_t^{(l)}$ and $\Sigma_t^{(l)}$ in the first layer:

$$M_t^{(1)} = \theta(X W_\mu^{(0)}) \quad (8)$$

$$\Sigma_t^{(1)} = \theta(X W_\sigma^{(0)}). \quad (9)$$

¹ Here σ is used instead of σ^2 to denote variances for the ease of presentation.

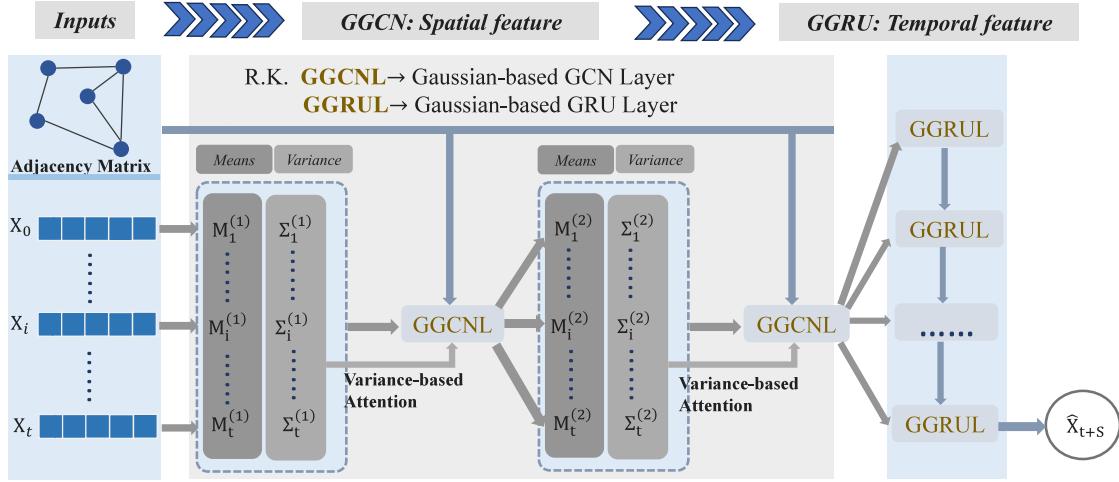


Fig. 1. The framework of RT-GCN with two layers.

Specifically, RT-GCN contains Gaussian-based GCN layers (GGCN) and Gaussian-based GRU (GGRU) layers.

Gaussian-based GCN Layer (GGCN Layer). The output of the GCN layer can be parameterized by mean $M_t^{(l)}$ and variance $\Sigma_t^{(l)}$. Formally, the matrix form is expressed as follows:

$$M_t^{(l+1)} = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} M_t^{(l)} W_\mu^{(l)} \right) \quad (10)$$

$$\Sigma_t^{(l+1)} = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \Sigma_t^{(l)} W_\sigma^{(l)} \right), \quad (11)$$

where $W_\mu^{(l)}$ and $W_\sigma^{(l)}$ are weight parameters associated with the mean and variance of the means and variances, respectively. The GCN model with two hidden layers is chosen to capture spatial dependencies since experiment results in Kipf and Welling [72] show that a two-layer network is the most effective setting.

In addition, a variance-based attention mechanism is integrated to mitigate the effects of perturbed data on information propagation, enhancing the robustness of the model. This mechanism assigns higher weights to hidden states with low variance and lower weights to those with high variance, thereby preventing abnormal or missing data from having an undue impact on other nodes and timestamps. To mitigate the influence of noisy data, an attention mechanism based on variance is introduced. Here, a smoothing exponential function is adopted to convert the variances into weights:

$$\mathcal{A}_t^{(l)} = \exp \left(-\gamma \Sigma_t^{(l)} \right), \quad (12)$$

where $\mathcal{A}_t^{(l)}$ denotes attention weights and γ is a hyper-parameter. With the integration of Eq. (12) into Eqs. (10) and (11):

$$M_t^{(l+1)} = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \left(M_t^{(l)} \odot \mathcal{A}_t^{(l)} \right) W_\mu^{(l)} \right) \quad (13)$$

$$\Sigma_t^{(l+1)} = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \left(\Sigma_t^{(l)} \odot \mathcal{A}_t^{(l)} \odot \mathcal{A}_t^{(l)} \right) W_\sigma^{(l)} \right). \quad (14)$$

The output of the last layer of the GGCN is a latent representation of the nodes with length S , which corresponds to the prediction time length.

Gaussian-based GRU Layer (GGRU Layer). Considering that the covariance and mean of hidden state ($\Sigma_{h(t)}$ and $M_{h(t)}$) are learned from the dataset independently, the independent gates are also used for updating and resetting $\Sigma_{h(t)}$ and $M_{h(t)}$. Formally, given the outputs of the last GGCN layer L , $M_t^{(L)}$ and $\Sigma_t^{(L)}$, the GGRU layer process is formulated as follows:

$$z_t^m = \theta \left(W_z^m \left[M_t^{(L)}, M_{h(t-1)} \right] + b_z^m \right) \quad (15a)$$

$$r_t^m = \theta \left(W_r^m \left[M_t^{(L)}, M_{h(t-1)} \right] + b_r^m \right) \quad (15b)$$

$$z_t^\Sigma = \theta \left(W_z^\Sigma \left[M_t^{(L)}, M_{h(t-1)} \right] + b_z^\Sigma \right) \quad (15c)$$

$$r_t^\Sigma = \theta \left(W_r^\Sigma \left[M_t^{(L)}, M_{h(t-1)} \right] + b_r^\Sigma \right) \quad (15d)$$

$$M_{c(t)} = \text{ReLU} \left(W_\mu \left[M_t^{(L)}, r_t^m \odot M_{h(t-1)} \right] + b_\mu \right) \quad (15e)$$

$$\Sigma_{c(t)} = \text{ReLU} \left(W_\sigma \left[\Sigma_t^{(L)}, r_t^\Sigma \odot \Sigma_{h(t-1)} \right] + b_\sigma \right) \quad (15f)$$

$$M_{h(t)} = z_t^m \odot M_{h(t-1)} + (1 - z_t^m) \odot M_{c(t)} \quad (15g)$$

$$\Sigma_{h(t)} = z_t^\Sigma \odot \Sigma_{h(t-1)} + (1 - z_t^\Sigma) \odot \Sigma_{c(t)}, \quad (15h)$$

where $M_{h(t-1)}$ and $\Sigma_{h(t-1)}$ are the hidden state transmitted from the previous GGRU layer at time $t-1$. *Sigmoid* is applied as an activation function in Eqs. (15a) ~ (15h). Since the mean and variance should remain positive numbers, the activation function *ReLU* is used in the memory cell.

To avoid failure of the back-propagation algorithm due to the fact that the distribution is not differentiable, a “reparameterization trick” was implemented [73]. This trick involves expressing the prediction \hat{X}_t as follows:

$$\hat{X}_t = M_{h(t)} + \epsilon \Sigma_{h(t)}, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}). \quad (16)$$

During the inference phase, the prediction is:

$$\hat{X}_t = M_{h(t)}. \quad (17)$$

4.1. Loss functions

The training loss for the RT-GCN model consists of two components: the task loss \mathcal{L}_{task} and the \mathcal{L}_2 regularization loss:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \beta_1 \mathcal{L}_2, \quad (18)$$

where β_1 is the hyperparameters for balancing the strength of \mathcal{L}_2 .

Specifically, these components are defined as follows:

(1) The Euclidean distance ($\|\cdot\|_2$) between the observed speed (X_t) and predicted speed $f_{RTGCN}(\cdot)$ is used as the loss function to minimize the prediction error. *Batch Random Noise* techniques are applied by adding an on-the-fly random noise δ into each batch of input in the training process. The task loss is as described in Eq. (19):

$$\mathcal{L}_{task} = \|X_{t:t+S} - f_{RTGCN}(A, X_{t-T+1:t} + S_1 \odot \delta)\|_2, \quad \delta \sim \mathcal{U}(-1, 1), \quad (19)$$

where S_1 is a random binary mask consisting of 0 and 1. S is the prediction length. \mathcal{U} is a uniform distribution. δ represents the added random noise into the input.

(2) In addition, following the T-GCN [45], we add \mathcal{L}_2 regularization on parameters from GGCN and GGRU layers. It is expressed as follows:

$$\begin{aligned} \mathcal{L}_2 = \sum_{l=0}^L & \left(\|W_\mu^{(l)}\|_2 + \|W_\sigma^{(l)}\|_2 + \|W_z^m\|_2 + \|W_r^m\|_2 + \|W_z^\Sigma\|_2 \right. \\ & \left. + \|W_r^\Sigma\|_2 + \|W_\mu\|_2 + \|W_\sigma\|_2 \right), \end{aligned} \quad (20)$$

where $W_\mu^{(l)}$ and $W_\sigma^{(l)}$ are learnable parameters in the l th layer in convolution step. W_z^m , W_r^m , W_z^Σ , W_r^Σ , W_μ , W_σ are learnable parameters as shown in Eqs. (15a) ~ (15h).

5. RT-GCN experiments

In this section, the effectiveness of the RT-GCN model is showcased by using real-world traffic datasets. The experiments are meticulously designed to address two research questions:

RQ1 what is the robustness on the raw dataset?

RQ2 what is the robustness on the trained clean dataset?

5.1. Experiment settings

All the experiments are executed under a platform with one NVIDIA A100 GPU 40 GB card. The training processes are implemented in Python with PyTorch 1.12.1.

5.1.1. Perturbation settings

The effectiveness of the proposed RT-GCN model is investigated on real-world traffic network datasets and its robustness is verified under two common abnormal data patterns, noisy perturbation and missing perturbation as described below:

Gaussian noisy perturbation. Gaussian noise is added to the randomly selected subset of the inputs. Formally, given the inputs $X_{(t-T+1:t)}$, the Gaussian perturbed inputs are expressed as:

$$\tilde{X}_{(t-T+1:t)} = X_{(t-T+1:t)} + S_2 \odot \hat{\tau}, \quad (21)$$

where S_2 is a random binary mask to control the percentage of added noise. The percentage of 1 in S_2 is controlled with the experiment definition. The noise $\hat{\tau}$ is scaled by the speed of the set:

$$\hat{\tau} = (\text{speed}_{\max} - \text{speed}_{\min}) \times \tau \times k, \quad \tau \sim \mathcal{N}(0, 1), \quad (22)$$

where τ represents the noise sampled from the standard Gaussian distribution. k is the noise severity that controls the magnitude of noise.

Random missing perturbation. The randomly selected subsets of the inputs are replaced as missing values. Formally, given the inputs $X_{(t-T+1:t)}$, the random missing inputs are expressed as:

$$\tilde{X}_{(t-T+1:t)} = X_{(t-T+1:t)} \odot S_2, \quad (23)$$

where S_2 is a random binary mask that dictates the percentage of added missing values. The ratio of values 0 and 1 in S_2 is administrated depending on the experimental setup. We use 0 by default for the missing value.

Besides, given the intricate nature of missing patterns in real-world transportation systems, which can include time-consecutive or spatially correlated instances, three types of intricate missing patterns are designed and tested:

Temporally correlated missing (TCM). Missing values are temporally correlated and appear as consecutive intervals for each sensor. This type of corrupted data may be a result of long-term physical damage or maintenance backlog. Setting the proportion of nodes with perturbed data as 50%. For each node, we pick the same time window with various ratios of historical time T (ranging from 20% to 80% in our experiment) and impute missing values in this time slot.

Spatially correlated missing (SCM). Missing values are dependent on the spatial dimension and tend to occur at neighboring sensors

or connected road links within each time slot. This may be caused by regional power outages or communication problems. Setting the proportion of perturbed historical data per selected nodes as 50%. For each time slot within the imputation window, we randomly select a node v in the traffic network, find sensors closest to v with a share of 20% to 80% of nodes N , and hide their data,

Spatio-temporally correlated missing (STCM). Missing values are dependent on both spatial and temporal dimensions. This means values are corrupted in consecutive time intervals and neighboring locations. This often happens caused by long-term regional malfunctions. We randomly select a node v in the traffic network, choose $N * 50\%$ sensors closest to v , and mask the data points of these nodes within the same time slots ranging from 20% to 80% of the input time steps T .

5.1.2. Evaluation metric

Two metrics, RMAE and RRMSE are introduced to measure the performance of the models in terms of prediction accuracy:

Robust Mean Absolute Error (RMAE):

$$\text{RMAE} = \frac{1}{KN} \sum_{t=1}^K \sum_{i=1}^N |X_{i(t)} - f_i(A, \tilde{X}_{t-T+1:t})|. \quad (24)$$

Robust Root Mean Square Error (RRMSE):

$$\text{RRMSE} = \sqrt{\frac{1}{KN} \sum_{t=1}^K \sum_{i=1}^N (X_{i(t)} - f_i(A, \tilde{X}_{t-T+1:t}))^2}. \quad (25)$$

where $X_{i(t)}$ and $f_i(A, \tilde{X}_{t-T+1:t})$ represent the actual traffic speed and the predicted speed of node i at the t th time stamp. $\tilde{X}_{t-T+1:t}$ is the perturbed input. K is the number of time samples and N is the number of traffic nodes. RMAE and RRMSE measure the deviation between the predicted values and the actual values *under the perturbed input*. Smaller values indicate higher accuracy.

Correspondingly, by replacing $\tilde{X}_{t-T+1:t}$ with the original input $X_{t-T+1:t}$, RMAE and RRMSE are equivalent to **Mean Absolute Error (MAE)** and **Root Mean Square Error (RMSE)**.

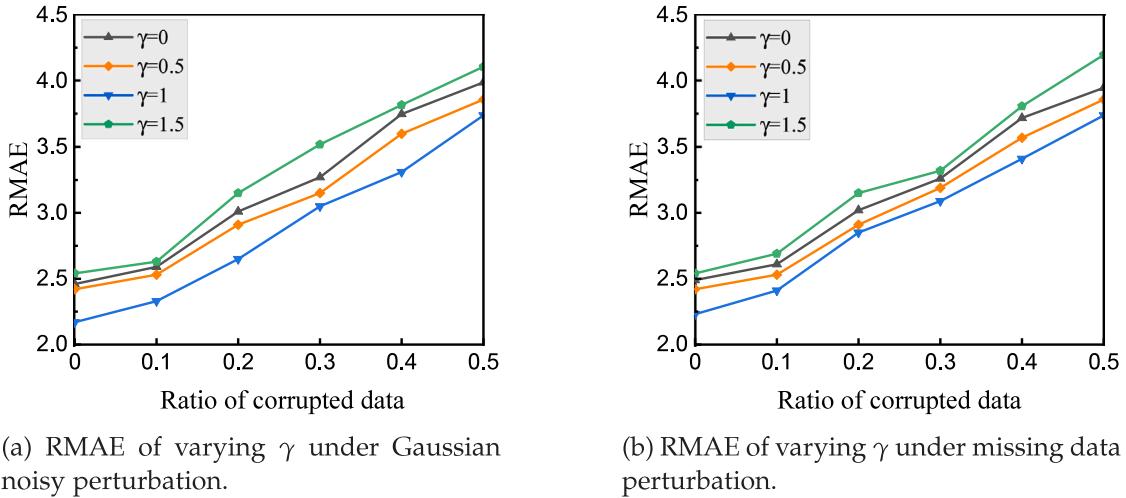
For task performance, MAE and RMSE values are reported based on the unperturbed input. For robust performance, RMAE and RRMSE are reported based on Gaussian noisy perturbation and Random missing perturbation, respectively.

5.1.3. Datasets

Four real-world datasets are used in our experiments:

- **PEMS-BAY:** covers a duration of six months from Jan. 1 to May 31, 2017 of 325 sensors on the Bay Area, collected from California Transportation Agencies' Performance Measurement System (PeMS), published by Li et al. [74] on the GitHub (<https://github.com/liyaguang/DCRNN>).
- **METR-LA:** comprises traffic information gathered from 207 loop detectors located on the highways of Los Angeles County, from Mar. 1 to Jun. 27, 2012, published by Li et al. [74] on the GitHub (<https://github.com/liyaguang/DCRNN>).
- **Los-loop:** contains traffic speed data collected by 207 loop detectors in the highway of Los Angeles, from March 1–7, 2012, published by Zhao et al. [45] on the GitHub (<https://github.com/lehaifeng/T-GCN>).
- **HK:** includes 179 sensors in the Kowloon area and Hong Kong island, having collected speed data from the open data platform provided by the Hong Kong government from May 1–31, 2020, published by Zhu et al. [15] on the GitHub (https://github.com/LYZ98/diffusion_attack).

It is notable that there are 0.0031% missing values in PEMS-BAY, and 8.11% missed in METR-LA. The Los-loop and HK datasets have already been preprocessed by a linear interpolation method to fill in the missing values. In the main experiments, we split Los-loop and

Fig. 2. Analysis of hyper-parameter γ on PEMS-BAY dataset.

HK datasets following Zhao et al. [45] where 80% for the training set and 20% for the test set, and split PEMS-BAY and METR-LA dataset following Li et al. [74] where 70% for training set, 20% for test set, 10% for validation set. We did not use the validation set for selecting the best checkpoint to report the best performance. Instead, we report the test accuracy at the last checkpoint for all the results in our main experiments. For the selection of the best hyperparameters (λ , β_1 , hidden size) of RT-GCN, we utilize a 10% hold-out set for choosing the best hyperparameters ahead for each dataset.

5.2. Baseline models

For model comparison, we only consider baseline models for which the source code is publicly accessible. The training of each model adheres to the default parameter settings outlined in the respective papers. Baseline models can be categorized into two groups based on their approach to handling missing values:

Group 1: This category involves jointly inferring the missing values and performing the prediction task.

- **GRU-I:** A variant of GRU [24], utilizes predictions from previous steps to infer the missing values.
- **GRU-D:** An extension of GRU [24], enhances prediction performance by incorporating missing patterns, including masking information and time intervals between missing and observed values.
- **LSTM-M:** A neural network architecture specifically designed for traffic prediction with missing data [75], utilizing the LSTM framework. It incorporates multiscale temporal smoothing techniques to infer missing data.
- **SGMN:** Spectral Graph Markov Network [70], defines the transition between network-wide traffic states at consecutive time steps as a graph Markov process and incorporates spectral graph convolution to infer missing data step by step.
- **GCN-M:** Graph Convolutional Network model that effectively handles complex missing values [71], takes into account both local spatiotemporal features and global historical patterns by utilizing an attention-based memory network.

Group 2: This category either preprocessed the missing data or treated missing values within the input sequence as zero values and masked them out during the computation of the loss error.

- **GRU:** Gated recurrent unit [76], see details in Section 3;

- **DCRNN:** Convolutional recurrent neural network [74], incorporates bidirectional random walks to embed the graph and utilizes gated recurrent units in an encoder-decoder model to capture spatial-temporal dependencies.
- **ASTGCN:** Attention-based spatiotemporal graph convolutional networks [77], which includes spatiotemporal attention mechanisms to effectively capture the correlations of traffic data;
- **AGCRN:** Adaptive graph convolutional recurrent network [78], learns an adaptive graph and incorporates recurrent graph convolutions with node parameter learning.
- **GMAN:** Graph multi-attention network [55], which employs a novel spatiotemporal attention mechanism that is computed using input data and pre-defined spatiotemporal embedding information.
- **STNN:** Local spacetime neural network [79], which obtains spatiotemporal correlations by using a novel spacetime convolution and attention mechanism.
- **Ada-STNet:** Adaptive spatiotemporal graph neural network [80], which realizes multistep traffic prediction through a dedicated spatiotemporal convolution architecture.

5.3. Hyperparameter of RT-GCN

In this experiment, we set the batch size to 32, the learning rate to 0.001, and the number of training epochs to 5000. The Adam optimizer is adopted to train the RT-GCN model.

To investigate whether the imposed variance-based attention weights in the RT-GCN can help prevent the propagation of perturbations, we vary hyperparameter γ in Eq. (12), which refers to the setting of attention weights. The tests are conducted by adding Gaussian noise and random missing data perturbation to different ratios of historical data of randomly selected 10% nodes. The RMAE on the PEMS-BAY and Los-loop datasets are reported, while the other two datasets show consistent results. As shown in Figs. 2 and 3, $\gamma = 1$ is the optimum value for this parameter and is selected as such for the analysis. We can observe that when using variance-based attention weights ($0 < \gamma < 1$), the model performs better than without the attention mechanism ($\gamma = 0$).

We do tests for 15 minute prediction tasks on clean datasets for β_1 in Eq. (18). According to the results, the optimum value for β_1 for four datasets is $5 \cdot 10^{-4}$. We train the models with 8, 16, 32, 64, 100, 128, 256, and 512 hidden units and select the optimal model with the lowest error. 64 hidden units are the best value for the Los-loop, PEMS-BAY, and METR-LA, while 100 is the optimum number for the HK dataset.

The model details of RT-GCN on four datasets are listed in Table 1.

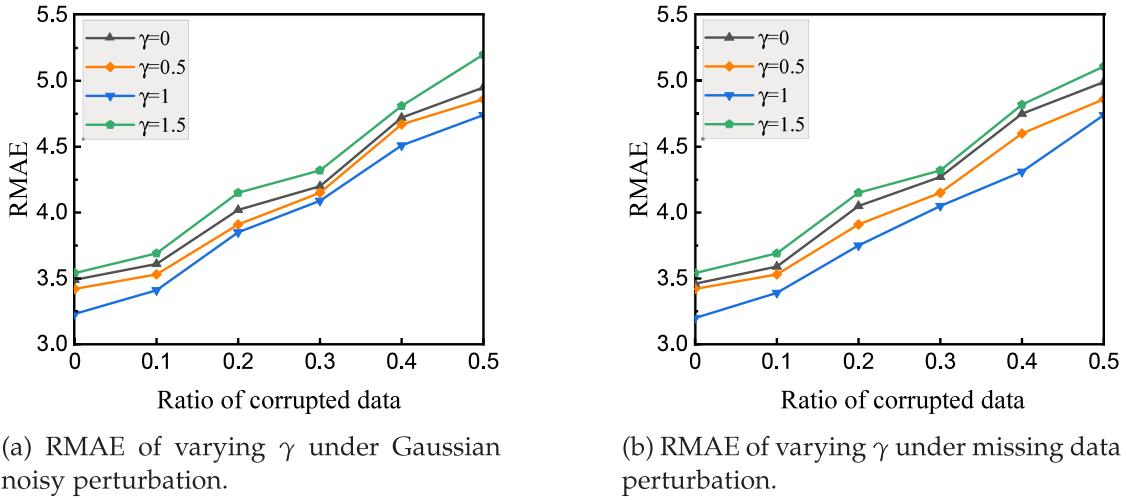
Fig. 3. Analysis of hyper-parameter γ on Los-loop dataset.

Table 1
Model details of RT-GCN on four datasets.

Dataset	Layers of GGCN	Layers of GGRU	Hidden size of GGCN	Hidden size of GGRU	Number of parameters	Training time (s/epoch)			
						RTGCN	GRU	DCRNN	GMAN
PEMS-BAY	2	1	64	64	25346	6.5	5.8	375.8	125.7
METR-LA	2	1	64	64	25346	4.3	3.6	318.4	97.8
Los-Loop	2	1	64	64	25346	1.2	0.9	107.7	37.3
HK	2	1	100	100	61202	1.8	1.2	216.1	73.5

Table 2
Performance comparison on the raw PEMS-BAY and METR-LA datasets.

Horizon	PEMS-BAY						METR-LA					
	15 min		30 min		60 min		15 min		30 min		60 min	
	MAE	RMSE										
GRU-I	1.89	3.52	2.26	4.22	2.62	4.89	3.49	5.83	3.97	6.74	4.60	7.88
GRU-D	5.34	9.25	5.42	9.26	5.41	9.27	7.43	11.85	7.45	11.84	7.47	11.86
LSTM-M	1.87	3.39	2.33	4.33	3.45	8.32	3.46	5.74	4.08	6.86	4.63	7.83
SGMN	1.63	3.40	2.29	4.91	3.31	6.86	4.23	8.54	5.46	10.88	7.37	13.78
GCN-M	1.33	2.72	1.62	3.64	1.95	4.40	2.74	5.21	3.12	6.18	3.54	7.12
GRU	1.89	3.53	2.27	4.24	2.65	4.90	3.48	5.80	3.97	6.74	4.65	7.86
DCRNN	1.38	2.95	1.74	3.97	2.07	4.74	2.77	5.38	3.15	6.45	3.60	7.59
ASTGCN	1.52	3.13	2.01	4.27	2.61	5.42	4.86	9.27	5.43	10.61	6.51	12.52
AGCRN	1.36	2.88	1.69	3.87	1.98	4.59	2.86	5.55	3.25	6.57	3.68	7.56
GMAN	1.35	2.90	1.65	3.82	1.92	4.49	2.80	5.55	3.12	6.49	3.44	7.35
STNN	1.28	2.71	1.67	3.73	2.03	4.93	2.64	5.30	3.10	6.55	3.51	7.56
Ada-STNet	1.30	2.73	1.62	3.67	1.89	4.36	2.65	5.06	3.03	6.08	3.47	7.18
RT-GCN	1.40	2.89	1.65	3.81	1.92	4.47	2.73	5.28	3.14	6.26	3.42	7.31

5.4. RQ1: Robustness on raw datasets

In this section, we perform experiments using the raw **PEMS-BAY** (contains 0.0031% missing value) and **METR-LA** (contains 8.11% missing value) datasets to exhibit the performance of the RT-GCN model and baselines. The results were averaged over 5 independent runs.

5.4.1. Prediction results without perturbation

We first conduct experiments without perturbation to build a reference, the prediction results on raw datasets are shown in Table 2. We highlight the best results across all baselines by using bold formatting. The results were averaged over 5 independent runs.

Our objective is not to outperform all baseline models in terms of forecasting accuracy for the raw dataset, instead, we aim to demonstrate that our model exhibits high performance especially with a relatively high share of noisy/missing data.

5.4.2. Prediction results on noisy perturbation

In this section, Gaussian noise is added with different rates and the performance of RT-GCN and baselines are compared. For example, a 40% proportion signifies that 40% of the data in the test set has been subjected to noise disturbances. The magnitude of the noise is determined by Eq. (21). Table 3 shows the prediction results of RT-GCN and baseline models for 60 min prediction tasks on PEMS-BAY and METR-LA datasets.

5.4.3. Prediction results on random missing perturbation

In order to evaluate the model's ability to handle missing values, we devised three scenarios with different random missing rates 10%, 20%, and 40% according to Eq. (23). The traffic speed for one hour (12 intervals) is predicted, and the results are as shown in Table 4.

5.4.4. Robustness on temporally and spatially correlated missing patterns

The performance comparison of RT-GCN with other baseline models on the raw datasets with respect to TCM, SCM and STCM conditions are plotted in Figs. 4–6.

Table 3

Performance comparison on the PEMS-BAY and METR-LA datasets with Gaussian noisy perturbation.

Noisy data rate	PEMS-BAY						METR-LA					
	20%		40%		60%		20%		40%		60%	
	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE
GRU-I	2.95	4.97	3.34	5.51	3.66	5.82	4.75	7.92	4.98	8.07	5.13	8.32
GRU-D	6.02	9.93	6.58	10.12	6.79	10.83	7.51	11.88	7.63	11.90	7.68	11.93
LSTM-M	3.69	8.50	3.88	8.61	3.97	8.72	4.75	7.91	4.98	8.07	5.13	8.25
SGMN	4.21	7.13	4.33	7.18	4.46	7.25	7.56	13.94	7.72	14.02	7.86	14.35
GCN-M	2.02	4.48	2.47	4.69	2.66	4.90	3.57	7.39	3.72	7.63	3.91	7.86
GRU	2.73	4.96	2.86	5.11	2.91	5.24	4.74	7.93	4.88	7.97	4.93	8.02
DCRNN	2.33	4.80	2.57	4.84	2.72	4.93	3.73	7.74	3.81	7.82	3.95	7.94
ASTGCN	2.72	5.54	2.83	5.66	2.94	5.98	6.91	12.92	7.13	12.70	7.48	12.83
AGCRN	2.15	4.82	2.74	5.12	3.03	5.36	3.72	7.73	3.95	7.89	4.05	7.95
GMAN	2.03	4.67	2.57	5.03	2.89	5.44	3.53	7.40	3.71	7.63	3.86	7.72
STNN	2.07	4.96	2.53	5.62	2.93	5.88	3.60	7.78	3.81	7.86	4.03	7.91
Ada-STNet	2.09	4.61	2.89	5.56	3.10	5.82	3.65	7.77	3.97	7.82	4.11	7.89
RT-GCN	1.97	4.32	2.28	4.55	2.39	4.87	3.47	7.35	3.59	7.51	3.53	7.48

Table 4

Performance comparison on the PEMS-BAY and METR-LA datasets with various missing rates.

Missing rate	PEMS-BAY						METR-LA					
	10%		20%		40%		10%		20%		40%	
	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE
GRU-I	2.81	4.52	3.49	5.75	4.22	6.63	4.72	7.93	4.94	8.05	5.09	8.21
GRU-D	5.60	9.76	6.46	10.97	7.22	11.58	7.50	11.87	7.65	11.89	7.73	11.96
LSTM-M	3.78	8.57	4.56	9.70	5.34	10.11	4.71	7.92	4.75	7.98	4.86	8.18
SGMN	3.44	6.95	4.34	7.03	4.45	7.21	7.54	13.93	7.61	13.97	7.81	14.12
GCN-M	2.04	4.56	2.21	4.89	2.63	5.09	3.57	7.35	3.63	7.47	3.86	7.71
GRU	2.71	4.88	2.82	5.08	3.05	5.43	4.70	7.89	4.87	8.16	4.95	8.41
DCRNN	2.78	4.84	2.92	4.96	3.12	5.26	3.71	7.73	3.84	7.80	3.90	7.93
ASTGCN	2.74	5.73	2.86	5.91	2.97	5.98	6.78	12.63	6.84	12.86	6.93	12.95
AGCRN	2.15	4.62	2.39	4.94	2.82	5.39	3.69	7.60	3.74	7.75	3.88	7.81
GMAN	2.09	4.51	2.17	4.92	2.68	5.11	3.46	7.41	3.62	7.58	3.89	7.73
STNN	2.07	4.49	2.23	4.98	2.71	5.75	3.58	7.74	3.72	7.89	4.33	7.89
Ada-STNet	2.08	4.47	2.27	4.95	2.75	5.46	3.63	7.76	3.89	7.76	4.37	7.92
RT-GCN	2.10	4.57	2.16	4.87	2.57	5.08	3.44	7.43	3.60	7.44	3.84	7.69

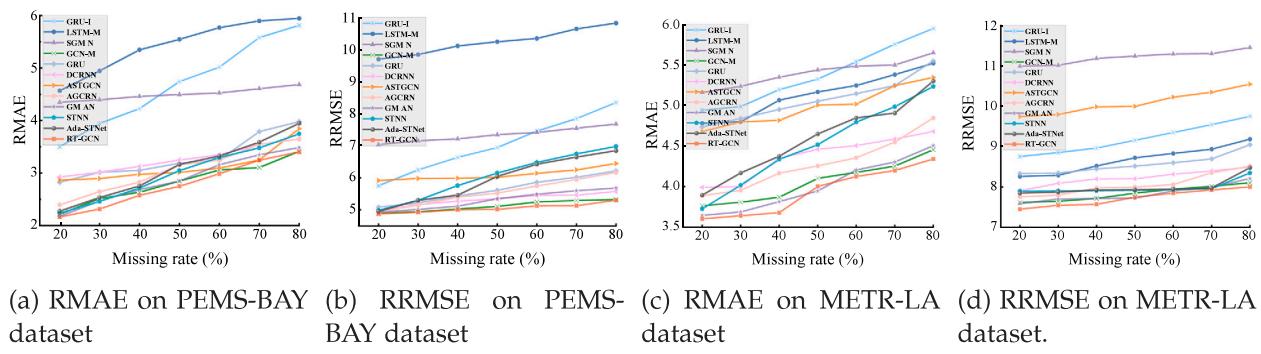
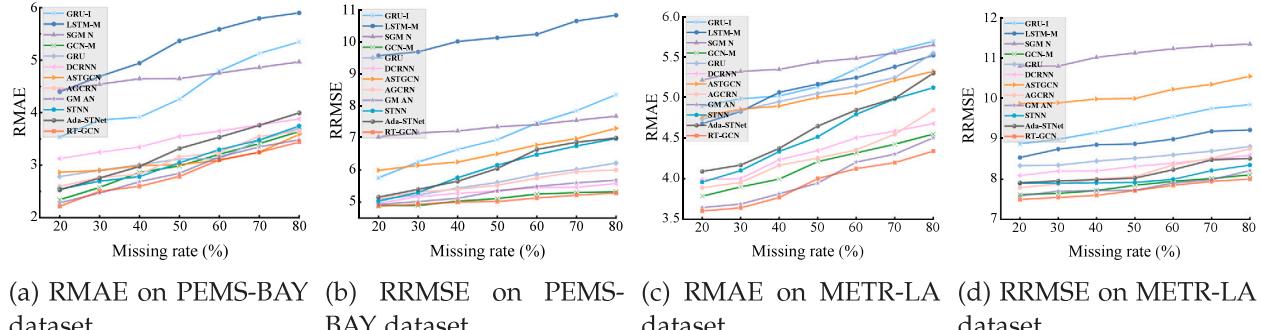
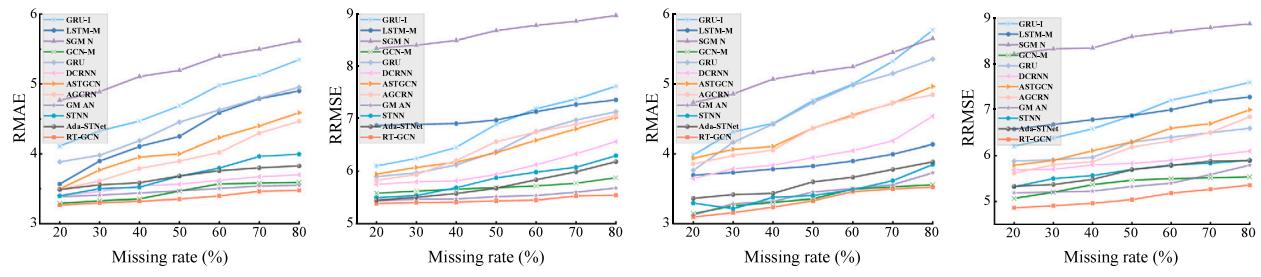
**Fig. 4.** Performance comparison with various missing rates under TCM.**Fig. 5.** Performance comparison with various missing rates under SCM.

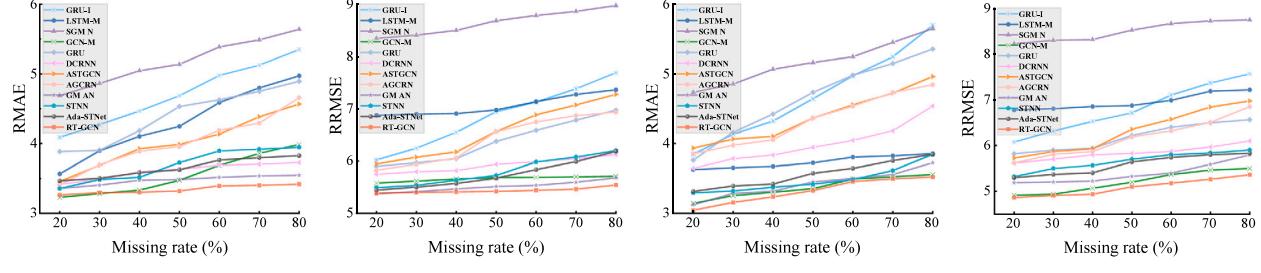
Table 7
Performance comparison on the Los-loop and HK datasets with various missing rates.

Missing rate	Los-loop						HK					
	10%		20%		40%		10%		20%		40%	
	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE	RMAE	RRMSE
GRU-I	4.12	6.31	4.78	6.87	5.31	7.29	3.84	6.07	4.32	6.52	4.97	7.10
GRU-D	7.30	10.89	7.49	11.02	7.64	11.34	6.37	10.21	6.51	10.39	6.54	10.81
LSTM-M	3.68	6.87	3.73	6.90	3.85	7.13	3.62	6.77	3.69	6.85	3.80	7.02
SGMN	4.88	8.34	5.11	8.49	5.36	8.78	4.73	8.22	5.06	8.31	5.24	8.66
GCN-M	3.27	5.57	3.31	5.68	3.39	5.69	3.14	4.91	3.30	4.93	3.49	5.19
GRU	3.97	5.89	4.57	6.04	5.22	6.58	3.76	5.81	4.42	5.93	4.98	6.37
DCRNN	3.81	5.74	4.03	5.81	4.10	5.96	3.64	5.61	3.83	5.79	4.04	5.82
ASTGCN	4.08	5.93	4.27	6.17	4.61	6.88	3.93	5.72	4.10	5.91	4.55	6.57
AGCRN	4.01	5.82	4.23	6.05	4.62	6.74	3.85	5.61	4.05	5.86	4.53	6.31
GMAN	3.36	5.37	3.47	5.46	3.51	5.53	3.12	5.18	3.31	5.22	3.50	5.39
STNN	3.35	5.49	3.51	5.63	3.89	5.98	3.29	5.31	3.37	5.56	3.48	5.77
Ada-STNet	3.46	5.44	3.58	5.57	3.76	5.83	3.31	5.29	3.42	5.40	3.64	5.72
RT-GCN	3.26	5.38	3.23	5.30	3.27	5.28	3.04	4.86	3.28	4.93	3.45	5.17



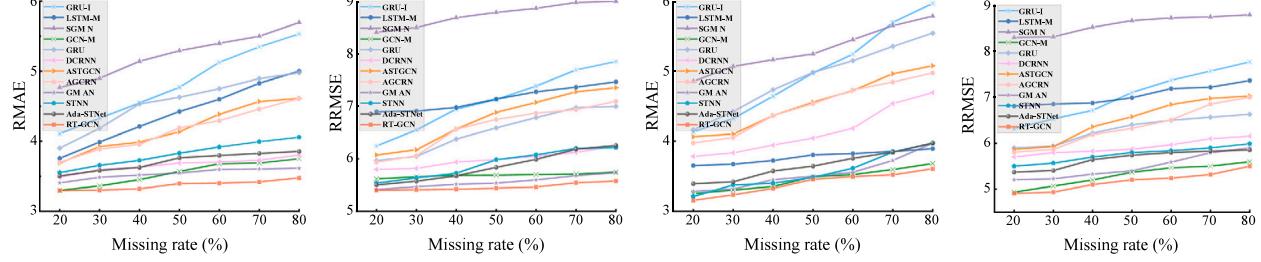
(a) RMAE on Los-loop dataset (b) RRMSE on Los-loop dataset (c) RMAE on HK dataset (d) RRMSE on HK dataset.

Fig. 7. Performance comparison with various missing rates under TCM.



(a) RMAE on Los-loop dataset (b) RRMSE on Los-loop dataset (c) RMAE on HK dataset (d) RRMSE on HK dataset.

Fig. 8. Performance comparison with various missing rates under SCM.



(a) RMAE on Los-loop dataset (b) RRMSE on Los-loop dataset (c) RMAE on HK dataset (d) RRMSE on HK dataset.

Fig. 9. Performance comparison with various missing rates under STCM.

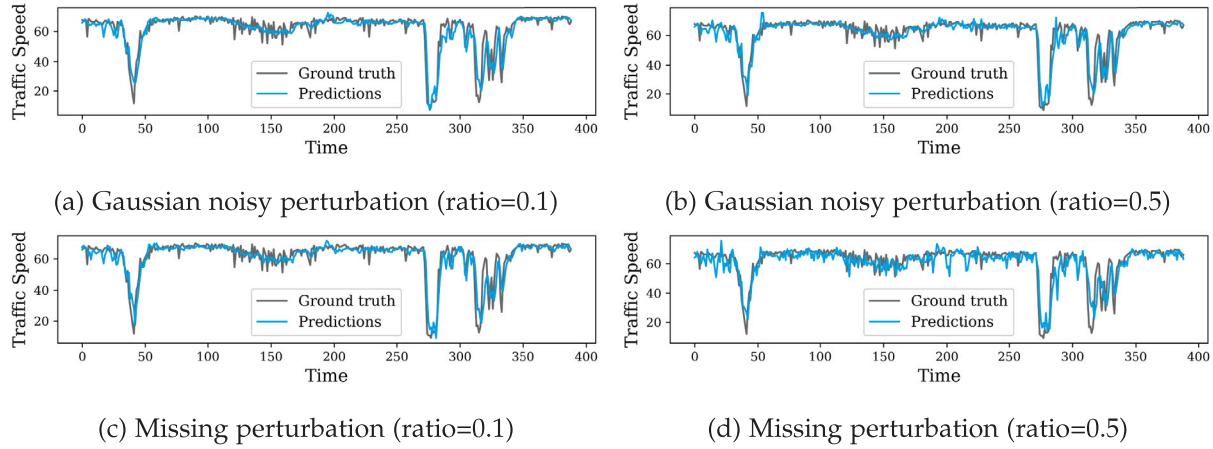


Fig. 10. Visualization of predictions from Los-loop dataset with Gaussian noisy and missing perturbation affecting 10% and 50% of randomly selected nodes respectively, as well as 10% of the historical data.

the PEMS-BAY dataset, STNN, GCN-M, and GMAN emerge as the best models for 15 min, 30 min, and 60 min prediction tasks respectively. However, the primary objective of this study is not to outperform all baseline models in terms of prediction accuracy but rather to demonstrate that RTGCN is capable of handling datasets with noise or missing data, making it more suitable for real-world scenarios.

Gaussian noisy perturbation. It can be observed from [Tables 6](#) and [3](#) that the RT-GCN outperforms the other baselines in terms of RMAE and RRMSE under almost all scenarios with Gaussian noisy perturbation, except for the 15 min prediction task on the HK dataset with 20% corrupted data, where the RMAE of RT-GCN is marginally higher than GCN-M and GMAN. Notably, as the proportion of corrupted data increases, the performance of RT-GCN remains consistently stable, while the prediction accuracy of other baselines deteriorates significantly. For instance, on the Los-loop dataset (as presented in [Table 6](#)): when the ratio of corrupted historical data is 60%, the RMAE of RT-GCN is 10.05% and 16.83% lower than GCN-M and GMAN respectively. Similarly, in the HK dataset, the RMAE of the competing baselines, GCN-M, ASTGCN, GMAN, STNN and Ada-STNet are 11.18%, 24.77%, 20.24%, 4.83% and 10.88% higher than RT-GCN when 40% of historical data is corrupted.

In some cases, the performance of RT-GCN shows even improvement when the proportion of noisy data increases. For instance, in the 15 min prediction task on the HK dataset presented in [Table 6](#), when the proportion of noisy historical data increases from 40% to 60%, the RMAE of RT-GCN decreases from 3.35 to 3.30. Similarly, in the 60 min forecasting task on the METR-LA dataset in [Table 3](#), RMAE slightly reduces when the ratio of corrupted traffic nodes increases from 40% to 60%. This demonstrates that our model possesses a remarkably robust ability to handle datasets contaminated with noise.

Missing perturbation. The superior performance of the RT-GCN model over the other baseline models under missing perturbation is evident from the results presented in [Tables 4](#) and [7](#). Generally, RT-GCN in most cases outperforms other baseline models, achieving lower RMAE and RRMSE. In cases where the missing rate is 0 and 10% in the PEMS-BAY dataset, GCN-M, GMAN, STNN and Ada-STNet typically outperform RT-GCN. While RT-GCN emerges as the best forecasting model as the missing rate increases to 20% and 40%, where the missing values become a more critical factor affecting the performance of forecasting models. Besides, the results demonstrate the consistent superiority of RT-GCN over other baselines when applied to the METR-LA dataset, which contains a higher share of missing data. Similar results also emerge on the clean Los-loop and HK datasets, where RT-GCN exhibits the lowest RMAE and RRMSE compared to all baseline models at various levels of missing data. This observation suggests that RT-GCN exhibits pronounced advantages in extracting complex

spatiotemporal correlation information from historical traffic data, no matter whether they contain noisy/missing values.

Additionally, according to [Figs. 4, 5, 6, 7, 8, 9](#), we can observe that the proposed RT-GCN outperforms all baselines across the different missing patterns at all missing rates under both raw and clean datasets. This indicates that regardless of whether the missing data is temporally or spatially correlated, our model can maintain prediction stability. In contrast, other baselines do not exhibit this performance; their RMAE and RRMSE values consistently increase as the missing ratio grows, and they consistently remain higher than RTGCN across various scenarios. Our proposed framework demonstrates a higher level of robustness when confronted with missing data with complex patterns. This is because the node representation layers based on the Gaussian distribution update the mean and variance values at each hidden layer. Additionally, the variance attention mechanism prevents the negative impact caused by missing data from spreading. Consequently, the model remains unaffected by the patterns of missing data.

Regarding the comparison of the baseline models, in Group 1, all models incorporate a process for handling missing data. Among them, GRU-D [24] exhibits significantly poorer performance compared to other algorithms, possibly because it was originally designed for health-care applications, where data tends to be more stable than dynamic traffic data. GRU-I [24], LSTM-M [75], and SGMN [70] are traffic prediction models designed to handle missing values. However, their predictive results in our experiments with added noise and missing values are not very favorable, generally falling short of baselines in Group 2. GCN-M [71], a recent model designed to effectively address complex missing values within the spatiotemporal context. It consistently delivers robust results in the presence of missing or noisy data across different scenarios. For instance, on the Los-loop dataset, GCN-M's performance is second only to RT-GCN, surpassing all other baselines. Within Group 2, GRU [76] performs moderately, possibly because it primarily extracts temporal features from traffic data while overlooking spatial characteristics. DCRNN [74] and ASTGCN [77] are two widely used and classic traffic prediction models. DCRNN, based on the spatial GCN model, constructs hidden feature representations by considering the diffusion process of each node as input. However, it has the drawback of high computational complexity and limited capacity for large-scale graphs. ASTGCN improves prediction performance by adding attention mechanisms, but both models experience a sharp decline in performance as the proportion of noisy/missing data increases. Moreover, three of the most recent models, namely AGCRN [78], GMAN [55], and STNN [79], are focused on the intricate task of spatiotemporal modeling for traffic data. In addition, Ada-STNet [80] excels in enabling dynamic learning and multistep prediction. These models exhibit commendable performance on clean datasets, with no

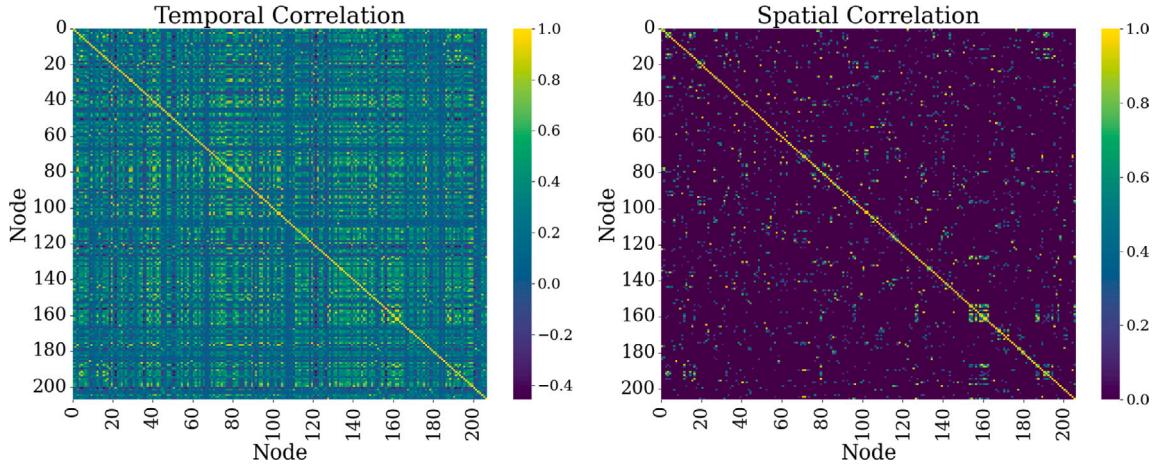


Fig. 11. Heatmap illustrating both temporal and spatial correlations using the Los-loop dataset. Spatial correlations refer to the adjacency matrix.

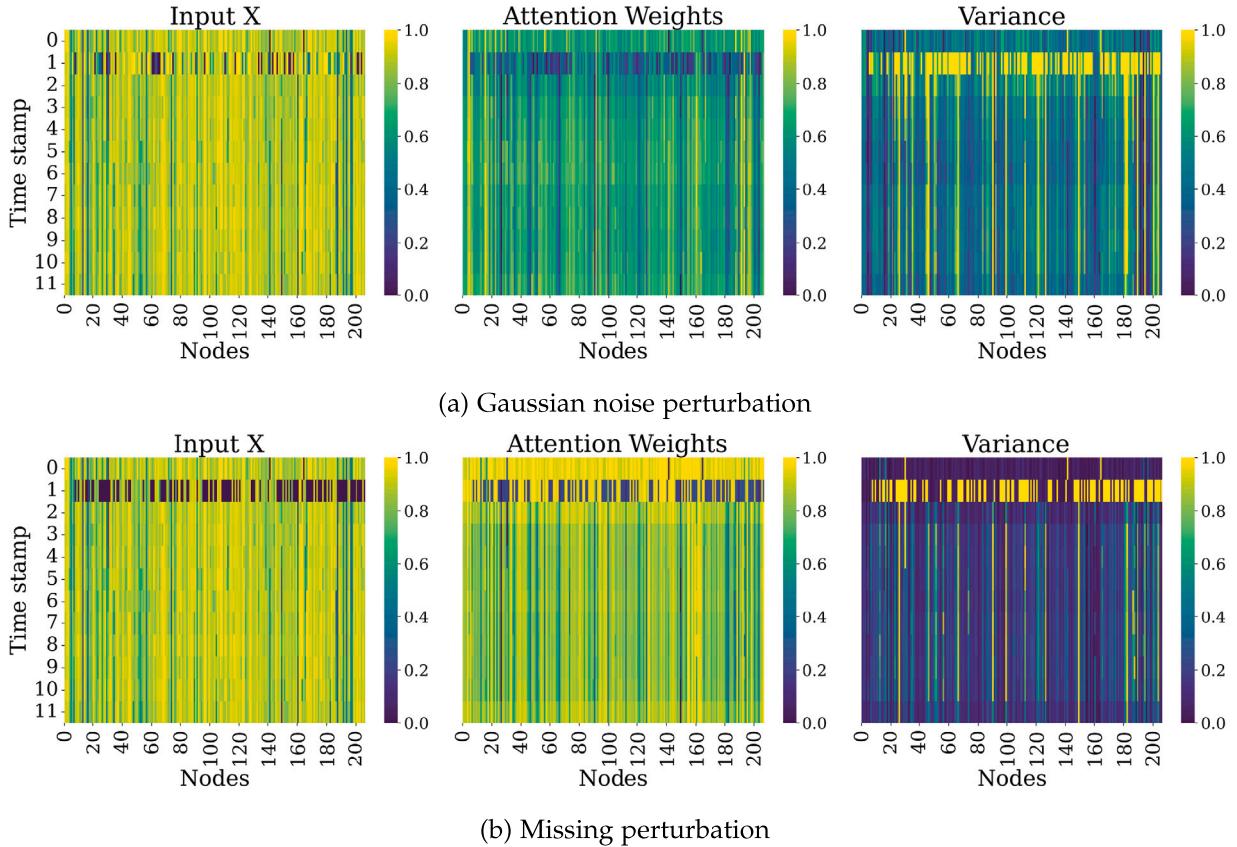


Fig. 12. Visualization of the input X, attention weights, and variance for the Los-loop dataset under two perturbation scenarios: Gaussian noisy perturbation and missing data perturbation, applied at timestamp 1 and 50% of randomly selected nodes. Data are normalized to the 0-1 range.

substantial differences among them. However, these models generally neglect missing values during training. Therefore, their performance deteriorates when noise or missing data is introduced, all of which are outperformed by RT-GCN.

5.7. Model analysis

Visualization of predictions. We present a visualization of the 15 min ahead predictions generated by RT-GCN on the Los-loop dataset in Fig. 10.

From the visualization, we can observe: ① Prediction of peaks: The RT-GCN model exhibits a tendency for slight under-predictions or

temporal shifts concerning peak values. This observation aligns with findings from Zhao et al. [45] wherein the T-GCN (original GCN-GRU model) was reported to have challenges in accurately predicting peak values. A potential reason for this behavior might be the inherent integration operation in GCN. The process of averaging node representations from neighboring nodes can introduce a “smoothing effect”, potentially leading to more moderated or “smoothed” peak values. ② Performance under perturbation: Notably, the RT-GCN model displays commendable resilience and accuracy under various perturbations, whether it is Gaussian noise or data missing. This robust performance is evident across both minor perturbations (e.g., ratio = 0.1) and more significant disruptions (e.g., ratio = 0.5). Such results underscore the

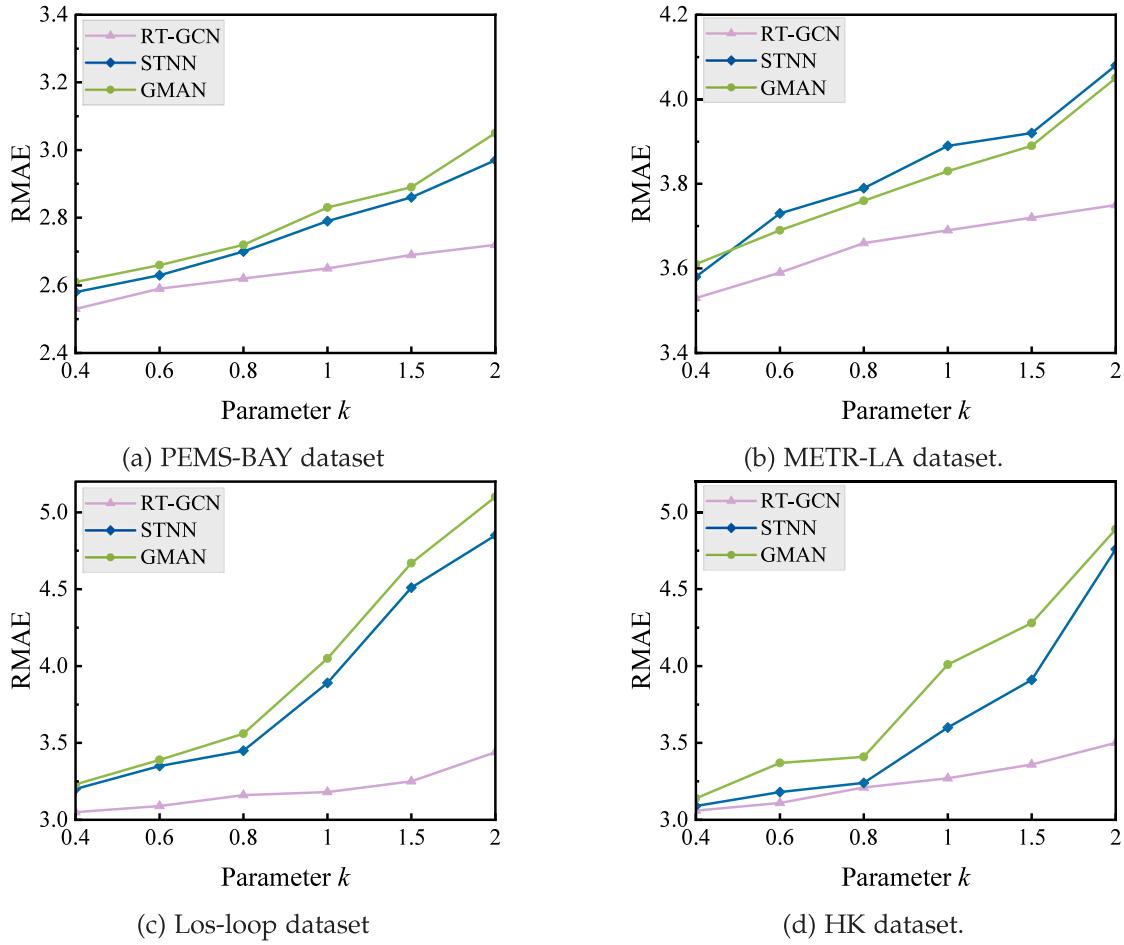


Fig. 13. The predicted performance with different magnitudes of Gaussian noise.

robustness of RT-GCN in handling perturbations, thereby highlighting the method's advanced capability.

Visualization of the affinity matrix regarding the temporal and spatial information. In order to delve deeper into the analysis of the affinity matrix regarding the temporal and spatial information, we expand to the visualization and analysis of the affinity matrix. As shown in Fig. 11, we include two types of affinity matrix: temporal correlation and spatial correlation. Temporal correlation is calculated from the Pearson Correlation for every node pair. Spatial correlation is denoted by the adjacency matrix directly.

By visually comparing the two heatmaps, we can identify nodes or regions where temporal correlations align with spatial correlations. Consistency between these two matrices might suggest that the spatial structure of the network effectively captures the temporal patterns in the speed data, indicating the potent efficacy of merging graph neural networks with time series neural networks.

Visualization of attention weights and variance. We discuss why the proposed RT-GCN model could counteract the impact of noisy and missing data corruption in this subsection.

One of the primary rationales behind introducing RT-GCN is that it would be able to capture the large variance induced by perturbations and utilize the attention weights based on the variance to mitigate the impact of perturbations. To empirically validate these suppositions, we present visual representations of the perturbed input X , alongside the associated attention weights and variances. The visualization is calculated based on the Los-loop dataset under two perturbation scenarios: Gaussian noisy perturbation and missing data perturbation, applied at timestamp 1 and 50% of randomly selected nodes. The results are shown in Fig. 12.

From the presented figure, it is evident that the attention weights associated with the perturbed data are considerably low, while there is a pronounced variance. This observation substantiates our assertion that our RT-GCN effectively captures the variance inherent in the perturbed data. Furthermore, through the proposed attention mechanism, the model mitigates the influence of such perturbed data.

Noise severity impact analysis. To investigate the influence of the magnitude of added Gaussian noise on the model, we conduct experiments using different values of the parameter k in Eq. (22). Specifically, we test values of 0.4, 0.6, 0.8, 1, 1.5, and 2, by adding the corresponding Gaussian noise to the randomly selected 20% historical data of 10% nodes. The predicted performance of representative models GMAN, STNN, and RT-GCN for various noise magnitudes on four datasets are compared in Fig. 13.

As shown in the results, RT-GCN consistently outperforms both GMAN and STNN models, regardless of the magnitude of the added noise. It is worth noting that RT-GCN demonstrates a stable performance, maintaining its high accuracy even when exposed to high levels of noise. To ensure that our experiments align with real-world scenarios, we set $k = 0.8$ in the Gaussian noisy perturbation tests. That is because too small of the value for k , leads to minor changes in the prediction results when the share of perturbed historical data changes. While very large values of k may not be consistent with reality.

6. Conclusions and future research

This study proposes a Gaussian-based Spatiotemporal Graph Convolution Network model for traffic prediction (RT-GCN), which aims to address the vulnerability of deep learning models to corrupted

traffic data. Our objective is to enhance the intrinsic robustness of the model itself, rather than dealing with the dataset in a laborious and suboptimal manner, as is the case with most other studies. Specifically, the RT-GCN model utilizes Gaussian distributed layers instead of plain vectors used in the original GCN and GRU methods, to mitigate the negative effects of outliers or missing information in the dataset. Furthermore, a variance-based attention mechanism is integrated to prevent the propagation of perturbed information through time and space. The experimental results demonstrate that RT-GCN outperforms several competitive models in terms of prediction accuracy under different levels of both noisy and missing data conditions. The authors have also thoroughly examined and validated that RT-GCN remains unaffected by various missing patterns, including temporally correlated and spatially correlated missing modes. Furthermore, it remains robust regardless of the proportion of manipulated nodes or the extent of perturbed historical data per node. This empirical evidence showcases the extensive practical applicability of RT-GCN. One notable advantage of employing RT-GCN for real-time prediction tasks is its ability to bypass the need for preprocessing the dataset to address challenges like missing data, uncertainty, and noise. This characteristic leads to improved efficiency and optimal prediction outcomes.

In terms of future research, there are several potential avenues. Firstly, it would be beneficial to examine the robustness of the proposed model under more complex and diverse scenarios, including changes in the road network structure or node relationships that may arise due to road maintenance or traffic accidents. Additionally, subsequent research will consider exploring the integration of some external factors into the RT-GCN, such as weather, point of interest (POI), periodicity, and events, which could potentially contribute to the development of more efficient transportation systems. Lastly, RT-GCN could be extended to facilitate uncertainty quantification. RT-GCN encapsulates two fundamental facets: the mean and the variance. The variance provides an inherent measure of the model's confidence in its predictions. A smaller variance indicates a tight range around the mean, suggesting the model is more certain about its prediction. Conversely, a larger variance denotes greater uncertainty.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] I. Okutani, Y.J. Stephanedes, Dynamic prediction of traffic volume through Kalman filtering theory, *Transp. Res.* B 18 (1984) 1–11.
- [2] S.V. Kumar, L. Vanajakshi, Short-term traffic flow prediction using seasonal ARIMA model with limited input data, *Eur. Transp. Res. Rev.* 7 (2015) 1–9.
- [3] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, *Transp. Res. C* 90 (2018) 166–180.
- [4] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, B. Yin, Deep learning on traffic prediction: Methods, analysis and future directions, *IEEE Trans. Intell. Transp. Syst.* (2021).
- [5] B. Yu, Y. Lee, K. Sohn, Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN), *Transp. Res. C* 114 (2020) 189–204.
- [6] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, H. Li, A3t-gcn: Attention temporal graph convolutional network for traffic forecasting, *ISPRS Int. J. Geo-Inform.* 10 (2021) 485.
- [7] J. Ye, J. Zhao, K. Ye, C. Xu, How to build a graph-based deep learning architecture in traffic domain: A survey, *IEEE Trans. Intell. Transp. Syst.* 23 (2022) 3904–3924, <http://dx.doi.org/10.1109/TITS.2020.3043250>.
- [8] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, G. Bai, Deep temporal convolutional networks for short-term traffic flow forecasting, *IEEE Access* 7 (2019) 114496–114507, <http://dx.doi.org/10.1109/ACCESS.2019.2935504>.
- [9] G. Guo, W. Yuan, Short-term traffic speed forecasting based on graph attention temporal convolutional networks, *Neurocomputing* 410 (2020) 387–393, <http://dx.doi.org/10.1016/j.neucom.2020.06.001>, <https://www.sciencedirect.com/science/article/pii/S0925231220309504>.
- [10] H. Zhu, Y. Xie, W. He, C. Sun, K. Zhu, G. Zhou, N. Ma, A novel traffic flow forecasting method based on RNN-GCN and BRB, *J. Adv. Transp.* 2020 (2020).
- [11] D. Liu, S. Hui, L. Li, Z. Liu, Z. Zhang, A method for short-term traffic flow forecasting based on GCN-LSTM, in: 2020 International Conference on Computer Vision, Image and Deep Learning, *CVIDL*, IEEE, 2020, pp. 364–368.
- [12] Y. Gui, D. Wang, L. Guan, M. Zhang, Optical network traffic prediction based on graph convolutional neural networks, in: 2020 Opto-Electronics and Communications Conference, *OECC*, 2020, pp. 1–3, <http://dx.doi.org/10.1109/OECC48412.2020.9273638>.
- [13] B.L. Smith, W.T. Scherer, J.H. Conklin, Exploring imputation techniques for missing data in transportation management systems, *Transp. Res.* 1836 (2003) 132–142.
- [14] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, L. Song, Adversarial attack on graph structured data, in: International Conference on Machine Learning, *PMLR*, 2018, pp. 1115–1124.
- [15] L. Zhu, K. Feng, Z. Pu, W. Ma, Adversarial diffusion attacks on graph-based traffic prediction models, 2021, <arXiv:2104.09369>.
- [16] J. Dai, W. Zhu, X. Luo, A targeted universal attack on graph convolutional network, 2020, arxiv preprint <arXiv:2011.14365>.
- [17] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, *PMLR*, 2019, pp. 7472–7482.
- [18] Z. Zhou, Z. Yang, Y. Zhang, Y. Huang, H. Chen, Z. Yu, A comprehensive study of speed prediction in transportation system: From vehicle to traffic, *Iscience* (2022) 103909.
- [19] X. Chen, S. Wu, C. Shi, Y. Huang, Y. Yang, R. Ke, J. Zhao, Sensing data supported traffic flow prediction via denoising schemes and ANN: A comparison, *IEEE Sens. J.* 20 (2020) 14317–14328, <http://dx.doi.org/10.1109/JSEN.2020.3007809>.
- [20] J. Zhao, Y. Gao, Z. Bai, H. Wang, S. Lu, Traffic speed prediction under non-recurrent congestion: Based on LSTM method and beidou navigation satellite system data, *IEEE Intell. Transp. Syst. Mag.* 11 (2019) 70–81, <http://dx.doi.org/10.1109/MITSM.2019.2903431>.
- [21] A. Cini, I. Marisca, C. Alippi, Multivariate time series imputation by graph neural networks, 2021, arxiv preprint <arXiv:2108.00298>.
- [22] Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, *Transp. Res. C* 118 (2020) 102674, <http://dx.doi.org/10.1016/j.trc.2020.102674>, <https://www.sciencedirect.com/science/article/pii/S0968090X20305891>.
- [23] J. Liu, G.P. Ong, X. Chen, Graphsage-based traffic speed forecasting for segment network with sparse data, *IEEE Trans. Intell. Transp. Syst.* 23 (2022) 1755–1766, <http://dx.doi.org/10.1109/TITS.2020.3026025>.
- [24] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, *Sci. Rep.* 8 (2018) 6085.
- [25] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, LSTM-based traffic flow prediction with missing data, *Neurocomputing* 318 (2018) 297–305.
- [26] Z. Zhang, X. Lin, M. Li, Y. Wang, A customized deep learning approach to integrate network-scale online traffic data imputation and prediction, *Transp. Res. C* 132 (2021) 103372.
- [27] X. Chen, L. Sun, Bayesian temporal factorization for multidimensional time series prediction, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (2022) 4659–4673, <http://dx.doi.org/10.1109/TPAMI.2021.3066551>.
- [28] Y. Liang, Z. Zhao, L. Sun, Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns, *Transp. Res. C* 143 (2022) 103826, <http://dx.doi.org/10.1016/j.trc.2022.103826>, <https://www.sciencedirect.com/science/article/pii/S0968090X22002479>.
- [29] A. Wang, Y. Ye, X. Song, S. Zhang, J.J.Q. Yu, Traffic prediction with missing data: A multi-task learning approach, *IEEE Trans. Intell. Transp. Syst.* 24 (2023) 4189–4202, <http://dx.doi.org/10.1109/TITS.2022.3233890>.
- [30] D. Xu, C. Wei, P. Peng, Q. Xuan, H. Guo, GE-GAN: A novel deep learning framework for road traffic state estimation, *Transp. Res. C* 117 (2020) 102635.
- [31] Z. Wang, J. Hu, G. Min, Z. Zhao, J. Wang, Data-augmentation-based cellular traffic prediction in edge-computing-enabled smart city, *IEEE Trans. Ind. Inform.* 17 (2020) 4179–4187.
- [32] S. Sun, C. Zhang, G. Yu, A bayesian network approach to traffic flow forecasting, *IEEE Trans. Intell. Transp. Syst.* 7 (2006) 124–132, <http://dx.doi.org/10.1109/TITS.2006.869623>.
- [33] Y. Wu, J.J. Yu, A bayesian learning network for traffic speed forecasting with uncertainty quantification, in: 2021 International Joint Conference on Neural Networks, *IJCNN*, 2021, pp. 1–7, <http://dx.doi.org/10.1109/IJCNN52387.2021.9533457>.
- [34] J. Xia, S. Wang, X. Wang, M. Xia, K. Xie, J. Cao, Multi-view bayesian spatio-temporal graph neural networks for reliable traffic flow prediction, *Int. J. Mach. Learn. Cybern.* (2022) 1–14.
- [35] R.G. Lopes, D. Yin, B. Poole, J. Gilmer, E.D. Cubuk, Improving robustness without sacrificing accuracy with patch gaussian augmentation, 2019, arxiv preprint <arXiv:1906.02611>.

- [36] Z. He, A.S. Rakin, D. Fan, Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 588–597.
- [37] M. Van Der Voort, M. Dougherty, S. Watson, Combining Kohonen maps with ARIMA time series models to forecast traffic flow, *Transp. Res. C* 4 (1996) 307–318.
- [38] B.L. Smith, M.J. Demetsky, Traffic flow forecasting: comparison of modeling approaches, *J. Transp. Eng.* 123 (1997) 261–266.
- [39] M. Castro-Neto, Y.S. Jeong, M.K. Jeong, L.D. Han, Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions, *Expert Syst. Appl.* 36 (2009) 6164–6173.
- [40] C.P. Van Hinsbergen, T. Schreiter, F.S. Zuurbier, J. Van Lint, H.J. Van Zuylen, Localized extended kalman filter for scalable real-time traffic state estimation, *IEEE Trans. Intell. Transp. Syst.* 13 (2011) 385–394.
- [41] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction, *Sensors* 17 (2017) 818.
- [42] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, S. Ji, Temporal multi-graph convolutional network for traffic flow prediction, *IEEE Trans. Intell. Transp. Syst.* 22 (2020) 3337–3348.
- [43] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2018, <http://dx.doi.org/10.24963/ijcai.2018/505>.
- [44] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, 2019, <arXiv:1906.00121>.
- [45] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-GCN: A temporal graph convolutional network for traffic prediction, *IEEE Trans. Intell. Transp. Syst.* 21 (2020) 3848–3858, <http://dx.doi.org/10.1109/TITS.2019.2935152>.
- [46] R. Vinayakumar, K.P. Soman, P. Poornachandran, Applying deep learning approaches for network traffic prediction, in: 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI, 2017, pp. 2353–2358, <http://dx.doi.org/10.1109/ICACCI.2017.8126198>.
- [47] Z. Zhao, W. Chen, X. Wu, P.C. Chen, J. Liu, LSTM network: A deep learning approach for short-term traffic forecast, *IET Intell. Transp. Syst.* 11 (2017) 68–75.
- [48] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014, arxiv preprint <arXiv:1406.1078>.
- [49] R. Fu, Z. Zhang, L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC, 2016, pp. 324–328, <http://dx.doi.org/10.1109/YAC.2016.7804912>.
- [50] Y. Shi, H. Feng, X. Geng, X. Tang, Y. Wang, A survey of hybrid deep learning methods for traffic flow prediction, in: Proceedings of the 2019 3rd International Conference on Advances in Image Processing, 2019, pp. 133–138.
- [51] M. Cao, V.O. Li, V.W. Chan, A CNN-LSTM model for traffic speed prediction, in: 2020 IEEE 91st Vehicular Technology Conference, VTC2020-Spring, IEEE, 2020, pp. 1–5.
- [52] Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, F.Y. Wang, A hybrid deep learning approach with GCN and LSTM for traffic flow prediction, in: 2019 IEEE Intelligent Transportation Systems Conference, ITSC, IEEE, 2019, pp. 1929–1933.
- [53] L. Tao, Y. Gu, W. Lu, X. Rui, T. Zhou, Y. Ding, An attention-based approach for traffic conditions forecasting considering spatial-temporal features, in: 2020 IEEE 5th International Conference on Intelligent Transportation Engineering, ICITE, 2020, pp. 117–122, <http://dx.doi.org/10.1109/ICITE50838.2020.9231367>.
- [54] T. Bogaerts, A.D. Masegosa, J.S. Angarita-Zapata, E. Onieva, P. Hellinckx, A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data, *Transp. Res. C* 112 (2020) 62–77.
- [55] C. Zheng, X. Fan, C. Wang, J. Qi, GMAN: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 1234–1241, <http://dx.doi.org/10.1609/aaai.v34i01.5477>.
- [56] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, *Proc. AAAI Conf. Artif. Intell.* 34 (2020) 914–921, <http://dx.doi.org/10.1609/aaai.v34i01.5438>.
- [57] C. Li, H. Zhang, Z. Wang, Y. Wu, F. Yang, Spatial-temporal attention mechanism and graph convolutional networks for destination prediction, *Front. Neurorobotics* 16 (2022) 925210.
- [58] H. Liu, C. Zhu, D. Zhang, Q. Li, Attention-based spatial-temporal graph convolutional recurrent networks for traffic forecasting, 2023, <arXiv:2302.12973>.
- [59] J. Jiang, C. Han, W.X. Zhao, J. Wang, PDFomer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction, 2023, <arXiv:2301.07945>.
- [60] Z.A. Sahili, M. Awad, Spatio-temporal graph neural networks: A survey, 2023, <arXiv:2301.10569>.
- [61] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013, arxiv preprint <arXiv:1312.6199>.
- [62] L. Sun, Y. Dou, C. Yang, J. Wang, P.S. Yu, L. He, B. Li, Adversarial attack and defense on graph data: A survey, 2018, arxiv preprint <arXiv:1812.10528>.
- [63] X. Wang, M. Cheng, J. Eaton, C.J. Hsieh, F. Wu, Attack graph convolutional networks by adding fake nodes, 2018, arxiv preprint <arXiv:1810.10751>.
- [64] D. Zügner, A. Akbarnejad, S. Günnemann, Adversarial attacks on neural networks for graph data, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2847–2856.
- [65] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, J. Tang, Graph structure learning for robust graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 66–74.
- [66] L. Qu, L. Li, Y. Zhang, J. Hu, PPCA-Based missing data imputation for traffic flow volume: A systematical approach, *IEEE Trans. Intell. Transp. Syst.* 10 (2009) 512–522, <http://dx.doi.org/10.1109/TITS.2009.2026312>.
- [67] G. Li, V.L. Knoop, H. van Lint, Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations, *Transp. Res. C* 128 (2021) 103185.
- [68] A. Liu, C. Li, W. Yue, X. Zhou, Real-time traffic prediction: A novel imputation optimization algorithm with missing data, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7.
- [69] L. Ge, H. Li, J. Liu, A. Zhou, Traffic speed prediction with missing data based on TGCN, in: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, 2019, pp. 522–529, <http://dx.doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00130>.
- [70] Z. Cui, L. Lin, Z. Pu, Y. Wang, Graph Markov network for traffic forecasting with missing data, *Transp. Res. C* 117 (2020) 102671, <http://dx.doi.org/10.1016/j.trc.2020.102671>, <https://www.sciencedirect.com/science/article/pii/S0968090X20305866>.
- [71] J. Zuo, K. Zeitouni, Y. Taher, S. Garcia-Rodriguez, Graph convolutional networks for traffic forecasting with missing values, *Data Min. Knowl. Discov.* 37 (2023) 913–947.
- [72] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arxiv preprint <arXiv:1609.02907>.
- [73] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arxiv preprint <arXiv:1312.6114>.
- [74] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018, <arXiv:1707.01926>.
- [75] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, LSTM-based traffic flow prediction with missing data, *Neurocomputing* 318 (2018) 297–305, <http://dx.doi.org/10.1016/j.neucom.2018.08.067>, <https://www.sciencedirect.com/science/article/pii/S0925231218310294>.
- [76] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, 2014, arxiv preprint <arXiv:1409.1259>.
- [77] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 922–929.
- [78] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 17804–17815, https://proceedings.neurips.cc/paper_files/paper/2020/file/c1eaad92b939420fc17005e5461e6f48-Paper.pdf.
- [79] S. Yang, J. Liu, K. Zhao, Space meets time: Local spacetime neural network for traffic flow forecasting, in: 2021 IEEE International Conference on Data Mining, ICDM, 2021, pp. 817–826, <http://dx.doi.org/10.1109/ICDM51629.2021.00093>.
- [80] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, B. Du, Adaptive spatio-temporal graph neural network for traffic forecasting, *Knowl.-Based Syst.* 242 (2022) 108199, <http://dx.doi.org/10.1016/j.knosys.2022.108199>, <https://www.sciencedirect.com/science/article/pii/S0950705122000508>.