Information Sciences 677 (2024) 120911

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Di-GraphGAN: An enhanced adversarial learning framework for accurate spatial-temporal traffic forecasting under data missing scenarios

Lincan Li [a], Jichao Bi [b,c,*], Kaixiang Yang [d], Fengji Luo [e]

[a] School of Computer Science and Engineering, University of New South Wales, Sydney, 2052, Australia
[b] Zhejiang Institute of Industry and Information Technology, Hangzhou, 310012, China
[c] School of Big Data and Software Engineering, Chongqing University, Chongqing, 400044, China
[d] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510641, China
[e] School of Civil Engineering, The University of Sydney, Sydney, 2006, Australia

ARTICLE INFO

Keywords:
Spatial-temporal traffic forecasting
Sequential data imputation
Graph attention networks
Generative adversarial networks
Dynamic spatiotemporal dependency modeling

ABSTRACT

Nowadays, various disturbances in urban transportation data acquisition/processing/storage lead to the inevitable data missing problem, which undermines the valuable traffic information and greatly threats the reliability of existing benchmark traffic prediction models. Inspired from the powerful generative learning ability of GANs, we propose an integrated spatiotemporal Data imputation Graph Attention Generative Adversarial Networks (Di-GraphGAN) for accurate and efficient spatial-temporal traffic forecasting under data missing scenarios. Specifically, we first propose a traffic data imputation module named DI-LSTM, which adopts the architecture of LSTM Network with an extra Time Damping unit to accurately estimating the missing values. Then, we facilitate Di-GraphGAN with an original developed Task-Efficient Graph Attention Networks (TE-GAT) for better graph representation learning and a Temporal Contextual Attention (TCA) mechanism to capture the dynamic spatiotemporal traffic patterns. Finally, extensive evaluations are conducted on two real-world traffic speed datasets from China, demonstrating that Di-GraphGAN achieves state-of-the-art performance in both traffic forecasting and spatiotemporal data imputation tasks.

1. Introduction

Intelligent Transportation System (ITS) is an advanced paradigm for traffic management and future smart cities, especially in combination with Internet of Vehicles (IoV) [1,2]. IoV is useful for estimating traffic network capacity, preventing traffic accidents, and guiding every participating vehicles. With the intelligent vehicular technology, traffic managers can easily assess traffic capacity and provide early traffic route guidance for each vehicle [3,4]. Therefore, accurate traffic prediction is critical in urban transportation systems.

The accessibility of massive traffic network data has been enabled by the emerging mobile Internet and intelligent vehicular technologies. However, the inevitability of data missing problem impairs the valid information contained in traffic data. For trans-

* Corresponding author at: Zhejiang Institute of Industry and Information Technology, Hangzhou, 310012, China.
E-mail addresses: lincan.li@unsw.edu.au (L. Li), jonny.bijichao@zju.edu.cn (J. Bi), yangkx@scut.edu.cn (K. Yang), fengji.luo@sydney.edu.au (F. Luo).

https://doi.org/10.1016/j.ins.2024.120911
Received 1 October 2023; Received in revised form 16 March 2024; Accepted 4 June 2024

Available online 7 June 2024

portation systems using speed detectors, speed observations may be lost because of hardware or communication failures [5]. For crowdsourcing systems, traffic data cannot be acquired if no participating vehicles are located on a certain road within a certain time interval. Hence, imputing the incomplete traffic data is critical for improving data quality and making better traffic predictions. Missing data can cause a significant decrease in model performance. The majority of existing traffic prediction models have a limited application scope because they fail to handle missing data problem [6–8]. Some literature adopted two separate procedures to address data imputation and traffic prediction independently [9]. The main limitations of this approach are that not only the whole procedure is time-wasting but also the traffic data cannot be modeled effectively in the prediction stage.

In recent years, both intelligent urban computing and computer science communities have been actively studying on spatial-temporal traffic forecasting. Liang et al. [10] proposed STRN, which adopts a backbone network then a global spatial-relationship module and meta learning to perform fine-grained urban flow forecasting. Guo et al. [11] proposed ASTGNN, which explicitly captures the temporal and spatial traffic dynamics via attention mechanism and dynamic GCN module, respectively. Bi-STAT [12] is an adaptive Transformer model which adopts two separate decoders for bi-directional horizon modeling in traffic forecasting. In recent top AI conferences, researchers employed various kinds of Graph Neural Networks (GNN) in combination with other advanced deep learning technologies to improve spatiotemporal forecasting performance [13–15]. Zhang et al. presented FASTGNN [16], which adopts federated learning for privacy-based traffic network topology graph construction. The authors of STAN [17] proposed to use transfer learning framework with elaborately designed spatial and temporal adaptation modules for traffic prediction in a new city. Nevertheless, there are still some critical challenges waiting for further investigation: (i) The mainstream traffic forecasting models usually concentrate on predicting short-term one-step futures, whereas the industry urgently calls for an efficient and accurate model addressing the Long Sequence Time-series Forecasting (LSTF) task. (ii) Spatial-temporal traffic prediction becomes more challenging under the influence of incomplete traffic data. Such situations strongly call for an integrated deep learning approach for End2End traffic data imputation and prediction. (iii) The patterns of citywide traffic network data are complex. In reality, neither a single CNN/RNN nor GCN-based modeling methods can fully extract the explicit and implicit spatiotemporal dependencies amongst the road segments or sensors in an urban network.

To address the above uncovered research gaps, in this article we propose Di-GraphGAN (which stands for spatial-temporal **D**ata **i**mputation **Graph** Attention **G**enerative **A**dversarial **N**etworks) for integrated traffic data imputation and multi-scale traffic forecasting. Specifically, the generator of Di-GraphGAN is formed as an Encoder-Decoder structure for efficient LSTF task. Moreover, we propose DI-LSTM as the data imputation mechanism, which incorporates a novel imputation unit into LSTM network. DI-LSTM is added into the generator to estimate missing traffic values in the raw input incomplete dataset. In addition, a Task-Efficient Graph Attention Network (TE-GAT) is designed to remove the redundancy in neighborhood aggregation found in the existing graph attention networks [14,15] while simultaneously improving accuracy and computational efficiency. The main contributions of this research are summarized in the following:

- We propose an enhanced spatial-temporal adversarial learning framework named Di-GraphGAN for end2end accurate traffic data imputation and multi-scale prediction under various data missing scenarios.
- An elaborately designed data imputation module called DI-LSTM is proposed, which adopts the architecture of LSTM network with an internal Time Damping unit to impute missing traffic values, thus help improving traffic prediction accuracy.
- We introduce Task Efficient Graph Attention Networks (TE-GAT), which employs a Filter Gate to determine the importance of each edge and retain only the necessary edges to be used for neighbor aggregation, thus reducing the computational overhead in existing graph attention and improving accuracy.
- Comprehensive experiments and evaluations are conducted on two modern city traffic speed datasets from Hangzhou and Guangzhou city, demonstrating that Di-GraphGAN consistently outperforms other benchmark methods.

The remainder of this paper is organized as follows. Section 2 elaborates the existing study on the research related to this work. Section 3 defines the basic preliminaries in this study. Section 4 formulates the model architecture of Di-GraphGAN and introduces related technical details, including the proposed DI-LSTM, TE-GAT, the generator and discriminator of Di-GraphGAN, the model optimization and training methods. Following that, we carry out extensive real-world experiments and performance evaluations in section 5. Finally, in section 6, we conclude this study and suggest potential research directions.

2. Literature review

2.1. Sequential missing data estimation methods

As a key factor in spatial-temporal analysis, missing data imputation has received considerable attention from researchers. Classical data imputation approaches can be categorized into three types: interpolation-based model, statistical-based model, and matrix/tensor factorization model [18]. K-nearest neighbors (KNN) [19] is a popular interpolation-based method that employs the mean value of neighboring data points to interpolate the missing data points. However, interpolation-based methods can only apply to a single traffic sensor or a road segment but are not feasible for large-scale traffic networks [18]. In order to establish the probability distribution, statistical-based models such as ARIMA [20] and its variants require a complete dataset (i.e. data missing rate is 0%) in advance, which is not feasible in real-world applications. Matrix/tensor factorization is another approach to achieve satisfactory data imputation results. In [21], the authors proposed a low-rank matrix factorization method which adopted temporal regularization constraints. Chen et al. [22] proposed Bayesian Temporal Matrix Factorization (BTMF) model to estimate the missing values

in spatial-temporal tasks. The major drawbacks of Matrix/tensor factorization models are that they require expensive computing resources and cannot reconstruct missing data in real-time.

More recently, researchers have adopted deep learning techniques to address time-series missing data imputation. Yoon et al. [23] presented an adversarial learning data imputation network named GAIN. The GAIN model feeds a hint vector into the discriminator to provide additional information for missing data. Luo et al. [24] proposed E2GAN as an end2end adversarial learning model for multivariate sequential data imputation. However, most existing deep learning models focus on capturing the global temporal correlation but ignore the hidden correlation between different hierarchy time scales.

2.2. Spatial-temporal traffic forecasting approaches

Generally speaking, traffic forecasting approaches can be classified as statistical methods and deep learning methods. Traditional statistical-based methods have made great contributions to traffic prediction. Ahmed et al. [25] first adopted ARIMA model to solve traffic prediction problems. Following that, Williams et al. [26] applied seasonal ARIMA to predict short-term traffic flows of transportation networks. However, statistical methods heavily rely on the assumption of stationary time series data, which results in extreme failure when applying to sequential data with relatively high fluctuations or large data missing rates.

Deep learning-based traffic prediction has become popular and attracted great attention in the last decade. Deep learning-based methods can be further classified as spatial dependency modeling and temporal dependency modeling.

Methods for modeling spatial dependencies. Convolutional Neural Network (CNN) and its variants are widely employed to extract spatial correlations from Euclidean data [27–29]. [29,30] adopted graph embedding to model spatial dependencies and generate embedded low-dimensional vectors for subsequent network processing. Graph convolutional network (GCN) is an efficient and flexible approach to capture the complex spatial correlations for non-Euclidean data. In [31–33], the complex spatial correlations in transportation networks were modeled using a variety of GCN layers. ASTGCN [34] utilized attention-based graph convolution networks to model the spatiotemporal traffic flow dynamics. STSGCN [32] introduced a synchronous modeling mechanism to capture the heterogenous urban spatiotemporal correlations simultaneously. RGSL [15] introduced a regularization method for implicit graph structure generation and combine the generated implicit graph knowledge with explicit traffic graph information using a Laplacian module.

Methods for modeling temporal dependencies. Recurrent Neural Networks (RNN) and the variants are proposed for sequential data modeling and have powerful abilities to capture temporal dependency [35,36,29]. Following that, GRU and LSTM further improved the long-term temporal modeling ability while avoided the gradient vanishing issue. CNNs also demonstrate powerful ability in capturing temporal correlations. For example, Temporal Convolutional Network (TCN) [37] has more simple architecture than recurrent architectures such as GRUs and LSTMs, and outperforms various RNN-based methods. Recently, attention mechanism has been adopted for modeling sequential data. Transformer is a representative self-attention model for Natural Language Processing and it has recently been employed to address spatiotemporal traffic prediction tasks [38–40]. However, it is hard for Transformer models to capture long-term temporal dependencies.

3. Preliminaries

Definition 1: Transportation Network g. We use graph $g = (V, E, A)$ to represent a transportation network. Here, V denotes a collection of vertices representing all the nodes. In this paper, a node is an individual road segment in the transportation network. E denotes a collection of edges, and $A = (a_{ij})^{N \times N}$ denotes the adjacency matrix of g. Each element a_{ij} of A indicates the calculated proximity between two vertices (e.g. distance between two road segments, geographical connectivity, POI similarity, etc.).

Definition 2: Multivariate Spatial-Temporal Data with Missing Values. The training historical traffic data can be represented as a three-dimensional tensor $X \in R^{T \times N \times C}$. Here, T is the temporal length of traffic data; C is the number of traffic features, which includes the ground-truth traffic sequential data and various semantic features; N is the total number of node individuals. Time sliding window is used in this work for slicing the sequential data into foxed-length smaller chunks. Let's set the length of sliding window as l. Then, at time step t, the input data X_t is in the shape of $X_t \in R^{l \times N \times C}$. To represent the missing traffic data at time interval t, the masking matrix $m_t \in \{0, 1\}^{l \times N \times C}$ is used to indicate whether traffic data X_t is missing. The masking matrix for X_t is formulated as:

$$m_t^c = \begin{cases} 1, & \text{if } X_t^c \text{ is observed} \\ 0, & \text{if } X_t^c \text{ is missing} \end{cases} \tag{1}$$

In multivariate sequential data, a feature in dimension $c (c \in C)$ can be missing for several continuous timestamps. Thus, we employ φ_t^c to record the time lag between the last observed data point to the present timestamp s_t, as shown in Eq. (2):

$$\varphi_t^c = \begin{cases} s_t - s_{t-1}, & \text{if } t > 1 \text{ and } m_{t-1}^c = 1 \\ \varphi_{t-1}^c + s_t - s_{t-1}, & \text{if } t > 1 \text{ and } m_t^c = 0 \\ 0, & \text{if } t = 1 \end{cases} \tag{2}$$

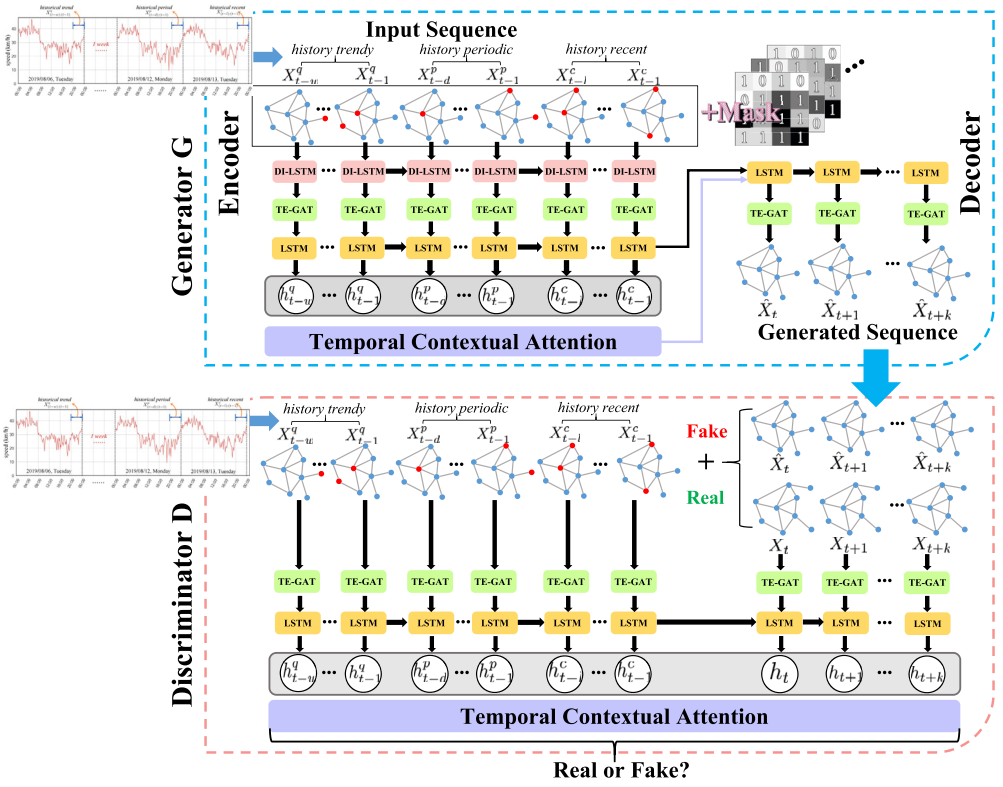

Fig. 1. Detailed architecture of the proposed Di-GraphGAN model.

Problem Statement. Given historical observed traffic matrix series $[X_{t-p}, X_{t-p+1}, ..., X_t]$ and the graph $g = (V, E, A)$, our goal is to train a neural network model with a learned mapping function f, which can generate the next k-step traffic predictions and impute the missing traffic data in the meantime:

$$[X_{t-p}, X_{t-p+1}, ..., X_t; g] \xrightarrow{f} [\hat{X}_{t+1}, ...\hat{X}_{t+k}] \tag{3}$$

4. Methodology

In this section, we introduce the proposed Di-GraphGAN in detail. As shown in Fig. 1, Di-GraphGAN is a GAN-based framework which consists of a generator G and a discriminator D as the primary components. G is developed as an encoder-decoder structure to produce the future k-step forecasts based on historical traffic data. Then the generated traffic forecasts and the corresponding real traffic speed data are jointly fed into D for training.

4.1. DI-LSTM: data-imputation LSTM module

Previous literature [41,42] suggested to use some pre-defined values such as zeros or the last observed values to impute missing data points. However, pre-defined imputation methods always lead to biased parameter estimation, resulting in unsatisfied prediction performances. In this research, we propose DI-LSTM (as shown in Fig. 2), a data imputation module using LSTM architecture with an internal time damping unit θ_{TD} to control the influence of different historical observations on the current timestamp data so as to achieve accurate missing data imputation.

In order to make the gradients in our deep model can be computed in implementation, the masking matrix m_t is first applied to the incomplete traffic matrix X_t to replace the missing data points in X_t with 0:

$$X'_t = m_t \odot X_t \tag{4}$$

Following that, we present the time damping vector θ to adjust the impacts of historical data on the current time step observation. To be specific, if the recorded time lag between the last valid observation and the present timestamp is very large, there must be a large time gap between the last valid data point and the current timestamp observation. Under this circumstance, the last valid data point should give less influence on the current step data. Also, each element value of θ should be in range $[0, 1]$. In light of this, we formulate the time damping vector θ as follows:

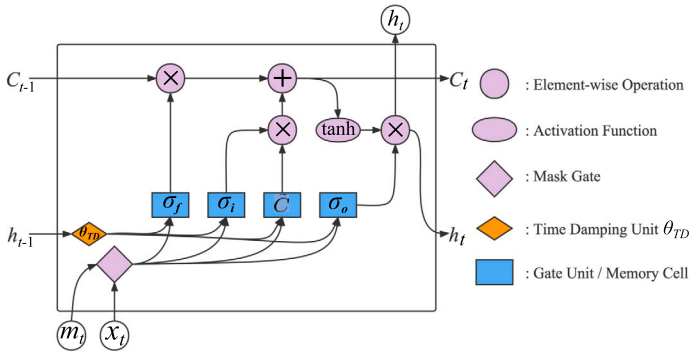

Fig. 2. Structure illustration of the proposed Data Imputation LSTM Module (DI-LSTM).

$$\theta_t = \frac{1}{e^{max(0, W_\theta \varphi_t + b_\theta)}} \tag{5}$$

where W_θ and b_θ are learnable parameters of the DI-LSTM. We adopt the exponential formulation $e^{max(\cdot)}$ in the denominator so as to ensure the time damping vector $\theta_t \in (0, 1]$.

After we obtain θ_t, it is then used to update the DI-LSTM cell hidden state h_{t-1} by element-wise multiplication as in Eq. (6), which is easy to understand and implement.

$$h'_{t-1} = \theta_t \odot h_{t-1} \tag{6}$$

The following procedure of DI-LSTM after we obtain the updated hidden state h'_{t-1} can be formulated as Eq. (7) - Eq. (12):

$$f_t = \sigma_f(W_f \cdot X'_t + P_f \cdot h'_{t-1} + Q_f \cdot m_t + b_f) \tag{7}$$

$$i_t = \sigma_i(W_i \cdot X'_t + P_i \cdot h'_{t-1} + Q_i \cdot m_t + b_i) \tag{8}$$

$$o_t = \sigma_o(W_o \cdot X'_t + P_o \cdot h'_{t-1} + Q_o \cdot m_t + b_o) \tag{9}$$

$$\tilde{C}_t = \tanh(W_{\tilde{C}} \cdot X'_t + P_{\tilde{C}} \cdot h'_{t-1} + Q_{\tilde{C}} \cdot m_t + b_{\tilde{C}}) \tag{10}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{11}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{12}$$

where Q_f, Q_i, Q_o, Q_C denote the learnable weights of masking matrix m_t in units $\sigma_f, \sigma_i, \sigma_o, \sigma_C$, respectively. If the original input traffic data X_t is complete with no missing values, all the items of m_t are set as 1 and the operations are resemble to LSTM network.

4.2. TE-GAT: task efficient graph attention networks

Existing standard GATs use attention mechanism to calculate edge weights at each layer based on node features, and interact among all the neighbors for graph representation learning, which not only limit the algorithm performance, but also increase the computational cost. Furthermore, standard GATs use multi-head attention to enhance the expressiveness of the model, resulting in multiple sets of redundant attention weights. To alleviate the above issues, we propose a task-efficient GAT by simplifying the multiple attention weights calculation and constraint the participated graph edges. The insight behind task-efficient graph attention networks (TE-GAT) is that not all node neighbors are equally important for graph aggregation, and we wish to select the significant neighbors but ignore the less important individuals for graph attention. The key technology is that we employ a Filter Gate to each edge individual to determine if that edge should be used for graph neighbor aggregation or not.

To identify the important edges of a graph and remove irrelevant/noisy edges, we apply a binary-value Filter Gate $u_{ij} \in \{0, 1\}$ to each edge $e_{ij} \in E$. Under this setting, the filter gate u_{ij} will determine whether an edge e_{ij} will be used for neighbor aggregation or not, which can be formulated as follows:

$$A^\star = A \odot U, \quad U \in \{0, 1\}^M \tag{13}$$

where M demotes the total number of edges in graph g.

Since our objective is to use less edges and meanwhile encourage the model prediction accuracy to be even better, we hereby train the TE-GAT's weight parameters W_{tegat} and Filter Gate U by minimizing a L_0-norm regularization loss:

$$L_{reg} = \frac{1}{n} \sum_{i=1}^{n} L(f_i(X_i, A^\star, W_{tegat}), y_i) + \lambda ||U||_0 \tag{14}$$

$$= \frac{1}{n}\sum_{i=1}^{n} L(f_i(X_i, A^{\star}, W_{tegat}), y_i) + \lambda \sum_{(i,j)\in E} 1_{[u_{ij}\neq 0]} \tag{15}$$

where $||U||_0$ denotes the L_0-norm of the Filter Gate U, i.e., the number of non-zeroes elements in U, $1_{[con]}$ is an indicator function which equals to 1 if the condition con is satisfied, and 0 otherwise, y_i is the ground-truth prediction of X_i, and λ is a regularization hyper-param that balances between data loss and edge sparsity.

The insight of our approach is that we employ a binary Filter Gate U that serves as a proxy to L_0 norm, which allows us to determine the presence or absence of edges in the graph efficiently. This is not the actual L_0 optimization but a simplification, where U consists of binary values that are much less computationally demanding to optimize. We use a regularized loss function that includes L_0 norm of U as a penalty term to encourage sparsity in the graph structure. The L_0 norm in this context is used as an indicator function, counting the number of non-zero elements.

Following that, TE-GAT's attention-based learning functions $f(X, A^{\star}, W_{tegat})$ can be formulated as:

$$h_i^{(l+1)} = \sigma\Big(\sum_{j\in N_i} \alpha_{ij} h_j^{(l)} W_{tegat}^{(l)}\Big) \tag{16}$$

where α_{ij} is the identical coefficient assigned to edge e_{ij} across all layers. This is in contrast with common GATs, where a layer-dependent attention coefficient $\alpha_{ij}^{(l)}$ is assigned for edge e_{ij} at the l-th layer.

The attention coefficient $\alpha_{ij}^{(l)}$ is calculated as follows:

$$\alpha_{ij} = Norm(A_{ij}u_{ij}) = \frac{A_{ij}u_{ij}}{\sum_{k\in N_i} A_{ik}u_{ik}} \tag{17}$$

where $Norm(A_{ij}u_{ij})$ stands for the row-scale normalization of $A^{\star}, (A^{\star} = A \odot U)$.

Compared with GATs, we do not use SoftMax function, because by setting $u_{ij} \in \{0, 1\}$ and $A_{ij} \geq 0$ we can obtain reasonable attention coefficients. Intuitively, a node j is always important for itself, thus we set $u_{jj} = 1$. Finally, we also employ multi-head attention to improve the representation learning ability of TE-GAT. The multi-head TE-GAT can be formulated as:

$$h_i^{(l+1)} = Concat\left[\sigma\Big(\sum_{j\in N_i} \alpha_{ij} h_j^{(l)} W_s^{(l)}\Big)\right]_{s=1}^{S} \tag{18}$$

where S denotes the total number of attention heads, $Concat[\cdot]$ denotes the concatenation function, α_{ij} denotes the attention coefficients computed by Eq. (17), and $W_s^{(l)}$ is the weight matrices of k-th head at l-th layer.

4.3. Generator of di-GraphGAN

In our task, the generator G is formulated as Encoder-Decoder structure to efficiently produce multi-scale traffic predictions. The input historical data of Di-GraphGAN contains multi-horizon temporal scales from coarse to fine-grained (i.e. trendy, periodic and recent) for more accurate traffic forecasting. In the Encoder part, as the raw input traffic data is incomplete with missing values, it is first processed by DI-LSTM module to obtain an imputed complete traffic data $X' = \{X'_{t-T+1}, ..., X'_t\} \in R^{T\times N\times C}$. Following that, the imputed data sequence is fed into Task-Efficient GAT network to capture the heterogeneous spatial dependencies, then sent to LSTM network to capture the temporal dependencies. At time interval t, the hidden states of traffic data sequence at LSTM network can be represented as $H = [h_{t-T+1}, ..., h_t]$.

Traffic features from different historical timestamps have different degrees of influence on the prediction time step, and the influence change dynamically under different external contexts. RNN networks always face the challenges in (i) Capturing the multi-scale temporal patterns due to fixed-length time steps that may not align well with the varying traffic conditions, and (ii) Accounting for the external dynamic factors that significantly influence traffic capacity, such as weather conditions, holidays, and events. In considering that, the Temporal Contextual Attention (TCA) mechanism is designed to overcome these limitations by providing a more nuanced attention scheme that considers the heterogeneity of traffic patterns and the influence of external contextual factors. It dynamically assigns attention weights to different historical time steps, thereby enhancing the LSTM's ability to generate more accurate traffic predictions.

TCA is integrated into both G and D after the LSTM layer. TCA mechanism is capable of capturing informative local contextual knowledge for traffic forecasting and can be fit into every time step of RNN network to improve the attention performance. In our case, TCA captures the dynamic correlations of different historical steps on the next LSTM step by computing the attention scores through probability distribution then integrate with input data at the present time step. The architecture of temporal contextual attention is shown in Fig. 3. To be specific, the hidden state of LSTM at time step t (denote as h_t) is firstly input to a fully-connected network in order to generate a probability distribution according to the previously learned features of the model. Following that, the sigmoid function is adopted to re-scale the probability distribution into range $[0, 1]$ thus produce suitable values to be used as attention weights. At time step t, the attention weights $\Gamma_t = [\gamma_1, \gamma_2, ..., \gamma_{t-1}, \gamma_t]$ produced by TCA mechanism can be formulated as:

$$\Gamma_t = Sigmoid(FC(h_t; \theta_t)) \tag{19}$$

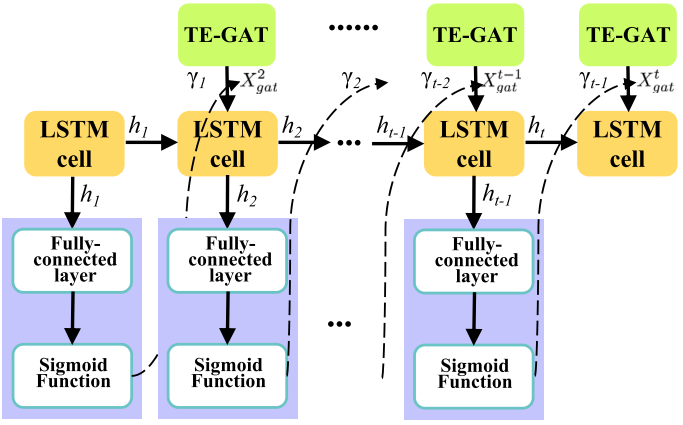

Fig. 3. Visualization of the temporal contextual attention mechanism.

where $FC(\cdot)$ represents the fully-connected layer, $Sigmoid(\cdot)$ denotes sigmoid activation function. θ_t represents the learned model parameters.

Let X^t_{gat} be the output of the TE-GAT layer at time step t. X^t_{gat} is fed into LSTM network as the original traffic feature data. Meanwhile, the temporal contextual attention recurrently assigns corresponding weights to different elements of X^t_{gat}, making the model pay attention to relevant important factors. The proposed attention weight scheme can be formulated as:

$$X^t_{att} = X^t_{gat} \cdot \Gamma_{t-1} \tag{20}$$

where $\cdot$ is the element-wise multiplication, X^t_{att} is the input for next step LSTM cell representing the original input data been processed with attention weights. Γ_{t-1} denotes the learned temporal contextual weight tensor.

For the decoder part of G, the hidden state h_t and the traffic embedding X^t_{LSTM} learned by LSTM are input to the decoder part LSTM network, then we gain the output $\hat{Y}_{de} = \{y_{t+1}, ..., y_{t+k}\}$. $\hat{Y}_{de}$ is processed by TE-GAT layer to generate the final prediction of k time steps future traffic data $\hat{X} = \{\hat{X}_{t+1}, ..., \hat{X}_{t+k}\}$.

4.4. Discriminator of di-GraphGAN

The architecture of discriminator is illustrated in the bottom half of Fig. 1, it mainly consists of TE-GAT, LSTM and TCA layers. Before introducing the discriminative procedure of D, we first elaborate the graph construction in both G and D. Graph adjacency matrix $A = (a_{ij})^{N\times N}$ is the critical component in graph neural networks to measure the spatial correlations among nodes in a traffic network. We adopt the connectivity graph adjacency matrix [14,32]. If road segment i (node V_i) and road segment j (node V_j) are geographical neighbors, the corresponding element a_{ij} in $A = (a_{ij})^{N\times N}$ is set as 1, otherwise 0. i.e., $a_{ij} = 1$ if node V_i connects to node V_j; $a_{ij} = 0$ otherwise.

Given a set of real-world traffic data and generated traffic data $\{X_i, \hat{X}_i\}^n_{i=1}$, the discriminator D attempts to discriminate between the two. D's distinguish error can rectify G to approximate the ground-truth data distribution thus producing more authentic data. Prior to entering D, historical traffic data $\{X_{t-T+1}, ..., X_t\}$ are merged with the ground-truth future data $\{X_{t+1}, ..., X_{t+k}\}$ and the generated future data $\{\hat{X}_{t+1}, ..., \hat{X}_{t+k}\}$ respectively to form the input real traffic sequence and fake traffic sequence. The two sets of samples are fed into TE-GAT network and LSTM network to capture spatiotemporal dependencies. At last, the temporal contextual attention (mentioned above in Generator G) is employed to process the outcome of LSTM. The discriminator of Di-GraphGAN adopts a new objective function, therefore it is no longer a direct criticizer but rather an assistant in measuring the Wasserstein distance between generated data distribution and real data distribution.

4.5. Objective function of di-GraphGAN

The overall objective function of Di-GraphGAN is an integrated and weighted loss, which consists of generative adversarial learning loss, data reconstruction loss, and the L_0 regularization loss function L_{reg} for TE-GAT. We take the form of Wasserstein GAN (WGAN) [43] framework, which employs Wasserstein distance to estimate the proximity of generated data distribution to the ground-truth data distribution. Suppose we have n pairwise traffic speed samples $\{X_i, \hat{X}_i\}^n_{i=1}$, the adversarial loss of Di-GraphGAN (denoted as L_{GAN}) can be formulated as:

$$L_{GAN} = \min_{\theta} \max_{w} \sum_{i=1}^{n} (f_w(X_i)) - \sum_{i=1}^{n} (f_w(g_\theta(\hat{X}_i))) \tag{21}$$

where θ denotes the generator's learnable parameters, $f_w(\cdot)$ is a class of parameterized mapping functions which are all K-Lipschitz for K.

Given an incomplete traffic data series X, how to impute the missing data points in X with the most authentic values is an essential problem. Therefore, we introduce the **masked reconstruction loss**, which can measure the degree of data imputation fitness by calculating the masked square error between the incomplete traffic data X and the generated sample $G(X)$:

$$L_{recon} = \|X \odot M - G(X) \odot M\|_2^2 \tag{22}$$

where $\|\cdot\|_2^2$ denotes the squared L2-norm function, $\odot$ denotes the element-wise multiplication. L_{recon} calculates the errors for both observed traffic data and the missing traffic values.

The L_0 regularization loss function L_{reg} for our task-efficient GAT has already been elaborated in Eq. (15). When fitting it into the overall objective function, its importance should be considered. Intuitively, L_{reg} is less significant than L_{GAN} and L_{recon}, which focus on the major tasks. Considering that, we assign a small weight parameter to L_{reg}, and search from $\{0.01, 0.05, 0.08, 0.1, 0.15\}$ to find the best weight setting. Through our experiments, we find $0.08L_{reg}$ shows the optimal performance.

Moreover, to flexibly tune-up the significance of L_{GAN} and the L_{recon}, we assign a weight parameter ϵ to the overall loss function. Integrating L_{GAN}, L_{recon}, L_{reg} and weight parameters, the overall objective function is:

$$L_{\text{Di-GraphGAN}} = \epsilon L_{GAN} + (1-\epsilon)L_{recon} + 0.08L_{reg} \tag{23}$$

The training procedure can be briefly described as follows: for an incomplete input traffic sequential data X, it is fed into the generator to obtain $G(X)$. Then, $G(X)$ and the corresponding real data X are compared using the discriminator. Di-GraphGAN model is trained by optimize $L_{\text{Di-GraphGAN}}$ through back-propagation. Finally, when the total loss function $L_{\text{Di-GraphGAN}}$ is converged, we consider the model is well-trained and replace the missing data points in X with the generated data points in $G(X)$: $X_{imputed} = X \odot M + (1-M) \odot G(X)$.

5. Experiments

In this section, we first introduce the two real-world traffic speed datasets, experiment configurations and baseline methods. Then we show the experimental results under different traffic data missing scenarios and analyze the performances. Finally, we carry out ablation study and sensitive study to evaluate the effectiveness of each model component and the proposed loss function.

HangZhou-speed dataset. This is an originally made dataset by us, which consists of 30000 vehicles' trajectory data in Gongshu District of Hangzhou city from Jan/01/2019 to May/31/2019. During the five-month period, there are regular working days and weekends, as well as major holidays such as Chinese New Year Festival. We carefully divide the traffic network of Gongshu District into 116 road segments using 72 major road intersections. The traffic network is represented by a 116×116 graph adjacency matrix, in which each individual is a unique road segment. For each road segment, we collect its average traffic speed values at 10-minute time interval, and the total length of traffic speed time series is 21,744.

GuangZhou-speed dataset. This dataset records traffic speed data from the transportation network in Guangzhou, China. In GuangZhou-speed, 214 road segments in the transportation network (include urban expressways and arterials) are selected within two months from August 1st, 2016 to September 30th, 2016. The time interval of traffic speed records is 10 minutes. The connectivity of this traffic network is represented by a 214×214 adjacency matrix, and the total length of traffic speed time series is 8784.

Implementation Details. Di-GraphGAN is implemented with PyTorch framework on two NVDIA RTX 4090 80 GB GPU. To begin with, the original traffic speed data is normalized into range $[0, 1]$ using the Min-Max Normalization, we then re-scale the prediction results back to their normal values for assessment. After normalization, we add masking vectors to represent the corresponding missing scenario and construct the input for Di-GraphGAN. The multi-horizon historical time step length for recent/periodic/trendy is set to 8,4,4, respectively. The hidden state size of LSTM is 200. For the TE-GAT network in both generator and discriminator, we stack two attention layers with 8-head attention (i.e., head number $S = 8$) for each layer. In terms of model training, Di-GraphGAN is trained using RMSProp optimization with the learning rate set as $1 \times e^{-5}$ and the batch size set as 32. Besides, after 100 epochs, we decay the learning rate to 90% of its previous values every 5 epochs. The following three widely used evaluation metrics are adopted in our experiments: (1) Mean Average Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Square Error (RMSE). For multiple features c, this work considers 6 types of weather conditions: Sunny/Cloudy/Rainy/Thunderstorm/Snow/Foggy (encoded by One-Hot encoding method), generalized time features: IsWeekend (True = 0/False = 1) and IsHoliday (True = 0/False = 1). Thus, there are 8 dimension of semantic features, plus the ground-truth traffic sequential data, the total dataset dimension is $c = 9$.

Data Missing Scenarios. Both the amount and the distribution of missing data have a significant impact on traffic prediction performances in industrial applications [22,44,42]. Therefore, we consider (1) random missing scenario (RM) and (2) non-random missing scenario (NM) when creating traffic datasets with missing patterns. The RM scenario is generated by randomly assigning a given percentage of data points from all time steps as 0. In contrast, the NM scenario is generated by randomly assigning all the data points of a given length of consecutive time steps as 0 (e.g. set the data points within a randomly chosen hour from Monday to Sunday as 0). We set datasets with 0%/5%/10%/20%/30%/40%/50%/60% missing values in experiments to evaluate different missing proportions on model performance. To guarantee all experiments and models are examined using the same datasets, we employ identical random seed when generating data missing scenarios.

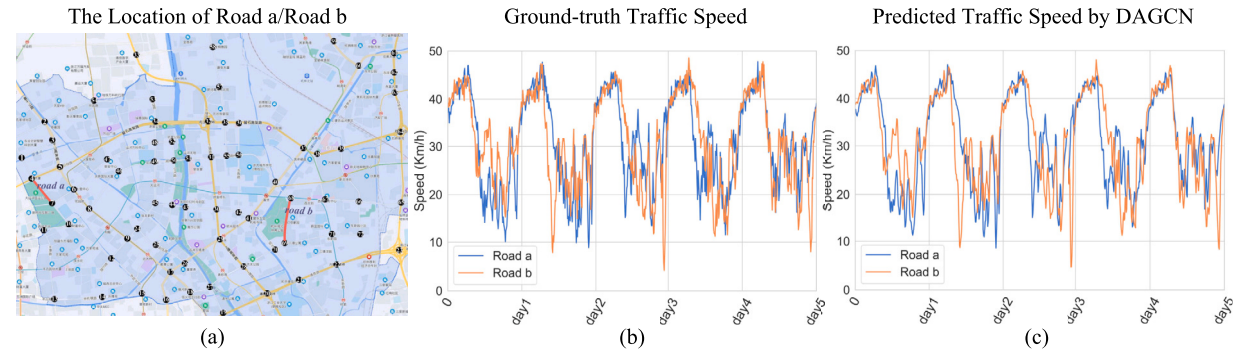

Fig. 4. Di-GraphGAN can capture the implicit semantic relationships between roads in a traffic network. Road *a* and road *b* have analogous POIs around them, thus they share similar traffic patterns.

5.1. Baseline methods

Our experiments mainly focus on two tasks: 1) spatiotemporal traffic prediction performance evaluation and 2) traffic data imputation performance evaluation. Thus, we compare Di-GraphGAN with different baseline methods for the two different tasks. Because of the page limitation, we cannot give each baseline method a specific description but only provide the model name and source article here.

1) *Baseline Methods for Spatiotemporal Traffic Prediction* We consider the following baseline methods, including statistic models and advanced deep learning traffic prediction models: (I). ARIMA [26], (II). GRU [45], (III). DCRNN [31], (IV). STSGCN [32], (V). STFGNN [33], (VI). GCGAN [46], (VII). GE-GAN [47], (VIII). RGSL [15], (IX). STIDGCN [48].
2) *Baseline Methods for Traffic Data Imputation* For traffic data imputation task, we consider classical models, the latest deep learning models and two novel matrix factorization-based models: (I). k-NN(I) [41], (II). Matrix Factorization [41], (III). GRU-D [49], (IV). BRITS [44], (V). NAOMI [42], (VI). TRMF [21], (VII). BTMF [22].

5.2. Experiment results and analysis

1) Traffic Prediction with Complete Data under Different Prediction Scales

To start with, we evaluate model performances on multi-scale traffic speed forecasting with complete traffic data as input (i.e. missing rate is 0%). We set the future prediction step $k = 1, 3, 4, 8, 12$. Here, $k = 1$ equals to one-step prediction and $k = 3, 4, 8, 12$ studies the impact of different prediction lengths. Table 1 displays the RMSE, MAPE, and MAE results of HangZhou-speed dataset and Fig. 5 visualizes the RMSE results change according to prediction scale k on GuangZhou-speed dataset. As can be seen from Table 1 and Fig. 5 that with the growth of prediction scale k, MAE, RMSE and MAPE for all methods also keep rising. This phenomenon can be interpreted as: making accurate forecasts for a large time horizon is more difficult than predicting for a small time horizon. Our Di-GraphGAN consistently surpasses other baseline methods among all prediction scales, demonstrating its effectiveness and superiority. Furthermore, Di-GraphGAN is capable of capturing the semantic similarity of roads even if they are not geographically connected. Fig. 4 (a) shows that road *a* and road *b* are not spatially adjacent to each other, but Di-GraphGAN is able to model their analogous traffic speed patterns as in Fig. 4 (b) and Fig. 4 (c). Finally, we randomly select one road segment from the two studied traffic datasets, and plot the ground-truth traffic speed along with the predictions made by Di-GraphGAN model for one-week time horizon as shown in Fig. 6 and Fig. 7.

It can be seen from Table 1 and Fig. 5 that the values of RMSE are always larger in Guangzhou-speed than in HangZhou-speed. The reason is that traffic speed data in Guangzhou city covers more road segments (214 road segments in total) with more fluctuations and diversity, where as traffic speed data in Hangzhou city (116 road segments in total) is slightly smaller than Guangzhou and less fluctuating. There are also some similarities between the two datasets. For example, during weekday traffic peak time (8:00 a.m.∼10:00 a.m. and 16:00 p.m.∼ 18:00 p.m.) the traffic speed data is always smaller than in other time of a day. Also, the traffic speed data shows different patterns during holiday periods compared with normal days. These phenomena reveal some underlying common traffic patterns in modern Chinese cities.

Among the baseline methods, ARIMA and GRU are two classical time-series forecasting methods. They have inferior performances than other deep learning spatial-temporal models. GCGAN and GE-GAN are two GAN-based traffic prediction methods which shown comparable performance. GCGAN outperforms STSGCN and STFGNN when prediction step $k \geq 4$, proving the advantage of adversarial learning models in combination with other deep learning techniques. Nevertheless, the drawbacks are that GCGAN fails to consider multi-scale temporal dependencies and the model architecture is too simple to modeling the complex urban traffic networks.

For GNN-based methods, STSGCN and STFGNN show excellent results in one-step prediction, but lose their advantages when the prediction scale becomes large. In contrast, DCRNN adopts Seq2Seq-based framework to perform multi-scale prediction and is more robust than STSGCN and STFGNN in LSTF task. RGSL and STIDGCN are two latest graph structure learning/dynamic graph

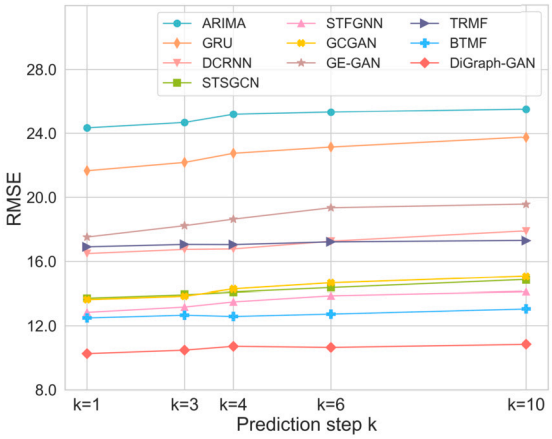

Fig. 5. The RMSE results of GuangZhou-speed dataset under no missing values and different prediction step k.

Table 1
Results on HangZhou-speed dataset with complete data (i.e., missing rate = 0%) and different prediction length k.

k	Metrics	ARIMA	GRU	DCRNN	STSGCN	STFGNN	GCGAN	GE-GAN	RGSL	STIDGCN	Di-GraphGAN
1	RMSE	9.18	8.94	7.23	4.49	3.68	4.69	7.56	4.82	3.95	**2.74**
	MAPE(%)	11.34	11.08	8.74	5.12	3.93	5.27	8.98	5.45	4.07	**3.38**
	MAE	5.92	5.81	4.67	2.53	2.05	2.62	4.83	2.78	2.24	**1.69**
3	RMSE	10.25	9.37	8.54	7.63	6.35	4.85	8.72	5.15	4.06	**3.12**
	MAPE(%)	12.26	11.23	10.51	9.15	7.42	5.39	10.45	5.83	4.28	**3.92**
	MAE	6.35	5.86	5.69	5.06	3.88	2.91	5.37	3.22	2.32	**1.88**
4	RMSE	11.82	9.84	9.67	9.96	8.83	5.47	10.92	5.23	4.31	**3.38**
	MAPE(%)	14.07	11.84	11.49	11.95	10.87	6.08	12.38	5.91	4.55	**4.27**
	MAE	7.93	6.13	6.08	5.97	5.65	3.63	6.74	3.29	2.69	**2.14**
8	RMSE	13.18	11.77	10.83	11.36	10.74	5.92	11.26	5.45	4.53	**3.47**
	MAPE(%)	15.53	13.82	12.95	13.05	12.68	6.56	13.43	6.07	4.72	**4.58**
	MAE	8.24	7.91	7.26	7.33	6.52	4.25	7.69	3.52	2.84	**2.28**
12	RMSE	14.33	12.54	11.58	12.42	12.75	6.69	13.07	5.63	4.95	**3.82**
	MAPE(%)	16.25	14.05	12.87	14.37	13.96	7.92	15.28	6.14	5.16	**4.83**
	MAE	9.41	8.08	7.62	8.14	7.45	4.83	8.15	3.59	3.06	**2.54**

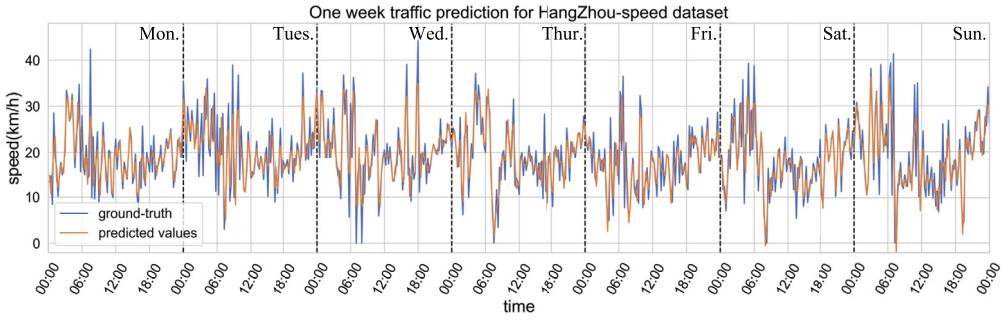

Fig. 6. Visualized ground-truth traffic speed data and corresponding predictions by our Di-GraphGAN for one-week time horizon.

convolution models. RGSL outperforms some deep learning models because it integrates explicit traffic features and implicit correlations when construct graph information. However, our Di-GraphGAN is also capable of learning implicit spatiotemporal traffic correlations by the powerful generative ability of GAN. STIDGCN takes advantage of its dynamic learning GCN network, whereas our Di-GraphGAN also achieves dynamic learning by introducing the Temporal Contextual Attention mechanism.

2) Traffic Prediction with Incomplete Data under Different Data Missing Scenarios

In this part, we conduct experiments under four synthetic traffic data missing situations with data missing rates changing from 5% to 60%. Both random missing (RM) scenario and non-random missing (NM) scenario are considered. We use the aforementioned

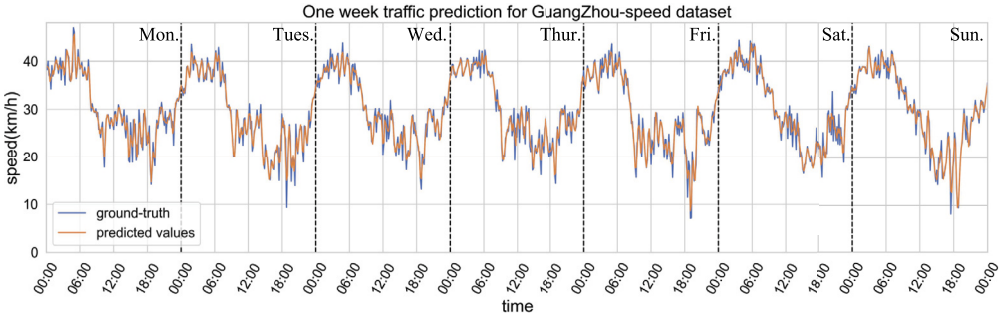

Fig. 7. Visualized ground-truth traffic speed data and corresponding predictions by our Di-GraphGAN for one-week time horizon.

Table 2
Traffic Prediction MAPE(%) comparison in Hangzhou city under random missing scenario. (prediction step $k = 1$).

Model	Missing Rate						
	5%	10%	20%	30%	40%	50%	60%
k-NN(I)+Di-GraphGAN	5.13	5.28	5.79	6.06	6.24	6.52	6.83
MF+Di-GraphGAN	4.87	5.03	5.36	5.83	6.17	6.38	6.72
GRU-D+Di-GraphGAN	4.24	4.49	4.82	5.12	5.48	5.75	5.96
BRITS+Di-GraphGAN	4.07	4.18	4.57	4.86	5.11	5.45	10.87
NAOMI+Di-GraphGAN	3.89	4.02	4.28	4.59	4.96	5.25	5.73
TRMF+Di-GraphGAN	3.92	4.06	4.33	4.75	5.08	5.37	5.65
BTMF+Di-GraphGAN	3.85	3.96	4.13	4.51	4.82	5.03	5.29
Di-GraphGAN	**3.63**	**3.77**	**3.98**	**4.12**	**4.35**	**4.56**	**4.78**

Table 3
Traffic Prediction MAPE(%) comparison in Hangzhou city under non-random missing scenario. (prediction step $k = 1$).

Model	Missing Rate						
	5%	10%	20%	30%	40%	50%	60%
k-NN(I)+Di-GraphGAN	4.81	5.12	5.37	5.68	5.93	6.22	6.49
MF+Di-GraphGAN	4.59	4.74	4.98	5.20	5.45	5.71	5.93
GRU-D+Di-GraphGAN	4.12	4.37	4.56	4.82	5.14	5.35	5.62
BRITS+Di-GraphGAN	3.97	4.15	4.41	4.63	4.82	5.09	5.37
NAOMI+Di-GraphGAN	3.72	3.94	4.17	4.38	4.57	4.76	5.05
TRMF+Di-GraphGAN	3.85	4.03	4.28	4.49	4.65	4.88	5.14
BTMF+Di-GraphGAN	3.58	3.79	3.96	4.15	4.39	4.63	4.88
Di-GraphGAN	**3.43**	**3.67**	**3.84**	**4.06**	**4.23**	**4.49**	**4.67**

baseline methods 2) for performance evaluation. To begin with, we drop out a specific proportion of data to create the required data missing scenario, then we apply different data imputation models to estimate the missing data points. Finally, the imputed full Hangzhou-speed data is trained using our Di-GraphGAN model for traffic prediction, and we collect the traffic prediction MAPE results for comparison. For instance, "k-NN(I)+Di-GraphGAN" means we use k-NN(I) model to realize traffic data imputation and then use Di-GraphGAN for full data traffic prediction, while "Di-GraphGAN" means we use our Di-GraphGAN model to realize end-to-end traffic data imputation and traffic prediction. Table 2 and Table 3 display the MAPE results of HangZhou-speed dataset under RM scenario and NM scenario, respectively. The results show that Di-GraphGAN consistently surpasses other data imputation methods for traffic prediction. It also proves the imputed data quality and reliability various a lot by using different imputation methods.

Next, we purely evaluate traffic data imputation performance on the two traffic speed datasets using MAE as metrics. Firstly, we drop out a specific proportion of data to create the required data missing scenario, then we apply different data imputation models to impute the missing values, and finally we calculate the errors between the imputed full traffic data and the ground-truth traffic data. The MAPE results of Guangzhou-speed dataset under RM scenario and NM scenario are displayed in Table 4 and Table 5. Furthermore, Fig. 8(a) and Fig. 8(b) depict the traffic data imputation MAE comparison of different models under NM scenario for Hangzhou-speed and Guangzhou-speed datasets. We also visualize the data imputation MAE results for the two cities under RM scenario as shown in Fig. 8(c) and Fig. 8(d).

We can derive the following listed conclusions from the above experimental results:

(i) Di-GraphGAN significantly outperforms other baseline methods in terms of both RM and NM scenarios with different data missing rates. Di-GraphGAN model is superior than other methods because first, we introduce a novel data imputation module

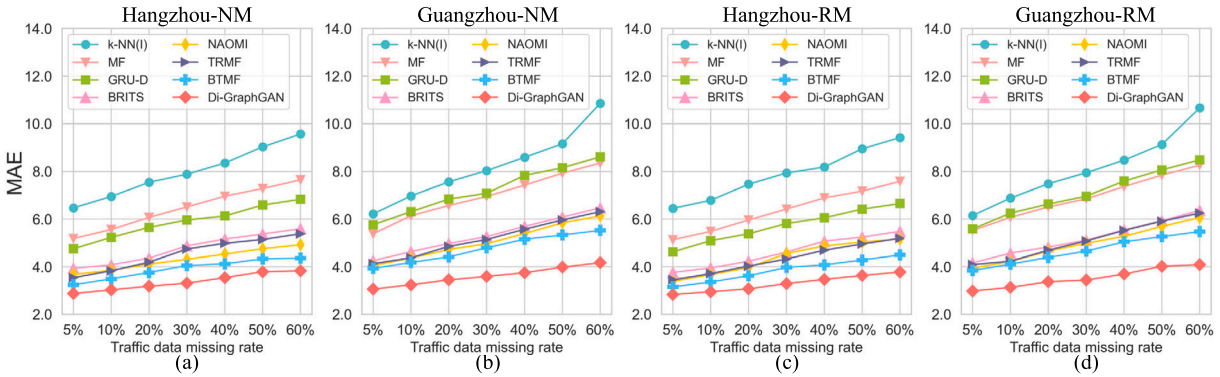

Fig. 8. Traffic data imputation MAE results for Hangzhou and Guangzhou city under RM and NM missing scenarios.

Table 4
Data imputation MAPE(%) performance comparison in Guangzhou city under random missing scenario.

Model	Missing Rate						
	5%	10%	20%	30%	40%	50%	60%
k-NN(I)	8.92	10.16	12.53	13.74	15.25	16.48	17.31
MF	8.15	8.63	9.38	9.73	10.09	10.82	11.58
GRU-D	7.93	8.47	9.36	9.85	10.69	11.23	12.59
BRITS	6.06	6.28	6.55	6.97	7.51	8.36	9.72
NAOMI	5.38	5.65	6.12	6.71	6.89	7.42	9.23
TRMF	6.56	6.83	7.14	7.68	8.13	8.57	8.94
BTMF	5.24	5.49	5.76	6.05	6.28	6.63	7.15
Di-GraphGAN	**4.87**	**5.03**	**5.18**	**5.31**	**5.54**	**5.79**	**5.83**

Table 5
Data imputation MAPE(%) performance comparison in Guangzhou city under non-random missing scenario.

Model	Missing Rate						
	5%	10%	20%	30%	40%	50%	60%
k-NN(I)	9.17	10.53	11.97	13.25	13.92	15.53	17.48
MF	8.23	8.65	9.28	9.74	10.33	10.89	11.64
GRU-D	7.98	8.51	9.36	10.17	11.58	12.37	12.95
BRITS	6.03	6.25	6.42	6.85	7.19	7.64	8.07
NAOMI	5.48	5.72	5.95	6.22	6.58	7.06	7.83
TRMF	5.92	6.13	6.55	6.84	7.15	7.23	7.58
BTMF	5.36	5.69	6.14	6.35	6.67	7.04	7.49
Di-GraphGAN	**4.95**	**5.14**	**5.26**	**5.39**	**5.62**	**5.87**	**5.91**

(DI-LSTM), which realizes accurate traffic data imputation using the time damping unit. Second, Di-GraphGAN is formulated as Seq2Seq architecture for more flexible prediction scales. Third, the weighted and integrated loss function of Di-GraphGAN better facilitates model training process and enhances model learning ability.

(ii) Classical data imputation models (k-NN(I) and MF) have inferior performance than other deep learning-based imputation methods, because they are incapable of modeling the complex spatiotemporal dependencies in urban traffic network data.

(iii) Deep learning models with Seq2Seq architecture or form like RNN structure have better performances in multi-scale prediction and long sequence time-series forecasting (LSTF) task.

(iv) Some GAN-based models show satisfactory performances in spatiotemporal traffic prediction (e.g. GCGAN) and traffic data imputation (e.g. NAOMI), respectively. Nevertheless, only using a simple GAN framework results in limited performance and cannot surpass the SOTA models in a specific task. In light of that, we should integrate the wisdom of GAN framework with other advanced technologies for further improvements.

(v) Recent matrix factorization-based methods (TRMF and BTMF) have slightly worse performance in traffic data imputation compared to deep learning-based imputation models under small missing rates, but they surpass many deep learning models with the increase of data missing rate. In general, they show better robustness to resist the change of data missing rate.

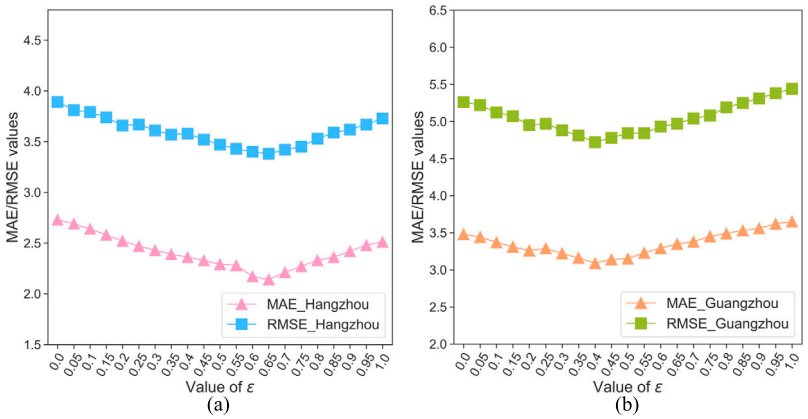

Fig. 9. The influence of weight-param ϵ on Di-GraphGAN's performance. We find that the optimal ϵ value for Hangzhou and Guangzhou is 0.65 and 0.4, respectively.

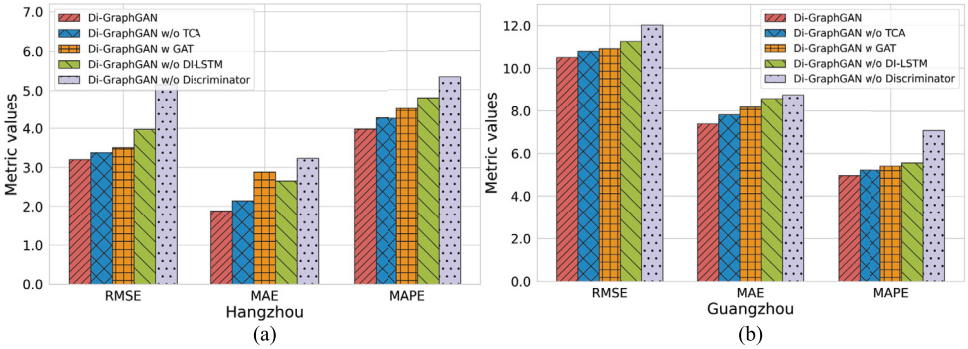

Fig. 10. The performance comparison of different model variants on Hangzhou and Guangzhou (prediction step $k = 1$).

5.3. Ablation study

This subsection dedicates to investigating the effectiveness of both Di-GraphGAN's main components and the weighted objective function.

5.3.1. Sensitive study

In this part, we study how the weight parameter ϵ in the overall objective function influences Di-GraphGAN's performance. $\epsilon \in [0, 1]$ is employed to tune-up the significance of generative adversarial loss and the masked reconstruction loss. We carry out experiments on $k = 4$ multi-scale traffic prediction with complete data and set ϵ's values changing from 0 to 1 with an increasing step of 0.05. Fig. 9(a) and Fig. 9(b) plot the traffic prediction MAE/RMSE change with different ϵ on Hangzhou-speed and Guangzhou-speed, respectively. The experiment analysis confirms that a suitable ϵ contributes a lot to Di-GraphGAN's performance.

5.3.2. Model components analysis

To study how each component influences Di-GraphGAN's performance, we employ the optimal ϵ value for the two datasets and set 25% missing values under random missing scenario. The four kinds of model variants are described as follows:

- **Di-GraphGAN w/o TCA**: We remove the proposed Temporal Contextual Attention in Di-GraphGAN.
- **Di-GraphGAN w GAT**: Instead of using our proposed Task-Efficient Graph Attention Networks (TE-GAT), we replace it with a standard GAT with 8-head attention mechanism and 2 layers.
- **Di-GraphGAN w/o DI-LSTM**: The data imputation module DI-LSTM is replaced with a vanilla LSTM layer.
- **Di-GraphGAN w/o Discriminator**: We remove the Discriminator and only retain the Generator. In this case, the model is formed as an Encoder-Decoder architecture.

The performance comparison of Di-GraphGAN's variants are shown in Fig. 10 (a) and Fig. 10 (b). We can derive the following findings from the results: First, Di-GraphGAN w/o TCA has slightly worse performance than Di-GraphGAN, proving that TCA mechanism is able to effectively integrate the temporal dynamics with semantic context information. Second, Di-GraphGAN w GAT shows inferior performance than Di-GraphGAN, which proves our task-efficient graph attention networks gains stronger representation learning ability to aggregate useful spatiotemporal correlations within graph structure data than the vanilla graph attention

Table 6
Efficiency Comparison Experiments of TE-GAT and standard GAT model.

Model	Dataset	Inference Time	Model Parameter Size	Resource Comparison
standard GAT	Guangzhou-speed	1.8 s	1.96M	9.53 GB
TE-GAT		1.3 s	1.42M	8.66 GB
standard GAT	Hangzhou-speed	1.0 s	1.96M	8.25 GB
TE-GAT		0.8 s	1.42M	7.69 GB

networks. Third, for traffic datasets containing relatively large percent of missing values, a precise data imputation method is a prerequisite to promote the performance of other applications, such as the proposed DI-LSTM. Fourth, the results of Di-GraphGAN w/o Discriminator shows that generative adversarial learning framework contributes a lot to learn the hidden traffic patterns and the real-world traffic data distribution, which is superior than a plain Encoder-Decoder model.

5.3.3. Computational efficiency evaluation

To investigate the computational efficiency of our TE-GAT, we conduct a set of experiments to compare the performance of TE-GAT with standard GAT network. The evaluation measurements include: (1) Inference Time, (2) Model Parameter Size, (3) Model Resource Consumption. Experimental results are shown in Table 6. Here, standard GAT means we replace the proposed TE-GAT with vanilla GAT network in the Di-GraphGAN model. The experiments are performed using the same hardware and software environment as introduced in the Implementation paragraph to eliminate external variables that could affect computation time. The results shown in Table 6, clearly demonstrate that the TE-GAT model achieves a reduction in reference time of 20% and 26% on Hangzhou-speed and Guangzhou-speed dataset, respectively; model parameter size reduction of 27.55%; Resource Consumption reduction of 9.12% and 6.79% on Hangzhou-speed and Guangzhou-speed dataset, respectively. The results indicating a certain improvement in computational efficiency.

6. Conclusion

In this work, we reconsider the way for accurate multi-scale spatiotemporal traffic forecasting under real-world data missing scenarios. An enhanced and integrated spatial-temporal data imputation graph attention generative adversarial networks (Di-GraphGAN) is proposed for end2end traffic data imputation and prediction. For the first step, we propose DI-LSTM, a novel data imputation module which accurately estimates missing data by a designed time damping unit. Based on this, we formulate Di-GraphGAN as a Seq2Seq architecture with DI-LSTM, Task-Efficient Graph Attention Networks (TE-GAT), and Temporal Contextual Attention (TCA) as main modules. Specifically, our Task-Efficient GAT simplifies the computational overhead caused by existing GAT's graph neighbor aggregation method and improves its representation learning ability; the proposed TCA mechanism captures informative local contextual knowledge and can be fit into every time step of RNN network to improve the attention performance. Finally, model performances are evaluated on two large-scale traffic speed datasets in Hangzhou and Guangzhou city with different prediction scales and data missing scenarios. Di-GraphGAN is compared with various state-of-the-art methods and the results show the superiority of Di-GraphGAN.

In the near future, we intend to conduct more comprehensive studies based on Di-GraphGAN. Further investigations may concentrate on introducing external semantic information into our model to generate more accurate traffic predictions. Aside from this, other potential applications such as traffic congestion detection, urban crowd-flow prediction, and traffic accident forecasting will also be explored.

CRediT authorship contribution statement

Lincan Li: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Jichao Bi:** Writing – review & editing, Supervision, Project administration, Conceptualization, Funding acquisition. **Kaixiang Yang:** Writing – review & editing, Resources, Funding acquisition, Formal analysis. **Fengji Luo:** Writing – review & editing, Formal analysis, Conceptualization.

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.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62302069, Grant 62106224, and in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ24F030015.

References

[1] W. Zhang, S. Yan, J. Li, Tcp-bast: a novel approach to traffic congestion prediction with bilateral alternation on spatiality and temporality, Inf. Sci. 608 (2022) 718–733.
[2] F. Kong, J. Li, B. Jiang, H. Song, Short-term traffic flow prediction in smart multimedia system for internet of vehicles based on deep belief network, Future Gener. Comput. Syst. 93 (2019) 460–472.
[3] A. Sumalee, H.W. Ho, Smarter and more connected: future intelligent transportation system, IATSS Res. 42 (2018) 67–71.
[4] R. Tian, C. Wang, J. Hu, Z. Ma, Multi-scale spatial-temporal aware transformer for traffic prediction, Inf. Sci. 648 (2023) 119557.
[5] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, Y. Zhang, Deep and embedded learning approach for traffic flow prediction in urban informatics, IEEE Trans. Intell. Transp. Syst. 20 (2019) 3927–3939.
[6] Y. Tashiro, J. Song, Y. Song, S. Ermon, Csdi: conditional score-based diffusion models for probabilistic time series imputation, in: Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 24804–24816.
[7] T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, Dynamic route planning with real-time traffic predictions, Inf. Syst. 64 (2017) 258–265.
[8] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Li, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (2018) 6085.
[9] Z. Cui, L. Lin, Z. Pu, Y. Wang, Graph Markov network for traffic forecasting with missing data, Transp. Res., Part C, Emerg. Technol. 117 (2020) 102671.
[10] Y. Liang, K. Ouyang, J. Sun, Y. Wang, J. Zhang, Y. Zheng, D. Rosenblum, R. Zimmermann, Fine-grained urban flow prediction, in: Proc. Web Conf., WWW '21, 2021, pp. 1833–1845.
[11] S. Guo, Y. Lin, H. Wan, X. Li, G. Cong, Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting, IEEE Trans. Knowl. Data Eng. 34 (2022) 5415–5428.
[12] C. Chen, Y. Liu, L. Chen, C. Zhang, Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–13.
[13] Y. Bao, J. Liu, Q. Shen, Y. Cao, W. Ding, Q. Shi, Pket-gcn: prior knowledge enhanced time-varying graph convolution network for traffic flow prediction, Inf. Sci. 634 (2023) 359–381.
[14] 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.
[15] H. Yu, T. Li, W. Yu, J. Li, Y. Huang, L. Wang, A. Liu, Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting, in: Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI'22, 2022, pp. 2362–2368.
[16] C. Zhang, S. Zhang, J.J.Q. Yu, S. Yu, Fastgnn: a topological information protected federated learning approach for traffic speed forecasting, IEEE Trans. Ind. Inform. 17 (2021) 8464–8474.
[17] Z. Fang, D. Wu, L. Pan, L. Chen, Y. Gao, When transfer learning meets cross-city urban flow prediction: spatio-temporal adaptation matters, in: Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI'22, 2022, pp. 2030–2036.
[18] 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., Part C, Emerg. Technol. 132 (2021) 103372.
[19] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, J. Sun, A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting, Transp. Res., Part C, Emerg. Technol. 62 (2016) 21–34.
[20] H. Wang, L. Liu, S. Dong, Z. Qian, H. Wei, A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid emd–arima framework, Transportmetrica B: Transp. Dyn. 4 (2016) 159–186.
[21] H.-F. Yu, N. Rao, I.S. Dhillon, Temporal regularized matrix factorization for high-dimensional time series prediction, in: Advances in Neural Information Processing Systems, 2016.
[22] X. Chen, L. Sun, Bayesian temporal factorization for multidimensional time series prediction, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1.
[23] J. Yoon, J. Jordon, M. van der Schaar, GAIN: missing data imputation using generative adversarial nets, in: Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 5689–5698.
[24] Y. Luo, Y. Zhang, X. Cai, X. Yuan, E^2gan: end-to-end generative adversarial network for multivariate time series imputation, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 2019, pp. 3094–3100.
[25] M.S. Ahmed, A.R. Cook, Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques, vol. 722, Transportation Research Board, 1979.
[26] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results, J. Transp. Eng. 129 (2003) 664–672.
[27] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, Dnn-based prediction model for spatio-temporal data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPACIAL '16, 2016.
[28] Z. Lin, J. Feng, Z. Lu, Y. Li, D. Jin, Deepstn+: context-aware spatial temporal neural network for crowd flow prediction in metropolis, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 1020–1027.
[29] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 3656–3663.
[30] 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, 2020, pp. 1234–1241.
[31] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: data-driven traffic forecasting, in: International Conference on Learning Representations (ICLR), 2018.
[32] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
[33] M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 4189–4196.
[34] 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, 2019, pp. 922–929.
[35] Z. Zhang, M. Li, X. Lin, Y. Wang, F. He, Multistep speed prediction on traffic networks: a graph convolutional sequence-to-sequence learning approach with attention mechanism, arXiv preprint arXiv:1810.10237, 2018.
[36] J.J.Q. Yu, J. Gu, Real-time traffic speed estimation with graph convolutional generative autoencoder, IEEE Trans. Intell. Transp. Syst. 20 (2019) 3940–3951.
[37] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271, 2018.

[38] L. Cai, K. Janowicz, G. Mai, B. Yan, R. Zhu, Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting, Trans. GIS 24 (2020) 736–755.
[39] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A.X. Liu, S. Dustdar, Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting, in: International Conference on Learning Representations, 2022.
[40] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: decomposition transformers with auto-correlation for long-term series forecasting, in: Advances in Neural Information Processing Systems, 2021.
[41] T. Hastie, R. Tibshirani, J.H. Friedman, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 2, Springer, 2009.
[42] Y. Liu, R. Yu, S. Zheng, E. Zhan, Y. Yue, Naomi: non-autoregressive multiresolution sequence imputation, in: Advances in Neural Information Processing Systems (NeurlPS), 2019.
[43] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875, 2017.
[44] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, Brits: bidirectional recurrent imputation for time series, in: Advances in Neural Information Processing Systems, 2018.
[45] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: Advances in Neural Information Processing Systems, 2014.
[46] Y. Zhang, S. Wang, B. Chen, J. Cao, Gcgan: generative adversarial nets with graph cnn for network-scale traffic prediction, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.
[47] D. Xu, C. Wei, P. Peng, Q. Xuan, H. Guo, Ge-gan: a novel deep learning framework for road traffic state estimation, Transp. Res., Part C, Emerg. Technol. (2020).
[48] A. Liu, Y. Zhang, Spatial-temporal interactive dynamic graph convolution network for traffic forecasting, arXiv preprint arXiv:2205.08689, 2022.
[49] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (2018) 1–12.