**This is the Pre-Published Version.**

The following publication Z. Zheng, Y. He, Z. Wang and W. Ma, "A Physics-Regularized Multiscale Attention Network for Spatiotemporal Traffic Data Imputation," in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 11, pp. 19522-19537, Nov. 2025 is available at https://doi.org/10.1109/TITS.2025.3595779.



# A Physics-Regularized Multiscale Attention Network for Spatiotemporal Traffic Data Imputation

Zhenjie Zheng, Yulin He, Zhengli Wang, Wei Ma *Member, IEEE*

***Abstract*—Spatiotemporal traffic data imputation is a fundamental task in numerous smart mobility applications. Existing studies indicate that accurately estimating missing values from observed data relies on capturing the spatiotemporal dependencies in traffic data. However, such dependencies may exhibit distinct characteristics across varying spatiotemporal areas, such as local short-term traffic fluctuations versus global long-range periodic commuting patterns. The comprehensive multiscale nature of dependencies in traffic data, encompassing more than just local and global levels, has not been well explored in the literature. To address this issue, we propose a physics-regularized multiscale attention network (PRMAN) that hierarchically extracts spatiotemporal features from local dynamics to global trends. Specifically, the proposed PRMAN builds upon the novel Swin Transformer and introduces a hierarchical architecture that performs self-attention in local spatiotemporal windows. By systematically expanding the window size across layers, this hierarchical design explicitly addresses the distinct characteristics between local and global spatiotemporal dependencies at different scales. Meanwhile, a physics-regularized loss function is developed to align learned spatiotemporal dependencies with traffic dynamics described by the fundamental diagram. This improves the model's generalizability beyond the training data, ensuring robust performance on unseen datasets. Numerical experiments on multiple benchmark datasets demonstrate that our proposed PRMAN achieves state-of-the-art performance in handling diverse and complex missing data patterns. The code and model are publicly available at https://github.com/2222ad/PRMAN.**

***Index Terms*—data imputation, spatiotemporal dependencies, multiscale attention, hierarchical architecture, physics-regularized neural network**

## I. Introduction

SPATIOTEMPORAL traffic data, which capture traffic states across different time intervals and road segments, are fundamental to modern smart cities. However, such data are often incomplete across both spatial and temporal dimensions due to various factors, such as sensor malfunctions that lead to missing spatial coverage and random communication failures that affect temporal consistency [1], [2]. For example, it is reported that approximately 30% of the traffic sensors in the California Performance Measurement System (PeMS) are malfunctioning [3]. Similarly, traffic sensors in Beijing have been reported to experience intermittent failures, resulting in up to 25% missing data [4], [5]. These missing data impair the quality of traffic datasets, reducing their reliability and effectiveness in various smart mobility applications [6]. Therefore, it is essential to develop effective imputation methods to estimate missing values and reconstruct complete traffic datasets.

Existing studies have widely recognized that capturing the inherent spatiotemporal dependencies in traffic data is fundamental to traffic data imputation [7]–[9]. Early studies primarily utilize time series methods, such as the autoregressive integrated moving average (ARIMA) model [10] and vector autoregressive (VAR) model [11] to characterize the temporal dependencies of traffic data while neglecting the spatial correlations. Later, various matrix or tensor decomposition methods [12], [13] are developed to address both spatial and temporal dependencies, which effectively enhance the data imputation performance. However, these methods typically rely on the low-rank assumption and overlook the critical domain knowledge, such as the spatiotemporal traffic dynamics described by the fundamental diagram [3], [14]. Consequently, they may struggle to capture the underlying causal mechanisms that govern the evolution of traffic states across temporal and spatial dimensions. Meanwhile, with the success of neural networks, various deep learning models, such as Autoencoders (AE) [15], [16], Convolutional Neural Networks (CNN) [17], and Graph Neural Networks (GNN) [18], have been widely used to capture spatiotemporal dependencies in traffic data and produce impressive results. More recently, Transformer-based approaches have gained significant attention and demonstrated strong performance in capturing long-range or global spatiotemporal dependencies [19], [20]. The self-attention mechanism, which dynamically weighs input features based on their relevance, enables attention-based methods to effectively handle complex missing data patterns [21], [22].

While existing studies have systematically examined the spatiotemporal dependencies in traffic data, the distinct characteristics that these dependencies exhibit across different spatiotemporal scales remain underexplored. Note that some research leverages both global and local information [14], [23] for traffic data imputation. However, their methods lack explicit mechanisms to hierarchically model spatiotemporal dependencies across multiple scales. Furthermore, they do not guarantee that the learned spatiotemporal dependencies align with established domain knowledge, which may compromise the model generalizability. More specifically, local short-term congestion propagation between adjacent roads caused by incidents or

Zhenjie Zheng and Wei Ma are with the Department of Civil and Environmental Engineering at the Hong Kong Polytechnic University, Hong Kong, SAR, China (E-mail: zzj17.zheng@polyu.edu.hk; wei.w.ma@polyu.edu.hk).

Yulin He and Zhengli Wang are with the School of Management and Engineering, Nanjing University, Nanjing 210093, China (E-mail: 502023150020@smail.nju.edu.cn; zhlwang@nju.edu.cn).

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

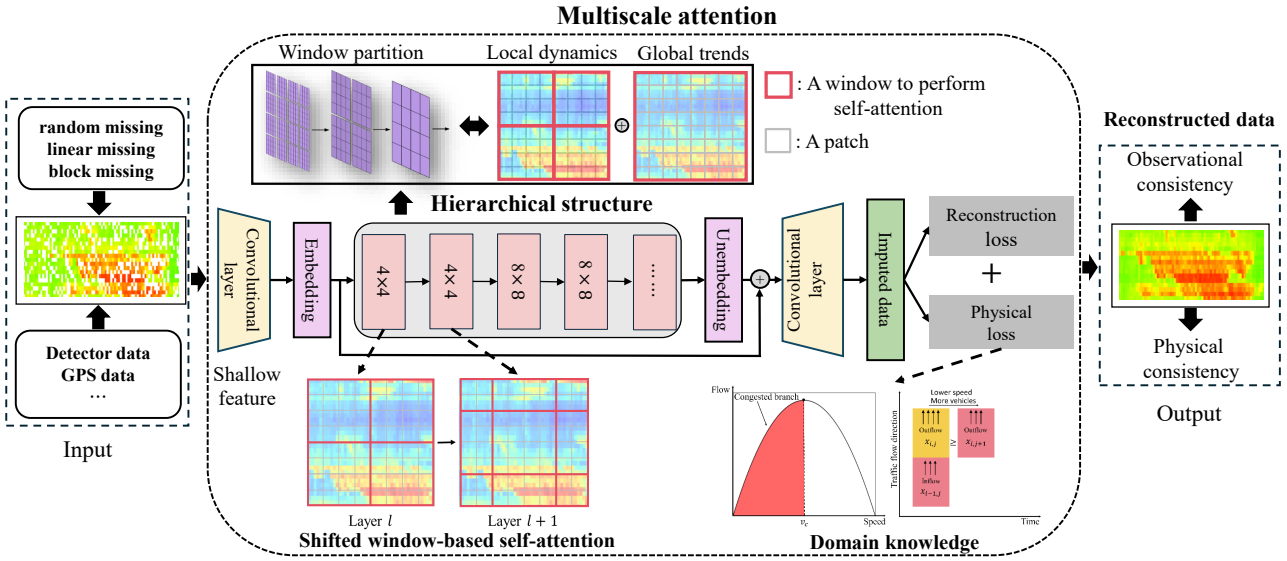



Fig. 1. An overview of the proposed physics-regularized multiscale attention network.

bottlenecks usually exhibits rapid and spatially contiguous interactions. These local traffic dynamics are dominated by physical propagation rules, like car-following theory [14]. In contrast, global long-term traffic patterns reflect overall conditions and trends of the traffic system, such as gradual traffic slowdowns during adverse weather events (e.g., heavy snowfall) and widespread city-wide congestion during peak hours. This indicates that global trends often involve delayed and spatially dispersed dependencies, unlike local dynamics. Consequently, existing methods [20], [24], [25] that do not explicitly handle multiscale spatiotemporal dependencies or incorporate domain knowledge warrant further investigation. In light of this, applying physics-regularized self-attention across varying spatiotemporal areas appears to perfectly address the above issues.

In this study, we propose a physics-regularized multiscale attention network (PRMAN) to address the aforementioned challenges. The overview of our work is illustrated in Fig. 1. The proposed PRMAN builds upon the Swin Transformer while introducing a modified self-attention mechanism that explicitly incorporates spatiotemporal coordinates. Specifically, it first partitions the input data into non-overlapping spatiotemporal windows and performs local self-attention within each window. To capture the spatiotemporal dependencies at different scales, the PRMAN employs a hierarchical architecture that progressively expands the window size across layers. This allows the network to effectively characterize local traffic dynamics within individual windows while gradually integrating global trends over broader spatiotemporal areas. Subsequently, windows are shifted between successive layers to enable cross-window interactions and information exchange between adjacent roads and time intervals. Meanwhile, a physics-regularized loss function is developed to enforce consistency between the learned spatiotemporal dependencies and the traffic dynamics described by the fundamental diagram. Such domain knowledge improves the model's generalization to unseen datasets and enhances its robustness against data anomalies. We validate our proposed PRMAN by conducting numerical experiments on multiple benchmark datasets. The inaccuracies in these data, such as those arising from atmospheric delays and signal reflections in urban environments [26], [27], are carefully examined and effectively mitigated through the integration of domain knowledge [3]. Results demonstrate that our model achieves state-of-the-art performance under various complex missing data patterns. .

We summarize the main contributions of this research as follows:

- We develop a multiscale attention network with a novel hierarchical architecture that progressively expands window sizes across layers for traffic data imputation. This effectively captures local traffic dynamics within individual windows while gradually integrating global trends over broader spatiotemporal areas;
- A physics-regularized loss function is incorporated into the network to enforce consistency between the learned spatiotemporal dependencies and the traffic dynamics described by the fundamental diagram, thereby improving model generalization and robustness; and
- Numerical experiments on multiple benchmark datasets show that our model consistently achieves state-of-the-art performance under a variety of complex missing data patterns.

The remainder of this paper is organized as follows. Section II reviews the related literature on traffic data imputation. Section III outlines our problem setting. Section IV elaborates on the proposed PRMAN. Section V presents the implementation details and training procedures. Section VI presents the numerical experiments, which demonstrate that our model outperforms the baseline methods across multiple benchmark datasets. Finally, the conclusions and future research directions are discussed in Section VII.

## II. Literature review

In this section, we provide a detailed review of existing studies on spatiotemporal traffic data imputation. Early studies primarily address missing values by treating the spatial and temporal dimensions separately. For example, some studies employed autoregressive models, such as ARIMA [10] and VAR [11], to capture temporal autocorrelation, while others used spatial interpolation techniques like k-nearest neighbors (KNN) [28] to estimate missing values. However, these methods fail to simultaneously capture the spatial and temporal dependencies in traffic data. Additionally, they often rely on oversimplified assumptions of linearity [29] and local smoothness [30], leading to suboptimal performance in scenarios that require integrated spatiotemporal reasoning [31], [32].

Later, various matrix and tensor decomposition methods have been proposed to model spatiotemporal dependencies and address the above limitations. For example, Tan et al. [33] employed a truncated nuclear norm to approximate tensor rank and formulate an efficient iterative algorithm based on the alternating direction method of multipliers (ADMM) to capture complex spatiotemporal patterns in traffic data. Jia et al. [34] integrated periodicity, road similarity, and temporal coherence into the matrix factorization method to further enhance the characterization of spatiotemporal dependencies. Lei et al. [35] integrated Gaussian process (GP) priors into the matrix factorization framework to reconstruct missing data while preserving spatial and temporal dependencies. For high-dimensional traffic data spanning multiple days, Chen et al. [13] extended the Bayesian probabilistic matrix factorization model to higher-order tensor decomposition methods. To efficiently process large-scale data, Chen et al. [36] proposed the low-tubal-tank smoothing tensor completion (LSTC) model that decomposes the large problem into a series of smaller subproblems. However, these methods often assume low-rank data structures and fail to account for critical traffic-specific correlations, such as spatiotemporal traffic dynamics governed by the fundamental diagram. Consequently, they may struggle to capture the intrinsic spatiotemporal evolution of traffic states governed by traffic domain knowledge.

Meanwhile, driven by the success of neural networks, deep learning methods have been widely adopted for traffic data imputation. Preliminary studies mainly employed simple neural networks to capture spatiotemporal relationships in traffic data. For example, Duan et al. [15] developed a denoising stacked autoencoder for traffic data imputation with consideration of both temporal and spatial factors. Che et al. [37] utilized a Recurrent Neural Network (RNN) model that accounts for spatial and temporal missing data patterns. Chen et al. [38] leveraged generative adversarial networks (GANs) to generate realistic traffic flow data, thereby improving the accuracy of traffic flow imputation and enhancing the reliability of traffic predictions. Subsequently, researchers have explored more advanced architectures by integrating multiple fundamental neural network models to further improve the model performance [8], [39], [40]. For example, Jin et al. [41] developed a spatiotemporal graph neural network (STGCN), which combined a graph convolutional network (GCN) for spatial dependency extraction with recurrent convolutional modules for temporal modeling. Similarly, Liang et al. [42] proposed an inductive graph neural network model that integrates a residual gated temporal convolutional network to capture temporal patterns from long sequences. The model also exploited the generalization of GNNs by incorporating masked learning and an adaptive memory-based attention mechanism to learn the implicit spatial correlations within networks.

Recently, Transformer-based methods have gained significant attention for their ability to capture global spatiotemporal dependencies. For example, Zhang et al. [43] introduced a self-attention generative adversarial network that integrates a self-attention mechanism, an autoencoder, and a GAN for traffic data imputation. Yang et al. [44] proposed a novel approach that combines spatiotemporal learnable bidirectional attention mechanisms with a GAN to estimate missing values. Similarly, Liu et al. [24] introduced spatio-temporal adaptive embedding (STAE), a novel component that enhances the performance of vanilla Transformers for traffic data imputation. Subsequent innovations, such as spatiotemporal attention mechanisms [25] and low-rankness-induced Transformers [20], have also been developed to further improve data imputation accuracy and robustness.

Despite the promising performance of these deep learning models, their ability to capture spatiotemporal dependencies in traffic data remains limited due to two key aspects. First, most deep learning models in the literature overlook the multiscale nature of spatiotemporal traffic dependencies. Consequently, they cannot well address the unique characteristics of multiscale dependencies across varying spatiotemporal areas. For example, Transformer-based methods [20], [25] applied self-attention globally across all tokens, limiting their ability to effectively model spatiotemporal dependencies at multiple scales. Although some studies have leveraged both global and local information for traffic data imputation [14], [23], their methods lacked explicit mechanisms to hierarchically model spatiotemporal dependencies across multiple scales. Second, the robustness and generalization of deep learning models highly rely on data quality and availability. Consequently, their performance may degrade in the presence of data anomalies and limited training data. Incorporating traffic domain knowledge helps mitigate the challenge by filtering out inconsistencies and guiding the model toward more robust representations.

## III. Problem statement

In this study, we focus on estimating missing traffic speed values from partially observed data under complex missing patterns. Let $E = \{e_1, \cdots, e_i, \cdots e_I\}$ be the set of loop detectors. These loop detectors are sequentially installed on roads from upstream to downstream. Similarly, let $T = \{t_1, \cdots, t_j, \cdots t_J\}$ denote the set of time intervals. For any given $i$ and $j$, a unique detector at a specific time interval can be determined. The observed traffic data is represented as a two-dimensional matrix $X$ of size $I \times J$, where $x_{i,j}$ denotes the speed recorded by

detector $i$ during time interval $j$. Furthermore, we define a mask matrix $M = \{m_{i,j}\}$ to indicate whether the data at row $i$ and column $j$ is observed. Mathematically, this can be formulated as follows:

$$m_{i,j} = \begin{cases} 1, & \text{if } x_{i,j} \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

To summarize, the input to our model is the speed matrix $X$, which contains missing values and is similar to those used in existing studies [45], [46]. The cell size in the speed matrix is determined by the reporting interval and the spacing between detectors. It is important to note that this study addresses a variety of missing data patterns, including random missing, linear missing, block missing, and their combinations. Accordingly, the mask matrix $M$ is introduced in the PRMAN to provide a unified method for characterizing the diverse missing patterns. This mask matrix enables the model to effectively characterize and accommodate multiple complex missing patterns, allowing data with different missing patterns to be utilized simultaneously during the training process. Our objective is to recover the missing values in $X$ based on the observed data, represented as $X \odot M$, where $\odot$ represents the Hadamard product.

## IV. A PHYSICS-REGULARIZED MULTISCALE ATTENTION NETWORK

In this section, we first present the architecture of the proposed PRMAN in Section IV-A. We then present the computational details of PRMAN, including modified self-attention computation and a shifted window mechanism in Section IV-B. The loss function that consists of reconstruction loss and physics-regularized loss is introduced in Section IV-C.

### A. Model architecture

The proposed PRMAN framework processes a dynamically partitioned two-dimensional spatiotemporal matrix as input. It employs a hierarchical architecture by initially segmenting the input into small patches and progressively merging neighboring patches in deeper layers (see Fig. 2). The framework consists of three modules: shallow feature extraction, hierarchical deep feature extraction, and spatiotemporal data reconstruction. Additionally, the mathematical equations supporting the model are detailed in the Appendix.

*1) Shallow feature extraction:* Initially, a $3 \times 3$ convolutional layer is employed to extract the shallow features from the input. Let $Input = (X \odot M,\ M)$ denote the model input, which has a shape of $(I, J, 2)$. In this context, $X$ is an $I \times J$ matrix containing the observed traffic speed data. The shallow feature extraction $H_{SF}$ is formulated as follows:

$$H_{SF} = Conv2d(Input), \tag{2}$$

where $Conv2d(\cdot)$ is a convolutional layer. Typically, convolutional layers effectively capture early spatiotemporal features, which improves overall model performance.

*2) Hierarchical deep feature extraction:* Subsequently, we develop a Residual Swin Transformer Block (RSTB) to enable hierarchical deep feature extraction. The RSTB contains multiple Swin Transformer layers (STL) that partition the input into non-overlapping spatiotemporal windows and progressively expand the window size across layers (see the right-hand black box in Fig. 2). As shown in the figure, the initial patch size is $4 \times 4$ at the first layer. Then, the patch merging layer concatenates the features of each group of $2 \times 2$ neighboring patches. This operation reduces the number of patches while yielding a larger patch size of $8 \times 8$ at the second layer. Similarly, a patch size of $16 \times 16$ is achieved at the third layer. Importantly, this hierarchical architecture effectively captures the spatiotemporal features of traffic data at different scales, which can enhance data imputation accuracy by leveraging both local and global dependencies.

To facilitate deep feature extraction, we employ $N$ RSTBs in this module. Let $H_{\text{DF}}$ denote the output of the module. Let $H_0$ denote the initial input to the RSTB, and $H_1, \ldots, H_n, \ldots, H_N$ denote the intermediate feature representations. The output $H_{DF}$ can be expressed as follows:

$$H_n = RSTB_n(H_{n-1}), \quad n = 1, 2, ..., N; \tag{3}$$

$$H_{DF} = Conv2d(H_N). \tag{4}$$

*3) Spatiotemporal data reconstruction:* Finally, we impute the missing data by aggregating the extracted shallow and deep features (i.e., $H_{SF}$ and $H_{DF}$). The reconstructed data $\hat{X}$ is formulated as follows:

$$\hat{X} = Conv2d(Conv2d(H_{DF}) + H_{SF}). \tag{5}$$

It is worth noting that shallow features primarily capture low-frequency information, whereas deep features focus on high-frequency information. The proposed PRMAN utilizes a long skip connection to directly transmit low-frequency information to the data reconstruction module. This fusion of both low-frequency and high-frequency features enhances the propagation of long-range dependencies and stabilizes the training process.

### B. Shifted window mechanism for self-attention

*1) Architectural composition and feature representation in RSTB:* As shown in Fig. 2, the RSTB is composed of $L$ stacked STLs followed by a convolutional layer, with residual connections integrated throughout the architecture. Let $STL_{n,l}(\cdot)$ denote the $l$-th STL within the $n$-th RSTB. Let $H_{n,0}$ denote the input to the $n$-th RSTB. The intermediate feature representations following each STL are denoted as $H_{n,1}, \cdots, H_{n,l}, \cdots, H_{n,L}$. Mathematically, the $H_{n,l}$ is formulated as follows:

$$H_{n,0} = \text{Conv2d}(H_{n-1,L}) + H_{n-1,0}, \quad n = 1, 2, ..., N; \tag{6}$$

$$H_{n,l} = STL_{n,l}(H_{n,l-1}), \quad n = 1, 2, ..., N; l = 1, 2, ..., L. \tag{7}$$

*2) Window partitioning in STL:* For STL, it does not rely on global self-attention that computes the relationships between a token and all other tokens. Instead, it performs self-attention within partitioned local windows, complemented by shifted window strategies [47].

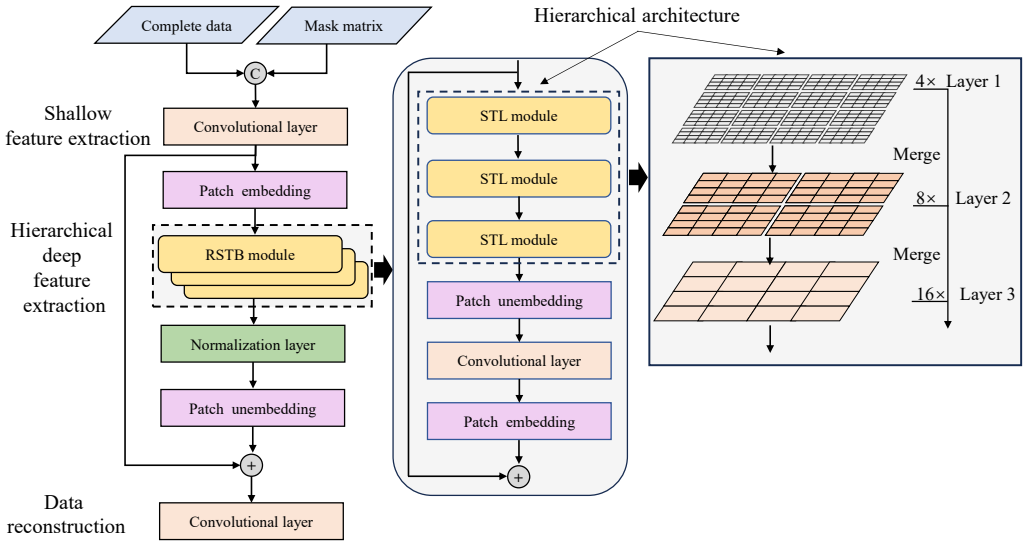



Fig. 2. The overall model architecture of our proposed PRMAN framework.

To achieve this, we first partition the input into non-overlapping windows, where each window consists of $M \times M$ patches. Following window partitioning, the dimension of each patch is transformed into $\lceil \frac{I}{M} \rceil \times \lceil \frac{J}{M} \rceil \times M^2$, where $\lceil \rceil$ represents the ceiling function. Let $C$ denote the number of feature dimensions after the embedding phase. Finally, these patch tokens are projected through a 2D convolutional layer, then flattened and transposed into a tensor of dimensions $(\lceil \frac{I}{M} \rceil \times \lceil \frac{J}{M} \rceil) \times M^2 \times C$. It is worth noting that the window size gradually expands across successive STLs, forming a hierarchical representation that captures both local and global spatiotemporal dependencies, as detailed in Section IV-A2.

*3) Shifted window-based self-attention in STL:* The above window-based self-attention only captures spatiotemporal dependencies within local windows, which overlooks the dependencies across different windows. To address this issue, we implement a shifted window partitioning strategy that alternates between two configurations across successive STLs. As shown in Fig. 3, the first STL uses a regular window partitioning strategy which starts from the top-left pixel, and the 8 × 8 feature map is evenly partitioned into 2 × 2 windows of size 4 × 4 (i.e., $M = 4$). Then, the next STL adopts a shifted configuration relative to the preceding layer, displacing the windows by $\lfloor \frac{H}{2} \rfloor$ pixels from the regular partitioning, where $\lfloor \rfloor$ represents the floor function.

As shown in Fig. 4, local self-attention is performed in two successive STLs: one using window-based multi-head self-attention (W-MSA) and the other using shifted window-based multi-head self-attention (SW-MSA). For each local window $Z \in R^{M^2 \times C}$,the multi-head self-attention is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + P}{\sqrt{d_k}}\right)V; \quad (8)$$

$$Q = ZW_Q, \quad K = ZW_K, \quad V = ZW_V. \quad (9)$$

$P$ represents the learnable relative two-dimensional positional encoding, while $Q$, $K$, $V$ denote the *query, key* and *value* matrices, respectively. In the multi-head self-attention mechanism, this process is repeated multiple times, with each iteration corresponding to a distinct attention head that focuses on different aspects of the input data. Once all heads have computed their respective attention outputs, these outputs are concatenated to form a comprehensive representation.

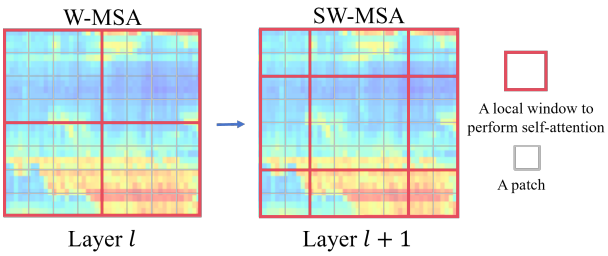



Fig. 3. An illustration of the shifted window mechanism across two successive STLs.

Before performing local self-attention, STL first normalizes the features using layer normalization (LN). After self-attention computation, the features pass through another LN, followed by a multi-layer perceptron (MLP). To address the vanishing gradient problem, residual connections are incorporated after both the self-attention and MLP computations. The complete process is formulated as follows:

$$\hat{Z}_n = \text{W-MSA}(\text{LN}(Z_{n-1})) + Z_{n-1}; \quad (10)$$

$$Z_n = \text{MLP}(\text{LN}(\hat{Z}_n)) + \hat{Z}_n; \quad (11)$$

$$\hat{Z}_{n+1} = \text{SW-MSA}(\text{LN}(Z_n)) + Z_n; \quad (12)$$

$$Z_{n+1} = \text{MLP}(\text{LN}(\hat{Z}_{n+1})) + \hat{Z}_{n+1}. \quad (13)$$

### C. Loss function

Our loss function is composed of two components: the reconstruction loss and the physics-regularized loss. The reconstruction loss quantifies the discrepancy between the imputed values and the corresponding ground truth. The physics-regularized loss quantifies the inconsistency

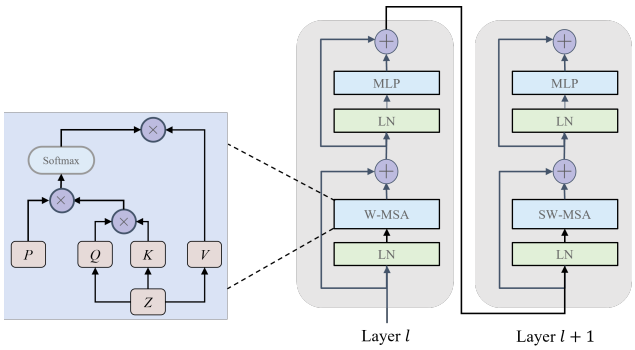



Fig. 4. Two successive STLs for performing W-MSA and SW-MSA.

between the reconstructed data and the spatiotemporal traffic dynamics.

*1) Reconstruction loss:* We use mean squared error (MSE) to quantify the reconstruction loss, which is defined as follows:

$$\mathcal{L}_r = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{x}_{i,j} - x_{i,j})^2}{IJ}, \tag{14}$$

where $x_{i,j}$ represents the ground true speed from detector $i$ at interval $j$ and $\hat{x}_{i,j}$ represents the corresponding imputed value. By minimizing Equation (14), the model learns to improve its data imputation accuracy.

*2) Physics-regularized loss:* An essential issue in traffic data imputation is ensuring that the reconstructed data adheres to established traffic domain knowledge. To this end, we build upon our previous study [3] and design a similar physics-regularized loss function based on the fundamental diagram, ensuring that the reconstructed data is statistically consistent with the observed data and physically plausible.

Let $v_c$ denote the critical speed, with speeds greater than $v_c$ corresponding to free-flow states and speeds lower than $v_c$ corresponding to congested states. According to the free-flow branch shown in Fig. 5, traffic flow decreases as traffic speed increases. This relationship implies that if the upstream speed collected by detector $i-1$ at time interval $j$ is greater than the downstream speed at the same time (i.e., $x_{i-1,j} \geq x_{i,j}$), then the upstream inflow should be lower than the downstream outflow. Consequently, the number of vehicles at detector $i$ and time interval $j+1$ should decrease, that is, the corresponding speed collected by detector $i$ and time interval $j+1$ should increase (i.e., $x_{i,j+1} \geq x_{i,j}$). To enforce this relationship, any reconstructed data in free-flow states that violate the constraint should be penalized using the following loss function:

$$\mathcal{L}_f = \frac{1}{(I-1)(J-1)}\sum_{i=2}^{I}\sum_{j=1}^{J-1} \max\left\{-(\hat{x}_{i-1,j} - \hat{x}_{i,j})(\hat{x}_{i,j+1} - \hat{x}_{i,j}),\ 0\right\}. \tag{15}$$

A similar loss function can be formulated for the reconstructed data in the congested states. According to the congested branch in Fig. 6, traffic flow decreases as traffic speed decreases. Following the same logic, if the upstream speed collected by detector $i$ at time interval $j$ is lower than the downstream speed at the same time (i.e., $x_{i,j} \leq x_{i-1,j}$), then the speed collected by detector $i$ should further decrease at time interval $j+1$ (i.e., $x_{i,j+1} \leq x_{i,j}$). The corresponding loss function is formulated as follows:

$$\mathcal{L}_c = \frac{1}{(I-1)(J-1)}\sum_{i=2}^{I}\sum_{j=1}^{J-1} \max\left\{-(\hat{x}_{i-1,j} - \hat{x}_{i,j})(\hat{x}_{i,j} - \hat{x}_{i,j+1}),\ 0\right\}. \tag{16}$$

Additionally, considering the inherent continuity of traffic states, the traffic conditions recorded by adjacent detectors over consecutive time periods should not exhibit significant variations. In view of this, we design a smooth loss function to ensure gradual and consistent fluctuations in the reconstructed traffic data, formulated as follows:

$$\mathcal{L}_s = \frac{1}{I(J-1)}\sum_{i=1}^{I}\sum_{j=1}^{J-1} max\left\{(x_{i,j+1} - x_{i,j})^2 - \gamma_t, 0\right\} + \frac{1}{(I-1)J}\sum_{i=1}^{I-1}\sum_{j=1}^{J} max\left\{(x_{i+1,j} - x_{i,j})^2 - \gamma_s, 0\right\}, \tag{17}$$

where $\gamma_t$ and $\gamma_s$ are two hyper-parameters used to measure the acceptable discontinuity of traffic speeds.

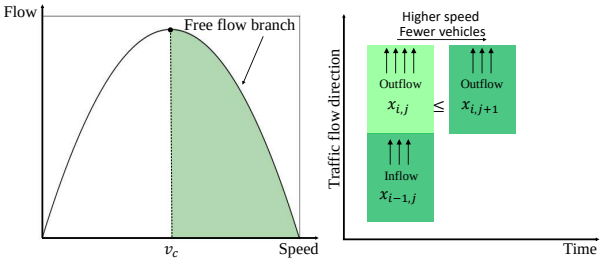



Fig. 5. Illustration of the physics-regularized loss in free-flow states.

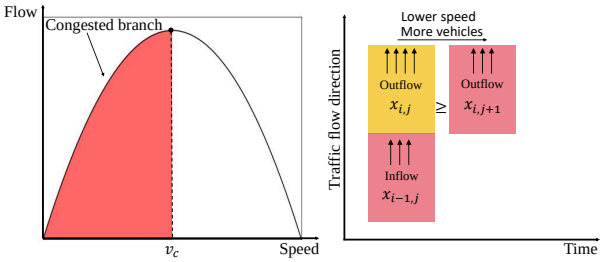



Fig. 6. Illustration of the physics-regularized loss in congested states.

*3) Total loss function:* The total loss function is defined as a weighted sum of the four aforementioned loss components, formulated as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{r}} + \lambda_p(\mathcal{L}_f + \mathcal{L}_c + L_s), \tag{18}$$

where $\lambda_p$ is a hyper-parameter that balances the importance of different loss terms. By minimizing the reconstruction loss and physics-regularized loss, our model not only ensures accurate imputation of missing data but also improves its generalization to unseen datasets. It is worth noting that the physics-regularized loss function is not used as a strict constraint but as a soft regularization term in the above equation. It serves as a physics-guided prior that encourages the reconstructed data to be consistent

with general traffic flow dynamics while allowing flexibility for data-driven learning. This design does not enforce a strict alignment to the expected relationship but penalizes deviations, thereby accounting for the uncertainty in real-world traffic data to some extent.

## V. Implementation of PRMAN

We use the Adam optimizer to train the proposed PRMAN [48]. In the training process, a learning rate decay mechanism is incorporated to speed up convergence and reduce the risk of overfitting. During the early training phase, we observe that the model struggles with data imputation, which may result in a relatively high physics loss. To resolve this issue, we implement a saturation mechanism that caps the physics-regularized loss when it exceeds a predefined threshold. We also incorporate a Gaussian noise matrix into the input to mitigate gradient computation anomalies that may occur when directly processing missing values (i.e., empty entries). The detailed training procedure is outlined in Algorithm 1.

**Algorithm 1** PRMAN training process

**Input:** Observed speed matrix: $X$;
Binary mask matrix: $M$;
Critical speed: $v_c$;
Hyper-parameters: $\lambda_p, \gamma_s, \gamma_t$;
Number of iterations: $Iter$.

1: Initialize the weight of all trainable parameters PRMAN.
2: **for** $i = 1, 2, \cdots, Iter$ **do**
3: Generate a Gaussian noise matrix $Z \sim N(0, 1)$.
4: Perturb the model input:$(X \odot M + Z \odot (1 - M)$ , $M)$.
5: Obtain the model output: $\hat{X}$.
6: Calculate the total loss of $\hat{X}$ using Equation (IV-C3).
7: Update network parameters with the Adam optimizer.
8: **end for**

**Output:** Reconstructed speed matrix $\hat{X}$.

Regarding the implementation details, our proposed PRMAN builds upon the Swin Transformer [47]. Specifically, we first utilize a 3x3 convolutional layer to extract low-level features with the embedding layer set to 96 channels. We then configure four RSTBs, each consisting of six STLs with a window size of 4, to extract spatiotemporal features. In each RSTB, the number of attention heads is set to 6. The data reconstruction layer employs a 3x3 convolutional kernel combined with absolute positional embeddings applied to each element. Additionally, the batch size and the initial learning rate are set to 16 and $10^{-4}$, respectively. The critical speed is set to 30 km/h. The model is implemented in PyTorch, and all training and evaluations are performed on an NVIDIA GeForce RTX 4070 GPU.

## VI. Numerical experiments

In this section, we evaluate our model on multiple real-world traffic datasets and compare its performance with various traffic data imputation methods. We find that our proposed PRMAN consistently outperforms these methods and achieves state-of-the-art performance across diverse and complex missing data patterns. The datasets used in our experiments are introduced in Section VI-A. The baseline methods are described in Section VI-B. The experimental settings are outlined in Section VI-C, and the results are presented in Section VI-D. Finally, we conduct an ablation study and a comprehensive sensitivity analysis to evaluate the contribution of key model components, as detailed in Sections VI-E and VI-F, respectively.

### A. Dataset description

We collect four publicly available real-world traffic datasets and select data from six highways to construct our benchmark datasets. The cell size (i.e., reporting interval and the spacing between loop detectors) in the speed matrix is directly derived from the datasets, and we do not adjust it manually. These datasets have been widely used in previous traffic studies [7], [20], ensuring their relevance and validation in the field. Table I provides a summary of their key characteristics. To ensure consistency across all datasets, our analysis is confined to the period from 7:00 a.m. to 11:00 p.m., during which data are generally available. Additionally, all data are aggregated into 5-minute intervals.

### B. Baseline methods

We compare the proposed PRMAN framework with the following data imputation methods:

- Historic Average (HA): The HA method is a straightforward imputation technique that replaces missing values with the mean of historical data. It is suitable for time series data when the data exhibit stable patterns and the proportion of missing values is relatively low. This method calculates the average of historical observations and uses it to fill in missing values, maintaining consistency while preserving overall trends.
- K-Nearest Neighbor (KNN) [49]: The KNN method estimates missing values by identifying the $K$ nearest neighbors based on similarity measures, such as Euclidean distance. It selects the $K$ most similar data points and imputes the missing value using their average or majority value. Essentially, this approach leverages local correlations in the data to provide more informed imputations.
- Self-Attention Generative Adversarial Imputation Network (SA-GAIN) [50]: The SA-GAIN integrates a self-attention mechanism, autoencoders, and Generative Adversarial Networks (GANs) to handle missing data effectively. The self-attention mechanism enables the model to capture correlations among spatially distributed sensors over time, enhancing its ability to process spatiotemporal data.
- ImputeFormer (IF) [20]: This model not only employs Transformer architectures but also enforces a low-rank structure on spatiotemporal data to regularize the data imputation. It uses projected attention for

TABLE I
OVERVIEW OF THE UTILIZED DATASETS.

| Dataset | Access link | Description |
|---|---|---|
| METR-LA | https://github.com/liyaguang/DCRNN | The METR-LA dataset is recorded from 207 loop detectors deployed on highways in Los Angeles County. The dataset is collected from March 1st, 2012, to June 27th, 2012. It provides a comprehensive representation of highway traffic conditions, offering detailed measurements of traffic speed and flow. |
| PEMS04 | https://github.com/Davidham3/ASTGCN/tree/master/data/PEMS04 | The PEMS04 dataset is derived from the Performance Measurement System (PeMS) in California. It includes data from over 300 sensors and offers comprehensive traffic speed and flow information across various road segments. The traffic data are recorded at 5-minute intervals from January 1, 2018, to February 28, 2018. |
| Seattle dataset | https://github.com/zhiyongc/Seattle-Loop-Data | The Seattle dataset is collected from loop detectors installed on major roads in Seattle, USA. It covers the entire year of 2005, with traffic speed and flow recorded at 5-minute intervals, offering a detailed overview of urban traffic conditions. |
| SanDiego dataset | https://data.transportation.gov/Automobiles/San-Diego-Test-Data-Sets | The SanDiego dataset, provided by the U.S. Department of Transportation, is sourced from the traffic monitoring system in SanDiego, California. It contains data collected from 2007 to 2010, covering traffic speed, flow, and density. |

temporal dynamics and embedded attention for spatial correlations. Additionally, a Fourier sparsity loss is applied to constrain the spectral structure learned by the model. By combining low-rank modeling with deep learning, this approach strikes a balance between inductive bias and high expressivity, enhancing its generalizability for a wide range of imputation tasks.

- Bayesian Augmented Tensor Factorization (BATF) [51]: BATF is an imputation method based on Bayesian tensor decomposition. It also integrates traffic domain knowledge from transportation systems. Consequently, BATF automatically learns model parameters within a fully Bayesian framework for data imputation.
- Bayesian Gaussian CP Decomposition (BGCP) [13]: BGCP is also an imputation method founded on Bayesian tensor decomposition. It extends the Bayesian probabilistic matrix factorization model to higher-order tensors and applies this advanced framework to spatiotemporal traffic data imputation.
- Low-Tubal-Rank Smoothing Tensor Completion (LSTC) [36]: LSTC is a scalable low-rank tensor learning model designed for large-scale spatiotemporal traffic data imputation. In particular, it is well-suited for multidimensional structures such as $location \times time \times day$ in traffic data. Unlike traditional tensor decomposition methods, LSTC incorporates linear unitary transformations to enable scalable tensor nuclear norm minimization.
- Pyramid Vision Transformer (PVT) [52]: PVT is a well-recognized approach that reformulates image-based learning tasks as time series modeling problems through a Transformer architecture. It also introduces a pyramid structure to generate multi-scale feature maps for efficient learning.
- Spatio-Temporal Denoising Graph Autoencoder (STDGAE) [53]: STDGAE effectively leverages temporal correlations and spatial coherence in the observed data to recover missing values. By integrating spatiotemporal graph convolution layers with a denoising autoencoder, it achieves state-of-the-art performance in data imputation tasks.

### C. Experimental setup

Our proposed PRMAN and the baseline methods are trained on identical datasets. Specifically, we split the traffic dataset into training and test sets using an 80:20 ratio. For each dataset, we adopt a method similar to that outlined in existing studies to generate three distinct missing data patterns: random, linear, and block missing. Importantly, to capture the diverse and complex nature of missing data in real-world applications, we also merge these patterns to generate mixed missing scenarios for model evaluation. We use the Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) to evaluate the data imputation performance. Mathematically, these metrics are defined as follows:

$$\text{MAPE} = \sum_{i=1}^{I}\sum_{j=1}^{J}(1-m_{i,j})\left|\frac{\hat{x}_{i,j}-x_{i,j}}{x_{i,j}}\right| \times \frac{100\%}{\sum_{i=1}^{I}\sum_{j=1}^{J}(1-m_{i,j})}; \tag{19}$$

$$\text{MSE} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(1-m_{i,j})(\hat{x}_{i,j}-x_{i,j})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}(1-m_{i,j})}, \tag{20}$$

where $\sum_{i=1}^{I}\sum_{j=1}^{J}(1-m_{i,j})$ indicates the total number of unobserved data points.

### D. Experimental results

*1) Comparative analysis of our model and baseline methods:* We first conduct benchmark experiments with a missing data rate of 30%. Table II presents the data imputation performance for our proposed model alongside the baseline methods. Specifically, Columns 1 and 2 report the data imputation methods and data missing patterns, respectively. Columns 3 through 14 report the MSE and MAPE of these methods across different datasets. Note that both the Seattle dataset (i.e., Seattle005 and Seattle405) and the SanDiego dataset (i.e., SanDiego005 and SanDiego015) provide two sets of traffic speed data from different highways. For the HA method, imputation values are computed as the average of the first five days of the corresponding data. For the KNN method, the number of clusters is set to five. The tensor decomposition methods employ the same parameter settings as reported in their

TABLE II
THE PERFORMANCE (MSE/MAPE (%)) OF OUR PROPOSED MODEL AND THE BASELINE METHODS.

| Model | Type | Seattle005 | | Seattle405 | | SanDiego005 | | SanDiego015 | | METR-LA | | PEMS04 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE |
| **PRMAN** | block | 46.31 | **17.40** | **18.56** | **6.98** | 4.28 | 1.89 | **5.61** | 2.17 | **30.24** | **10.81** | **9.81** | **3.02** |
| | linear | 72.72 | **21.68** | **21.88** | **7.34** | 6.94 | 2.66 | **17.36** | 4.52 | **43.01** | **14.31** | **32.75** | **5.81** |
| | random | 17.42 | 9.86 | **8.41** | **4.89** | **1.33** | 1.20 | **1.74** | **1.41** | **14.15** | **6.99** | **2.41** | **1.79** |
| | mixed | **32.31** | **10.59** | 14.34 | **5.95** | 5.23 | 2.32 | **10.07** | **3.03** | **25.75** | 8.49 | **19.80** | **4.81** |
| IF | block | 55.05 | 18.51 | 22.38 | 7.72 | 4.29 | 1.70 | 9.48 | 2.68 | 62.74 | 15.64 | 11.13 | 3.14 |
| | linear | 89.96 | 24.02 | 22.99 | 7.65 | **4.93** | **1.99** | 21.10 | **4.41** | 77.17 | 17.02 | 36.55 | 6.06 |
| | random | 26.66 | 12.46 | 11.60 | 5.67 | 2.36 | 1.39 | 6.38 | 2.39 | 46.04 | 12.48 | 5.25 | 2.28 |
| | mixed | 78.08 | 11.64 | 17.04 | 10.51 | 4.97 | 1.83 | 14.17 | 3.48 | 32.57 | **8.06** | 39.73 | 6.87 |
| SA-GAIN | block | 73.08 | 21.59 | 30.03 | 9.99 | 6.53 | 2.62 | 11.62 | 3.53 | 71.49 | 21.52 | 22.64 | 5.43 |
| | linear | 81.59 | 22.83 | 31.35 | 9.98 | 6.24 | 2.52 | 17.94 | 4.35 | 71.89 | 19.04 | 37.57 | 7.31 |
| | random | 53.68 | 17.48 | 24.98 | 9.16 | 4.61 | 2.16 | 10.89 | 3.35 | 64.21 | 18.74 | 20.04 | 5.11 |
| | mixed | 62.63 | 15.79 | 30.70 | 13.42 | 6.46 | 2.63 | 14.37 | 3.96 | 69.64 | 23.42 | 27.53 | 5.59 |
| HA | block | 102.37 | 25.70 | 190.33 | 29.99 | 20.43 | 4.86 | 60.38 | 10.50 | 224.68 | 41.52 | 70.82 | 10.96 |
| | linear | 116.33 | 27.82 | 190.21 | 30.18 | 20.56 | 4.88 | 64.93 | 10.86 | 217.08 | 38.98 | 72.89 | 10.87 |
| | random | 98.08 | 24.42 | 189.05 | 29.93 | 18.76 | 4.69 | 66.31 | 11.04 | 234.63 | 41.42 | 69.41 | 10.78 |
| | mixed | 111.84 | 26.57 | 187.92 | 30.23 | 20.43 | 4.85 | 62.42 | 10.58 | 247.26 | 34.43 | 71.15 | 10.89 |
| KNN | block | 168.18 | 29.54 | 47.12 | 12.20 | 5.76 | 1.82 | 20.84 | 3.58 | 75.27 | 14.98 | 21.92 | 4.65 |
| | linear | 124.01 | 32.19 | 59.38 | 16.37 | 14.76 | 4.16 | 44.34 | 8.33 | 232.24 | 41.47 | 88.64 | 13.65 |
| | random | 77.90 | 19.53 | 22.13 | 8.18 | 1.79 | 1.13 | 1.93 | **1.41** | 35.11 | 10.23 | 6.74 | 2.66 |
| | mixed | 124.79 | 32.56 | 54.10 | 15.53 | 14.54 | 4.21 | 47.11 | 8.63 | 280.99 | 45.60 | 67.78 | 10.65 |
| BATF | block | 60.90 | 19.45 | 34.10 | 10.75 | 3.89 | **1.56** | 5.90 | **1.96** | 56.35 | 16.42 | 25.95 | 5.06 |
| | linear | 94.16 | 23.88 | 119.58 | 18.42 | 17.36 | 2.99 | 47.77 | 6.01 | 131.31 | 23.39 | 266.09 | 17.64 |
| | random | 27.11 | 11.87 | 20.24 | 8.42 | 1.47 | 1.08 | 2.54 | **1.40** | 34.68 | 12.11 | 8.32 | 3.39 |
| | mixed | 77.30 | 23.77 | 53.68 | 12.08 | 6.70 | 1.96 | 21.26 | 4.52 | 89.09 | 14.94 | 60.01 | 6.68 |
| BGCP | block | 69.99 | 21.83 | 37.50 | 11.43 | 5.33 | 1.79 | 8.40 | 2.36 | 57.57 | 17.65 | 25.16 | 5.49 |
| | linear | 149.71 | 30.84 | 124.18 | 16.42 | 18.98 | 4.35 | 42.93 | 6.67 | 141.89 | 25.13 | 347.51 | 18.41 |
| | random | 28.13 | 12.65 | 21.16 | 8.59 | 1.77 | 1.18 | 3.50 | 1.59 | 39.64 | 12.55 | 9.21 | 3.59 |
| | mixed | 81.01 | 23.23 | 63.60 | 14.99 | 5.95 | 1.98 | 13.98 | 3.19 | 110.60 | 13.99 | 51.22 | 6.66 |
| LSTC | block | 52.58 | 19.55 | 28.06 | 10.30 | 4.99 | 1.67 | 5.87 | 2.19 | 53.45 | 16.87 | 13.55 | 4.40 |
| | linear | 91.77 | 26.83 | 168.29 | 32.65 | 9.15 | 2.82 | 25.00 | 5.33 | 150.93 | 26.10 | 55.87 | 10.34 |
| | random | 26.17 | 12.17 | 22.27 | 7.14 | 1.41 | **1.07** | 2.57 | 1.45 | 23.73 | 9.18 | 4.71 | 2.48 |
| | mixed | 59.23 | 20.51 | 37.70 | 12.26 | **4.86** | **1.84** | 12.29 | 3.12 | 63.29 | 14.86 | 20.63 | 4.89 |
| PVT | block | **46.17** | 18.41 | 27.99 | 10.94 | 9.53 | 3.30 | 21.51 | 5.39 | 145.97 | 28.36 | 46.88 | 9.49 |
| | linear | **51.64** | 26.47 | 26.47 | 10.58 | 7.91 | 3.03 | 17.99 | 4.88 | 134.15 | 27.32 | 38.45 | 8.40 |
| | random | 13.71 | 4.50 | 28.07 | 11.00 | 9.05 | 3.22 | 21.24 | 5.37 | 144.38 | 28.39 | 44.10 | 9.09 |
| | mixed | 41.34 | 15.92 | 29.11 | 11.13 | 9.29 | 3.29 | 21.05 | 5.33 | 154.20 | 29.47 | 49.39 | 9.61 |
| STDGAE | block | 48.83 | 18.16 | 23.06 | 8.23 | **3.42** | 1.63 | 6.82 | 2.14 | 34.87 | 11.42 | 25.74 | 6.47 |
| | linear | 63.14 | 28.91 | 26.32 | 9.11 | 9.27 | 3.18 | 18.74 | 4.63 | 47.92 | 15.24 | 52.21 | 10.01 |
| | random | **12.68** | **4.14** | 19.84 | 7.76 | 1.82 | 1.43 | 2.31 | 1.58 | 15.36 | 7.12 | 31.22 | 7.19 |
| | mixed | 37.34 | 11.04 | **12.25** | 6.15 | 5.87 | 2.38 | 11.29 | 3.27 | 33.47 | 12.13 | 20.83 | 5.63 |

Note: values highlighted in bold represent the best performance for each scenario.

respective papers. Similarly, the SA-GAIN and ImputeFormer models use configurations consistent with their original implementations, with only minor modifications to the learning rate to ensure stable and better performance.

From Table II, we find that our model generally outperforms the baseline methods under various missing data patterns across multiple datasets. Even when our model is not the top performer for a specific missing pattern within a dataset, our performance gap with the best model remains minimal. We also find that Transformer-based methods, specifically our proposed PRMAN and IF, perform best in nearly all datasets. This indicates that Transformer architectures are indeed effective in capturing the spatiotemporal dependencies inherent in traffic data. However, our PRMAN outperforms IF by leveraging a hierarchical structure that enables the model to capture both local and global dependencies at different scales of granularity. Additionally, we find that methods based on tensor decomposition show the best performance on some datasets but exhibit inconsistent performance across others. This is likely because the underlying low-rank or related assumptions may not strictly hold for all datasets. Furthermore, for different missing data patterns, the results reveal that both the MSE and MAPE for random missing values are significantly lower than those observed for other patterns. In contrast, the linear, block, and mixed missing patterns are considerably more challenging to address.

*2) Results under different missing rates:* We then test the performance of our model and the baseline methods under different missing rates. For the sake of clarity, only those baselines that exhibit strong performance in Section VI-D1 are reported. Specifically, we evaluate our model against IF and LSTC on the METR-LA dataset and examine their performance across missing data rates of 30%, 40%, 50%, 60%, and 70%. The results are summarized in Table III. We find that our model outperforms the baseline methods in most cases. Importantly, the MSE and MAPE of our model exhibit greater stability than those of IF and LSTC even at a 70% missing rate. This suggests that our proposed PRMAN can fully utilize the observed data to maintain robust performance, as long as there is enough information for accurate imputation. In contrast, other methods tend to be more negatively affected. These results demonstrate that the proposed PRMAN can accurately and robustly impute missing values across diverse missing rate scenarios.

We further visualize our results using speed contour plots. Figs. 7 and 8 illustrate the ground truth, input, and output of our model under different missing data patterns, with missing rates set at 30% and 70%, respectively. By comparing the model output with the ground truth, we find that the proposed PRMAN accurately estimates missing values even when the missing patterns are complex and the observed data is limited (see Figs. 8f and 8h). This demonstrates that the proposed PRMAN exhibits remarkable performance in capturing spatiotemporal dependencies and reconstructing traffic data.

### E. Ablation study

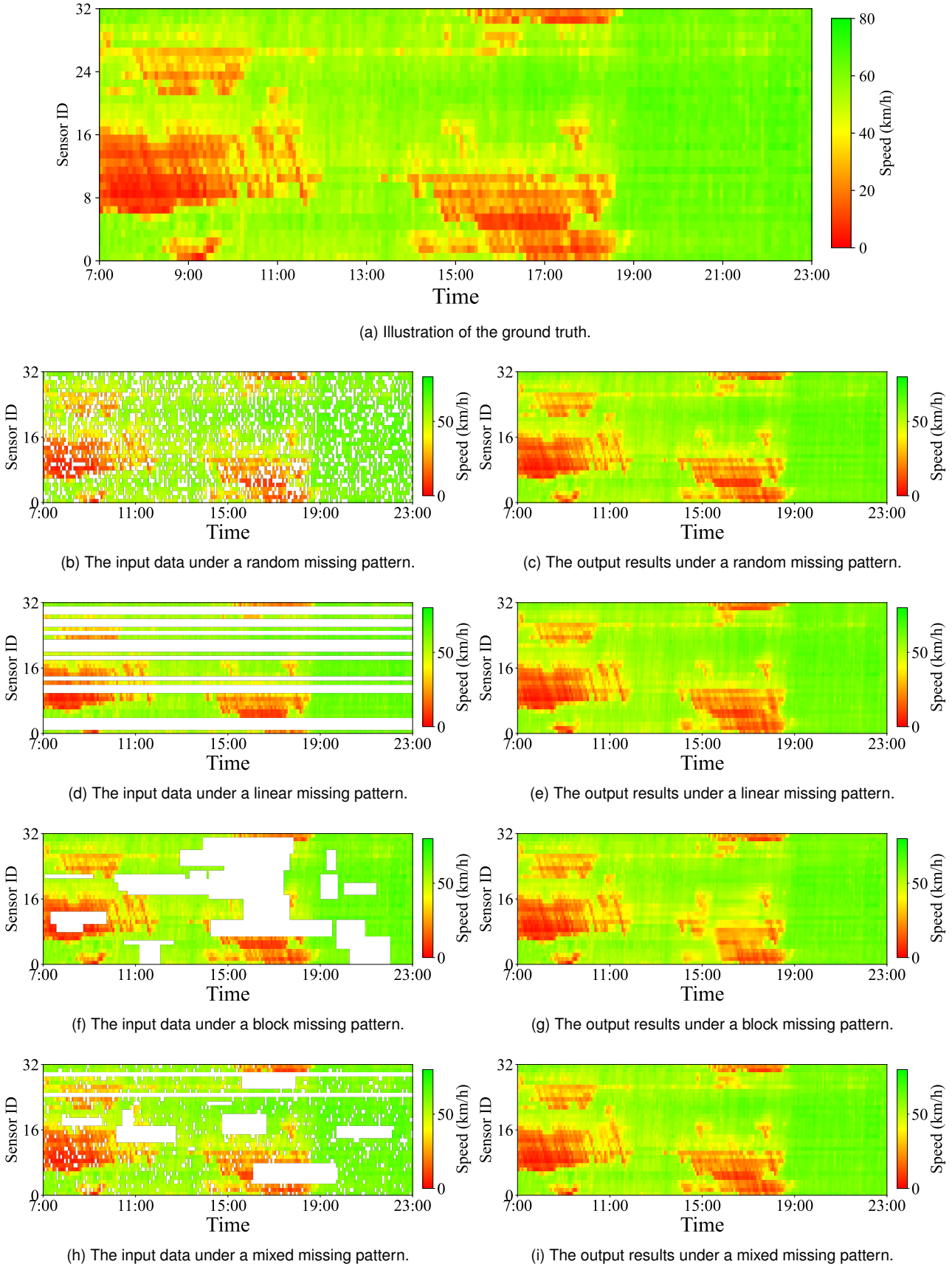
To enhance model generality, we integrate a physics-regularized loss function into the Swin Transformer by leveraging the fundamental diagram. This incorporation enhances the model's ability to learn beyond the training data, which ensures robust performance in previously unseen scenarios. Additionally, it guides and constrains the model to produce results that align with physical laws, which guarantees that the proposed PRMAN remains effective even when applied to limited and sparse datasets. In this section, we perform an ablation study to quantitatively assess and analyze the contribution of the physics-regularized loss.

Specifically, we test the model performance by adjusting the weight of physics-regularized loss $\lambda_p$ (0, 0.1, 0.2, 0.5, 1) under a 70% missing rate. The results are summarized in Table IV. A $\lambda_p$ value of 0 indicates that the physics-regularized loss is not employed in the model. As the value of $\lambda_p$ increases, the influence of the traffic domain knowledge on data imputation becomes more pronounced. We find that model with the physics-regularized loss ($\lambda_p > 0$) consistently outperform the model without it. Notably, for complex missing patterns (e.g., mixed) where observed data is limited, models with higher physical loss weights (e.g, $\lambda_p = 0.5$ and $\lambda_p = 1$) have a better performance, which aligns with the analysis above. To summarize, the physics-regularized loss significantly improves the accuracy and robustness of the model.

### F. Sensitivity analysis

In this section, we examine how the neural network's parameters, such as the local window size and the number of RSTB units, influence data imputation performance. The dataset utilized in this experiment is SanDiego015. We first test the performance of our model using local window sizes of 2, 4, 8, and 16, which control the granularity of feature extraction. In general, a small window size may limit the model's ability to capture global features, while an excessively large window size may introduce redundant information and impair the model's generalizability. The performance of our model under different local window sizes is summarized in Table V. We find that variations in the local window size do not significantly affect the MSE and MAPE, which suggests that the proposed PRMAN is robust with respect to this parameter. To achieve stable and satisfactory imputation performance, we recommend selecting a local window size of 4.

We further examine model performance with respect to the number of RSTBs and the number of STLs within each RSTB. In this experiment, the number of RSTBs is set to 3, 4, 5, 6, and 7, while the number of STLs in each RSTB is set to 4, 5, and 6. This implies that a total of 15 distinct configurations are employed to conduct the experiments. The utilized dataset is SanDiego015, with all missing data patterns taken into account. As shown in Fig. 9, we find that neither the number of RSTB nor the number of STL significantly affects the performance of our model. This suggests that our model is robust to these parameter variations, likely due to its hierarchical architecture, which enables effective extraction of both local and global features.




(a) Illustration of the ground truth.

(b) The input data under a random missing pattern.

(c) The output results under a random missing pattern.

(d) The input data under a linear missing pattern.

(e) The output results under a linear missing pattern.

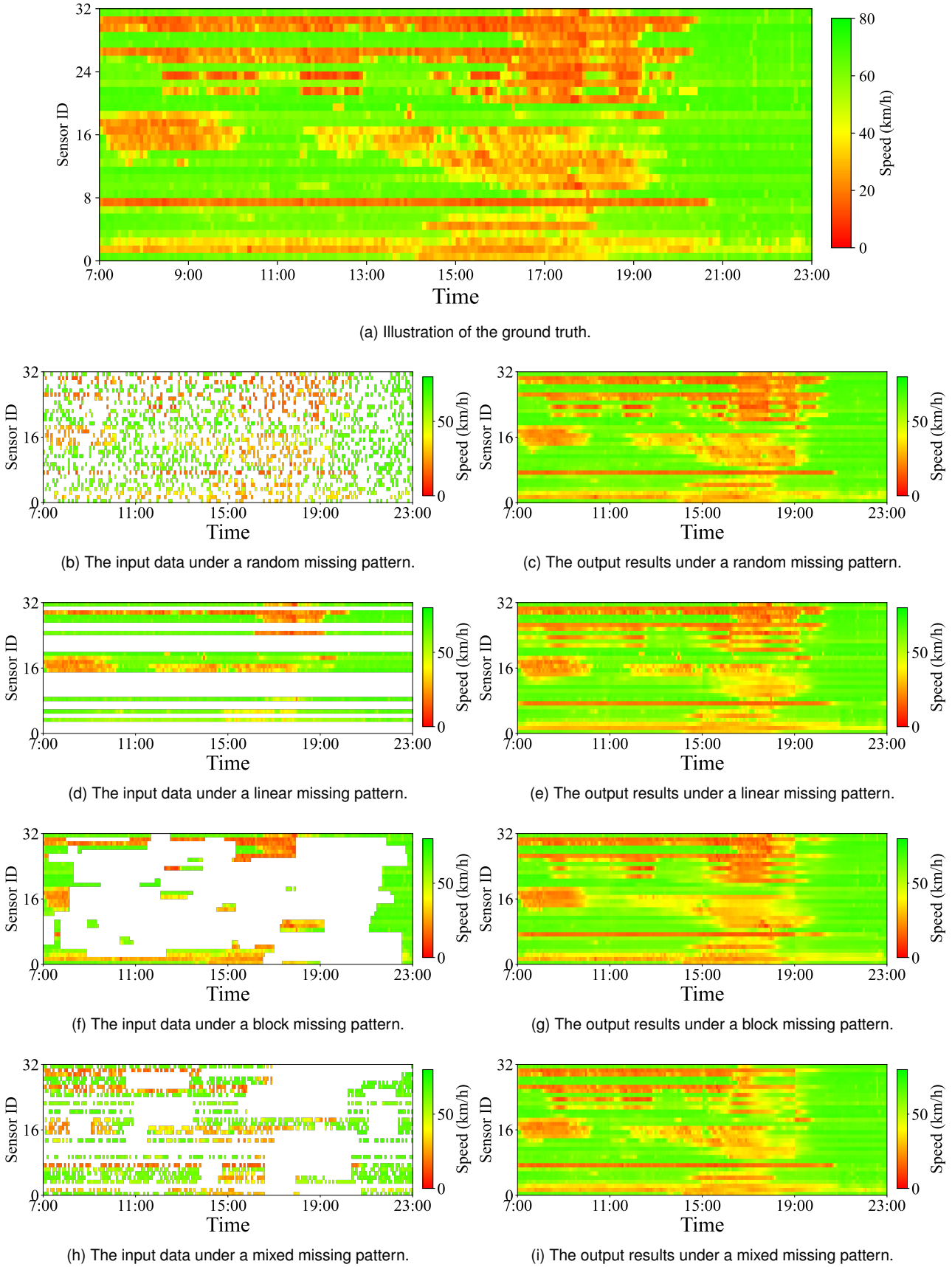
(f) The input data under a block missing pattern.

(g) The output results under a block missing pattern.

(h) The input data under a mixed missing pattern.

(i) The output results under a mixed missing pattern.

Fig. 7. Visualizations of ground truth, input, and output of our model under different missing patterns (30% missing rate).




(a) Illustration of the ground truth.

(b) The input data under a random missing pattern.

(c) The output results under a random missing pattern.

(d) The input data under a linear missing pattern.

(e) The output results under a linear missing pattern.

(f) The input data under a block missing pattern.

(g) The output results under a block missing pattern.

(h) The input data under a mixed missing pattern.

(i) The output results under a mixed missing pattern.

Fig. 8. Visualizations of ground truth, input, and output of our model under different missing patterns (70% missing rate).

TABLE III
THE PERFORMANCE (MSE/MAPE (%)) OF OUR PROPOSED MODEL AND THE BASELINE METHODS UNDER DIFFERENT MISSING RATES.

| Model | Type | 30% | | 40% | | 50% | | 60% | | 70% | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE |
| **PRMAN** | block | **30.24** | **10.81** | **35.97** | **10.88** | **37.73** | **11.68** | **41.76** | **11.44** | **50.58** | **13.61** |
| | linear | **43.01** | **14.31** | **50.43** | **13.13** | **51.18** | **15.25** | **66.38** | **14.93** | **75.15** | **17.28** |
| | random | **14.15** | **6.99** | **17.28** | **7.43** | **18.30** | **7.99** | **22.14** | **8.41** | **27.75** | **10.00** |
| | mixed | **25.75** | 8.49 | **31.89** | **9.44** | **35.33** | **12.44** | **40.76** | **11.39** | **47.74** | **14.92** |
| IF | block | 62.74 | 15.64 | 61.38 | 14.16 | 63.94 | 14.77 | 72.44 | 15.02 | 86.14 | 17.63 |
| | linear | 77.17 | 17.02 | 80.11 | 16.44 | 85.10 | 18.45 | 101.72 | 18.21 | 108.78 | 20.11 |
| | random | 46.04 | 12.48 | 52.20 | 12.13 | 56.79 | 12.54 | 65.99 | 13.66 | 82.25 | 15.72 |
| | mixed | 32.57 | **8.06** | 37.48 | 11.18 | 42.55 | 13.76 | 46.76 | 14.95 | 52.54 | 18.57 |
| LSTC | block | 53.45 | 16.87 | 53.05 | 15.35 | 56.18 | 16.91 | 64.11 | 16.46 | 72.80 | 18.38 |
| | linear | 150.93 | 26.10 | 143.17 | 23.00 | 146.67 | 25.69 | 177.14 | 25.89 | 189.61 | 28.85 |
| | random | 23.73 | 9.18 | 24.75 | 8.78 | 25.18 | 9.73 | 29.66 | 10.14 | 35.04 | 11.42 |
| | mixed | 63.29 | 14.86 | 66.49 | 14.17 | 73.39 | 17.94 | 85.77 | 16.93 | 96.52 | 21.43 |

TABLE IV
THE PERFORMANCE (MSE/MAPE (%)) OF OUR MODEL UNDER DIFFERENT $\lambda_p$ WITH A 70% MISSING RATE.

| Type | 0 | | 0.1 | | 0.2 | | 0.5 | | 1.0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE |
| block | 51.57 | 14.05 | 52.60 | 14.75 | 58.27 | 17.14 | **49.77** | **13.99** | 50.84 | 14.16 |
| linear | 75.53 | 17.11 | 77.38 | 18.11 | 79.70 | 19.78 | **71.95** | **16.92** | 72.91 | 17.27 |
| mixed | 35.53 | 13.54 | 36.99 | 13.97 | 37.68 | 14.23 | **31.46** | **12.65** | 32.82 | 13.09 |
| random | 28.03 | **10.12** | 30.27 | 11.15 | 34.38 | 12.73 | **27.81** | 10.26 | 27.83 | 9.98 |

TABLE V
THE PERFORMANCE (MSE/MAPE (%)) OF OUR MODEL UNDER DIFFERENT LOCAL WINDOW SIZES.

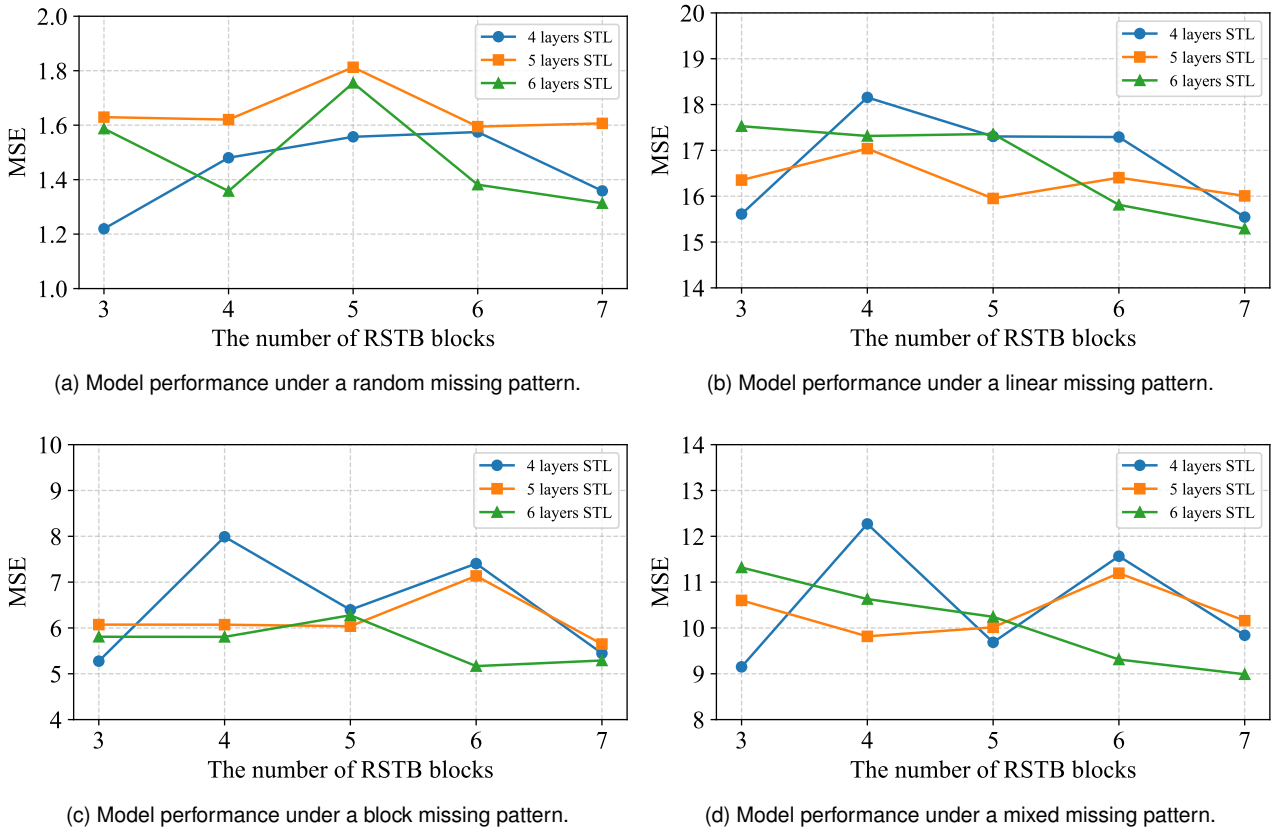
| Type | window size = 2 | | window size = 4 | | window size = 8 | | window size = 16 | |
|---|---|---|---|---|---|---|---|---|
| | MSE | MAPE | MSE | MAPE | MSE | MAPE | MSE | MAPE |
| block | 6.10 | 2.27 | **5.28** | **1.98** | 6.80 | 2.48 | 6.45 | 2.39 |
| linear | 16.53 | 4.27 | **15.61** | **3.86** | 15.50 | 4.08 | 17.52 | 4.26 |
| mixed | 10.22 | 3.05 | **9.15** | **2.75** | 10.47 | 3.11 | 12.23 | 3.36 |
| random | 1.53 | 1.30 | **1.22** | **1.11** | 1.52 | 1.29 | 1.63 | 1.36 |

### G. Extended discussion on our model

In this study, we mainly focus on developing accurate imputation methods to reconstruct the spatiotemporal traffic data. In real-world applications, our model can be further improved by investigating the following aspects. First, to improve the practical deployment of the proposed model in real-time applications, it is essential to consider not only algorithmic efficiency but also the integration of system-level computation and intelligent resource management. Latency-aware resource allocation techniques can help mitigate computational bottlenecks and enable real-time inference at the network edge [54]. Second, traffic data are often spatially skewed due to heterogeneous sensor deployments (e.g., urban vs. highway areas), and skew-aware strategies can enhance the model's scalability in such uneven environments [55]. Such strategies can also improve data quality and enhance the accuracy of data imputation. Finally, the proposed physics-regularized loss function primarily captures static traffic dynamics. To enhance the model's responsiveness to sudden changes in traffic patterns and bursty congestion events, adaptive resource allocation methods may be incorporated as a complementary mechanism [56]. These system-level strategies represent valuable extensions to our current framework and will be explored in future work.

## VII. CONCLUSIONS

In this study, we develop a physics-regularized multiscale attention network to estimate the missing values from partially observed data. The contributions of our proposed PRMAN come from two aspects. First, the PRMAN introduces a novel hierarchical architecture that extracts spatiotemporal dependencies of traffic data in multiple scales, which is new to the literature. The proposed PRMAN does not rely on global self-attention that computes the relationships between all tokens. Instead, it performs self-attention within partitioned local windows. By progressively expanding the window size across layers, this hierarchical design effectively captures both local and global dependencies inherent in traffic data at different scales, while accounting for their distinct characteristics. Second, we propose a physics-regularized loss function based on the fundamental diagram to enforce consistency between the reconstructed data and the underlying spatiotemporal traffic dynamics. The incorporation of traffic domain knowledge enhances the model's generalization ability beyond the training data, ensuring robust performance in unseen scenarios. We conduct extensive numerical experiments on six benchmark datasets to evaluate the model performance in handling diverse and complex missing data patterns. The results demonstrate that the proposed PRMAN consistently achieves state-of-the-




(a) Model performance under a random missing pattern.

(b) Model performance under a linear missing pattern.

(c) Model performance under a block missing pattern.

(d) Model performance under a mixed missing pattern.

Fig. 9. Impact of RSTB and STL configurations on the model performance.

art performance compared to existing imputation methods, including Transformer-based approaches and tensor decomposition-based techniques.

We outline the future research directions as follows:

- To enhance the scalability and applicability of PRMAN, it is of great interest to explore transfer learning techniques that adapt a pre-trained model from one road network to another with minimal fine-tuning. This is particularly useful when high-quality labeled data are scarce in the target domain. Incorporating domain adaptation strategies could further address distributional shifts in spatiotemporal traffic patterns across different scenarios; and
- To enhance the robustness of PRMAN, another promising direction is to extend it into a unified framework that simultaneously removes corrupted noises and imputes missing values. However, this problem is particularly challenging due to the limited information available from observed data with irregular measurement errors. To address this, incorporating multi-source data, such as points of interest, population density, and network topology may provide a feasible solution.

## ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China [Grant 72101012, 72394362&72394363/72394360]. (Corresponding authors: Zhengli Wang; Wei Ma).

## APPENDIX

The mathematical equations used in the proposed model are detailed as follows.

### A. 2D Convolution (Conv2d)

$$
\begin{aligned}
&\mathcal{X} \in \mathbb{R}^{N \times C_{\text{in}} \times H \times W}; \quad \text{(input)} \\
&\mathcal{K} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_h \times k_w}; \quad \text{(kernel weights)} \\
&\mathcal{B} \in \mathbb{R}^{C_{\text{out}}}; \quad \text{(bias)} \\
&\text{padding} \in \mathbb{N}; \text{(Padding added to both sides of the input)} \\
&\text{stride} \in \mathbb{N}; \text{(Stride of the convolution)} \\
&H_{\text{out}} = \left\lfloor \frac{H + 2\text{padding} - k_h}{\text{stride}} \right\rfloor + 1; \quad \text{(output dimension)} \\
&W_{\text{out}} = \left\lfloor \frac{W + 2\text{padding} - k_w}{\text{stride}} \right\rfloor + 1; \quad \text{(output dimension)} \\
&\mathcal{Y}_{n,c,i,j} = \sum_{c'=0}^{C_{\text{in}}-1} \sum_{p=0}^{k_h-1} \sum_{q=0}^{k_w-1} \mathcal{X}^{n,c'}_{i \cdot s+p-p_{\text{pad}}, j \cdot s+q-p_{\text{pad}}} \\
&\cdot \mathcal{K}_{c,c',p,q} + \mathcal{B}_c. \quad \text{(output)}
\end{aligned}
\tag{21}
$$

### B. Layer Normalization (LN)

$$
\begin{aligned}
&\mathcal{X} \in \mathbb{R}^{N \times C \times H \times W}; \quad \text{(input)} \\
&\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{C}; \quad \text{(affine parameters)} \\
&\mu_{n,h,w} = \frac{1}{C} \sum_{c=1}^{C} \mathcal{X}_{n,c,h,w}; \\
&\sigma^2_{n,h,w} = \frac{1}{C} \sum_{c=1}^{C} \left(\mathcal{X}_{n,c,h,w} - \mu_{n,h,w}\right)^2; \\
&\hat{\mathcal{X}}_{n,c,h,w} = \frac{\mathcal{X}_{n,c,h,w} - \mu_{n,h,w}}{\sqrt{\sigma^2_{n,h,w} + \epsilon}}; \\
&\mathcal{Y}_{n,c,h,w} = \gamma_c \cdot \hat{\mathcal{X}}_{n,c,h,w} + \beta_c. \quad \text{(output)}
\end{aligned}
\tag{22}
$$

### C. Multilayer Perceptron (MLP)

$$
\begin{aligned}
&\mathcal{X} \in \mathbb{R}^{d_{\text{in}}}; \quad \text{(input)} \\
&\mathcal{W}^{(1)} \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{in}}}, \mathcal{B}^{(1)} \in \mathbb{R}^{d_{\text{hid}}}; \\
&\mathcal{W}^{(2)} \in \mathcal{R}^{d_{\text{out}} \times d_{\text{hid}}}, \mathcal{B}^{(2)} \in \mathbb{R}^{d_{\text{out}}}; \\
&\mathcal{H} = \text{ReLU}\left(\mathcal{W}^{(1)}\mathcal{X} + \mathcal{B}^{(1)}\right) \quad \text{(Rectified Linear Unit)}; \\
&\mathcal{Y} = \mathcal{W}^{(2)}\mathcal{H} + \mathcal{B}^{(2)}. \quad \text{(output)}
\end{aligned}
\tag{23}
$$

### D. Softmax Function (softmax)

$$
\begin{aligned}
&\mathcal{Z} \in \mathbb{R}^{K}; \quad \text{(input)} \\
&\mathcal{P} \in \mathbb{R}^{K}; \quad \text{(output)} \\
&\mathcal{P}_i = \frac{\exp(\mathcal{Z}_i)}{\sum_{j=1}^{K} \exp(\mathcal{Z}_j)} \quad \text{for } i = 1, 2, \ldots, K.
\end{aligned}
\tag{24}
$$

## REFERENCES


[1] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, p. 1 – 29, 2009, cited by: 360; All Open Access, Green Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-67651030467&doi=10.1145%2f1525856.1525863&partnerID=40&md5=24fd74d2a299b7e24e6e0675d628dfc1

[2] X.-Y. Liu and X. Wang, "Ls-decomposition for robust recovery of sensory big data," *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 542–555, 2018.

[3] Z. Zheng, Z. Wang, Z. Hu, Z. Wan, and W. Ma, "Recovering traffic data from the corrupted noise: A doubly physics-regularized denoising diffusion model," *Transportation Research Part C: Emerging Technologies*, vol. 160, p. 104513, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X24000342

[4] L. Qu, L. Li, Y. Zhang, and J. Hu, "Ppca-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 512–522, 2009.

[5] H. Li, M. Li, X. Lin, F. He, and Y. Wang, "A spatiotemporal approach for traffic data imputation with complicated missing patterns," *Transportation Research Part C: Emerging Technologies*, vol. 119, p. 102730, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20306458

[6] R. Rahman and S. Hasan, "Short-term traffic speed prediction for freeways during hurricane evacuation: A deep learning approach," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1291–1296.

[7] X. Chen and L. Sun, "Bayesian temporal factorization for multidimensional time series prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4659–4673, 2022.

[8] X. Chen, C. Zhang, X.-L. Zhao, N. Saunier, and L. Sun, "Nonstationary temporal matrix factorization for multivariate time series forecasting," *arXiv preprint arXiv:2203.10651*, 2022.

[9] X. Chen, X.-L. Zhao, and C. Cheng, "Forecasting urban traffic states with sparse data using hankel temporal matrix factorization," *INFORMS Journal on Computing*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272713976

[10] W. Velicer and S. Colby, "A comparison of missing-data procedures for arima time-series analysis," *Educational and Psychological Measurement*, vol. 65, pp. 596–615, 08 2005.

[11] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling financial time series with S-PLUS®*, pp. 385–429, 2006.

[12] X. Chen, Z. He, and J. Wang, "Spatial-temporal traffic speed patterns discovery and incomplete data recovery via svd-combined tensor decomposition," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 59–77, 2018.

[13] X. Chen, Z. He, and L. Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 73–84, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X1830799X

[14] Y. Liang, Z. Zhao, and L. Sun, "Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns," *Transportation Research Part C: Emerging Technologies*, vol. 143, p. 103826, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22002479

[15] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transportation Research Part C*, vol. 72, pp. 168–181, 2016.

[16] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *CoRR*, vol. abs/2003.05991, 2020. [Online]. Available: https://arxiv.org/abs/2003.05991

[17] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[18] Y. Chen and X. M. Chen, "A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 143, p. 103820, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22002431

[19] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423001203

[20] T. Nie, G. Qin, W. Ma, Y. Mei, and J. Sun, "Imputeformer: Low rankness-induced transformers for generalizable spatiotemporal imputation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2260–2271. [Online]. Available: https://doi.org/10.1145/3637528.3671751

[21] T. Wang, J. Chen, J. Lü, K. Liu, A. Zhu, H. Snoussi, and B. Zhang, "Synchronous spatiotemporal graph transformer: A new framework for traffic data prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 589–10 599, 2023.

[22] K. Zhang, F. Zhou, L. Wu, N. Xie, and Z. He, "Semantic understanding and prompt engineering for large-scale traffic data imputation," *Information Fusion*, vol. 102, p. 102038, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523003548

[23] X. Chen, Z. Cheng, H. Cai, N. Saunier, and L. Sun, "Laplacian convolutional representation for traffic time series imputation," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[24] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, ser. CIKM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 4125–4129. [Online]. Available: https://doi.org/10.1145/3583780.3615160

[25] P. Wang, T. Zhang, Y. Zheng, and T. Hu, "A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation," *International Journal of Geographical Information Science*, vol. 36, pp. 1–27, 02 2022.

[26] A. Aggarwal, "Atmospheric delay correction of rinex gps data," *Test Engineering and Management*, pp. 24 877–24 882, 2020.

[27] A. Kumar, "Geo-tagged 3d geometric modeling of urban structures by mitigating reflected gps signals using a laser range sensor," in *Science and Information Conference*. Springer, 2024, pp. 396–414.

[28] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C Emerging Technologies*, vol. 62, pp. 21–34, 01 2016.

[29] C. F. Ansley and R. Kohn, "On the estimation of arima models with missing values," in *Time Series Analysis of Irregularly Observed Data*, E. Parzen, Ed. New York, NY: Springer New York, 1984, pp. 9–37.

[30] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning. 2001," *Journal of the Royal Statistical Society*, vol. 167, no. 1, pp. 192–192, 2004.

[31] H. Zou, Y. Yue, Q. Li, and A. Yeh, "An improved distance metric for the interpolation of link-based traffic data using kriging:a case study of a large-scale urban road network," *International Journal of Geographical Information Science*, vol. 26, no. 4, pp. 667–689, 2012.

[32] B. Shamo, E. Asa, and J. Membah, "Linear spatial interpolation and analysis of annual average daily traffic data," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, p. 04014022, 2015.

[33] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 15–27, 2013, euro Transportation: selected paper from the EWGT Meeting, Padova, September 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X12001532

[34] X. Jia, X. Dong, M. Chen, and X. Yu, "Missing data imputation for traffic congestion data based on joint matrix factorization," *Knowledge-Based Systems*, vol. 225, p. 107114, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121003774

[35] M. Lei, A. Labbe, Y. Wu, and L. Sun, "Bayesian kernelized matrix factorization for spatiotemporal traffic data imputation and kriging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 962–18 974, 2022.

[36] X. Chen, Y. Chen, N. Saunier, and L. Sun, "Scalable low-rank tensor learning for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103226, 08 2021.

[37] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, 04 2018.

[38] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624–1630, 2020.

[39] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[40] X. Kong, W. Zhou, G. Shen, W. Zhang, N. Liu, and Y. Yang, "Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data," *Knowledge-Based Systems*, vol. 261, p. 110188, 12 2022.

[41] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 10, pp. 5388–5408, 2023.

[42] W. Liang, Y. Li, K. Xie, D. Zhang, K.-C. Li, A. Souri, and K. Li, "Spatial-temporal aware inductive graph neural network for c-its data recovery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8431–8442, 2023.

[43] 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 Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7919–7930, 2022.

[44] B. Yang, Y. Kang, Y. Yuan, X. Huang, and H. Li, "St-lbagan: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation," *Knowledge-Based Systems*, vol. 215, p. 106705, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120308340

[45] Z. He, L. Zheng, P. Chen, and W. Guan, "Mapping to cells: A simple method to extract traffic dynamics from probe vehicle data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 3, pp. 252–267, 2017.

[46] K. Zhang, X. Feng, N. Jia, L. Zhao, and Z. He, "TSR-GAN: Generative adversarial networks for traffic state reconstruction with time space diagrams," *Physica A: Statistical Mechanics and its Applications*, vol. 591, p. 126788, 2022.

[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[48] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[50] 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 Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7919–7930, 2021.

[51] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[52] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[53] Y. Fan, X. Yu, R. Wieser, D. Meakin, A. Shaton, J.-N. Jaubert, R. Flottemesch, M. Howell, J. Braid, L. Bruckman *et al.*, "Spatio-temporal denoising graph autoencoders with data augmentation for photovoltaic data imputation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–19, 2023.

[54] H. Ahmadvand and F. Foroutan, "Latency and privacy-aware resource allocation in vehicular edge computing," *arXiv preprint arXiv:2501.02804*, 2025.

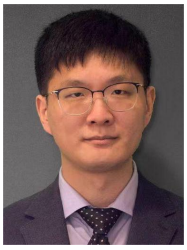
[55] H. Ahmadvand, T. Dargahi, F. Foroutan, P. Okorie, and F. Esposito, "Big data processing at the edge with data skew aware resource allocation," in *2021 IEEE conference on network function virtualization and software defined networks (NFV-SDN)*. IEEE, 2021, pp. 81–86.

[56] H. Ahmadvand and F. Foroutan, "Dv-arpa: data variety aware resource provisioning for big data processing in accumulative applications," *arXiv preprint arXiv:2008.04674*, 2020.

**Zhenjie Zheng** obtained the bachelor's degree in Industrial Engineering from Huazhong University of Science and Technology, China, and the Ph.D. degree in Industrial Engineering from Tsinghua University, China. He is currently a postdoctoral fellow with the Department of Civil and Environmental Engineering, the Hong Kong Polytechnic University. His research interests include machine learning in transportation management, AI-powered network-wide traffic state estimation, data-driven city emergency detection and management, and LoT-based intelligent traffic control.

**Yulin He** obtained the bachelor's degree in Management Science from Huazhong University of Science and Technology, China. He is currently pursuing his master's degree at the School of Management and Engineering, Nanjing University, China. His research interests include machine learning in transportation management and AI-powered traffic state estimation.

**Zhengli Wang** obtained the bachelor's degree in Management Science from Beijing Normal University, China, and the Ph.D. degree in Industrial Engineering from Tsinghua University, China. He is currently an Associate Professor with the School of Management and Engineering, the Nanjing University, China. His research interests include data-driven modeling and optimization of complex traffic systems, machine learning for smart urban traffic management, traffic state estimation with physical insights, and spatiotemporal analysis of traffic incident impacts.



**Wei Ma** (IEEE member) obtained the bachelor's degrees in Civil Engineering and Mathematics from Tsinghua University, China, master degrees in Machine Learning and Civil and Environmental Engineering, and PhD degree in Civil and Environmental Engineering from Carnegie Mellon University, USA. He is currently an assistant professor with the Department of Civil and Environmental Engineering at the Hong Kong Polytechnic University (PolyU). His research focuses on intersection of machine learning, data mining, and transportation network modeling, with applications for smart and sustainable mobility systems. Dr. Ma serves as the Associate Editor of IEEE T-ITS and OJ-ITS.