A Two-Stage Imputation Method for Urban Traffic Data Based on Sparse Spatiotemporal Attention

Jiyu Wang and Zhidan Liu, *Senior Member, IEEE*

***Abstract*—Accurate imputation of missing traffic data remains critical for intelligent transportation systems, but existing methods often inadequately balance global patterns and localized correlations. This study proposes a novel two-stage imputation framework that harmonizes tensor decomposition with deep learning mechanisms. In the first stage, low-rank tensor completion (LRTC) model extracts latent low-rank features from incomplete traffic data, establishing global priors. The second stage fuses the priors with raw observations via a fusion module, feeding the enhanced representation into a Spatial-Temporal Feature Enhancement Network (STFEN) with scalable window attention. Directed search for strongly relevant features in sparse spatiotemporal attention by introducing domain prior knowledge effectively improves accuracy and computational efficiency. A comprehensive evaluation of three real-world traffic datasets demonstrates the superiority of the two-stage imputation approach compared to state-of-the-art baselines under different missing scenarios. We further systematically evaluate seven matrix/tensor decomposition variants as potential candidates for the first stage processor, with experimental results demonstrating that superior first stage model accuracy enables greater precision recovery in the subsequent phase. This work establishes a new methodological framework to enhance sparse traffic data imputation, thereby facilitating the deployment of data-driven architectures in urban computing systems.**

***Index Terms*—Traffic data imputation, spatiotemporal dependence, sparse self-attention, tensor completion.**

I. Introduction

THE proliferation of smart city initiatives has exacerbated the reliance on data-driven models where the integrity of the data used for training is directly correlated to the performance of downstream tasks [1]. However, the acquisition of complete traffic data in real-world scenarios is constrained by two primary factors. First, budget limitations restrict infrastructure deployments, compelling transportation departments to focus sensor installations on key arterial roadways rather than achieving comprehensive network coverage. This inevitably results in observation gaps on non-instrumented roadways, thereby impacting system-wide traffic analysis. Second, even the road sections equipped with sensors are vulnerable to adverse weather conditions, communication failures, and other disruptive factors, which can lead to data loss and degradation [2]. Such incomplete data significantly undermines the effectiveness of downstream traffic applications, impeding the ability of traffic management systems to accurately assess and respond to real-time traffic conditions [3]. As a result, building an efficient methodology to estimate missing traffic data using observational data is a pressing issue that can assist minimize the concerns listed above.

Traditional statistics-based estimating approaches require data that meet statistical standards or certain assumptions. Their simplistic structure is incapable of capturing complex and dynamic traffic spatiotemporal patterns, leading to poor performance in traffic data imputation tasks [4]. Existing data-driven methods based on tensor completion and deep learning are able to achieve better performance by further mining spatiotemporal patterns. Tensor-based methodologies leverage low-rank approximations to impute missing entries, where a variety of works have explored the impact of different tensor ranks on data imputation tasks [5], [6]. Contemporary research extends beyond global low-rank pattern extraction by imposing linear regularization constraints across multidimensional data domains [7], [8]. The increasing popularity of deep learning techniques can be attributed to their capacity to effectively capture traffic data's complicated, dynamic, and nonlinear spatiotemporal dependence [9]. Several deep time series models have used the cumulative error propagation mechanism and GNN-based models to aggregate road network node characteristics for traffic data imputation effectively [10], [11]. Self-attention mechanism can effectively establish a dynamic and adaptive interaction method between missing data and observed data to meet various types of scenarios [12].

Although significant efforts have been made to improve the traffic data imputation results, existing models still face two key shortcomings:

First, prior studies fail to simultaneously account for both the global low-rank structure and local spatiotemporal nonlinear correlations inherent in traffic data. Due to the complex interactions between heterogeneous sensors and temporal dynamics, linear regularization in tensor completion may inadequately capture the local consistency patterns of real-world traffic systems [13]. This necessitates more sophisticated approaches, such as deep learning architectures, to effectively represent nonlinear spatiotemporal dependence. While

Received 18 March 2025; revised 29 July 2025, 9 November 2025, and 26 January 2026; accepted 22 March 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62572416 and in part by Guangdong Provincial Key Laboratory of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007. The Associate Editor for this article was W.-L. Shang. *(Corresponding author: Zhidan Liu.)*

The authors are with the Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China (e-mail: jwang738@connect.hkust-gz.edu.cn; zhidanliu@hkust-gz.edu.cn).

Digital Object Identifier 10.1109/TITS.2026.3677887

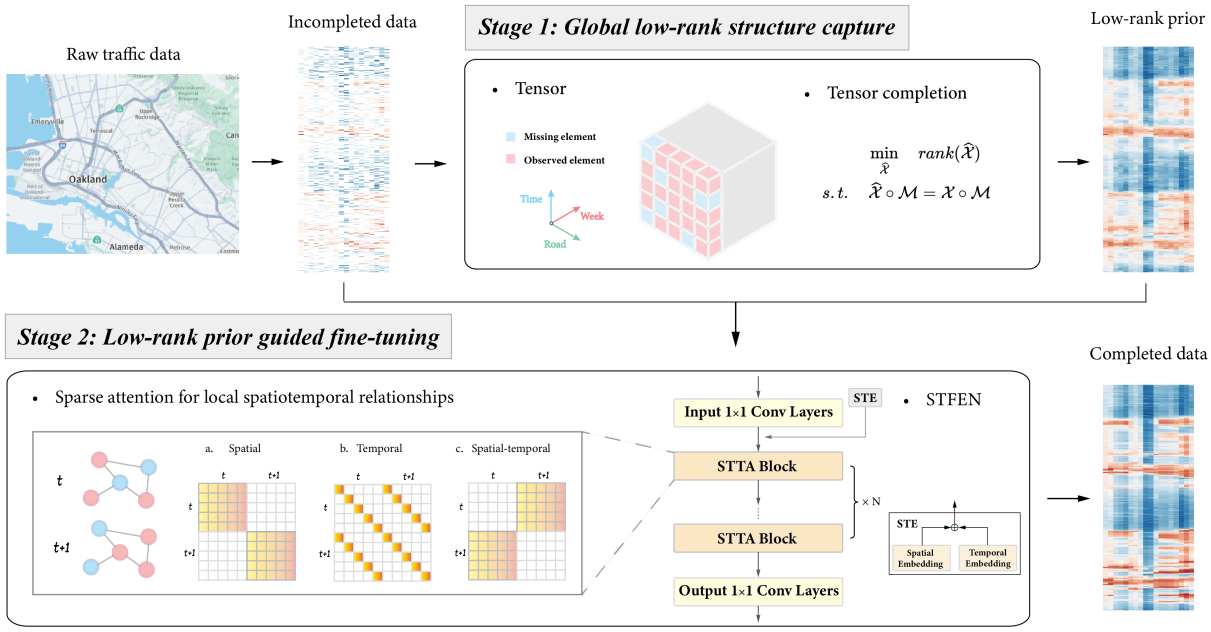


Fig. 1. Framework of two-stage traffic data imputation method. In the first stage, a tensor completion model is used to extract global low-rank priors from incomplete traffic data; in the second stage, STFEN, guided by the global low-rank priors, further captures local spatiotemporal relationships to complete the imputation of missing data.

deep learning techniques demonstrate strong capabilities in modeling localized nonlinear relationships (typically with sequence lengths ranging from 12 to 72 steps) [14], they lack explicit mechanisms to characterize the global low-rank patterns across the complete dataset.

Second, existing deep spatiotemporal networks applied to traffic data imputation tasks lack modeling of spatial-temporal dependence. The strategy of stacking temporal and spatial feature extraction modules used in prior researches cannot directly capture the spatial-temporal dependence [15], and the primitive traffic status has already been covered. Recent traffic prediction studies have proposed synchronous graph neural networks to capture nonlinear spatial-temporal dependence [16], [17]. However, their performance is constrained by a predefined topological network, making it difficult to pay attention to dynamic relationships accurately and lack scalability. Especially for traffic data imputation, the contextual information of temporal semantics should be considered.

Considering that traffic data has both global low-rank structure and local nonlinear spatiotemporal relationships, a two-stage traffic data imputation framework is proposed to deal with the task of traffic data, as shown in Fig. 1. In the first stage, we use low-rank tensor completion (LRTC) to use to obtain a global low-rank prior and input it to the next stage by fusing it with the observed data. In the second stage, a Spatial-Temporal Feature Enhancement Network (STFEN) is developed to capture all dynamic relationships and maintain local spatiotemporal consistency for data enhancement guided by the low-rank prior. Specifically, temporal, spatial, and spatial-temporal dependence are explored synchronized through scalable sparse spatiotemporal attention, all while reducing computing complexity and memory usage. The temporal attention is used to compensate for the lack of temporal dependence that may result and thereby enhance the temporal dependence. Based on the global low-rank initial value in the first stage, imputation accuracy is further improved. This framework demonstrates excellent compatibility and has been validated through integration with seven alternative global pattern extractors (matrix completion, CP decomposition, TNN, etc.) outside of LRTC. Therefore, the method we propose is sufficient to deal with a variety of missing data scenarios. Our contributions are summarized as follows:

- We develop a two-stage traffic data imputation method to estimate missing data by taking into account the global low-rank structure and local nonlinear spatio-temporal features of traffic data.This architecture enables flexible integration of diverse matrix/tensor factorization techniques to extract global features.
- We propose a scalable sparse spatiotemporal attention mechanism that captures three nonlinear spatiotemporal dependence underlying traffic data. At the same time, the memory overhead and computational complexity of the operation are reduced.
- Validated by three real-world traffic datasets with multiple scenarios, the two-stage data imputation method achieves better imputation accuracy than baseline methods.

The remainder of this paper is organized as follows: Section II offers an overview of relevant works includ ing traffic imputatioin method and self-attention mechanism. Section III introduces the problem and imputation process. Section IV details our proposed architecture. Subsequently, Section V presents a thorough comparison with existing methods, including ablation studies and sensitivity analysis. Finally, Section VI concludes the paper.

II. Literature Review

This section provides an overview of the data-driven techniques utilized for the imputation of missing traffic data, specifically focusing on tensor completion and deep learning approaches. We also review self-attention mechanisms applied to modeling spatiotemporal relationships in traffic data.

A. Traffic Data Imputation Approaches

The fundamental principle underlying the process of traffic data imputation remains the identification of traffic patterns or laws based on observed data, which are then utilized to estimate missing values [18]. Various data restoration methods in traffic analysis, such as mean, KNN, and ARIMA, rely on statistical principles and necessitate the fulfillment of specific assumptions regarding the data [19]. These methods have a simplistic structure and cannot effectively capture intricate and ever-changing traffic spatiotemporal patterns, resulting in subpar performance when applied to traffic data imputation tasks.

The tensor completion approach is proposed for traffic data imputation, taking into account the inherent low-rank nature of the traffic data. Reference [20] apply Tucker decomposition to a four-dimensional traffic data tensor for the imputation of missing traffic flows. Considering the uncertainty of traffic data, [21] propose a Bayesian tensor decomposition method to estimate the traffic state. To avoid the impact of the tensor rank hyperparameter on the modeling of traffic spatiotemporal data, the tensor completion method that minimizes the tensor rank is widely used. such as truncated nuclear norm [5], truncated schatten p-norm [6], weighted tensor nuclear norm [22]. Some studies [23] maintain the local consistency of spatial and temporal by imposing linear regularization on different dimensions of tensor data and limiting the degree of local variation. Reference [8] use temporal second-order difference and Hessian regularization to impose stricter restrictions on the data along the temporal and spacial dimensions.

Deep learning techniques leverage the nonlinear relationships present in data to provide accurate estimations for missing data [24]. Reference [25] propose a data imputation framework BRITS based on the cumulative propagation mechanism of short-term timing errors. Reference [10] have made improvements based on prior knowledge, such as the periodicity of traffic data time series. References [26] and [27] employ a self-attention method to extract dynamic and adaptive temporal and spatial patterns with minimal observational data. Reference [28] introduce a kriging model to reconstruct traffic signals at road nodes where no signal is detected at the same time. Furthermore, generative adversarial networks are employed for the estimation of missing values owing to their exceptional capacity to produce data distributions [29], recent research has found that adding multi-level composite spatiotemporal feature extractors is beneficial to improve its stability [30]. Reference [31] further introduce the constraints of conditional loops on the basis of the adversarial generative network and adopt a self-supervised training paradigm. When comparing the stability of the confrontation generation network with the diffusion model, it can be observed that the diffusion model exhibits greater robustness and accuracy in terms of data recovery. Since it takes a lot of time and computational cost to separately train the same model for different missing rates and missing data patterns, [32] propose a pre-training solution strategy. Although Imputeformer integrates low-rank structure with deep learning, it operates exclusively on localized time series segments [33]. Consequently, it inherently lacks the capability to capture and express the global low-rank structure present within the complete dataset, which is a key distinction addressed by our framework.

For completing the imputation task, it is important to take into account the low-rank structure and nonlinear spatiotemporal relationships of the traffic data. When it comes to traffic data modeling, however, tensor completion and deep learning approaches can only take one characteristic into account and leave the other out. In this research, we attempt to integrate deep learning approaches with tensor completion in a two-stage process to fully extract patterns from traffic data.

B. Self-Attention Mechanism

The attention mechanism is capable of identifying and emphasizing significant semantic information features through the identification of correlations within the data [34]. The early models emphasize exploring attention mechanisms as a means to enhance the performance of recurrent neural networks, specifically inside encoder-decoder frameworks [35]. After the introduction of the self-attention mechanism, it breaks the constraints of temporal auto-regression and road network topology, enabling the model to learn long-term dynamic and complex spatiotemporal features [36], [37]. Previous research pioneer stacking temporal self-attention modules and spatial self-attention modules in series or in parallel to integrate spatiotemporal features fully and approximately extract spatiotemporal dependence [38], [39]. Some models integrate networks like LSTM and GCN with self-attention mechanisms to concurrently capture long-term and short-term spatiotemporal patterns between data, avoiding the model's failure in short-term feature acquisition [40], [41]. In order to improve the model's robustness in dealing with emergencies in traffic scenes, a threshold strategy can be used to limit the attention mechanism's acceptance range of information, and multi-dimensional traffic features can be mined through a more refined attention extraction module [42], [43].

Despite the significant results made in the above studies, it is important to acknowledge that the modeling of the spatial-temporal dependence of traffic data remains unsatisfactory. Several studies, including STSGCN and STSGSA, which are respectively proposed by [16] and [17], have acknowledged the significance of spatial-temporal relationships in relation to local traffic patterns. However, their efficacy is constrained by the connectivity of the predefined adjacency matrix, and imputation tasks cannot account for long-term spatial-temporal context dependence.

III. Preliminaries

In this section, we first define the notations used in this paper and then formulate the traffic data imputation problem.

A. Notations and Definitions

Vectors are denoted by the boldface lowercase letter, e.g. $\boldsymbol{z} \in R^{I_1}$, matrices are denoted by the boldface uppercase letter, e.g. $\boldsymbol{Z} \in R^{I_1 \times I_2}$. Tensors are denoted in the calligraphic letters, a tensor of thrid-order is defined as $\mathcal{Z} \in R^{I_1 \times I_2 \times I_3}$, and an element of the tensor is denoted as z_{i_1,i_2,i_3}. A tensor can be unfolded into a matrix along different dimensions, for a third-order tensor $\mathcal{Z} \in R^{I_1 \times I_2 \times I_3}$, the operator for unfolding along the k-th dimension of the tensor is defined as $unfold_k(\mathcal{Z})$, which results in a matrix $\mathcal{Z}_{(k)} \in R^{I_k \times I_{k-1} I_{k+1}}$. The corresponding inverse operation is denoted as $fold_k(\mathcal{Z}_{(k)})$. For tensor $\mathcal{Z} \in R^{I_1 \times I_2 \times I_3}$, its tensor-SVD is denoted as $\mathcal{Z} = \mathcal{U} * \mathcal{S} * \mathcal{V}'$, where both $\mathcal{U} \in R^{I_1 \times I_1 \times I_3}$ and $\mathcal{V} \in R^{I_2 \times I_2 \times I_3}$ are orthogonal tensor and $\mathcal{S} \in R^{I_1 \times I_2 \times I_3}$ is the F-diagonal tensor. The singular value shrinkage operation of tensor $D_\tau(\mathcal{Z})$ is defined as $D_\tau(\mathcal{Z}) = \mathcal{U} * ifft_3(\max\{fft_3(\mathcal{S}) - \tau, 0\}) * \mathcal{V}'$, where fft_k denotes the Fourier transform along the k-th dimension and $ifft_k$ denotes the Fourier inverse transform.

B. Problem Formalization

The traffic road network including N road detectors can be modeled as a directed graph $G = \{V, E\}$, where V denotes the set of nodes and $e = (u, v) \in E$ is used to denote the connectivity between nodes. The data collected by the traffic detectors containing missing values for a total of D days can be represented as a tensor $\mathcal{X} \in R^{N \times T \times D}$, and the element $x_{n,t,d}$ denotes the traffic signal (speed, volume, or density) detected by the n-th road detector at the moment t of day d. A binary mask tensor $\mathcal{M} \in R^{N \times T \times D}$ is defined to indicate the location of missing traffic data, $m_{n,t,d} = 1$ when (n, t, d) belongs to the observed index set Ω; otherwise, $m_{n,t,d} = 0$ when $(n, t, d) \notin \Omega$.

The imputation task can be formulated as the problem of learning a function $\mathcal{F}(*)$ which uses the observed data and the global low-rank initial values to impute the missing data.

$$\widehat{\boldsymbol{y}}_{t_1}, \widehat{\boldsymbol{y}}_{t_2}, \cdots, \widehat{\boldsymbol{y}}_{t_P} = \mathcal{F}(\boldsymbol{y}_{t_1}, \boldsymbol{y}_{t_2}, \cdots, \boldsymbol{y}_{t_P}) \tag{1}$$

where, $\boldsymbol{y}_{t_p} = concat[\widehat{\boldsymbol{x}}_{t_p}, (\boldsymbol{x}_{t_p} \circ \boldsymbol{m}_{t_p})] \in R^P$. $\boldsymbol{x}_{t_p} \in R^P$ is the original observation value at t_p and $\widehat{\boldsymbol{x}}_{t_p} \in R^P$ is the low rank prior from LRTC. The symbol "$\circ$" denotes the Hadamard product and $\boldsymbol{m}_{t_p} \in R^P$ is the binary vector.

IV. Methodology

We propose a two-stage imputation method in this section to address the issue of missing traffic data. In the first stage, the LRTC technique is applied to learn the global low-rank structure of the data built on its low-rank properties. The global low-rank initial value is filled in the missing position as prior information. In the second step, we develop a STFEN model that aims to acquire a comprehensive understanding of the nonlinear spatiotemporal relationships, hence enhancing the accuracy of imputation.

A. The First Stage: LRTC Provides Global Low-Rank Prior Initial Values

To deal with the issue of tensor rank minimization, which has been proven as NP-hard. Reference [44] proposed a tensor completion model LRTC that minimizes the tensor nuclear norm, one defined based on the tensor-SVD decomposition as a foundation. We briefly introduce the LRTC model used as shown in Eq. (2), and the ADMM solving algorithm, as depicted in Algorithm. 1.

Algorithm 1 Algorithm of LRTC Method in the First Stage

Input: origin tensor $\mathcal{X} \in R^{N \times T \times D}$, binary mask tensor $\mathcal{M} \in R^{N \times T \times D}$, recovered tensor $\widehat{\mathcal{X}}$, penalty factor ρ, iteration K.
Initialize: $\widehat{\mathcal{X}} = \mathcal{X} \circ \mathcal{M}$, $\rho^0 = \rho$.
for $k = 1$ *to* K **do**
$\mathcal{K}^{k+1} = (\widehat{\mathcal{X}}^k + \mathcal{E} + \frac{1}{\rho^k}\mathcal{Y}^k)) \circ (1 - \mathcal{M}) + \mathcal{X} \circ \mathcal{M}$;
$\widehat{\mathcal{X}}^{k+1} = D_{\frac{1}{\rho^k}}(\mathcal{K}^{k+1} - \mathcal{E}^k - \frac{1}{\rho^k}\mathcal{Y}^k)$;
$\mathcal{E}^{k+1} = (\mathcal{K}^{k+1} - \widehat{\mathcal{X}}^{k+1} - \frac{1}{\rho^k}\mathcal{Y}^k) \circ \mathcal{M}$;
$\mathcal{Y}^{k+1} = \mathcal{Y}^k + \rho^k(\widehat{\mathcal{X}}^{k+1} + \mathcal{E}^{k+1} - \mathcal{K}^{k+1})$;
$\rho^{k+1} = \min(1.05 \times \rho^k, \rho_{max})$;
end
Output: recovered tensor $\widehat{\mathcal{X}}$.

$$\min_{\widehat{\mathcal{X}}, \mathcal{E}, \mathcal{K} \in R^{N \times T \times D}} \left\| \widehat{\mathcal{X}} \right\|_{Tensor\_NN} \quad s.t. \quad \begin{cases} \mathcal{K} \circ \mathcal{M} = \mathcal{X} \circ \mathcal{M} \\ \widehat{\mathcal{X}} + \mathcal{E} = \mathcal{K} \end{cases} \tag{2}$$

where $\mathcal{E} \in R^{N \times T \times D}$ denotes the random noise in the traffic data. In this objective function, the global low-rank structure of traffic data is captured by tensor nuclear norm. $Tensor\_NN$ is computed from the sum of the singular values of the tensor slices after Fourier transform and captures the major spatiotemporal patterns in the traffic data from the frequency domain perspective. In order to use the ADMM algorithm to solve, an intermediate variable $\mathcal{K} \in R^{N \times T \times D}$ is added to the constraints. In the first stage, we use the LRTC algorithm to perform preliminary imputation on the traffic data. The solution algorithm is as follows:

B. The Second Stage: Spatial-Temporal Feature Enhancement Network Captures Local Nonlinear Relationships

Fig. 1 shows the architecture of the proposed STFEN, which consists of three components: an input layer, L stacked spatial-temporal and temporal attention blocks (STTA), and an output Layer. To facilitate further spatiotemporal feature extraction, the observation records and recovery results Y from the previous stage are fed into the input layer, where a 1×1 convolutional neural network maps them to a high-dimensional space. Each STTA block includes a sparse spatiotemporal attention (SSA) and temporal attention (TA) respectively. The SSA can simultaneously and adaptively capture the information of temporal, spatial, and spatial-temporal dependence. Additionally, it offers the flexibility to expand the attention scope of spatial-temporal dependence by adjusting hyperparameters. To lessen the potential decrease in the proportion of temporal dependence caused by a limited number of temporal attention elements in SSA, TA is stacked to enhance temporal dependence. Finally, the output layer maps the result of STTA

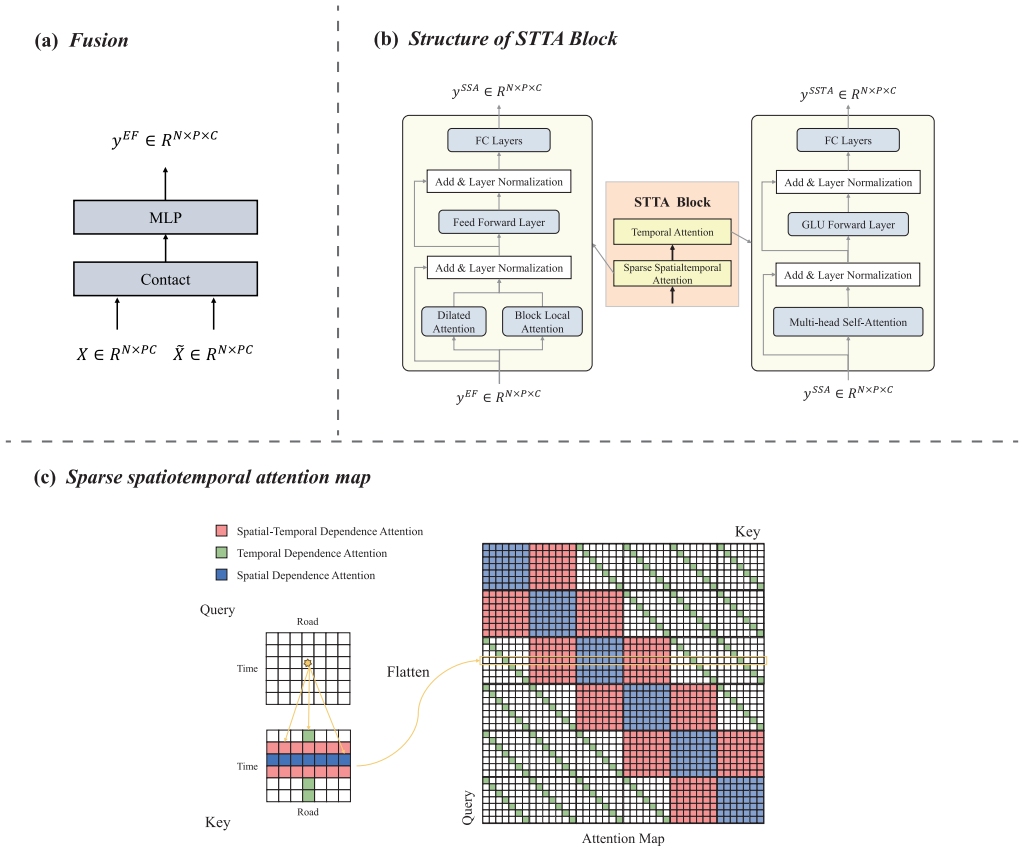


Fig. 2. Overall architecture of the spatial-temporal feature enhancement network framework. (a) ***Fusion***, used to fuse global low-rank priors and observation data; (b) ***Structure of STTA Block***, which consists of sparse spatiotemporal attention (left) and temporal attention (right). The sparse spatiotemporal attention mechanism consists of two attention modules: dilated attention and block local attention, which focus on the elements on the diagonal and block diagonal, respectively; (c) ***Sparse spatiotemporal attention map***, which serves as the core building blocks for SSA. Enable attention mechanisms to focus only on certain elements to establish effective spatiotemporal relationships and reduce redundant information. Blue represents spatial dependence, green represents temporal dependence, and red indicates spatial-temporal dependence.

to the data estimation result through a 1×1 convolutional neural network. Furthermore, spatial and temporal embedding enhances the model's ability to distinguish information from different time steps and nodes.

It is worth noting that the concept of local spatiotemporal attention proposed in our work is different from previous methods. In previous studies, local attention typically meant focusing on adjacent nodes or information from one or two immediate time steps. In contrast, the local attention mechanism discussed here is defined relative to the global low-rank modeling of the first stage, which refers to calculating attention weights by focusing on a fixed, short-term window. The low-rank model in the first stage extracts features from the entire dataset, while the design in the second stage captures spatiotemporal features from time series data spanning 12 time steps. Next, we will provide a detailed introduction to each module.

1) Spatial and Temporal Embedding: The aggregation of node features from different steps into a graph may result in the absence of temporal information. Meanwhile, the evolution of the traffic state is constrained by road topology and node information [38]. Hence, including temporal and spatial embedding enables the model to distinguish between information from different nodes and time steps. The formulation for temporal embedding and spatial embedding addition are as follows

$$\mathcal{Y}^{EF} = \mathcal{Y}^{data} + \mathcal{Y}^{se} + \mathcal{Y}^{te} \tag{3}$$

where, $\mathcal{Y}^{data} \in R^{N\times P\times C}$ is $\boldsymbol{Y}$ transformed from R to R^C by 1×1 convolutional network and $\boldsymbol{Y} = concat[\widehat{X}, X]$. The spatial embedding added is $\mathcal{Y}^{se} \in R^{N\times 1\times C}$ which guarantees that each node has its own unique feature information at different time steps. The temporal embedding added is $\mathcal{Y}^{te} \in R^{1\times P\times C}$ which ensures that values of all nodes at different time steps are different.

2) Sparse Spatiotemporal Attention for Spatial, Temporal, and Spatial-Temporal Dependence: Most previous studies employ the self-attention mechanism to capture nonlinear, complicated, and dynamic temporal dependence (shown in the green square in the Fig. 2) or spatial dependence (blue square) in traffic data. Though several existing models learn the spatial-temporal dependence (red square), their ability to establish long-term patterns is limited. We introduce sparse spatiotemporal attention(SSA), which can capture synchronized temporal, spatial, and spatial-temporal dependence.

A variable called correlation status $S = \{(v_m, t_j)|1 \le m \le N, 1 \le j \le P\}$, where S is the set of indices, is defined to determine which element need to be considered when establishing spatiotemporal relationships. In the case when $j = p$, the correlation status S denotes the set of elements

located at the index locations $\{(v_1, t_p), ((v_2, t_p), \cdots, (v_N, t_p)\}$. This implies that all road node information will be involved in the self-attention mechanism operation at time step t_p. The distinguishing factor between the sparse transformer and normal self-attention lies in utilizing the correlation status S to influence the operations on K and V. Consequently, the query elements are restricted to directing their attention solely toward the elements in the correlation status S. To provide a more succinct description of the formula, we designate $\mathcal{Y}^{EF}_{v_n,t_p,:}$ of the node v_n at time step t_p as $\xi_{n,p} \in R^C$. Then, the relationship between any element $\xi_{m,j}$ and $\xi_{n,p}$ in the correlation status S can be formally stated as follows:

$$\omega^h_{(n,p)(m,j)} = \frac{q_h k_h^T}{\sqrt{d_k}} \tag{4}$$

$$\beta^h_{(n,p)(m,j)} = \frac{exp(LeakyRelu(\omega^h_{(n,p)(m,j)}))}{\sum_{(m,j)\in S} exp(LeakyRelu(\omega^h_{(n,p)(m,j)}))} \tag{5}$$

$$where \quad q_h = \xi_{n,p} W_{q_h}, \quad k_h = \xi_{m,j} W_{k_h} \tag{6}$$

where $W_{q_h}, W_{k_h} \in R^{C\times d_k}$ are learnable parameters, representing the projection matrix of the h-th head. $\omega^h_{(n,p)(m,j)}$ denotes the relevance between the node v_n at time step t_p and the node v_m at time step t_j in h-th head. The attention score associated with this relevance is represented by $\beta^h_{(n,p)(m,j)}$. Subsequently, the features of all elements in the correlation status S can be aggregated based on the attention score magnitude. This process utilizing the multi-head sparse self-attention mechanism (MHSSA) can be represented as follows:

$$\mathcal{Y}^{SSA}_{v_n,t_p} = \left(\prod_{h=1}^{H}\left(\sum_{(m,j)\in S} \beta^h_{(n,p)(m,j)} \nabla h\right)\right) W_O \tag{7}$$

$$where \quad \nabla h = \xi_{m,j} W_{\nabla_h} \tag{8}$$

$$\mathcal{Y}^{SSA} = LayerNorm(\mathcal{Y}^{SSA} + \mathcal{Y}^{FE} W_1) \tag{9}$$

where $W_O \in R^{Hd_k\times d_k}, W_1 \in R^{C\times d_k}$, are learnable parameters. $\mathcal{Y}^{SSA} \in R^{N\times P\times d_k}$ is the feature after the information of all index elements in the correlation status S is aggregated. In order not to lose its own information, accelerate network convergence and improve generalization ability, we use residual connection and LayerNorm for normalization.

The key to sparse spatiotemporal attention is to determine the elements that need to be included in correlation status S when establishing nonlinear spatiotemporal relations. To improve the accuracy of traffic data imputation, it is critical that the correlation status S satisfies the following requirements: (1) correlation status S should cover elements that can represent three nonlinear relationships among traffic data: temporal, spatial, and spatiotemporal dependence. (2) correlation status S needs to pay attention to the information of timing context semantics. To modify the index range in the S correlation status, we employ two attention mechanisms: Block Local Attention and Dilated Attention.

Block Local Attention

The Block Local Attention mechanism has been specifically developed to retrieve the items inside a row effectively. This allows the SSA to effectively capture the dynamic and nonlinear spatial as well as spatial-temporal dependence. To establish a dynamic spatial dependence for the road section node v_n at time t_p, the Block Local Attention mechanism will retrieve information from all other road section nodes at the same time t_p. This set of indexes is denoted as $\{(v_1, t_p), (v_2, t_p), \cdots, (v_N, t_p)\}$, which aligns with the overall design of the normal spatial transformer. Regarding the establishment of spatial-temporal dependence, it is vital to consider the road node data from different temporal intervals. For instance, it is essential to retrieve the road node information at time t_{p+1}. The set of indexes is denoted as $\{(v_1, t_{p+1}), (v_2, t_{p+1}), \cdots, (v_N, t_{p+1})\}$. Similarly, we will ensure the preservation of temporal context consistency by considering the historical node information of all road sections up to the present moment, as well as the future node information of all road section nodes. The correlation status in Block Local Attention might be expressed as:

$$S = \{(v_m, t_j) | 1 \le m \le N, p - l \le j \le p + l\} \tag{10}$$

where l serves as a constraint that restricts the extent of focus on spatial-temporal dependence. When $l = 0$, the correlation status $S = \{(v_m, t_j) | 1 \le m \le N, j = p\}$ focuses solely on the spatial dependence and disregards the spatial-temporal dependence. When the value of l is greater than zero, the correlation status S allows SSA to acquire both the spatial dependence and the spatial-temporal dependence. These dependence are shown by the blue and red areas in Fig. 2.(c), respectively.

Dilated Attention

Dilated Attention is designed to focus on the temporal dependence in traffic data. In contrast to traffic prediction, which is limited to observing previous information at certain times, the imputation task requires consideration of the temporal context's information. For the link node v_n at time t_p, elements at index positions such as $\{(v_n, t_1), (v_n, t_2), \cdots, (v_n, t_P)\}$ are considered to be elements of (v_n, t_p) with a nonlinear temporal dependence. Hence, the correlation status of Dilated Attention can be formulated as:

$$S = \{(v_m, t_j) | m = n, 1 \le j \le P\} \tag{11}$$

The Fig. 2.(c) illustrates the range of temporal attention for variables S within the designated green region. The Attention Map exhibits a diagonal distribution in terms of the temporal dependence across all elements.

3) Temporal Attention for Temporal Dependence: In real-world traffic scenarios, missing data is random and might happen on any detector during any period. Three distinct categories of nonlinear traffic spatiotemporal relationships exhibit varying degrees of importance regarding missing data among different locations and periods. It is possible that the temporal dependence between the three dependence could be underestimated due to the limited focus on the P elements, which is considerably smaller in comparison to the N spatial dependence elements and the $2Nl$ spatial-temporal dependence elements. Consequently, following the computation of the sparse spatiotemporal attention, we add a temporal attention mechanism to compensate for the limitations of the sparse spatiotemporal attention and strengthen temporal dependence.

The entirety of the temporal attention mechanism has two distinct components: multi-head self-attention and the GLU

forwarding layer. The former procedure generates dynamic connections between traffic data at different periods by utilizing a self-attention mechanism. Including a gating mechanism in the latter approach enhances the capture of temporal correlations without increasing the parameters of the normal feed-forward layer. Applications in other fields have demonstrated the effectiveness of the gating mechanism.

Multi-head Self-Attention

The relationship between the t_p-th moment and the t_j-th moment at the h-th head for the node v_n is defined as $u^h_{(n,p)(n,j)}$ which is calculated as follows:

$$u^h_{(n,p)(n,j)} = \frac{q'_h (k'_h)^T}{\sqrt{d_k}} \tag{12}$$

$$\gamma^h_{(n,p)(n,j)} = \frac{exp(LeakyRelu(\gamma^h_{(n,p)(n,j)}))}{\sum_{(n,j)\in S} exp(LeakyRelu(\gamma^h_{(n,p)(n,j)}))} \tag{13}$$

$$where \quad q'_h = \xi_{n,p} W_{q'_h}, \quad k'_h = \xi_{n,j} W_{k'_h} \tag{14}$$

where $W_{q'_h}, W_{k'_h} \in R^{C\times d_k}$ are learnable parameters. $\gamma^h_{(n,p)(n,j)}$ is its corresponding attention score. It's important to emphasize that only the temporal dependence needs to be considered here. Accordingly, the correlation status is $S = \{(v_m, t_j)|m = n, 1 \le p \le P\}$. Then, data is transferred via a multi-headed sparse self-attention process:

$$\mathcal{Y}^{TA}_{v_n,t_p} = \left(\prod_{h=1}^{H} \left(\sum_{(n,j)\in S} \gamma^h_{(n,p)(n,j)} \nabla'_h \right) \right) W'_O \tag{15}$$

$$where \quad \nabla'_h = \xi_{n,j} W_{\nabla'_h} \tag{16}$$

where $W'_O \in R^{Hd_k\times d_k}, W'_\nabla \in R^{C\times d_k}$ are learnable parameters. $\mathcal{Y}^{TA} \in R^{N\times P\times d_k}$ is the feature after information aggregation in correlation status S.

GLU Forward Layer

Temporal attention involves substituting the feed-forward layers seen in conventional attention mechanisms with GLU forward layer. This substitution is aimed at improving the capacity to extract temporal relationships. The transformation of $\mathcal{Y}^{TA}$ will be carried out by two linear layers operating independently, with one of the layers activated by the sigmoid function.

$$\mathcal{Y}^{SSTA} = \left(Sigmoid\left(\mathcal{Y}^{TA} W_4\right) \bigotimes (\mathcal{Y}^{TA} W_5) \right) W_6 \tag{17}$$

$$\mathcal{Y}^{SSTA} = LayerNorm(\mathcal{Y}^{SSTA} + \mathcal{Y}^{TA} W_7) \tag{18}$$

$$\widehat{\boldsymbol{Y}} = \mathcal{Y}^{SSTA} W_8 + b_8 \tag{19}$$

where $W_4, W_5, W_6, W_7 \in R^{d_k\times d_k}$ are all trainable parameters. Finally, $\mathcal{Y}^{SSTA}$ will transfer from the high-dimensional space to the final data imputation result via a linear transformation of the data dimension at the FC layer.

4) Loss Function: We use a two-part loss function $\mathcal{L}$ to train our model, one that penalizes us for missing position information and another for reconstructing information. The proportion of loss experienced by these two components is equalized by the weight λ.

$$\mathcal{L} = (1-\lambda) \left\| \left((\widehat{\boldsymbol{Y}} - \boldsymbol{Y'}) \circ \boldsymbol{M} \right) \right\|_F^2 + \lambda \left\| \left((\widehat{\boldsymbol{Y}} - \boldsymbol{Y'}) \circ (1 - \boldsymbol{M}) \right) \right\|_F^2 \tag{20}$$

Algorithm 2 Algorithm of the Second Stage for STFEN Model Training

Input: Imputation results $\widehat{\mathcal{X}} \in R^{N\times T\times D}$ from the first stage, binary mask tensor $\mathcal{M} \in R^{N\times T\times D}$, size of the imputation windows P, batch size B, number of batches N_b, maximum iterations I.

Unfolding the tensor $\widehat{\mathcal{X}}, \mathcal{M}$ to matrix $\boldsymbol{Y}, \boldsymbol{M}$ and split samples for each batch according to sequence length P;

for $i = 1$ *to* I **do**

  for $j = 1$ *to* N_b **do**

    Map to high-dimensional space and add spatial, temporal embeddings: $\mathcal{Y}^{EF} \leftarrow \boldsymbol{Y}$ by Eq.(3);

    Clarify correlation status: $S_{spatial}$, $S_{temporal}$, $S_{spatiotemporal}$ by Eq.(10)(11);

    Learn all dynamic dependence with SSA: $\mathcal{Y}^{SSA} \leftarrow \mathcal{Y}^{EF}$ by Eq.(4)-(9);

    Enhance temporal dependence with TA: $\mathcal{Y}^{SSTA} \leftarrow \mathcal{Y}^{SSA}$ by Eq.(12)-(18);

    Map features into imputation results: $\widehat{\boldsymbol{Y}} \leftarrow \mathcal{Y}^{SSTA}$ by Eq.(19);

  end

  Compute the loss function by Eq.(20);

  Perform backward propagation to update the STFEN model parameters θ;

end

Output: Imputation results $\widehat{\boldsymbol{Y}}$ and STFEN model parameters θ.

Where $\widehat{\boldsymbol{Y}} \in R^{N\times P}$ is the ground truth, $\boldsymbol{Y'} \in R^{N\times P}$ is the final imputation results.

V. Experiments

In this section, we validate the proposed two-stage traffic data imputation method on two real-world datasets with different missing scenarios. The sections that follow introduce the chosen datasets and assessment measures before describing the baseline methodology and parameter settings. And, the experimental results are analyzed from five aspects: imputation performance, efficiency and computational complexity, ablation study, parameter sensitivity analysis, the importance of initial values, performance in newly developed sensors scenario and the interpretability of the sparse spatiotemporal attention map.

A. Dataset

To validate our model, we conducted comparative experiments on three real-world traffic datasets, namely PEMS04, PEMS08, and Seattle. The first two datasets were provided by the California Department of Transportation[1] And it includes information such as speed, traffic, and density. The latter dataset was collected from 323 detectors deployed on Seattle highways.[2] These systems summarize the data in 5-minute intervals, and Table I provides more detailed information for both datasets.

Two types of missing data scenarios are considered, random missing and non-random missing. For the random missing scenarios, we randomly mask the real data in the tensor according to the set missing rate. For non-random missing

[1]https://pems.dot.ca.gov/

[2]https://github.com/zhiyongc/Seattle-Loop-Data

TABLE I
THE DETAILS OF THE THREE DATASETS

Dataset	Time range	Node	Temporal resolution	Samples
PEMS04	1/1/2018-2/28/2018	307	5 min	16992
PEMS08	7/1/2016-8/31/2016	170	5 min	17856
Seattle	1/1/2015-3/31/2015	323	5 min	25920

Fig. 3. Illustration of random missing and non-random missing scenarios. Random missing involves random masking in the spatiotemporal matrix, while non-random missing involves continuous masking along the time dimension.

scenarios, we only consider the case of continuous missing traffic data along the time dimension, as shown in Fig. 3. We apply a masking procedure to the whole dataset, introducing missing values at predetermined rates (e.g., 10%, 30%, 50%, and 70%). This is done using both random and non-random mechanisms for missing data instances.

B. Baselines

We compare the proposed two-stage traffic data imputation method with state-of-the-art models, including low-rank tensor completion and deep learning methods.

1) **LRTC_TNN** [5]: a low-rank tensor completion method based on truncated nuclear norm, which obtains important low-rank patterns by controlling the truncation degree of tensor patterns of different dimensions.
2) **LATC** [45]: a low-rank tensor completion method that considers both the global and local structures of spatio-temporal traffic data. While capturing the global structure of traffic data with Truncated Nuclear Norm, an AR(p) model is applied to each time series to maintain local time consistency.
3) **GAIN** [46]: an adversarial generative network to generate correct distributions by distinguishing between observed and unknown data.
4) **MIWAE** [47]: a deep latent variable model based on an importance-weighted autoencoder that improves a potentially stringent lower bound on the log-likelihood of observations.
5) **AGCRN** [48]: an adaptive graph convolutional recurrent network that captures spatial and temporal correlations in traffic time series data, which can adaptively learn node-specific patterns and infer the inter-dependence among different time splot.
6) **GMAN** [38]: a transformer network based on the encoder-decoder framework, which adaptively captures the temporal and spatial relationships in traffic data through multi-head self-attention.
7) **CSDI** [49]: a conditional score-based diffusion model for estimation that can utilize correlations between observations.
8) **MBGAN** [50]: an adversarial generative network that can capture temporal dependence in multivariate time series data through bidirectional gating units, and learn relationships between different variables through multi-head self-attention mechanisms.
9) **MISIT** [51]: an autoencoder network that uses the transformer encoder to extract data features and trains in a self-supervised manner.
10) **SAITS** [52]: a self-attention method for multivariate temporal interpolation trained by joint optimization that explicitly captures temporal dependence and feature correlations.
11) **AGCRN (TNN_based)**: a hybrid two-stage method, in which LRTC_TNN is used to calculate the initial value of missing positions in the first stage, and AGCRN is used for estimation in the second stage.

C. Experimental Setting

1) Data Preprocessing: In the first stage, we organize the observed traffic data into a three-dimensional tensor and maintain the original data scale without any normalization operation. The imputation result processed by the LRTC model is still a three-dimensional tensor, so the tensor needs to be expanded into a two-dimensional node-time matrix before being input to the second stage. The second stage of STFEN is to mine local spatiotemporal relationships, so we divide the node-time matrix into training set, test set and validation set in a ratio of 7:2:1 according to the time window length of 12. Moreover, the data input into STFEN needs to be normalized according to the normalized transformation: $x' = (x - mean(x))/std(x)$.

2) Parameters Setting: In the first stage of the LRTC method, initial optimization step $\rho_0 = 10^{-5}$, and ρ is updated with $\rho = min\{1.05 \times \rho, \rho_{max}\}$ in each iteration, and the maximum iterations is 100. The STFEN in the second stage is implemented based on the PyTorch framework, and all experiments are conducted on an NVIDIA GeForce RTX 4090 GPU. The model is trained using the Adam optimizer and the batch gradient descent algorithm, with a batch size of 2. An early stopping strategy is adopted to avoid overfitting the model and the learning rate is fixed being 0.001. The main hyperparameter settings of STFEN in the experiment are as

TABLE II
PERFORMANCE COMPARISONS OF ALL METHODS FOR TRAFFIC DATA IMPUTATION TASK UNDER RANDOM MISSING

Dataset	Method	10% Random Missing			30% Random Missing			50% Random Missing			70% Random Missing		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS04	LRTC_TNN	0.970	1.653	1.852 %	1.145	2.045	2.247 %	1.413	2.648	2.887 %	1.751	3.321	3.654 %
	LATC	1.014	1.801	2.022 %	1.177	2.176	2.415 %	1.356	2.607	2.893 %	1.599	3.143	3.462 %
	GAIN	2.586	4.652	4.955 %	2.614	4.792	4.711 %	3.873	6.853	5.038 %	5.823	7.607	6.944 %
	MIWAE	2.215	5.401	5.059 %	2.260	5.430	5.165 %	2.297	5.394	5.211 %	2.372	5.520	5.366 %
	AGCRN	0.979	1.722	1.844 %	1.093	2.108	1.958 %	1.112	2.365	2.156 %	1.353	2.944	2.728 %
	GMAN	0.997	1.895	1.935 %	1.099	2.134	2.090 %	1.119	2.404	2.153 %	1.362	3.079	2.824 %
	CSDI	**0.831**	1.583	1.686 %	1.009	1.815	1.863 %	1.489	2.200	2.505 %	1.965	2.780	3.086 %
	MBGAN	1.442	2.584	3.276 %	1.996	2.784	3.171 %	1.991	3.673	4.091 %	1.995	3.605	4.042 %
	MISIT	1.285	2.023	2.444 %	1.342	2.240	2.729 %	1.494	2.571	3.082 %	1.911	3.399	4.188 %
	SAITS	0.886	1.562	1.694 %	1.058	1.868	2.092 %	1.237	2.264	2.567 %	1.590	3.004	3.467 %
	AGCRN (TNN_based)	0.880	1.594	**1.660** %	1.007	1.882	1.892 %	1.105	2.299	2.202 %	1.220	2.396	2.656 %
	Ours	0.930	**1.534**	1.711 %	**0.936**	**1.561**	**1.760** %	**1.026**	**1.805**	**1.938** %	**1.219**	**2.268**	**2.450** %
PEMS08	LRTC_TNN	0.744	1.360	1.387 %	0.861	1.653	1.644 %	1.028	2.078	2.037 %	1.267	2.629	2.571 %
	LATC	0.751	1.439	1.428 %	0.862	1.62	1.670 %	1.010	2.054	2.018 %	1.216	2.539	2.497 %
	GAIN	2.083	4.244	4.662 %	2.194	4.573	4.766 %	2.188	4.584	4.992 %	3.439	6.163	12.119 %
	MIWAE	1.849	4.781	4.108 %	1.961	4.936	4.287 %	2.024	5.103	4.462 %	2.170	5.244	4.682 %
	AGCRN	0.734	1.414	1.310 %	0.770	1.482	1.378 %	0.907	1.682	1.580 %	1.199	2.245	1.907 %
	GMAN	0.912	1.367	1.725 %	0.886	2.016	1.681 %	0.950	2.250	1.900 %	1.173	2.789	2.497 %
	CSDI	0.778	1.398	1.434 %	0.920	1.526	1.577 %	1.314	1.990	2.295 %	2.342	3.219	3.885 %
	MBGAN	1.138	1.925	2.391 %	1.250	2.266	2.482 %	1.374	2.663	3.021 %	1.465	3.557	3.252 %
	MISIT	1.161	1.950	2.176 %	1.215	2.152	2.343 %	1.373	2.502	2.729 %	1.599	3.034	3.266 %
	SAITS	0.808	1.502	1.530 %	0.959	1.833	1.867 %	1.149	2.277	2.297 %	1.453	2.918	2.975 %
	AGCRN (TNN_based)	0.811	1.415	1.379 %	0.816	1.479	1.403 %	0.889	1.674	1.555 %	1.109	2.181	1.935 %
	Ours	**0.722**	**1.280**	**1.296** %	**0.766**	**1.393**	**1.344** %	**0.829**	**1.566**	**1.520** %	**1.015**	**1.976**	**1.810** %
Seattle	LRTC_TNN	2.032	3.094	4.874 %	2.149	3.282	5.226 %	2.316	3.552	5.742 %	2.589	4.015	6.638 %
	LATC	2.091	3.168	5.031 %	2.257	3.436	5.559 %	2.532	3.856	6.377 %	3.060	4.666	8.029 %
	GAIN	4.303	7.532	13.321 %	5.225	8.874	13.953 %	6.123	9.072	17.389 %	7.754	11.347	9.023 %
	MIWAE	3.482	6.285	9.734 %	3.534	6.320	9.757 %	3.616	6.506	10.113 %	3.717	6.688	10.409 %
	AGCRN	2.470	3.875	6.131 %	2.575	3.968	6.973 %	2.587	4.162	6.256 %	2.884	4.945	7.830 %
	GMAN	2.278	3.321	5.602 %	2.306	3.475	5.632 %	2.459	3.783	5.816 %	2.620	4.142	6.198 %
	CSDI	2.686	4.142	6.140 %	2.918	4.584	7.818 %	3.180	4.699	8.152 %	3.381	4.943	9.150 %
	MBGAN	2.773	4.162	6.247 %	3.001	4.600	8.109 %	3.496	5.137	9.217 %	3.994	5.860	13.114 %
	MISIT	2.342	3.565	5.804 %	2.447	3.760	6.081 %	2.587	4.039	6.659 %	2.840	4.559	7.782 %
	SAITS	2.371	3.573	5.682 %	2.384	3.650	5.756 %	2.577	3.992	6.465 %	2.858	4.522	7.516 %
	AGCRN (TNN_based)	2.105	3.233	5.697 %	2.292	3.478	5.968 %	2.468	3.705	6.349 %	2.633	4.029	6.790 %
	Ours	**1.937**	**2.914**	**4.396** %	**1.946**	**2.925**	**4.373** %	**2.096**	**3.150**	**4.797** %	**2.271**	**3.449**	**5.307** %

follows. The number of STTA is 1, the hidden dimension $C = 64$, and the spatial-temporal attention range $l = 1$. In the loss function, λ is set to 0.7. The negative slope of LeakyReLU is set to 0.2. For the deep learning models in the baseline methods, we adopt the original settings from their papers.

3) Evaluation Metrics: The accuracy of traffic data imputation results would be measured by three commonly used evaluation metrics: $RMSE$, MAE, and $MAPE$:

$$RMSE(Y, \widehat{Y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{21}$$

$$MAE(Y, \widehat{Y}) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right| \tag{22}$$

$$MAPE(Y, \widehat{Y}) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right| \times 100\% \tag{23}$$

where $\widehat{y}_i \in R^N$ is the final imputation results, $y'_i \in R^N$ is the ground truth.

D. Result

1) Imputation Performance: In this subsection, we compare the performance of the proposed two-stage traffic data imputation method with other baseline models on three datasets. Table II and Table III show the performance of different models in the two cases of random missing and non-random missing respectively. Overall, the imputation difficulty of the non-random missing scenarios is greater than that of the random missing scenarios. Tensor completion techniques, such as LRTC_TNN and LATC, exhibit strong performance in both random and non-random missing scenarios, and may attain sub-optimal accuracy in some instances. These findings indicate that tensor completion techniques that only rely on the low-rank characteristics of traffic data have strong robustness. In many circumstances, LATC with autoregression as a linear constraint has worse performance compared to LRTC_TNN.

The performance of various deep learning techniques varies greatly in different missing data scenarios. For example, SAITS achieved excellent results under random missing conditions in both the PEMS04 and PEMS08 datasets, but its accuracy significantly decreased under non random missing conditions. On the other hand, AGCRN exhibits partially optimal but inconsistent performance in the random missing scenario of PEMS08, while performing poorly in other environments.

Models such as GAIN, MIWAE, and MBGAN that use adversarial generative training paradigms have encountered difficulties in interpolating large-scale traffic data. The inherent uncertainty and randomness of GAN based models hinder their

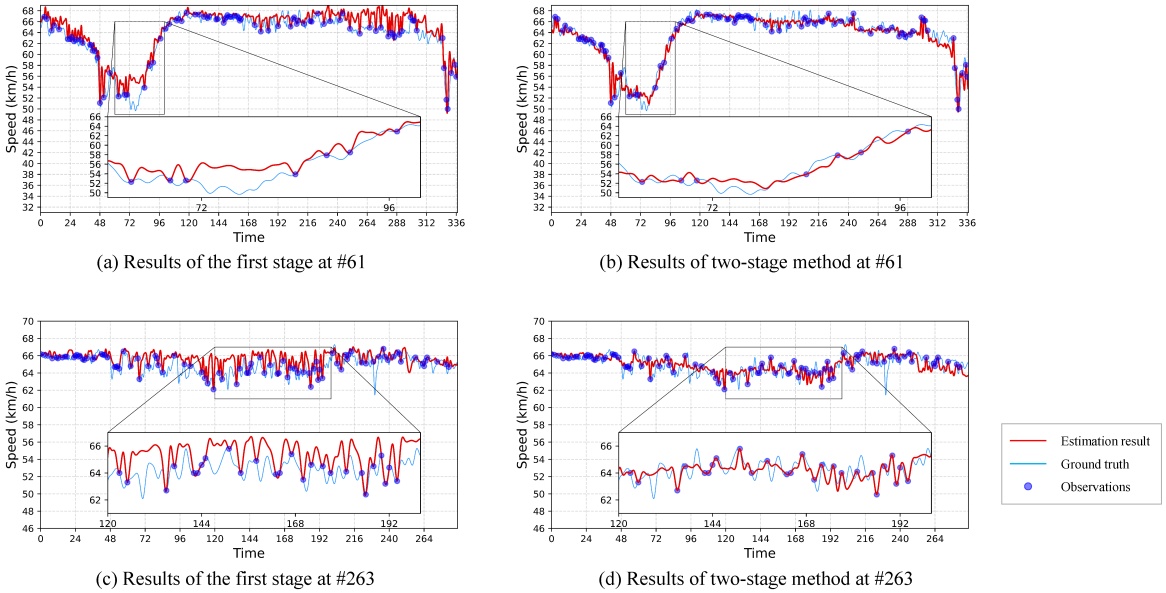
TABLE III
PERFORMANCE COMPARISONS OF ALL METHODS FOR TRAFFIC DATA IMPUTATION TASK UNDER NON-RANDOM MISSING

Dataset	Method	10% Non-random Missing			30% Non-random Missing			50% Non-random Missing			70% Non-random Missing		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS04	LRTC_TNN	2.078	4.255	4.663 %	2.126	4.318	4.791 %	2.194	4.428	4.903 %	2.312	4.639	5.120 %
	LATC	2.080	4.183	4.452 %	2.178	4.389	4.845 %	2.354	4.711	5.224 %	2.614	5.133	5.835 %
	GAIN	3.715	6.161	7.241 %	4.503	6.960	8.378 %	4.529	7.300	8.437 %	7.116	10.762	13.848 %
	MIWAE	2.519	5.925	5.916 %	2.592	5.938	5.818 %	2.649	5.948	5.992 %	2.763	6.114	6.629 %
	AGCRN	2.135	4.197	4.625 %	2.390	4.375	5.124 %	2.412	4.535	5.343 %	2.897	4.816	5.761 %
	GMAN	3.286	5.666	6.561 %	3.321	6.324	7.686 %	3.208	6.677	8.272 %	3.897	7.255	9.812 %
	CSDI	1.957	4.049	**3.933** %	2.187	4.254	4.857 %	2.276	4.466	5.152 %	2.815	5.139	6.184 %
	MBGAN	2.167	4.839	5.416 %	2.185	4.611	5.017 %	2.410	4.659	5.446 %	2.782	4.900	6.331 %
	MISIT	2.837	5.093	6.235 %	3.072	5.461	6.940 %	3.180	5.507	7.153 %	3.103	5.629	7.175 %
	SAITS	2.480	5.347	5.800 %	3.381	6.644	8.254 %	3.747	7.104	9.247 %	3.964	7.267	9.555 %
	AGCRN (TNN_based)	2.188	4.061	4.492 %	2.238	4.162	4.971 %	2.298	4.305	5.002 %	2.477	4.608	5.599 %
	Ours	**1.858**	**3.791**	4.173 %	**1.955**	**3.919**	**4.283** %	**2.050**	**4.121**	**4.690** %	**2.109**	**4.243**	**4.836** %
PEMS08	LRTC_TNN	1.665	3.705	3.572 %	1.820	3.957	3.895 %	1.951	4.233	4.213 %	2.020	4.326	4.201 %
	LATC	1.752	3.868	3.647 %	1.879	4.134	3.965 %	2.046	4.477	4.438 %	2.261	4.692	4.820 %
	GAIN	2.198	4.431	4.131 %	3.260	5.533	5.046 %	3.721	6.237	5.675 %	4.342	7.019	9.404 %
	MIWAE	2.061	5.173	4.298 %	2.025	5.112	4.423 %	2.141	5.331	4.968 %	2.285	5.605	4.990 %
	AGCRN	2.065	4.266	4.723 %	2.256	4.389	4.931 %	2.811	5.155	5.160 %	2.899	5.206	5.290 %
	GMAN	1.757	3.340	4.971 %	2.531	3.783	6.013 %	3.300	4.184	7.166 %	3.368	4.635	8.156 %
	CSDI	**1.353**	2.996	4.168 %	1.503	3.914	4.432 %	2.468	5.176	7.794 %	3.043	6.299	8.742 %
	MBGAN	1.875	3.711	4.296 %	1.931	4.341	4.138 %	1.615	3.930	3.625 %	2.591	4.884	5.795 %
	MISIT	2.628	5.168	5.395 %	2.961	5.646	6.176 %	3.241	5.892	6.858 %	3.329	6.113	7.194 %
	SAITS	1.934	4.244	4.134 %	2.658	5.436	5.710 %	3.006	5.884	6.581 %	3.182	6.149	7.004 %
	AGCRN (TNN_based)	2.113	4.305	4.881 %	2.208	4.088	4.156 %	2.166	4.274	4.305 %	2.470	4.839	4.516 %
	Ours	1.453	**2.953**	**2.844** %	**1.495**	**3.083**	**3.096** %	**1.604**	**3.357**	**3.362** %	**1.789**	**3.575**	**3.650** %
Seattle	LRTC_TNN	2.382	3.785	6.094 %	2.548	4.084	6.668 %	2.834	4.581	7.598 %	3.292	5.358	9.136 %
	LATC	2.537	3.917	7.164 %	2.776	4.323	7.888 %	3.034	4.786	8.871 %	3.501	5.627	10.608 %
	GAIN	4.135	5.328	11.921 %	4.895	6.050	15.951 %	5.475	6.968	19.155 %	7.604	10.694	21.880 %
	MIWAE	3.568	6.367	9.918 %	3.602	6.648	10.154 %	3.733	6.850	10.484 %	3.868	6.941	11.043 %
	AGCRN	3.987	6.093	11.221 %	4.150	7.065	11.676 %	4.167	7.885	14.345 %	4.238	8.081	14.995 %
	GMAN	2.464	3.984	6.424 %	2.952	4.585	7.019 %	3.305	5.170	9.074 %	4.150	5.539	11.849 %
	CSDI	2.653	4.224	6.810 %	2.939	4.820	7.486 %	3.559	5.358	11.749 %	3.996	5.650	12.679 %
	MBGAN	3.625	5.145	9.819 %	3.967	5.580	10.503 %	4.432	6.147	11.342 %	4.948	6.609	14.266 %
	MISIT	2.674	4.113	7.181 %	2.863	.443	7.971 %	3.033	4.764	8.636 %	3.487	5.543	10.538 %
	SAITS	2.480	3.745	6.168 %	2.596	3.975	6.563 %	2.795	4.333	7.264 %	3.092	4.900	8.497 %
	AGCRN (TNN_based)	2.342	3.809	6.912 %	2.778	4.315	7.574 %	3.071	4.925	8.294 %	3.438	5.981	10.643 %
	Ours	**2.254**	**3.494**	**5.348** %	**2.297**	**3.631**	**5.634** %	**2.428**	**3.897**	**5.974** %	**2.736**	**4.484**	**7.116** %

ability to provide accurate interpolation in specific spatiotemporal contexts. Although MBGAN achieves high accuracy at high loss rates, its training is hindered by instability and presents significant limitations. In contrast, the unsupervised CSDI diffusion model performs strongly under low missing rate conditions, which may be due to its excellent distribution fitting ability and the ability of its attention network to effectively capture spatiotemporal features. Meanwhile, AGCRN, GMAN, and SAITS - all capable of non-linear spatiotemporal feature mining - achieved optimal or near optimal accuracy in low random missing settings, highlighting the importance of modeling non-linear spatiotemporal interactions in traffic data imputation. Although MISIT can capture the interrelationships between variables in multivariate time series, it has limitations in extracting meaningful spatiotemporal patterns from traffic data. In addition, although LRTC_TNN provides initial estimates of missing values, in most cases, AGCNN (based on TNN) cannot achieve optimal or suboptimal performance. This is because directly using these initial estimates not only introduces low rank information, but also significant noise, which can affect the performance of the model when learning local spatiotemporal patterns.

By integrating tensor completion with deep learning, the proposed two-stage traffic data imputation method is able to represent spatiotemporal nonlinear relationships in the data using a sparse spatiotemporal attention mechanism and to mine global low-rank features. In the random missing scenarios, the two-stage traffic data imputation method not only achieves better performance than the baseline models but also achieves significant improvements and our model also achieves optimal and sub-optimal performance in most non-random missing scenarios.

Fig. 4 show real cases of data imputation of two road sections with different missing rates in the PEMS04 dataset. Estimate results given by LRTC in the first stage based on the global low rank of traffic data can be close to some real situations, but in many cases there are still deviations caused by ignoring local nonlinear spatiotemporal relationships. The STFEN in the second stage makes up for this very well. It further constrains the valuation based on the estimation results given in the first stage, thereby alleviating possible local discontinuity dilemmas. The two-stage traffic data imputation framework effectively handles scenarios with high missing rates. Table VI presents the performance comparison between our model and the baseline models under three high missing rate scenarios: 85%, 90%, and 95%. The two-stage traffic data imputation method achieves leading imputation performance. Furthermore, Fig. 5 illustrates the imputation results under both random and non-random missing scenarios with an 80%



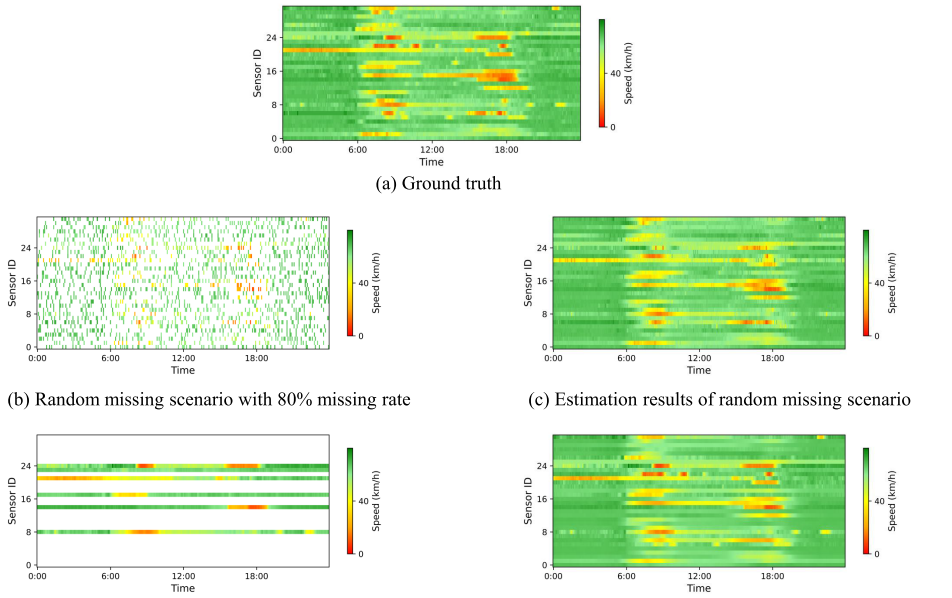
(a) Results of the first stage at #61

(b) Results of two-stage method at #61

(c) Results of the first stage at #263

(d) Results of two-stage method at #263

Fig. 4. Visualization of the imputation results of the two-stage traffic data imputation method for sensor 263 at 40% and sensor 61 at 50% random missing. The red line indicates the estimated results, the blue line represents the ground truth, and the blue circles denote the observations.



(a) Ground truth

(b) Random missing scenario with 80% missing rate

(c) Estimation results of random missing scenario

(d) Non-random missing scenario with 80% missing rate

(e) Estimation results of non-random missing scenario

Fig. 5. Visualizations of ground truth, missing scenario, and estimation results under different missing patterns (80% missing rate, and white represents missing values).

TABLE IV
ABLATION STUDY

	30% Random Missing			50% Non-random Missing		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
STFEN	**1.561**	**0.936**	**1.760** %	**4.121**	**2.050**	**4.690** %
w/o Dilated Attention	1.570	0.941	1.763 %	4.181	2.093	4.799 %
w/o Block Local Attention	1.865	1.117	2.073 %	4.387	2.138	4.839 %
w/o Sparse Spatiotemporal Attention	2.056	1.213	2.112 %	4.403	2.178	4.977 %
w/o Temporal Attention	1.675	1.081	1.846 %	4.126	2.094	4.710 %

high missing rate. The combination of tensor completion and deep learning enables our method to achieve excellent performance, improving the estimation accuracy compared to using either method alone.

TABLE V
STFEN PERFORMANCE UNDER DIFFERENT WINDOW LENGTH

Window length (P)	PEMS04		PEMS08	
	RMSE	Memory (MB)	RMSE	Memory (MB)
$P = 8$	3.991	3989	3.168	2561
$P = 12$	3.919	5131	3.083	3839
$P = 16$	4.215	6861	3.256	4931
$P = 20$	4.376	8651	3.241	6168

TABLE VI
PERFORMANCE COMPARISONS UNDER EXTREME MISSING RATES

Missing rate	Method	Metrics		
		MAE	RMSE	MAPE
85%	AGCRN	1.566	3.206	3.665 %
	CSDI	2.189	4.224	4.910 %
	Ours	**1.375**	**2.899**	**2.801** %
90%	AGCRN	1.993	3.832	3.673 %
	CSDI	2.347	4.434	5.215 %
	Ours	**1.790**	**3.210**	**3.546** %
95%	AGCRN	2.382	5.397	4.459 %
	CSDI	2.661	4.880	6.020 %
	Ours	**2.011**	**3.627**	**3.507** %

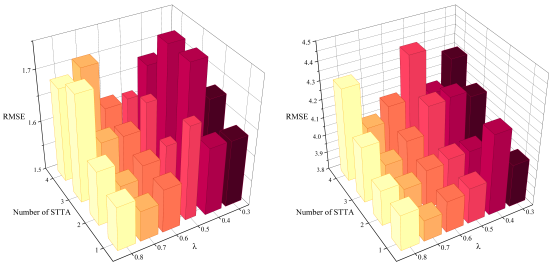


Fig. 6. Sensitivity experiment results of parameters the number of STTA and loss balance weight λ.

2) Efficiency and Computational Complexity: The sparsification operation selectively computes the similarity between components at certain places, hence reducing the computational load, as opposed to completely hiding them out as in the vanilla transformer. Given a spatiotemporal tensor data of size $N \times P \times C$, where N and P represent road segments and time periods respectively, and C represents the data dimension. When the vanilla transformer capture these three dependence in traffic spatiotemporal data at the same time, then it needs to pay attention to the information of all periods on any segment, which will lead to the computational complexity of $O(N^2P^2C)$. In fact, a lot of information is redundant, and taking it all into account would be a waste of computing resources [53]. The spatial-temporal dependence of traffic is usually short-term, so the proposed block local attention only focuses on elements within the scope l, and its computational complexity is $O(N^2(2l+1)^2C)$. Dalited attention establishes temporal dependence, so the computational complexity is $O(P^2C)$. Considering the number of road segments N is generally much larger than time periods P, and the overall computational complexity of the STFEN algorithm may be expressed as $O(N^2(2l+1)^2C + P^2C) \sim O(N^2(2l+1)^2C)$. From a floating point operations standpoint, each 5-minute interval of expanding the spatial-temporal attention scope will result in an increase in calculations by $2PC*(N^2-1)$ flops. Despite the sparsification operation we applied along the temporal dimension, the approach still faces complexity $O(N^2)$ when the road network is large. The training and inference times of the proposed model are shown in Fig 7.

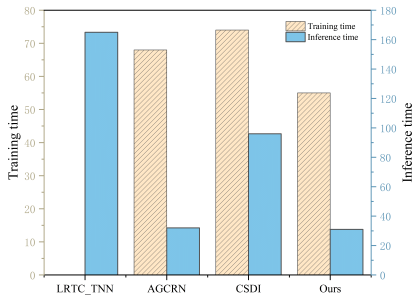


Fig. 7. Computation costs of the proposed framework on the Seattle dataset.

Furthermore, we present the computational resource consumption of the two-stage data imputation framework during actual operation as shown in TABLE V. Its runtime overhead is closely related to the time window length, with the required GPU memory increasing rapidly as the window length expands. A trade-off between computational accuracy and efficiency emerges at $P = 12$, corresponding to a local window length of one hour. At this configuration, the framework achieves optimal data imputation performance while maintaining relatively low memory consumption.

3) Ablation Study: Within the PEMS04 dataset, which has 30% random missing and 60% non-random missing scenarios, we conduct a series of ablation experiments to assess the efficacy of each primary module of STFEN. These modules include the sparse spatiotemporal attention module and the temporal attention module. More precisely, we use the LRTC method's imputation outputs from the first stage as the input for the second stage. Additionally, we establish four variations derived from STFEN:

- w/o Dilated Attention: remove the dilated attention used to model temporal dependence in the sparse spatiotemporal attention module.
- w/o Block Local Attention: remove the block local attention used to model spatial-temporal and spatial dependences in the sparse spatiotemporal attention module.
- w/o Sparse Spatiotemporal Attention: remove the entire sparse spatiotemporal attention module.
- w/o Temporal Attention: remove the temporal attention used to enhance temporal dependence.

Presented in Table IV is the experimental findings which obtained from the ablation study. This demonstrates that these fundamental modules have a major contribution to the overall performance enhancement of STFEN, as evidenced by the fact that the imputation accuracy of the four variations declined in comparison with STFEN in two random missing scenes.

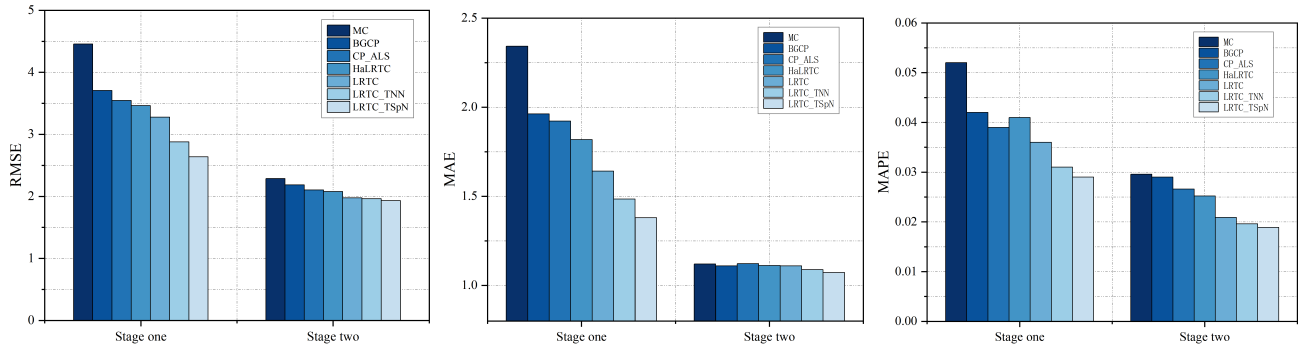


Fig. 8. Compare the impact of different first stage methods on the final repair accuracy. The methods we have selected cover seven models, including matrix decomposition, tensor decomposition, and tensor completion.

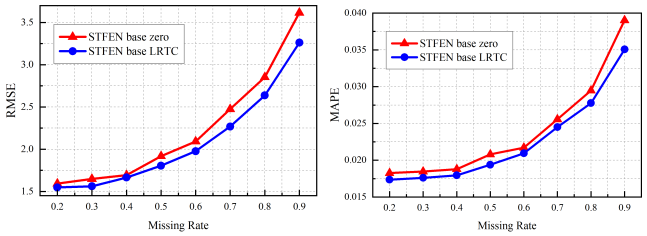


Fig. 9. Comparison of LRTC-initialized vs zero-initialized imputation performance.

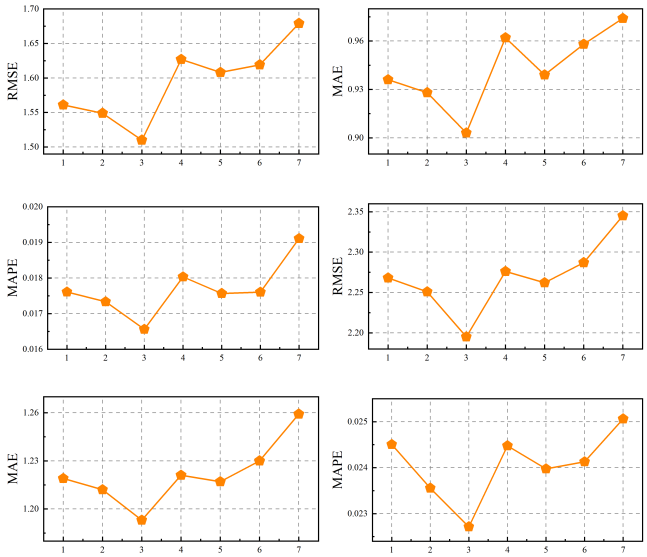


Fig. 10. Imputation accuracy at 30% and 70% random missing scenarios for different spatial-temporal attention scope l.

After removing sparse spatiotemporal attention, the overall error increases dramatically, which indicates that it has a considerable impact on model performance. This is due to the fact that it is able to completely evaluate the three spatiotemporal relationships that are present in traffic data. Secondly, the ablation study of block local attention demonstrates that dynamic spatial-temporal and spatial dependences are more significant for traffic data imputation tasks. This has the potential to be connected to the regional locally of traffic status.

4) Parameter Sensitivity Analysis: This section conducts a sensitivity analysis of key parameters in the STFEN model to determine the optimal parameter configuration for its best performance.

Fig. 6 demonstrates how the model's performance varies with changes in the balancing parameter λ within the loss function and the number of STTA layers. Increasing the number of layers does not enhance the model's imputation capability. Optimal imputation performance is achieved only when the number of STTA layers is set to 1 and parameter λ is 0.7.

The findings of the hyperparameter sensitivity analysis that carried out in the spatial-temporal attention scope are shown in Fig. 10. As the spatial-temporal attention scope l increases from 1 to 7, the imputation error of the approach exhibits a non-linear trend, initially decreasing and thereafter increasing. The interpolation accuracy reaches its maximum when the spatial-temporal attention scope $l = 3$. This finding provides more evidence that in the process of interpolating traffic data, it is advisable to concentrate on the traffic information within a 15-minute timeframe when establishing nonlinear spatial-temporal relationships. When the spatial-temporal attention scope is smaller than 10 minutes, the limited number of elements available for analysis hinders the improvement of traffic data imputation accuracy. Nevertheless, when the spatial-temporal attention scope is excessively broad, the abundance of information becomes redundant, hence hindering the construction of efficient nonlinear spatial-temporal interactions.

5) Importance of Initial Value: To assess the necessity for the first stage of the two-stage traffic data estimation approach and examine the impact of employing various tensor completion models in the first stage on the final estimation results. A variety of experiments are established in the scenario of the missing random situation.

We implemented seven different experimental approaches, picking different global models in the first stage: (a) MC, matrix completion model [54]; (b) BGCP: Bayesian Gaussian tensor decomposition [21]; (c) CP_ALS: CP decomposition based on alternating iterative squares [55]; (d) HaLRTC: tensor completion model minimizing the rank of each mode [56]; (e) LRTC: tensor completion model minimizing the rank of the tensor; (f) LRTC_TNN: truncated nuclear norm based tensor completion model [5]; (g) LRTC_TSpN: Truncated tensor Schatten-norm based tensor completion model [6].

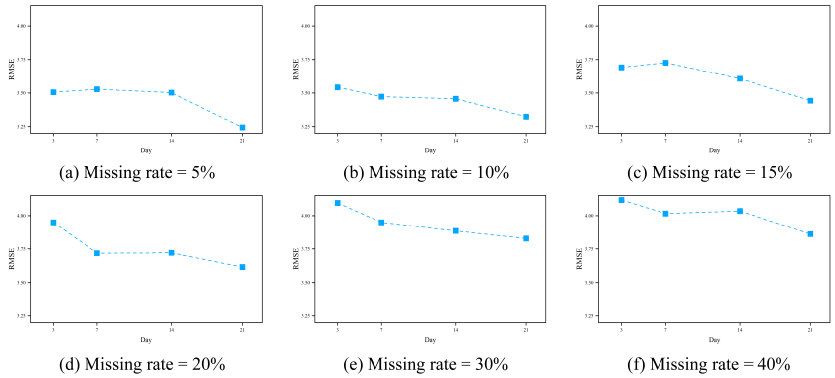


Fig. 11. Imputation accuracy for historical data on road segments with newly installed sensors.

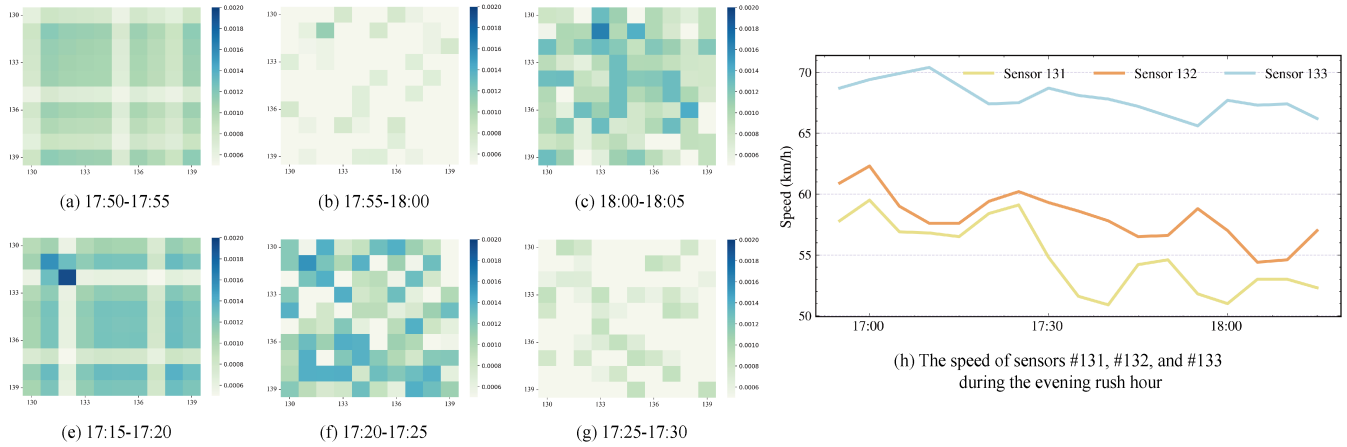


Fig. 12. The attention matrix obtained from the sparse spatiotemporal attention mechanism.

Figure 8 presents the comparative performance of seven constructed models under 60% random missing rate conditions. The experimental results demonstrate that the two-stage traffic data imputation framework is influenced by the precision of initial-stage estimation. The results reveal a positive correlation where higher first stage recovery accuracy leads to improved final imputation accuracy through the second stage. After a more comprehensive evaluation for more random missing cases, as shown in Fig. 9. We find that using tensor completion estimation to replace missing elements produces better results than a solution that assigns zero values without any further processing. These findings suggest that our proposed two-stage structure has beneficial effects and guarantees improved performance. The accuracy of the first estimate directly affects the size of the inaccuracy of the final estimate obtained through the two-stage process.

6) Performance in the Newly Developed Sensors Scenario: In this subsection, we focus on the following problem: under the premise that the road network topology remains unchanged, how can we utilize the limited data collected by newly installed sensors on previously unmonitored road segments to impute historical data from periods before these sensors were deployed.

In the experimental setup, we assume that a proportion p (which can also be interpreted as the missing rate) of road segments are newly equipped with sensors in Seattle dataset. For these newly installed sensors, we consider four different deployment durations: 3, 7, 14, and 21 days. The experimental results in Fig 11 demonstrate that the proposed two-stage traffic data imputation framework effectively addresses the scenario of newly installed sensors. Even under the challenging condition where only 3 days of data are available from 40% of the road segments, the framework achieves low error in reconstructing the historical data. Furthermore, longer deployment durations, which provide more collected information, further improve the accuracy of historical data estimation.

7) Interpretability of Sparse Spatiotemporal Attention Map: We conduct a focused analysis on traffic patterns during the afternoon peak period (17:00-18:30) within a local road network monitored by three adjacent detectors (Sensor 131, 132, and 133). Through visualization of attention maps, we demonstrate that the sparse attention mechanism in the SSA architecture effectively captures meaningful traffic dynamics. To enhance interpretability, an upper threshold of 0.002 is applied to attention scores, enabling clearer identification of critical correlations among the numerous elements in attention map. The experimental results demonstrate that the SSA mechanism effectively identifies two critical spatiotemporal patterns in traffic data through its sparse operations.

(1) **Localized synchronization in non-congested periods**. Fig. 12.(e)-(g) present attention heatmaps for Sensor 131 during 17:15–17:20 and its correlations with three consecutive time windows (17:15–17:20, 17:20–17:25, and 17:25–17:30). The analysis reveals strong synchronization between Sensor 131 and its downstream neighbor Sensor 132 during 17:15–17:20, as evidenced by high attention scores, indicating tightly coupled traffic state variations. In contrast, significantly reduced attention scores are observed at 17:25–17:30 when traffic states diverged between the two sensors. This phenomenon highlights the localized spatial coupling of adjacent nodes during non-congested periods.

(2) **Long-range dependence capture during peak congestion**. Under congested conditions, SSA successfully captures long-range spatiotemporal dependence through its cross-temporal propagation mechanism. As shown in Fig. 12.(a)-(c), strong correlations emerge between Sensor 131 (17:50–17:55), Sensor 132 (17:55–18:00), and Sensor 133 (18:00–18:05). Fig. 12.(h) illustrates that the sparse attention mechanism identifies the critical state transition from speed decline to recovery across these nodes.

The capture of the two key spatio-temporal patterns effectively enhances the spatio-temporal modeling capability of SSA for local road networks and improves the interpolation accuracy for sparse data.

VI. Conclusion and Future Work

In this study, we propose a two-stage traffic data imputation framework that can capture both the global structure and the local nonlinear spatiotemporal relationships. In the first stage, the LRTC method is used to capture the global low-rank structure of the data, and the global low-rank initial value is filled for missing elements. In the second stage, STFEN is proposed to learn local nonlinear spatiotemporal relationships. Specifically, we design a sparse spatiotemporal attention module, which can simultaneously capture all dynamic spatiotemporal nonlinear relationships and alleviate computational complexity. We also add temporal attention to enhance temporal dependence. Experiments on two public datasets show that the two-stage traffic data imputation method can achieve accuracy exceeding state-of-the-art methods. Although the two-stage framework offers significant advantages, it still faces numerous challenges in practical applications, most notably in large-scale road network scenarios. Although Section V-D2 discusses the sparse attention mechanism we designed to reduce computational complexity, the computational burden remains influenced by the scale of N. Therefore, the impact of node quantity must be taken into account before applying this method to address traffic data imputation in real-world scenarios

Several promising research directions emerge for the spatio-temporal traffic data imputation problem. Firstly, the proposed two-stage traffic data imputation framework has only been validated under single-source data missingness scenarios. Given the inherent correlations often present in real-world data, integrating multi-source data for joint imputation represents a significant potential research avenue. Secondly, leveraging LLMs for semantic-aware spatiotemporal dependence mining. LLMs can parse unstructured traffic data (e.g., event reports, weather forecasts) and generate contextual embeddings that direct the sparse spatiotemporal attention to focus on semantically relevant features. And pre-trained LLMs on transportation knowledge can supply explicit constraints, such as traffic flow laws and seasonal patterns, to the fusion module, bridging the gap between data-driven models and prior physical knowledge.

References

[1] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.

[2] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial–temporal correlation," *Phys. A, Stat. Mech. Appl.*, vol. 446, pp. 54–63, Mar. 2016.

[3] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining road network correlation for traffic estimation via compressive sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1880–1893, Jul. 2016.

[4] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 108–120, Sep. 2013.

[5] X. Chen, J. Yang, and L. Sun, "A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102673.

[6] T. Nie, G. Qin, and J. Sun, "Truncated tensor schatten p-norm based approach for spatiotemporal traffic data imputation with complicated missing patterns," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103737.

[7] L. Deng, X.-Y. Liu, H. Zheng, X. Feng, and Y. Chen, "Graph spectral regularized tensor completion for traffic data imputation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10996–11010, Aug. 2022.

[8] X. Xu, M. Lin, X. Luo, and Z. Xu, "HRST-LR: A Hessian regularization spatio-temporal low rank algorithm for traffic data imputation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 11001–11017, Oct. 2023.

[9] Z. Liu, P. Zhou, Z. Li, and M. Li, "Think like a graph: Real-time traffic estimation at city-scale," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2446–2459, Oct. 2019.

[10] D. Xu, H. Peng, C. Wei, X. Shang, and H. Li, "Traffic state data imputation: An efficient generating method based on the graph aggregator," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13084–13093, Aug. 2022.

[11] Y. Zhang, X. Wei, X. Zhang, Y. Hu, and B. Yin, "Self-attention graph convolution residual network for traffic data completion," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 528–541, Apr. 2023.

[12] Q. Xu, S. Ruan, C. Long, L. Yu, and C. Zhang, "Traffic speed imputation with spatio-temporal attentions and cycle-perceptual training," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 2280–2289.

[13] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[14] Y. Liang, Z. Zhao, and L. Sun, "Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns," *Transp. Res. C, Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103826.

[15] T. Wang et al., "Synchronous spatiotemporal graph transformer: A new framework for traffic data prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10589–10599, Dec. 2023.

[16] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 914–921.

[17] Z. Wei et al., "STGSA: A novel spatial–temporal graph synchronous aggregation model for traffic prediction," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 226–238, Jan. 2023.

[18] S. Zhang, "Shell-neighbor method and its application in missing data imputation," *Int. J. Speech Technol.*, vol. 35, no. 1, pp. 123–133, Aug. 2011.

[19] P. J. García-Laencina, J. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1483–1493, Apr. 2009.

[20] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.

[21] X. Chen, Z. He, and L. Sun, "A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transp. Res. C-Emerg. Technol.*, vol. 98, pp. 73–84, Jan. 2018.

[22] X. Chen, S. Liang, Z. Zhang, and F. Zhao, "A novel spatiotemporal data low-rank imputation approach for traffic sensor network," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20122–20135, Oct. 2022.

[23] H. Xie, Y. Gong, and X. Dong, "Spatial–temporal regularized tensor decomposition method for traffic speed data imputation," *Int. J. Data Sci. Anal.*, vol. 17, no. 2, pp. 203–223, Jul. 2023.

[24] Q. Ma et al., "End-to-end incomplete time-series modeling from linear memory of latent variables," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4908–4920, Dec. 2020.

[25] A. Collado-Villaverde, P. Muñoz, and M. D. R. Moreno, "Brati: Bidirectional recurrent attention for time series imputation," 2025, *arXiv:2501.05401*.

[26] X. Wu, M. Xu, J. Fang, and X. Wu, "A multi-attention tensor completion network for spatiotemporal traffic data imputation," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20203–20213, Oct. 2022.

[27] W. Zhang, P. Zhang, Y. Yu, X. Li, S. A. Biancardo, and J. Zhang, "Missing data repairs for traffic flow with self-attention generative adversarial imputation net," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7919–7930, Jul. 2022.

[28] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4478–4485.

[29] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.

[30] B. Zhang, R. Miao, and Z. Chen, "Spatial–temporal traffic data imputation based on dynamic multi-level generative adversarial networks for urban governance," *Appl. Soft Comput.*, vol. 151, Jan. 2024, Art. no. 111128.

[31] J. Li, R. Li, L. Xu, and J. Liu, "Self-supervised generative adversarial learning with conditional cyclical constraints towards missing traffic data imputation," *Knowledge-Based Syst.*, vol. 284, Jan. 2024, Art. no. 111233.

[32] Y. Qu, Z. Li, X. Zhao, and J. Ou, "Towards real-world traffic prediction and data imputation: A multi-task pretraining and fine-tuning approach," *Inf. Sci.*, vol. 657, Feb. 2024, Art. no. 119972.

[33] T. Nie, G. Qin, W. Ma, Y. Mei, and J. Sun, "ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 2260–2271.

[34] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[35] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[36] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[37] F. Huang, P. Yi, J. Wang, M. Li, J. Peng, and X. Xiong, "A dynamical spatial–temporal graph neural network for traffic demand prediction," *Inf. Sci.*, vol. 594, pp. 286–304, May 2022.

[38] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[39] X. Ye, S. Fang, F. Sun, C. Zhang, and S. Xiang, "Meta graph transformer: A novel framework for spatial–temporal traffic prediction," *Neurocomputing*, vol. 491, pp. 544–563, 2022.

[40] G. Huo, Y. Zhang, B. Wang, J. Gao, Y. Hu, and B. Yin, "Hierarchical spatio-temporal graph convolutional networks and transformer network for traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 3855–3867, Apr. 2023.

[41] Q. Ren, Y. Li, and Y. Liu, "Transformer-enhanced periodic temporal convolution network for long short-term traffic flow forecasting," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120203.

[42] L. Wang, D. Guo, H. Wu, K. Li, and W. Yu, "TC-GCN: Triple cross-attention and graph convolutional network for traffic forecasting," *Inf. Fusion*, vol. 105, May 2024, Art. no. 102229.

[43] Z. Geng et al., "STGAFormer: Spatial–temporal gated attention transformer based graph neural network for traffic flow forecasting," *Inf. Fusion*, vol. 105, May 2024, Art. no. 102228.

[44] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3842–3849.

[45] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12301–12310, Aug. 2022.

[46] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 5689–5698.

[47] P. Mattei and J. Frellsen, "MIWAE: Deep generative modelling and imputation of incomplete data sets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4413–4423.

[48] L. Bai, L. Yao, C. Li, and X. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. NIPS*, 2020, pp. 17804–17815.

[49] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 24804–24816.

[50] Q. Ni and X. Cao, "MBGAN: An improved generative adversarial network with multi-head self-attention and bidirectional RNN for time series imputation," *Eng. Appl. Artif. Intell.*, vol. 115, Oct. 2022, Art. no. 105232.

[51] A. Y. Yildiz, E. Koç, and A. Koç, "Multivariate time series imputation with transformers," *IEEE Signal Process. Lett.*, vol. 29, pp. 2517–2521, 2022.

[52] W. Du, D. Coté, and Y. Liu, "SAITS: Self-attention-based imputation for time series," *Expert Syst. Appl.*, vol. 219, Jun. 2023, Art. no. 119619.

[53] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 4, 2023, pp. 4365–4373.

[54] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.

[55] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *Siam Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[56] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

Jiyu Wang received the B.S. degree in traffic engineering from Beijing Jiaotong University, Beijing, China, in 2021, and the M.S. degree in transportation from Sun Yat-sen University, Shenzhen, China, in 2024. He is currently pursuing the Ph.D. degree with The Hong Kong University of Science and Technology (Guangzhou), China. His research interests include spatio-temporal data mining and mobility management.

Zhidan Liu (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2014. After that, he was a Research Fellow with Nanyang Technological University, Singapore, and a Faculty Member with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He is currently an Assistant Professor with the Intelligent Transportation Thrust, System Hub, The Hong Kong University of Science and Technology (Guangzhou). His research interests include the Internet of Things, mobile computing, urban computing, and big data analytics. He is a Senior Member of CCF.