Knowledge-Based Systems 261 (2023) 110188

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data

Xiangjie Kong [a], Wenfeng Zhou [a], Guojiang Shen [a,*], Wenyi Zhang [a], Nali Liu [a], Yao Yang [b]

[a] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[b] Zhejiang Lab, Hangzhou 311100, China

ARTICLE INFO

Article history:
Received 5 September 2022
Received in revised form 28 November 2022
Accepted 4 December 2022
Available online 7 December 2022

Keywords:
Spatiotemporal traffic data
Missing data imputation
Graph generator
Dynamic graph convolution
Recurrent neural networks

ABSTRACT

In real-world intelligent transportation systems, the spatiotemporal traffic data collected from sensors often exhibit missing or corrupted data, significantly hindering the development of traffic data research. Missing data imputation is a classic research topic that encompasses a wide range of methods. However, these methods are typically underdeveloped in two aspects: the dynamic spatial dependencies of the road network over time, and the information extraction and utilization of diverse data. In this study, we design a novel deep learning architecture – Dynamic Graph Convolutional Recurrent Imputation Network (DGCRIN) – as a tool to impute missing traffic data. The DGCRIN employs a graph generator and dynamic graph convolutional gated recurrent unit (DGCGRU) to perform fine-grained modeling of the dynamic spatiotemporal dependencies of road network. Additionally, an auxiliary GRU learns the missing pattern information of the data, and a fusion layer with a decay mechanism is introduced to fuse a diverse range of information. This architecture enables the DGCRIN to be highly adaptable to complex scenarios involving missing data. Extensive experiments on two datasets demonstrate the superiority of DGCRIN over multiple baseline models.

1. Introduction

Spatiotemporal traffic data collected from the real world via sensor devices are essential and fundamental for traffic research and applications [1]. By analyzing and mining traffic data, researchers can address a wide range of problems, including traffic speed prediction [2], traffic pattern recognition [3], and traffic data generation [4]. However, owing to the inherent unpredictability of the data collection and storage processes, the collected traffic information frequently contains missing data, which degrades model performance for downstream tasks if not handled properly. Therefore, it is necessary to perform imputation carefully on spatiotemporal traffic data.

Generally, the essence of traffic data imputation is to productively extract effective latent information, such as temporal correlations and spatial dependencies, from observed data to estimate the missing data. A wide range of methods have been developed to achieve this goal. Early approaches attempted to directly utilize statistical features, such as zeros, historical averages [5], and last observations [6], to fill in or simply eliminate the gaps in data. These rudimentary methods consider solely a site's own historical data, whereas a more effective approach is to combine information from multiple sites. A typical KNN-based imputation method estimated missing data by averaging the known values of the *k* neighbors. Recently, matrix- and tensor-based decomposition techniques [7–9] have exhibited great potential as tools to solve traffic data imputation problems. However, these methods rely on a global low rank while ignoring local spatiotemporal consistency. Therefore, these models may have certain limitations in capturing globally complex spatiotemporal dependencies. In contrast, deep learning-based approaches have exhibited extraordinary nonlinear modeling capabilities for various tasks [10–15]. Recent studies have attempted to use neural networks (NN), such as recurrent neural networks (RNN) [16], convolutional neural networks (CNN) [17], and graph neural networks (GNN) [18], to handle data imputation tasks. Among them, the GNN-based models are highly effective in capturing spatial dependencies among irregular road networks than other methods. Although these methods have achieved some success in addressing the issue of missing data, two important research gaps remain.

Dynamic spatial dependencies: Most existing GNN-based methods model spatial correlations via predefined static graph structures based on geographic distance or road connectivity. Consequently, these methods maintain a constant graphical structure over time. However, traffic data often exhibit strong dynamic correlations in the spatiotemporal dimension, which may not be fully modeled by a static graph structure. Several recent studies have proposed the construction of dynamic graph structures to

* Corresponding author.
E-mail addresses: xjkong@ieee.org (X. Kong), gjshen1975@zjut.edu.cn (G. Shen).

https://doi.org/10.1016/j.knosys.2022.110188
0950-7051/© 2022 Elsevier B.V. All rights reserved.

capture the dynamic dependencies of traffic data, and achieved promising performance in traffic prediction tasks [19–21]. However, this issue is rarely considered in the context of traffic data imputation. As a result, the construction of a dynamic graph that models dynamic dependencies, and provides more accurate and effective inference information for data reconstruction, remains a challenge.

Various complex temporal information: Traffic data are essentially time-series data collected through various physical devices at regular or irregular time intervals. In an environment prone to missing data, a variety of time-series datasets can be obtained from different perspectives. For example, according to the missing position and data collection timestamp, a masking matrix dataset and a time-lag matrix dataset can be generated. The former indicates the original data states that includes missing patterns, and the latter helps in studying the contributions of observations to the missing value estimates [22]. Both approaches are beneficial for the imputation task. Nevertheless, many prior studies [1,23,24] on the imputation of traffic data have neglected the analysis and utilization of such datasets. Therefore, the extraction and fusion of information from diverse and complex temporal data warrant further examination.

To address the aforementioned issues, a novel spatiotemporal deep learning approach for traffic data imputation called **D**ynamic **G**raph **C**onvolutional **R**ecurrent **I**mputation **N**etworks **(DGCRIN)** is proposed in this study. A graph generator was developed to model dynamic spatial correlations and a dynamic graph convolutional gated recurrent unit (DGCGRU) was used to capture spatiotemporal dependencies. Furthermore, to extract more useful information from different types of data, we employed an auxiliary gated recurrent unit (GRU) to model the missing patterns of masking data, and then introduced a fusion layer with a decay mechanism to fuse information from disparate data. The missing values of different road segments at each timestamp are imputed in a bidirectional process using the spatial dependencies and temporal correlations learned from historical observations. To evaluate the effectiveness of the proposed model, we conducted experiments on two real-world traffic datasets, demonstrating the superior performance of our method over that of existing approaches. In addition, corresponding hyperparameter sensitivity and ablation experiments were also carried out.

Our main contributions can be summarized as follows:

- We propose a novel traffic data imputation framework to infer missing values in the spatiotemporal input data. The model can effectively model the dynamic spatiotemporal characteristics of incomplete traffic data by considering the road network's dynamics and diverse temporal information including data missing patterns and observation slot, to achieve more accurate estimation.
- We propose a dynamic graph generation technique to perform fine-grained modeling of the spatial correlations between road network nodes with incomplete traffic data via using the recurrent generated imputation and historical information.
- We demonstrate the capabilities and advantages of the proposed model by applying it to two real-world traffic datasets under three types of missing patterns. Compared with the baselines, our model significantly reduced the imputation error while achieving a higher imputation accuracy.

The remainder of this paper is organized as follows. In Section 2, we first review existing studies pertaining to traffic data imputation by category. The imputation problem is formulated in Section 3, and the methodology is described in Section 4. The experiments conducted within this study are presented in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

This section provides a detailed review of existing studies related to traffic data imputation. We classified the literature among three categories: RNN-based, GAN-based, and GNN-based methods.

2.1. RNN-based methods

Traffic data represents sequential data collected over time. RNNs are generally effective in handling sequence data [25] by maintaining chain-like structures and special gate mechanisms. Accordingly, many RNN-based imputation methods have been proposed. The use of a vanilla RNN network, such as a GRU or long short-term memory (LSTM), represents the most naive approach to imputation [26]. However, such an approach generally yields suboptimal performance because the model initially fills in missing data with predefined values, such as zero or a historical average, causing it to learn biased parameters [27]. To improve network robustness and enable the capability to handle missing data, [28,29] proposed novel LSTM-based network structures. In [28], the masking vector was introduced directly into an ordinary LSTM structure to aid in modeling the missing patterns. Based on this, [29] implemented an imputation unit inside the LSTM structure, wherein the missing values in the current time step were filled in by the values inferred from the preceding cell and hidden states. Another method matches the attention mechanism with an RNN variant [30]. Although these methods are somewhat expressive, they ignore the spatial dependencies among traffic data.

2.2. GAN-based methods

With the recent growth of research in this field, deep generation-based imputation models represented by generative adversarial networks (GANs) [31] have received increasing attention. A GAN learns the real distribution of input data via confrontational training between two neural networks – a generator (G) and discriminator (D) – to generate synthetic data. The first application of a GAN to the task of data imputation was GAIN, proposed by Yoon et al. [32]. GAIN uses a hint matrix to guide the discriminator in evaluating the authenticity of generated data. Subsequently, several excellent models have been developed. For example, Zhang et al. [33] introduced a self-attention mechanism into GAIN to capture the correlations between different sensors at different time. Wang et al. [34] innovatively mined the potential category information of data to yield higher-quality imputation results. Their proposed method incorporates a classifier into the GAIN framework, which constrains the generator to impute missing data based on categorical features. Yuan et al. [35] proposed the STGAN model, which employs a center loss to ensure that the generator distribution. However, because these GAN-based methods primarily focus on data distribution, they are insufficient when the data distribution is unclear owing to a large quantity of missing data.

2.3. GNN-based methods

Recently, GNNs have exhibited remarkable performance in modeling non-Euclidean data – particularly graphs [36] – and achieved satisfactory performance in the field of transportation. Zhang et al. [37] developed a graph convolutional network (GCN)-based model for simultaneous traffic prediction and imputation tasks that uses a distance-based predefined fixed weighted adjacency matrix to characterize spatial dependencies. Yao et al. [38] considered the network structure of spatial flows and proposed a

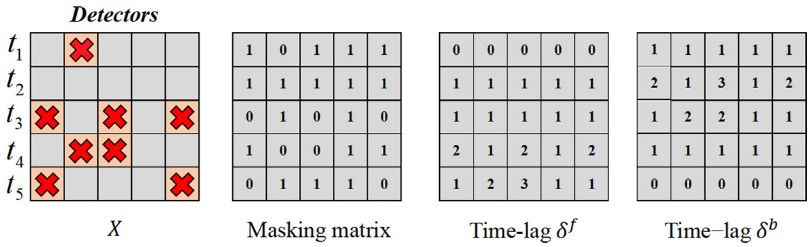


Fig. 1. Illustration of the mask matrix and time interval matrices.

spatial interaction GCN model for the spatial origin–destination flow imputation task. Because a single predefined static graph based on distance cannot reflect the changes in spatial correlations over time, Xu et al. [39] designed a model that employs GraphSAGE to aggregate spatiotemporal information from a graph constructed by correlation coefficients of historical data. With the development of heterogeneous graph research [40–42], a heterogeneous graph-based GCN model was proposed in [43]. This model builds a multigraph from geographical and historical data to explicitly model the dynamic dependencies between road segments. Another GNN variant called GAT, which incorporates the attention mechanism, was developed to address the problem of missing traffic data in [44]. Although these methods focus on modeling spatial dependencies, they are coarse-grained and rely on sufficient historical data.

3. Preliminaries and problem formulation

3.1. Preliminaries

Generally, a traffic network with N detectors can be defined as an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{SA}\}$, where $\mathcal{V} = \{v_i\}$ is a set of N detectors corresponding to the nodes in the graph, and $\mathcal{E} = e\{v_i, v_j\}$ represents the spatial connectivity between two detectors. The adjacency matrix of graph $\mathcal{G}$ is structured using matrix $\mathcal{SA} \in \mathbb{R}^{N\times N}$, according to the following rule:

$$\mathcal{SA}_{ij} = \begin{cases} 1, & e_{ij} = 1 \\ 0, & e_{ij} = 0 \end{cases} \tag{1}$$

where e_{ij} denotes the connectivity between the graph nodes. Because speed data is a widely-used form of traffic data, it is the primary focus of this study. Supposing the detectors collect speed data at $\mathscr{T} \in \{t_0, t_1, t_2, \dots, t_T\}^T$ time points, then the original observation dataset can be expressed as $\mathscr{X} = \{X_1, X_2, \dots, X_T\} \in \mathbb{R}^{T\times N}$, where the $X_t = \{x_{t,1}, x_{t,2}, \dots, x_{t,N}\} \in \mathbb{R}^{1\times N}$ denotes the data collected by all detectors at timestamp t. Additionally, we define a masking matrix $\mathscr{M} = (M_1, M_2, \dots, M_T) \in \mathbb{R}^{T\times N}$ that has the same dimension as the original $\mathscr{X}$ to indicate the existence of the data. The masking matrix is defined as

$$m_{ij} = \begin{cases} 1, & \textit{if } x_{ij} \textit{ is observed}. \\ 0, & \textit{if } x_{ij} \textit{ is missing}. \end{cases} \tag{2}$$

To explore the potential interaction information between the observed and missing data as much as possible, we also introduce a time-lag matrix $\triangle = (\delta_1, \delta_2, \dots, \delta_T) \in \mathbb{R}^{T\times N}$ to record the time gaps between the last observation and the present unobserved information, as in [45]. The elements of $\triangle$ can be formulated using the following equation:

$$\delta_{t,j} = \begin{cases} |s_t - s_{t-1}| + \delta_{t-1,j}, & \textit{if } t > 1, m_{t-1,j} = 0 \\ |s_t - s_{t-1}|, & \textit{if } t > 1, m_{t-1,j} = 1 \\ 0, & \textit{if } t = 1 \end{cases} \tag{3}$$

where $|s_t - s_{t-1}|$ is the time interval between two neighboring samples. By reversing $\mathscr{T}$, we can also obtain the backward time-lag matrix $\triangledown$. For convenience, we provide a specific example in Fig. 1, wherein we assumed that the data were sampled at equal intervals.

3.2. Problem formulation

Given the observed records $X_{1:t}$ with missing values $M_{1:t}$ in the time period $\{1, 2, \dots, t\}$, the purpose of data imputation is to learn a reconstruction function $\mathscr{RF}$ that satisfies the following constraints:

$$min : \{X_1, X_2, \dots, X_t\} - \mathscr{RF}\{\widetilde{X}_1, \widetilde{X}_2, \dots, \widetilde{X}_t\}|_{M_{1:t}=1} \tag{4}$$

4. Methodology

This section introduces the proposed missing imputation framework for traffic data. We first provide an overview of the proposed model, and then describe each component in detail.

4.1. Network architecture

An overview of the proposed DGCRIN framework is shown in Fig. 2(a). The network is a bidirectional structure that takes the observed records $X_{1:t}$ with missing values $M_{1:t}$ and the corresponding time-lag $\delta_{1:t}$ as input, and generates a recovery of incomplete data $\widehat{X}_{1:t}$ as output in a rolling manner. The unfolding of the forward process is depicted more specifically in Fig. 2(b). The overall model comprises three primary components: a graph generator, a DGCGRU, and a masking GRU. At each time step, the graph generator first uses the current imputed data and historical information to adaptively generate a dynamic graph that models the road network's spatial correlations. Then, the DGCGRU, which substitutes the fully connected layers of a traditional GRU with the dynamic graph convolution operation, is employed to effectively integrate the dynamic graph with the static graph, thereby capturing the spatiotemporal dependencies within the data. Furthermore, an auxiliary masking GRU is used to process the masking matrix separately to learn information from the missing patterns. Finally, a fusion layer with a temporal decay mechanism is employed to perform information fusion, and a fully connected layer is used for data inference. More details are provided in the subsequent subsections.

4.2. Graph generator

Road traffic conditions usually encompass complex spatiotemporal correlations that change over time, such as morning and evening peaks. It is therefore reasonable to model the traffic network dynamically. However, for the data imputation task, the data samples at each moment may be incomplete, which poses a challenge for modeling. Inspired by [21], we addressed this issue by designing a novel dynamic graph generator that utilizes iteratively generated imputation data and historical information

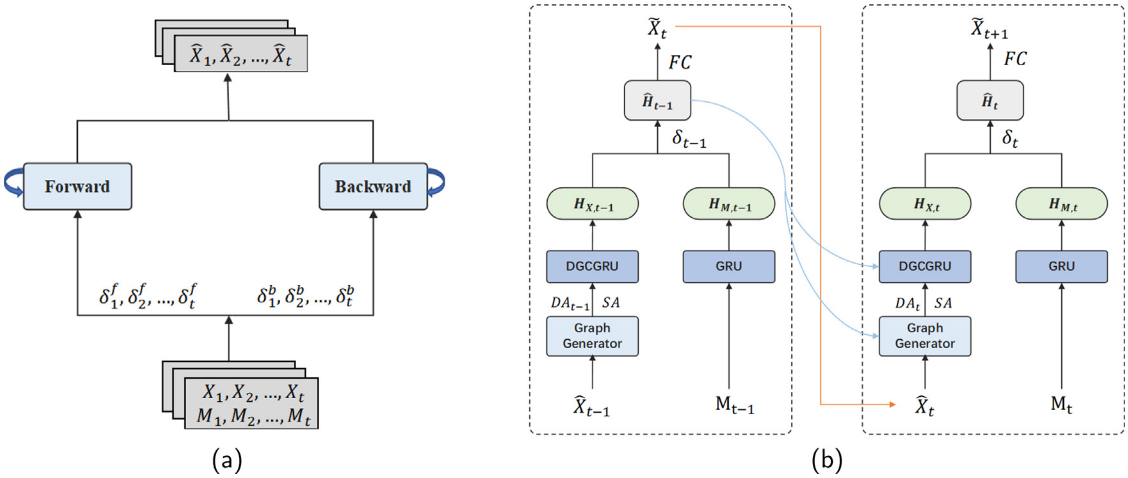


Fig. 2. Architecture of DGCRIN (left), and the unfolding of the forward process (right).

to obtain a dynamic adjacency matrix. At each time step, the generated imputation data $\widehat{X}_t$ and previous fusion hidden state $\widehat{H}_{t-1}$ are concatenated as the input:

$$con_t = \widehat{X}_t \parallel \widehat{H}_{t-1} \tag{5}$$

where $\mathbf{con}_t \in \mathbb{R}^{B\times N\times D_{con}}$, $\parallel$ denotes a concatenation operation; $\mathbf{con}_t$ is treated as a dynamic-node feature. To reasonably measure the correlations between nodes, we first use a graph convolution module to process $\mathbf{con}_t$ and learn spatial features based on the predefined graph:

$$DF^t = \Theta_{*G}(con_t) \tag{6}$$

where $*G$ is the graph convolution and Θ represents the learnable parameters. A distance-based predefined graph is a static structural map of a road network that reflects the basic spatial relationships between nodes, and can be used to conduct message passing. Then, a randomly initialized trainable parameter node embedding $\mathbf{E} \in \mathbb{R}^{N\times D_{em}}$, where D_{em} denotes the node embedding dimension, is employed to perform element-wise multiplication with $\mathbf{DF}^t$ to generate a dynamic representation [46] for the nodes:

$$DE^t = tanh(\alpha(DF^t \odot E)) \tag{7}$$

where $\odot$ denotes the Hadamard product, α and is a hyperparameter that controls the activation function's saturation rate. During training, $\mathbf{E}$ automatically updates the hidden dependencies among different data sequences.

Similar to defining a graph by node similarity as in [47], we adopted a self-multiplying approach to infer the dynamic correlations between each pair of nodes and generate a dynamic graph:

$$DA^t = ReLU(tanh(\alpha(DE^t DE^{t^T}))) \tag{8}$$

where $\mathbf{DA}^t$ represents the dynamic adjacency matrix at time step t, and the *ReLU* activation function is used to regularize the dynamic adjacency matrix by ensuring that all matrix values are non-negative.

4.3. Dynamic graph convolutional GRU

4.3.1. Dynamic graph convolution

Next, we obtain a distance-based static graph and node-attribute-based dynamic graph that reflect the correlations between nodes in the spatial and temporal dimensions, respectively. The latter provides an effective complement to the former. The

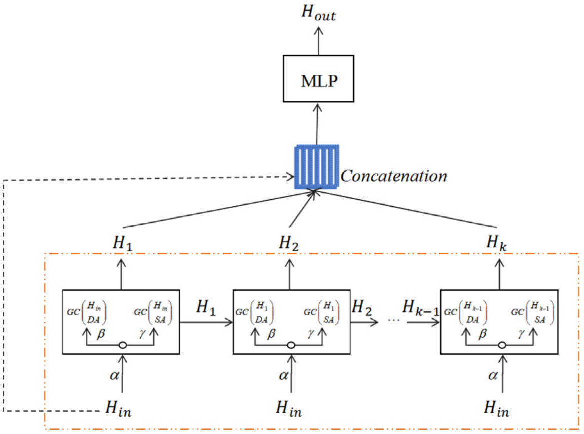


Fig. 3. Structure of dynamic graph convolution module. $GC()$ represents the graph convolution operation.

ensuing problem is the task of extracting and using the information between nodes based on the static and dynamic graphs to capture the spatial dependencies of the road network. We use a dynamic graph convolution module, as in [21], to perform a depth-weighted summation of the graph convolution results for the input feature, static graph $\mathbf{SA}$, and dynamic graph $\mathbf{DA}^t$. An example of a dynamic graph convolution module is illustrated in Fig. 3. The information propagation step is defined as follows:

$$H^k = \alpha H_{in} + \beta H^{k-1}\widetilde{DA^t} + \gamma H^{k-1}\widetilde{SA} \tag{9}$$

$$H_{out} = \sum_{i=0}^{K} H^k W^k, H^0 = H_{in} = con_t \tag{10}$$

where α, β, and γ are learnable hyperparameters that control the ratio for retaining the state of each component, K is the depth of propagation, W^k represents learnable parameters, and $\mathbf{H}_{out}$ denotes the output node state. In the process of in-depth information dissemination, we retain part of the original information at each depth to preserve the locality of the node state and explore a deep neighborhood [48]. Thus, the model obtains a broader view and learns richer spatial information at each moment.

4.3.2. DGCGRU

Because we employed the comprehensive information of the current moment to predict the missing data in the next moment to achieve step-by-step imputation, the model must capture the spatial and temporal dependencies of data concurrently. Therefore, following [21,49], we introduced a DGCGRU, wherein the matrix multiplication operation inside the GRU is replaced with the aforementioned dynamic graph convolution module. The DGCGRU is defined as follows:

$$z^t = \sigma(\Theta_{z*G}(\widehat{X}_t \parallel \widehat{H}_{t-1})) \tag{11}$$

$$r^t = \sigma(\Theta_{r*G}(\widehat{X}_t \parallel \widehat{H}_{t-1})) \tag{12}$$

$$c^t = tanh(\Theta_{c*G}(\widehat{X}_t \parallel (r^t \odot \widehat{H}_{t-1}))) \tag{13}$$

$$H_{X,t} = z^t \odot \widehat{H}_{t-1} + (1 - z^t) \odot c^t \tag{14}$$

where z^t and r^t are the reset gates and update gate at time t, respectively; σ denotes the sigmoid activation function; $*G$ represents the dynamic graph convolution module; Θ_z, Θ_r, and Θ_c are learnable parameters; and $\mathbf{H}_{X,t}$ are the output hidden states.

4.4. Information fusion

4.4.1. Masking GRU

In addition to mining information from traffic observation data, we considered the corresponding masking data; in terms of the type of data, they are both different—observation data are numeric, whereas masking data are Boolean [50]. Likewise, the former is a node feature, whereas the latter indicates the state (existing or missing) of the former, and reflects the missing data patterns. Thus, it is reasonable to process the two types of information using different components. In this study, we used an additional GRU to model the masking matrix:

$$H_{M,t} = GRU(M_t) \tag{15}$$

where $\mathbf{H}_{M,t}$ is the output hidden state of the masking GRU at time t.

4.4.2. Fusion layer with temporal decay

After obtaining the two hidden states, $\mathbf{H}_{X,t}$ and $\mathbf{H}_{M,t}$, representing different information, we applied a gated unit fusion strategy as in [51]. By applying a sigmoid activation function to $\mathbf{H}_{X,t}$, $\mathbf{H}_{M,t}$ is treated as a weighted filter gate, that selectively preserves the information of $\mathbf{H}_{X,t}$ according to the current and historical existing states of the observation data. The fusion strategy mechanism can be expressed as follows:

$$\widetilde{H}_{M,t} = \sigma(W_M H_{M,t} + b_M) \tag{16}$$

$$H_{F,t} = tanh(W_F(H_{X,t} \odot \widetilde{H}_{M,t}) + b_F) \tag{17}$$

where W_M, W_F, b_M and b_F are learnable parameters, and $\mathbf{H}_{F,t}$ denotes the fusion information at time t.

Intuitively, a larger time gap between the last observation and missing location minimizes the observation's contribution to the reconstruction of missing data. Accordingly, a temporal decay factor λ_t is also learned to capture this recession effect via a monotonically decreasing exponential function from δ_t as in [22,45,50]. The final hidden state $\widehat{H}_t$ is then generated as follows:

$$\lambda_t = \frac{1}{e^{max(0, W_\lambda \delta_t + b_\lambda)}} \tag{18}$$

$$\widehat{H}_t = \lambda_t \odot H_{F,t} \tag{19}$$

where W_λ and b_λ are learnable parameters.

4.5. Imputation

Based on $\widehat{H}_t$, an approximation $\widetilde{X}_{t+1}$ for the next timestamp $t + 1$ can be obtained using the following linear transformation:

$$\widetilde{X}_{t+1} = W_X \widehat{H}_t + b_X \tag{20}$$

$$\widehat{X}_{t+1} = X_{t+1} \odot M_{t+1} + \widetilde{X}_{t+1} \odot (1 - M_{t+1}) \tag{21}$$

where W_X and b_X are learnable parameters, and $\widehat{X}_{t+1}$ is the imputation result of $\mathbf{X}_{t+1}$. We replaced the missing values in $\mathbf{X}_{t+1}$ with the corresponding estimated values in $\widetilde{X}_{t+1}$ to explore the temporal dependencies of the observed component.

Finally, by considering the above with the given observations $\{X_s, X_{s+1}, \dots, X_{end}\}$, and corresponding related data $\{M_s, M_{s+1}, \dots, M_{end}\}$ and $\{\delta_s, \delta_{s+1}, \dots, \delta_{end}\}$, the model iteratively generates forward direction imputation $\{\widehat{X}^f_s, \widehat{X}^f_{s+1}, \dots, \widehat{X}^f_{end}\}$ and backward direction imputation $\{\widehat{X}^b_s, \widehat{X}^b_{s+1}, \dots, \widehat{X}^b_{end}\}$. The final imputation can be obtained by combining the following steps:

$$\widehat{X}_i = \begin{cases} \widehat{X}^b_s & i = s \\ \frac{1}{2}(\widehat{X}^f_i + \widehat{X}^b_i) & s < i < end \\ \widehat{X}^f_{end} & i = end \end{cases} \tag{22}$$

4.6. Loss

To optimize DGCRIN, we employed a loss function with two components to train the model as follows:

$$Loss = \frac{1}{T}\sum_{t=1}^{T} k_1 < M_t, L_e(X_t, \widehat{X}_t) > + k_2 < 1 - M_t, L_e(\widehat{X}^f_t, \widehat{X}^b_t) > \tag{23}$$

where k_1 and k_2 are hyper-parameters that control the proportion of the two-parts loss, and L_e represents the mean absolute error (MAE). In the formula, the first term measures the error between the observed and estimated values, ensuring that the model reconstructs the real observed value as much as possible. The second term enforces the missing estimation in each step to be consistent in both directions, thereby accelerating model convergence.

5. Experiment

To investigate the effectiveness of our proposed model, we conducted a series of experiments on two real traffic datasets with different missing data scenarios. The following section first introduces our dataset and evaluation metrics, and then provides a brief description of the comparison methods and experimental setting. Finally, relevant experimental results are analyzed and visualized.

5.1. Datasets

To evaluate the performance of DGCRIN, comparative experiments were conducted on two real-world traffic datasets: PeMS08 and PeMS04. Details regarding the datasets are presented in Table 1. Both represent public traffic speed datasets collected by the California Department of Transportation (Caltrans) at different times in different areas. We designed two methods to simulate missing data: random missing (**RM**) and continuous missing (**CM**). For RM, we randomly removed a certain amount of observed data in the matrix. For CM, which can be classified as temporal continuous missing (**TCM**) or spatially continuous missing (**SCM**), we randomly selected some contiguous regions

Table 1
Statistics of datasets.

Datasets	Time_range	Total_days	Samples	Nodes	Sample_rate	Input_len
PeMS-08	7/1/2016-8/31/2016	62	17856	170	5 min	12
PeMS-04	1/1/2018-2/28/2018	59	16992	307	5 min	12

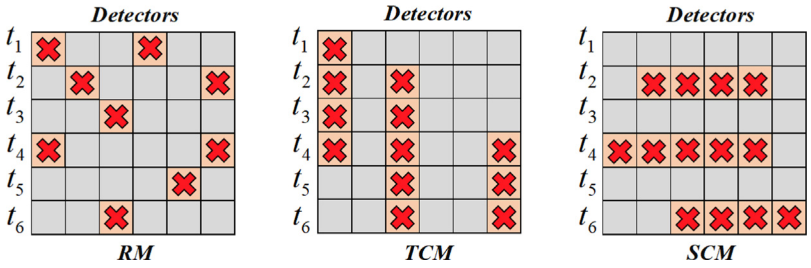


Fig. 4. Illustration of RM, TCM and SCM scenarios.

in the temporal or spatial dimensions, respectively, and removed all observations in those regions. Examples of all three methods are illustrated in Fig. 4. Following [51], we verified the model's validity with overall missing data rates of 10%, 30%, 50% and 70% Three commonly used evaluation metrics – **MAPE**, **RMSE** and **MAE** – were measured to test the accuracy of imputation:

$$MAPE(X,\widehat{X}) = \frac{1}{m}\sum_{i=1}^{m}|\frac{x_i - \hat{x}_i}{x_i}| \times 100\% \tag{24}$$

$$RMSE(X,\widehat{X}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{x}_i)^2} \tag{25}$$

$$MAE(X,\widehat{X}) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \hat{x}_i| \tag{26}$$

5.2. Benchmark imputation models

We tested a variety of state-of-the-art baselines to compare performance with DGCRIN. The following baselines were considered in the experiment:

(1) Traditional statistical method:

(a) HA: The historical average (HA) method uses the average of the previous five days to estimate the missing data.

(2) Bayesian matrix/tensor factorization methods:

(a) BGCP [7]: Bayesian Gaussian C/P (BGCP) decomposition uses the Markov chain algorithm to learn latent factor matrices, such as low-rank structures.
(b) BTTF [9]: Bayesian temporal tensor factorization, is a Bayesian probabilistic matrix factorization (BPMF) [52] model that integrates low-rank matrix/tensor factorization and vector autoregressive (VAR) processes into a single probabilistic graphical model that imposes temporal smoothness.

(3) Low-rank tensor completion method:

(a) LRTC-TNN [8]: In the Low-rank tensor completion (LRTC) with truncated nuclear norm (TNN), based on HaLRTC [53], a novel truncated kernel norm is defined on the tensor to solve the problem of missing spatiotemporal traffic data in a low-rank structure.

(4) GAN-based method:

(a) PC-GAIN [34]: A pseudo-label conditional GAIN is a GAN-based model that utilizes the potential category information of an entity to enhance the model's generation capabilities.

(5) Deep learning-based methods:

(a) BLSTM-I [29]: The bidirectional LSTM with an imputation mechanism is an RNN-based model that captures temporal dependencies using a bidirectional structure. An imputation unit was implemented inside the LSTM to handle missing values.
(b) GCBRNN [37]: A graph convolutional bidirectional recurrent neural network is a combination of GCN and GRU, which applies the graph convolution operation and 1 × 1 convolution module to capture the spatiotemporal dependencies in traffic data based on a static graph.

5.3. Experimental settings

The proposed model was implemented using PyTorch software. In the training stage, we utilized the Adam optimizer to update all model parameters, where the learning rate and batch size were set to 1*e*-3 and 128, respectively. We repeated the experiment five times with 200 epochs and reported the average values of all evaluation metrics. The node embedding dimensions and hidden-state size were set to 40 and 64, respectively. The proportions of the two losses *k1* and *k2* were assigned to 10 and 1, respectively. In addition, a 20-step early stopping method was adopted to optimize the model parameters and prevent overfitting.

For intelligent computing methods (PC-GAIN, BLSTM-I, and GCBRNN), we set the relevant hyperparameters according to the recommended values specified in the respective papers. For matrix/tensor methods (BGCP, BTTF, LRTC-TNN), we reconstructed all data into a third-order tensor (sensor × day × time slot) as input. The low ranks of BGCP and BTTF were set to 30.

5.4. Results

5.4.1. Imputation results and visualization

Tables 2 and 3 summarize the imputation performance of DGCRIN and its competing baselines for the two spatiotemporal datasets. We can observe from the tables that DGCRIN significantly outperformed all baseline models in different missing

Table 2
Performance comparison (in MAPE/RMSE/MAE) for imputation tasks with RM.

Datasets	PeMS08(RM)				PeMS04(RM)			
	10%	30%	50%	70%	10%	30%	50%	70%
HA	5.88% 5.493 2.591	5.83% 5.511 2.616	5.96% 5.587 2.647	6.11% 5.686 2.694	7.75% 6.518 3.435	7.83% 6.602 3.459	7.93% 6.681 3.486	7.99% 6.774 3.505
BGCP	3.4% 3.357 1.640	3.39% 3.379 1.650	3.44% 3.422 1.666	3.43% 3.434 1.665	4.52% 4.133 2.096	4.48% 4.106 2.080	4.52% 4.128 2.089	4.52% 4.130 2.097
BLSTM-I	2.92% 3.091 1.281	3.18% 3.333 1.398	3.62% 3.702 1.557	4.35% 4.413 1.837	4.17% 4.025 1.864	4.49% 4.304 1.995	4.71% 4.507 2.065	5.40% 5.074 2.329
BTTF	3.76% 3.724 1.845	3.76% 3.752 1.862	3.76% 3.754 1.852	3.81% 3.815 1.875	5.27% 4.881 2.435	5.28% 4.904 2.428	5.30% 4.905 2.436	5.28% 4.903 2.432
PC-GAIN	4.59% 4.437 2.155	4.56% 4.473 2.063	4.88% 4.675 2.230	10.47% 9.014 5.813	5.20% 4.482 2.454	4.80% 4.309 2.268	5.24% 4.696 2.487	13.02% 13.33 7.238
LRTC-TNN	1.38% 1.379 0.735	1.61% 1.662 0.846	2.01% 2.061 1.025	2.58% 2.648 1.289	1.95% 1.780 1.001	2.34% 2.181 1.178	2.94% 2.734 1.429	3.57% 3.313 1.722
GCBRNN	1.18% 1.304 0.660	1.48% 1.658 0.801	1.99% 2.140 1.053	2.92% 2.950 1.492	**1.51%** 1.552 0.822	1.91% 2.011 0.989	2.60% 2.640 1.270	4.10% 3.714 1.907
DGCRIN	**1.16%** **1.261** **0.648**	**1.34%** **1.562** **0.746**	**1.73%** **1.913** **0.882**	**2.34%** **2.512** **1.134**	**1.51%** **1.511** **0.817**	**1.82%** **1.863** **0.938**	**2.29%** **2.295** **1.121**	**3.07%** **3.019** **1.416**

The best results are highlighted in bold, and the second-best results are underlined.

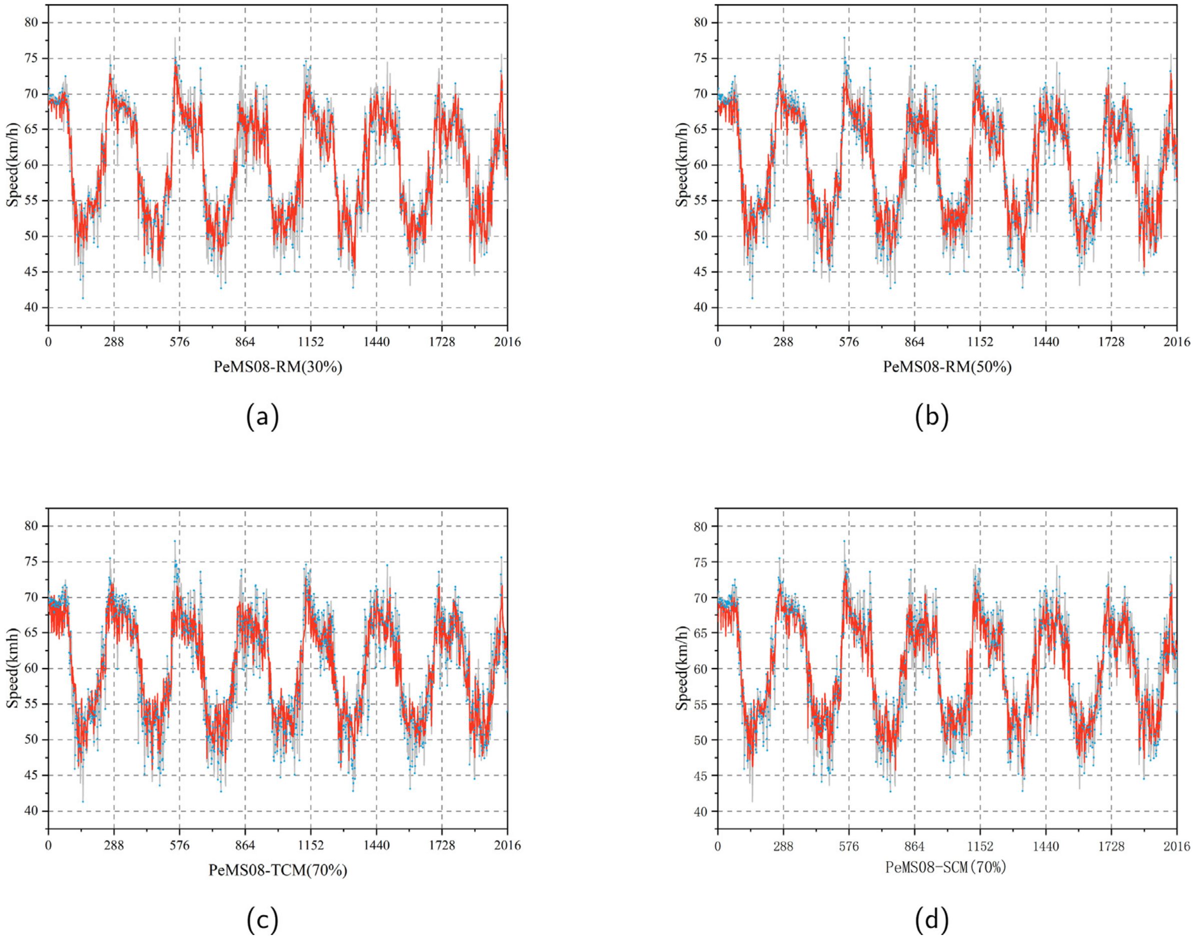
Table 3
Performance comparison (in MAPE/RMSE/MAE) under TCM and SCM for imputation tasks.

Datasets	PeMS08(TCM)		PeMS08(SCM)		PeMS04(TCM)		PeMS04(SCM)	
	30%	70%	30%	70%	30%	70%	30%	70%
HA	5.86% 5.527 2.616	6.11% 5.703 2.698	5.82% 5.511 2.615	6.22% 5.762 2.732	7.84% 6.607 3.463	8.02% 6.781 3.506	7.94% 6.633 3.474	8.00% 6.768 3.501
BGCP	3.42% 3.425 1.652	3.47% 3.485 1.680	3.43% 3.404 1.656	3.51% 3.493 1.697	4.51% 4.138 2.094	4.54% 4.142 2.093	4.53% 4.130 2.085	4.55% 4.143 2.102
BLSTM-I	4.36% 4.378 1.868	4.87% 4.702 2.107	3.23% 3.359 1.410	4.43% 4.321 1.841	4.82% 4.563 2.133	5.44% 5.050 2.351	4.48% 4.300 1.970	5.30% 4.962 2.294
BTTF	3.84% 3.795 1.887	3.83% 3.812 1.894	3.81% 3.780 1.877	3.86% 3.855 1.893	5.33% 4.908 2.461	5.32% 4.916 2.449	5.34% 4.910 2.436	5.30% 4.903 2.436
PC-GAIN	4.54% 4.433 2.072	11.63% 9.325 5.922	4.96% 4.770 2.282	13.24% 10.029 6.269	4.78% 4.265 2.233	13.20% 13.407 7.307	5.12% 4.411 2.334	16.10% 15.288 9.251
LRTC-TNN	2.42% 2.509 1.210	2.78% 2.881 1.386	1.64% 1.665 0.853	2.97% 3.622 1.489	3.39% 3.166 1.637	3.81% 3.562 1.834	2.39% 2.201 1.187	3.69% 3.689 1.782
GCBRNN	2.71% 2.811 1.371	3.28% 3.223 1.633	1.42% 1.643 0.840	3.03% 3.012 1.521	3.60% 3.385 1.667	4.52% 4.018 2.07	1.93% 2.032 0.995	4.13% 3.879 1.876
DGCRIN	**2.28%** **2.467** **1.114**	**2.70%** **2.732** **1.245**	**1.41%** **1.545** **0.756**	**2.55%** **2.797** **1.214**	**3.38%** **3.062** **1.466**	**3.67%** **3.409** **1.614**	**1.83%** **1.871** **0.944**	**3.15%** **3.070** **1.446**

The best results are highlighted in bold, and the second-best results are underlined.

scenarios. For most models, it is apparent that the CM scenario is more challenging than the RM scenario for the imputation task. The MAPE/RMSE/MAE values generally increased with missing rates, suggesting that the patterns and rates of missing data have a significant impact on model performance. Furthermore, all models performed worse on PeMS04 than PeMS08, which may be attributed to the higher number of road network nodes that increased the complexity of the former. In most cases, the data-distribution-based method PC-GAIN performed poorly and appeared insufficient with high rates of missing data. Matrix/tensor factorization methods (BGCP and BTTF) accumulated relatively large errors because they ignore temporal characteristics.

In particular, the performance of the BLSTM-I, GCBRNN, and DGCRIN models degraded the most in the TCM scenario, which



(a)

(b)

(c)

(d)

Fig. 5. Imputation examples for PeMS08 under different missing scenarios (8/20 − 8/26, 2016, sensor #133). In these panels, gray curves represent the ground truth, red curves denote the imputed values, and blue dots correspond to missing points.

may be related to the model structure: all three models are based on an iteratively generated pattern that is prone to error accumulation in the time dimension. One primary difference between DGCRIN and GCBRNN is that the former considers the dynamic correlation of the road network. These experimental results demonstrate the importance of dynamic spatiotemporal correlations for traffic data imputation, and the effectiveness of DGCRIN in modeling dynamic spatiotemporal dependencies.

To illustrate the proposed model's imputation effect more intuitively, Figs. 5 and 6 visualize imputation instances for PeMS08 and PeMS04 under different missing scenarios.

5.4.2. Ablation study

To examine the validity of the submodules (i.e., time-lag matrix, masking GRU, dynamic graph, and static graph) in the DGCRIN framework, we performed ablation experiments with four variants of PeMS08 in RM with a 30% missing rate:

(1) DGCRIN-I: DGCRIN with only a forward process to demonstrate the advantages of bidirectional learning.
(2) DGCRIN-II: DGCRIN without a time-lag matrix to decay the fusion hidden state.
(3) DGCRIN-III: DGCRIN without masking GRU to learn the missing pattern information.
(4) DGCRIN-IV: DGCRIN without a graph generator to learn the spatial correlations between road network nodes.
(5) DGCRIN-V: DGCRIN without a static graph in the information propagation process of the dynamic graph convolution module.

The experimental results of the ablation studies are shown in Fig. 7. The results illustrate that DGCRIN is superior to other variants. The following conclusions can be drawn from these results. First, the bidirectional structure has a significant impact on model performance, as it enables the model to comprehensively consider the forward and backward propagation of data. Second, the dynamic graph module is a critical component, as it extracts potential spatiotemporal dependency information by tracking the dynamic changes of the road network, which is beneficial for the imputation task. Third, the positive effect of the masking GRU is evident. Finally, both the decay mechanism of the time-lag matrix and the static graph have been demonstrated to improve performance.

5.4.3. Hyperparameter sensitivity

To further analyze DGCRIN, we evaluated model performance with different hyperparameters – the node embedding dimension (E) and size of hidden state (H) – on PeMS08 under the 20% RM scenario. We only reported the RMSE and MAE results, as the MAPE results are less volatile. As shown in Fig. 8, an increase in E correlates with a gradual decrease in MSE and MAE, as the complexity of road conditions requires a higher dimensionality of node embedding to store sufficient information. When E becomes very large, performance decreases, which may be a result of overfitting. We then changed H from 16 to 128, the results are shown in Fig. 9. RMSE and MAE showed similar variation patterns, as both initially decreased with an increase in H, and started to increase when H exceeded 64. One possible cause of the eventual decrease is the redundancy of hidden state information.

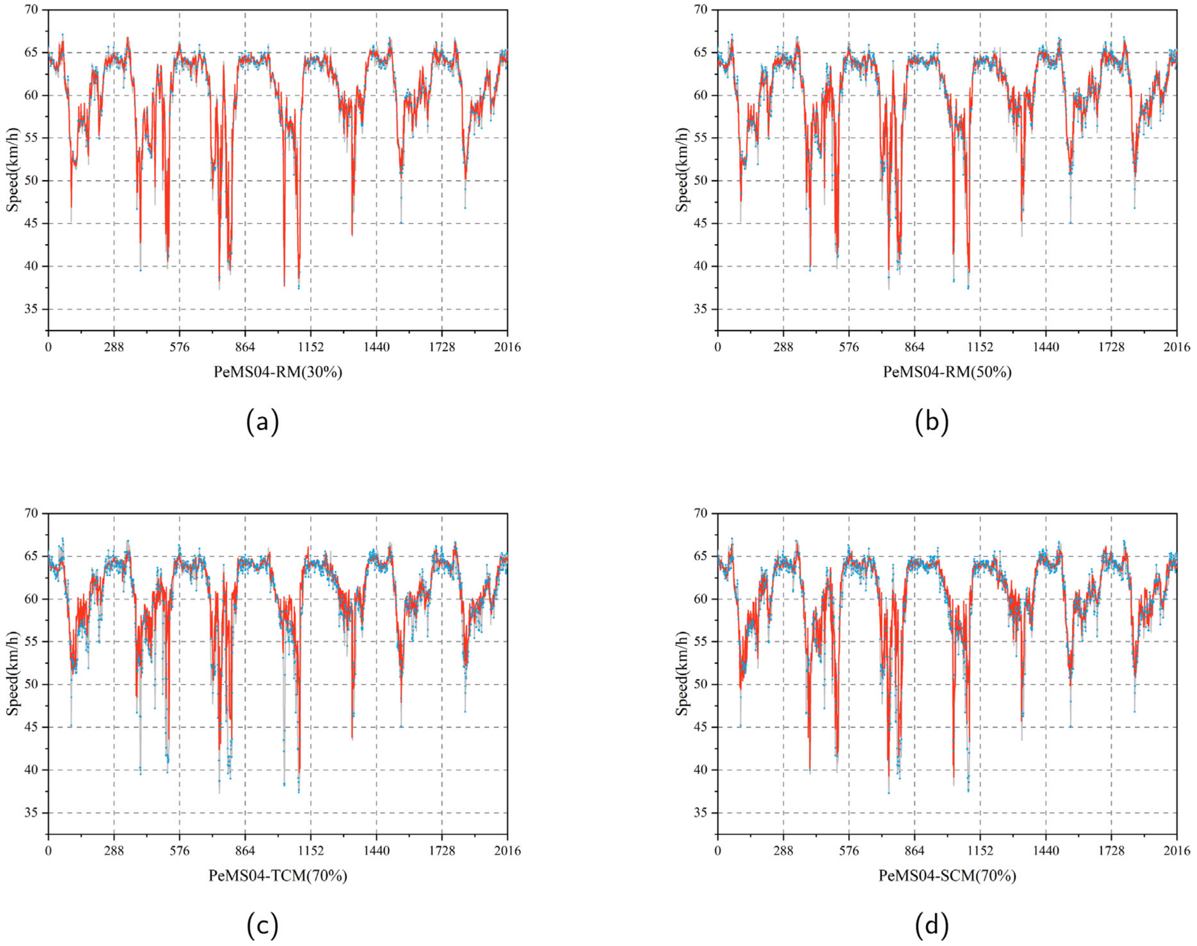


Fig. 6. Imputation examples for PeMS04 under different missing scenarios (2/15 − 2/21, 2018, sensor #96). In these panels, gray curves indicate the ground truth, red curves denote the imputed values, and blue dots represent missing data.

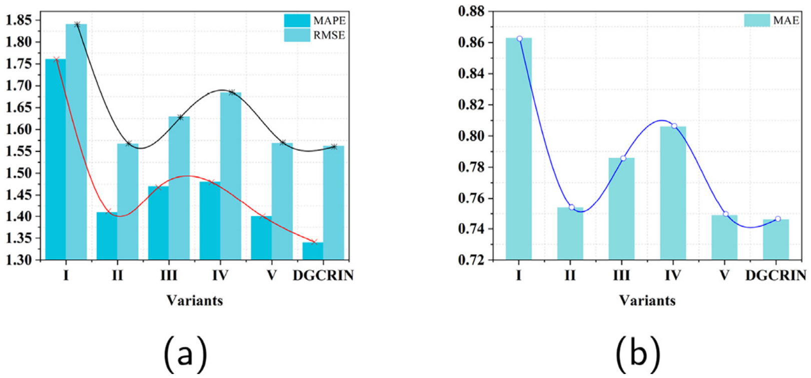


Fig. 7. Ablation results of DGCRIN on PeMS08 in RM with 30% missing-data rate.

6. Conclusion

In this study, we modeled the dynamic spatial dependencies of road networks under the conditions of incomplete traffic data, and developed DGCRIN for traffic data imputation. Inspired by the iterative generation characteristics of RNNs, we designed a novel graph generator to model the dynamic spatial correlations between road network nodes at each moment using the recurrent generated imputation data and historical information. Based on the dynamic graph, we employed a DGCGRU module to efficiently capture the spatiotemporal characteristics of data by applying graph convolution to static and dynamic graphs. In addition, we studied how to effectively mine information from a variety of diverse data to provide gains for imputation tasks. Extensive experiments on two real-world datasets proved the competitive computational performance of the proposed method over that of existing methods under different missing data scenarios.

In the future, we will continue to develop more efficient methods to model the dynamic spatial correlations of incomplete traffic data. We also plan to integrate additional external factors

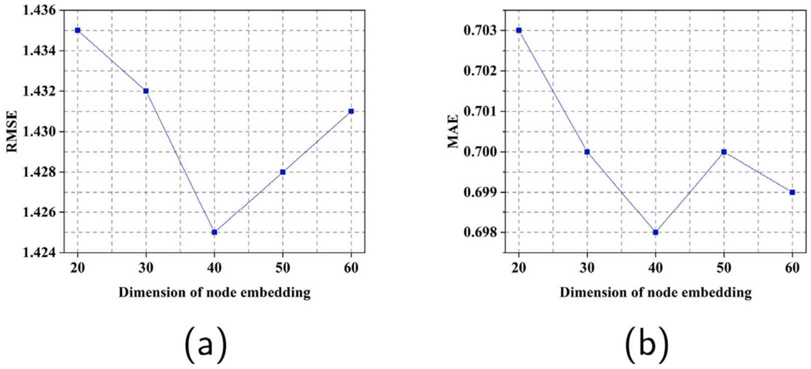


Fig. 8. Hyperparameter E sensitivity test.

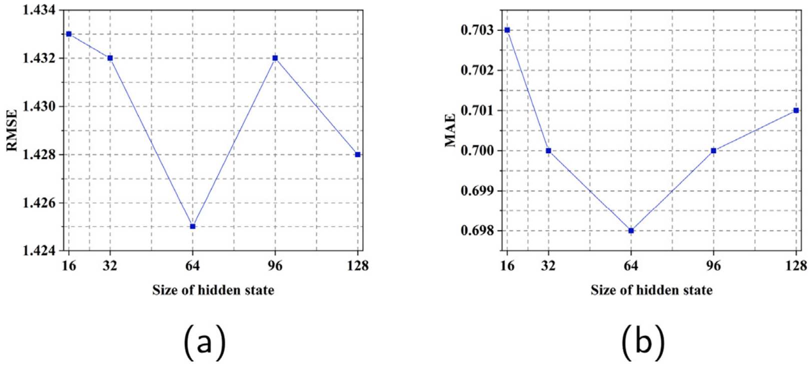


Fig. 9. Hyperparameter H sensitivity test.

(such as weather and POI similarity graphs) into the model, with the expectation that DGCRIN can perform data imputation and prediction tasks simultaneously.

CRediT authorship contribution statement

Xiangjie Kong: Conceptualization, Methodology, Investigation. **Wenfeng Zhou:** Methodology, Writing – original draft, Validation, Software. **Guojiang Shen:** Supervision, Project administration, Funding acquisition. **Wenyi Zhang:** Writing - review, Validation, Visualization. **Nali Liu:** Writing – review, Validation. **Yao Yang:** Validation.

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.

Acknowledgments

This work was supported in part by the Zhejiang Provincial Natural Science Foundation, China under Grant LR21F020003, in part by the National Natural Science Foundation of China under Grant 62072409 and Grant 62073295, in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2022C01050, and in part by Key Research Project of ZheJiang Lab under Grant 2022NF0AC01.

References

[1] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, Fei-Yue Wang, An efficient realization of deep learning for traffic data imputation, Transp. Res. C 72 (2016) 168–181.

[2] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, Hui Xiong, Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 547–555.

[3] Guojiang Shen, Difeng Zhu, Jingjing Chen, Xiangjie Kong, Motif discovery based traffic pattern mining in attributed road networks, Knowl.-Based Syst. (2022) 109035.

[4] Xiangjie Kong, Qiao Chen, Mingliang Hou, Azizur Rahim, Kai Ma, Feng Xia, RMGen: A tri-layer vehicular trajectory data generation model exploring urban region division and mobility pattern, IEEE Trans. Veh. Technol. (2022).

[5] Weihao Yin, Pamela Murray-Tuite, Hesham Rakha, Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods, J. Intell. Transp. Syst. 16 (3) (2012) 159–176.

[6] Roderick J.A. Little, Donald B. Rubin, Statistical Analysis with Missing Data, Vol. 793, John Wiley & Sons, 2019.

[7] Xinyu Chen, Zhaocheng He, Lijun Sun, A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation, Transp. Res. C 98 (2019) 73–84.

[8] Xinyu Chen, Jinming Yang, Lijun Sun, A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation, Transp. Res. C 117 (2020) 102673.

[9] Xinyu Chen, Lijun Sun, Bayesian temporal factorization for multidimensional time series prediction, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[10] Guojiang Shen, Xiao Han, KwaiSang Chin, Xiangjie Kong, An attention-based digraph convolution network enabled framework for congestion recognition in three-dimensional road networks, IEEE Trans. Intell. Transp. Syst. (2021).

[11] Cai Xu, Wei Zhao, Jinglong Zhao, Ziyu Guan, Xiangyu Song, Jianxin Li, Uncertainty-aware multi-view deep learning for Internet of Things applications, IEEE Trans. Ind. Inform. (2022).

[12] Dongda Li, Zhaoquan Gu, Yuexuan Wang, Changwei Ren, Francis CM Lau, One model packs thousands of items with recurrent conditional query learning, Knowl.-Based Syst. 235 (2022) 107683.

[13] Xiaokang Zhou, Wei Liang, Shohei Shimizu, Jianhua Ma, Qun Jin, Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems, IEEE Trans. Ind. Inform. 17 (8) (2020) 5790–5798.

[14] Xiangjie Kong, Bing Zhu, Guojiang Shen, Tewabe Chekole Workneh, Zhanhao Ji, Yang Chen, Zhi Liu, Spatial-temporal-cost combination based taxi driving fraud detection for collaborative Internet of Vehicles, IEEE Trans. Ind. Inform. 18 (5) (2021) 3426–3436.

[15] Xiangyu Song, Jianxin Li, Yifu Tang, Taige Zhao, Yunliang Chen, Ziyu Guan, JKT: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523.

[16] Yoshua Bengio, Francois Gingras, Recurrent neural networks for missing or asynchronous data, Adv. Neural Inf. Process. Syst. 8 (1995).

[17] Yifan Zhuang, Ruimin Ke, Yinhai Wang, Innovative method for traffic data imputation based on convolutional neural network, IET Intell. Transp. Syst. 13 (4) (2019) 605–613.

[18] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, Lijun Sun, Inductive graph neural networks for spatiotemporal kriging, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vool. 35, no. 5, 2021, pp. 4478–4485.

[19] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, Weiyang Kong, LSGCN: Long short-term traffic prediction with graph convolutional networks, in: IJCAI, 2020, pp. 2355–2361.

[20] Guopeng Li, Victor L. Knoop, Hans van Lint, Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations, Transp. Res. C 128 (2021) 103185.

[21] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, Yong Li, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, ACM Trans. Knowl. Discov. Data (TKDD) (2021).

[22] Shuo Yang, Minjing Dong, Yunhe Wang, Chang Xu, Adversarial recurrent time series imputation, IEEE Trans. Neural Netw. Learn. Syst. (2020).

[23] Dong Li, Linhao Li, Xianling Li, Zhiwu Ke, Qinghua Hu, Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation, Neurocomputing 411 (2020) 351–363.

[24] Difeng Zhu, Guojiang Shen, Jingjing Chen, Wenfeng Zhou, Xiangjie Kong, A higher-order motif-based spatiotemporal graph imputation approach for transportation networks, Wirel. Commun. Mob. Comput. 2022 (2022).

[25] Xiaokang Zhou, Yue Li, Wei Liang, CNN-RNN based intelligent recommendation for online medical pre-diagnosis support, IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (3) (2020) 912–921.

[26] Junhui Zhao, Yiwen Nie, Shanjin Ni, Xiaoke Sun, Traffic data imputation and prediction: An efficient realization of deep learning, IEEE Access 8 (2020) 46713–46722.

[27] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, Yan Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (2018) 1–12.

[28] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, Bailin Yang, LSTM-Based traffic flow prediction with missing data, Neurocomputing 318 (2018) 297–305.

[29] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Yinhai Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, Transp. Res. C 118 (2020) 102674.

[30] Yi-Fan Zhang, Peter J. Thorburn, Wei Xiang, Peter Fitch, SSIM-A deep learning approach for recovering missing time series sensor data, IEEE Internet Things J. 6 (4) (2019) 6618–6628.

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[32] Jinsung Yoon, James Jordon, Mihaela Schaar, Gain: Missing data imputation using generative adversarial nets, in: International Conference on Machine Learning, PMLR, 2018, pp. 5689–5698.

[33] Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, Salvatore Antonio Biancardo, Junyi Zhang, Missing data repairs for traffic flow with self-attention generative adversarial imputation net, IEEE Trans. Intell. Transp. Syst. (2021).

[34] Yufeng Wang, Dan Li, Xiang Li, Min Yang, PC-GAIN: Pseudo-label conditional generative adversarial imputation networks for incomplete data, Neural Netw. 141 (2021) 395–403.

[35] Ye Yuan, Yong Zhang, Boyue Wang, Yuan Peng, Yongli Hu, Baocai Yin, STGAN: Spatio-temporal generative adversarial network for traffic data imputation, IEEE Trans. Big Data (2022).

[36] Xiangyu Song, Jianxin Li, Qi Lei, Wei Zhao, Yunliang Chen, Ajmal Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.

[37] Zhengchao Zhang, Xi Lin, Meng Li, Yinhai Wang, A customized deep learning approach to integrate network-scale online traffic data imputation and prediction, Transp. Res. C 132 (2021) 103372.

[38] Xin Yao, Yong Gao, Di Zhu, Ed Manley, Jiaoe Wang, Yu Liu, Spatial origin-destination flow imputation using graph convolutional networks, IEEE Trans. Intell. Transp. Syst. 22 (12) (2020) 7474–7484.

[39] Dongwei Xu, Hang Peng, Chenchen Wei, Xuetian Shang, Haijian Li, Traffic state data imputation: An efficient generating method based on the graph aggregator, IEEE Trans. Intell. Transp. Syst. (2021).

[40] Yaming Yang, Ziyu Guan, Jianxin Li, Wei Zhao, Jiangtao Cui, Quan Wang, Interpretable and efficient heterogeneous graph convolutional network, IEEE Trans. Knowl. Data Eng. (2021).

[41] Xiaokang Zhou, Wei Liang, I. Kevin, Kai Wang, Laurence T. Yang, Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations, IEEE Trans. Comput. Soc. Syst. 8 (1) (2020) 171–178.

[42] Mingwei Zhang, Guiping Wang, Lanlan Ren, Jianxin Li, Ke Deng, Bin Zhang, METoNR: A meta explanation triplet oriented news recommendation model, Knowl.-Based Syst. 238 (2022) 107922.

[43] Weida Zhong, Qiuling Suo, Xiaowei Jia, Aidong Zhang, Lu Su, Heterogeneous spatio-temporal graph convolution network for traffic forecasting with missing values, in: 2021 IEEE 41st International Conference on Distributed Computing Systems, ICDCS, IEEE, 2021, pp. 707–717.

[44] Yongchao Ye, Shiyao Zhang, James J.Q. Yu, Spatial-temporal traffic data imputation via graph attention convolutional network, in: International Conference on Artificial Neural Networks, Springer, 2021, pp. 241–252.

[45] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, Yitan Li, Brits: Bidirectional recurrent imputation for time series, Adv. Neural Inf. Process. Syst. 31 (2018).

[46] Guotong Xue, Ming Zhong, Jianxin Li, Jia Chen, Chengshuai Zhai, Ruochen Kong, Dynamic network embedding survey, Neurocomputing 472 (2022) 212–223.

[47] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, Can Wang, Adaptive graph convolutional recurrent network for traffic forecasting, Adv. Neural Inf. Process. Syst. 33 (2020) 17804–17815.

[48] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, Chengqi Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 753–763.

[49] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, Haifeng Li, T-GCN: A temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 21 (9) (2019) 3848–3858.

[50] Jiawei Ouyang, Yuhao Zhang, Xiangrui Cai, Ying Zhang, Xiaojie Yuan, ImputeRNN: Imputing missing values in electronic medical records, in: International Conference on Database Systems for Advanced Applications, Springer, 2021, pp. 413–428.

[51] Xuesong Wu, Mengyun Xu, Jie Fang, Xiongwei Wu, A multi-attention tensor completion network for spatiotemporal traffic data imputation, IEEE Internet Things J. (2022).

[52] Ruslan Salakhutdinov, Andriy Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 880–887.

[53] Ji Liu, Przemyslaw Musialski, Peter Wonka, Jieping Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 208–220.