ImputeFormer: Low Rankness-Induced Transformers for Generalizable Spatiotemporal Imputation

Tong Nie[1], Guoyang Qin[1], Wei Ma[2], Yuewen Mei[1], Jian Sun[1,†]
[1]Tongji University, [2]The Hong Kong Polytechnic University
nietong@tongji.edu.cn, 2015qgy@tongji.edu.cn, wei.w.ma@polyu.edu.hk,
meiyuewen@tongji.edu.cn, sunjian@tongji.edu.cn

ABSTRACT

Missing data is a pervasive issue in both scientific and engineering tasks, especially for the modeling of spatiotemporal data. This problem attracts many studies to contribute to data-driven solutions. Existing imputation solutions mainly include low-rank models and deep learning models. The former assumes general structural priors but has limited model capacity. The latter possesses salient features of expressivity but lacks prior knowledge of the underlying spatiotemporal structures. Leveraging the strengths of both two paradigms, we demonstrate a low rankness-induced Transformer to achieve a balance between strong inductive bias and high model expressivity. The exploitation of the inherent structures of spatiotemporal data enables our model to learn balanced signal-noise representations, making it generalizable for a variety of imputation problems. We demonstrate its superiority in terms of accuracy, efficiency, and versatility in heterogeneous datasets, including traffic flow, solar energy, smart meters, and air quality. Promising empirical results provide strong conviction that incorporating time series primitives, such as low-rankness, can substantially facilitate the development of a generalizable model to approach a wide range of spatiotemporal imputation problems. The model implementation is available at: https://github.com/tongnie/ImputeFormer.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**.

KEYWORDS

Missing Data, Data Imputation, Transformers, Low-Rank Modeling, Spatiotemporal Data, Time Series.

ACM Reference Format:
Tong Nie, Guoyang Qin, Wei Ma, Yuewen Mei, and Jian Sun. 2024. ImputeFormer: Low Rankness-Induced Transformers for Generalizable Spatiotemporal Imputation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/xx

1 INTRODUCTION

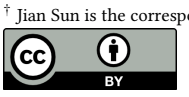
Missing data is a common challenge in detection systems, especially in high-resolution monitoring systems. Factors such as inclement weather, energy supply, and sensor service time can adversely affect the quality of monitoring data [5]. Given these factors, data missing rates can be quite high. For example, the air quality measurements in the Urban Air project [48] contain about 30% invalid records due to station malfunction. Similarly, subsets of Uber Movement data, such as New York City, have an approximate 85% of missing records. This problem encourages researchers to develop advanced models that can exploit limited observations to impute missing values. Extensive research has contributed to data-driven methods for this purpose, especially in the field of spatiotemporal data [5, 6, 28, 30, 40, 44].

Generally, there are two research paradigms for imputing missing data. The first paradigm uses low-rank and low-dimensional analytical models, such as [4, 6, 20, 28, 40], which assumes the data has a well-structured matrix or tensor form. These models utilize the algebraic properties of the assumed structure, such as low-rankness, low nuclear norm, and spectrum sparsity (we use the term "low-rank" as a proxy), to impute missing values [25]. While simple low-rank models like matrix factorization and tensor completion can effectively handle incomplete data, they may struggle with capturing complex patterns, such as nonlinearity and nonstationarity. With strong inductive bias, these models can excessively smooth the reconstructed data, filtering out informative signals, and generating oversmoothing reconstructions in some cases [27].

The second paradigm uses deep learning-based imputation models. These models learn the dynamics of the data-generating process and demonstrate improved performance [1, 2, 7, 10, 19, 26]. However, despite the success of these models on various benchmarks, there are still costs and difficulties that need further attention. First, such data-intensive methods require a substantial amount of training expenses due to the complex model structures in deep learning, such as probabilistic diffusion and bidirectional recurrence [1, 36]. This can consume significant computational memory and resources, making them less efficient for real-time deployment. Second, empirical loss-based learning methods, without the guidance of physics or data structures, are prone to overfitting and perform poorly when applied to tasks fall outside of the distribution of training data [47].

With the shift of focus in the field of deep imputation from RNNs and diffusion models to Transformers [10, 24, 26], Transformer-related architectures have gained significant attention due to their potential to provide efficient generative output and high expressivity, enabling more effective imputations compared to autoregression-based models. Additionally, Transformers are considered foundational architectures for general time series forecasting [11]. However, the effectiveness of applying Transformers to general data imputation tasks requires further investigation. Modern deep learning techniques associated with these architectures, such as self-attention and residual connections, can unintentionally preserve

† Jian Sun is the corresponding author.



KDD '24, August 25–29, 2024, Barcelona, Spain.

ACM ISBN xx.
https://doi.org/xx

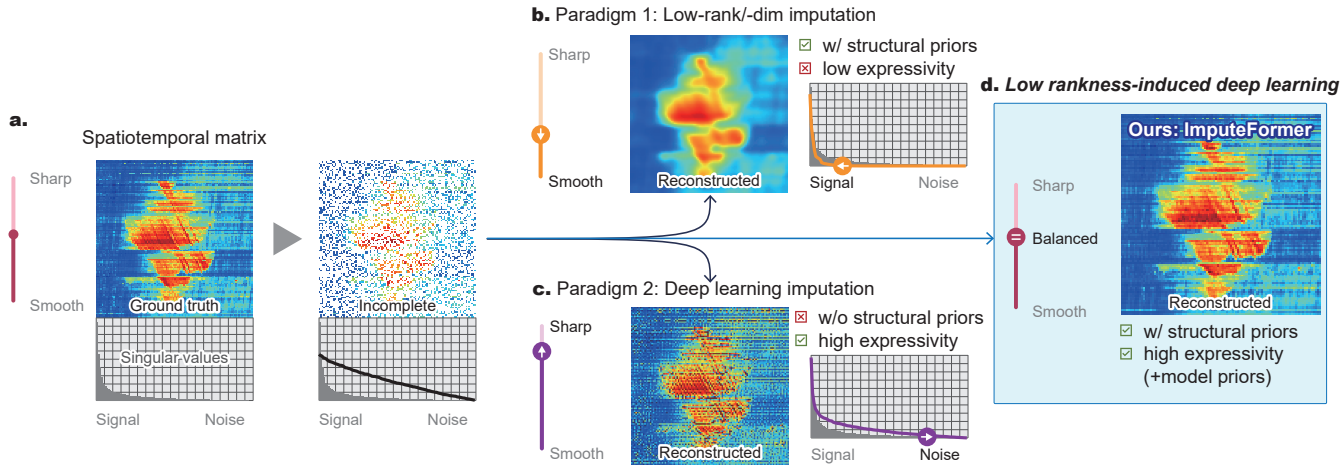


Figure 1: (a) The distribution of singular values in spatiotemporal data is long-tailed. The existence of missing data can increase its rank (or singular values). (b) Low-rank models can filter out informative signals and generate a smooth reconstruction, resulting in truncating too much energy in the left part of its spectrum. (c) Deep models can preserve high-frequency noise and generate sharp imputations, maintaining too much energy for the right part of the singular spectrum. With the generality of low-rank models and the expressivity of deep models, `ImputeFormer` achieves a signal-noise balance for accurate imputation.

high-frequency noise in data as informative signals [34]. This can lead the model to learn high-rank representations that violate the natural distribution of data. Furthermore, the existence of missing data can introduce spurious correlations between "tokens," posing challenges to these architectures. Considering the above-mentioned concerns, incorporating a low-rank inductive bias into the Transformer framework seems to provide a chance to improve both the effectiveness and efficiency in spatiotemporal imputation.

In summary, matrix- and tensor-based models offer useful priors for spatiotemporal data, such as low-rankness and sparsity. However, their ability to represent data is limited (see Fig. 1(b)). On the other hand, deep learning models, particularly Transformers, excel at learning representations but lack prior knowledge of data generation (see Fig. 1(c)). As the demand for a versatile and adaptable model that can handle various imputation problems in reality increases, such as cross-domain datasets, different observation conditions, highly sparse measurements, and different input patterns, it becomes apparent that existing advanced solutions, typically evaluated on limited tasks with simple settings, may not be generalizable. Hence, there is a temptation to merge these two paradigms and utilize their respective strengths to investigate an alternative paradigm that can effectively handle complex imputation scenarios.

To this end, in this paper we leverage the structural priors of low-rankness to generalize the canonical Transformer (see Fig. 1(d)) in general spatiotemporal imputation tasks. Our approach, referred to as *Imputation Transformers* (ImputeFormer), imposes low-rankness and achieves attention factorization equivalently by introducing a projected attention mechanism on the temporal dimension and an embedded attention on the spatial dimension. Additionally, we propose a Fourier sparsity loss to regularize the solution's spectrum. By inheriting the merits of both low-rank and deep learning models, it has achieved state-of-the-art imputation performance on various benchmarks. Our main contributions are summarized as follows:

(1) We are among the first to empower Transformers with low-rankness inductive bias to achieve a balance between signal and noise for general spatiotemporal data imputation;
(2) Compared to state-of-the-art benchmark models, we demonstrate the advantages of `ImputeFormer` in accuracy, efficiency, and versatility in diverse datasets, such as traffic flow, solar energy, electricity consumption, and air quality;
(3) Comprehensive case studies reveal the model's interpretability and provide insights into the deep imputation paradigm.

2 PRELIMINARY

Notations. This section first introduces some basic notations following [26]. In a continuously working sensor system with N static detectors at some measurement positions, spatiotemporal data with context information can be obtained: (1) $\mathbf{X}_{t:t+T} \in \mathbb{R}^{N\times T}$: The observed data matrix containing missing values collected by all sensors over a time interval $\mathcal{T} = \{t, \dots, t+T\}$, where T represents the observation period; (2) $\mathbf{Y}_{t:t+T} \in \mathbb{R}^{N\times T}$: The ground truth data matrix used for evaluation; (3) $\mathbf{U}_{t:t+T} \in \mathbb{R}^{T\times d_{\mathrm{u}}}$: Exogenous variables describe time series, such as the time of day, day of week, and week of month information; (4) $\mathbf{V} \in \mathbb{R}^{N\times d_{\mathrm{v}}}$: Meta information of sensors, such as detector ID and location of installation.

Problem Formulation. The multivariate time series imputation problem defines an inductive learning and inference process:

$$\begin{aligned} \texttt{Learning}\ \widehat{\Theta} &= \arg\min_{\Theta} \sum_{t\in\mathcal{T}} \ell\left(\texttt{NN}(\{\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t\}, \mathbf{V}|\Theta), \mathbf{x}_t\right), \\ \texttt{Inference}\ \widehat{\mathbf{x}}_{t'} &= \texttt{NN}(\{\mathbf{x}_{t'}, \mathbf{u}_{t'}, \mathbf{m}_{t'}\}, \mathbf{V}|\widehat{\Theta}), \forall\{t' \dots t'+T\}, \end{aligned} \tag{1}$$

where $\texttt{NN}(\cdot|\Theta)$ is the neural network model parameterized by Θ, and the indicator $\mathbf{m}_t$ denotes the locations of the masked values for training and the locations of the observed values for inference. After training the model on observed data, the imputation model can act in a different time span than the training set.

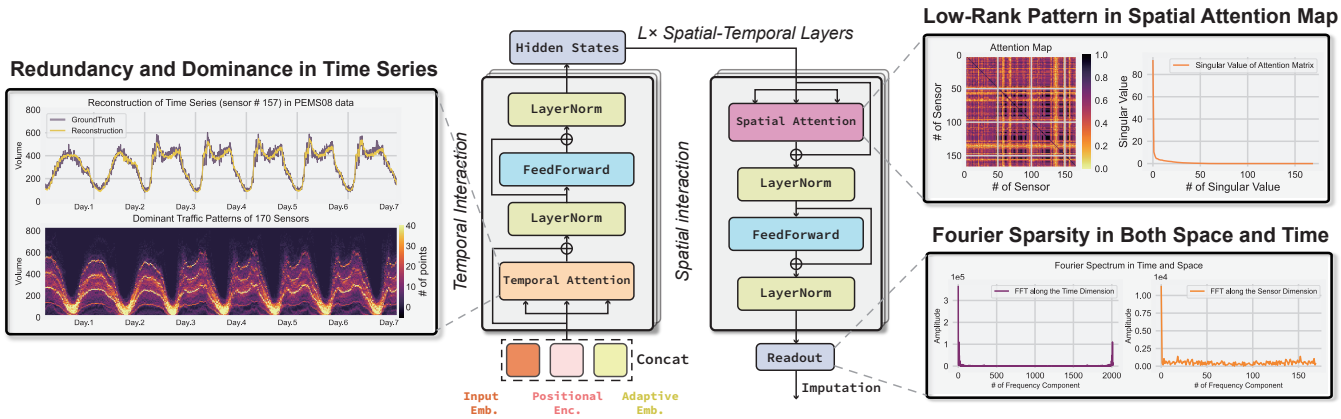


Figure 2: Low-rankness in time series and the induced `ImputeFormer`. (a) Redundancy in time series: PEMS08 data can be reasonably reconstructed using only five dominant patterns. (b) Low-rank spatial attention map: the singular values of the multivariate attention map show a long-tailed distribution and most of them are small values. (c) Fourier sparsity in both space and time axes: both the spatial and temporal signals possess a sparse Fourier spectrum, with most amplitudes close to zero.

3 RELATED WORK

Generally, there exist two series of studies on multivariate time series imputation, i.e., 1) low-dimensional/rank models and 2) deep imputation models. We particularly discuss existing Transformer-based solutions to clarify the connections between our models.

Low-Dimensional/Rank Imputation. Early methods addressed the data imputation problem by exploring statistical interpolation tools, such as `MICE` [37]. Recently, low-rank matrix factorization [20, 46] and tensor completion [4–6, 28, 30] have emerged as numerically efficient techniques for spatiotemporal imputation. To incorporate series-related features, `TRMF` [46] imposed autoregressive regularization on the temporal manifold. `TiDER` [20] decomposed the time series into trend, seasonality and bias components under the factorization framework. Despite being conceptually intuitive and concise, limited capacity hinder their practical effectiveness.

Deep Learning Imputation. Recent advances in neural time series analysis open a new horizon to improve imputation performance. Generally, deep imputation methods learn to reconstruct the distribution of observed data or aggregate pointwise information progressively [12]. Representative methods include `GRU-D` [2], `GRUI` [22], `BRITS` [1], `GAIN` [45], `E2GAN` [23] `NAOMI` [21], `CSDI` [36], and `PriSTI` [19]. To exploit the multivariate nature of spatiotemporal data, graph neural networks (GNNs) have been adopted to model sensor-wise correlations for more complicated missing patterns. For example, `MDGCN` [15] and `GACN` [44] applied GNNs with RNNs for traffic data imputation. `IGNNK` [42], `STAR` [14], and `STCAGCN` [31] further tackle the kriging problem, which is a special data imputation scenario. As a SOTA model and an architectural template for GNN-RNNs, `GRIN` [7] based on message-passing GRUs that progressively performed a two-stage forward and backward recurrent message aggregation with predefined relational biases.

Transformers for Time Series Imputation. Transformers [38] can aggregate abundant information from arbitrary input elements, becoming a natural choice for sequential data imputation. In particular, `CDSA` [24] developed a cross-channel attention that utilizes correlations in different dimensions. `SAITS` [10] combined the masked imputation task with an observed reconstruction task, and applied a diagonally-masked self-attention to hierarchically reconstruct sparse data. `SPIN` [26] achieved SOTA imputation performance by conducting a sparse cross-attention and a temporal self-attention on all observed spatiotemporal points. However, the high complexity of cross-attention hinders its application in larger graphs.

4 LOW RANKNESS-INDUCED TRANSFORMER

This section elaborates the `ImputeFormer` model. The major difference between our model and the canonical Transformer is the integration of low-rank factorization. Unlike GNNs, `ImputeFormer` does not require a predefined graph due to the global adaptive interaction between series. It also bypasses the use of intricate temporal techniques, such as bidirectional recurrent aggregation, sparse cross-attention, and attention masking. Furthermore, it achieves linear complexity with respect to spatial and temporal dimensions.

4.1 Architectural Overview

The overall structure of the proposed `ImputeFormer` is shown in Fig. 2. The input embedding layer projects sparse observations to hidden states in an additional dimension and introduces both fixed and learnable embedding into the inputs. Following a time-and-graph template [8], TemporalInteraction and SpatialInteraction perform global message passing alternatively at all spatiotemporal coordinates. Finally, a `MLP` readout is adopted to output the final imputation. This process can be summarized as follows:

$$
\begin{aligned}
\mathcal{Z}_{t:t+T}^{(0)} &= \texttt{InputEmb}(\mathbf{X}_{t:t+T}, \mathbf{U}_{t:t+T}, \mathbf{V}), \\
\mathcal{Z}_{t:t+T}^{(\ell+1)} &= \texttt{TemporalInteraction}(\mathcal{Z}_{t:t+T}^{(\ell)}), \\
\mathcal{Z}_{t:t+T}^{(\ell+1)} &= \texttt{SpatialInteraction}(\mathcal{Z}_{t:t+T}^{(\ell+1)}), \forall \ell \in \{0, \dots L\}, \\
\widehat{\mathbf{X}}_{t:t+T} &= \texttt{Readout}(\mathcal{Z}_{t:t+T}^{(L+1)}).
\end{aligned}
\tag{2}
$$

The canonical Transformer block [38] can be adopted to gather spatial-temporal information for imputation. However, we argue that directly applying it to the imputation problem is questionable,

and there exist three key concerns: (1) *Spurious correlations*: Short-term series within a window can be noisy and indistinguishable. Modeling relational structures using sparse input can cause spurious and misleading correlations. (2) *High-rank estimations*: Time series are typically low-rank in nature [16]. Full-attention computation on raw data can be overcorrelated and generate high-rank attention maps. (3) *Scalability issue*: All pairwise attention on large graphs is memory intensive and computationally inefficient. To address these issues, we start from time series primitives and enhance the Transformer using these structural priors.

4.2 Spatiotemporal Input Embedding

Input Embedding. We adopt a dimension expansion strategy [43] to preserve the information density of the incomplete time series. In practice, we expand an additional dimension of the input and project it into a hidden state along this new dimension:

$$\mathcal{Z}^{(0)}_{t:t+T} = \texttt{MLP}(\texttt{Unsqueeze}(\mathbf{X}_{t:t+T}, \text{dim=-1}), \tag{3}$$

where $\texttt{Unsqueeze}(\cdot) : \mathbb{R}^{N\times T} \to \mathbb{R}^{N\times T\times 1}$, and $\mathcal{Z}^{(0)}_{t:t+T} \in \mathbb{R}^{N\times T\times D}$ is the initial hidden representation. With this, we can aggregate message from other time points by learning data-dependent weights:

$$\mathbf{Z}^{i,(\ell+1)}_{t:t+T} = \mathcal{F}_\ell(\mathbf{Z}^{i,(\ell)}_{t:t+T})\mathbf{Z}^{i,(\ell)}_{t:t+T}, \tag{4}$$

where $\mathcal{F}_\ell(\cdot) : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times T}$ represents a data-driven function at the ℓ-th layer, such as self-attention. The rationale of this strategy is discussed in A.1.2.

Time Stamp Encoding. Time stamp encoding is adopted to handle the order-agnostic nature of Transformers [38]. As the input series covers a relatively short range, we only consider the time-of-day information. We adopt the sinusoidal positional encoding in [38] to inject the time-of-day information of each time series:

$$\begin{aligned} p^t_{\text{sine}} &= \texttt{sin}(p_t * 2\pi/\delta_D),\ p^t_{\text{cosine}} = \texttt{cos}(p_t * 2\pi/\delta_D), \\ \mathbf{u}_t &= \left[p^t_{\text{sine}} \| p^t_{\text{cosine}}\right], \end{aligned} \tag{5}$$

where p_t is the index of t-th time-of-day point in the series, and δ_D is the day-unit time mapping. We concatenate $\mathbf{p}_{\text{sine}}$ and $\mathbf{p}_{\text{cosine}}$ as the final time stamp encoding $\mathbf{U}_{t:t+T} \in \mathbb{R}^{T\times 2}$.

Node Embedding. Previous work has demonstrated the importance of node identification in distinguishing different sensors for spatiotemporal forecasting [8, 17, 29, 33]. Here we also recommend the use of learnable node embedding for imputation task. On the one hand, it benefits the adaptation of local components [8] in graph-based data structure. On the other hand, we highlight that node embedding can be treated as an abstract and low-dimensional representation of the incomplete series. To implement, we assign each series a randomly initialized parameter $\mathbf{e}^i \in \mathbb{R}^{D_s}$. We then split the hidden dimension of the static node embedding equally by the length of the time window as a *multi-head* node embedding and unfold it to form a low-dimensional and time-varying representation: $\mathbf{E}^i_{t:t+T} \in \mathbb{R}^{T\times D_s/T}$. Implicit interactions between node embedding, input data, and modular components are involved in the end-to-end gradient descent process. Finally, the spatiotemporal input embedding for each node can be formulated as follows:

$$\mathbf{Z}^{i,(1)}_{t:t+T} = \texttt{Concat}(\mathbf{Z}^{i,(0)}_{t:t+T}; \mathbf{U}_{t:t+T}; \mathbf{E}^i_{t:t+T}, \text{dim=-1}), \tag{6}$$

where $\mathbf{Z}^{i,(1)}_{t:t+T} \in \mathbb{R}^{T\times(D+D_s/T+2)}$ is input to the following modules.

4.3 Temporal Projected Attention

As is evident in Fig. 2, time series are supposed to be redundant in the time domain, that is, most of the information can be reconstructed using only a few dominant modes. However, as the hidden dimension D' is practically much larger than the sequence length T, the attention score $\mathbb{R}^{T\times D'} \times \mathbb{R}^{D'\times T} \to \mathbb{R}^{T\times T}$ can be a high-rank matrix, which is both adverse and inefficient to reconstruct incomplete hidden spaces. To address this concern, we propose a new projected attention mechanism to impose a low-rank constraint on the attentive process and efficiently model pairwise temporal interactions between time points in linear complexity.

To utilize this structural bias, we first project the initial features to dense representations by attending to a low-dimensional vector. Specifically, we first randomly initialize a learnable vector that is shared by all nodes with the gradient tractable as the *projector* $\mathbf{P}_{\text{proj}} \in \mathbb{R}^{C\times D'}$, where $C < T$ is the projected dimension. In order to represent the temporal message in a compact form, we then project the hidden states $\mathbf{Z}^{i,(\ell)} \in \mathbb{R}^{T\times D'}$ (subscripts are omitted for brevity) to the projected space by attending to the query projector:

$$\begin{aligned} \widetilde{\mathbf{Z}}^{i,(\ell)}_{\text{proj}} &= \texttt{SelfAtten}(\mathbf{P}^{(\ell)}_{\text{proj}}, \mathbf{Z}^{i,(\ell)}, \mathbf{Z}^{i,(\ell)})), \\ &= \texttt{Softmax}\left(\frac{\mathbf{P}^{(\ell)}_{\text{proj}}\mathbf{W}_{\text{Q}}\mathbf{W}^{\text{T}}_{\text{K}}\mathbf{Z}^{i,(\ell),\text{T}}}{\sqrt{D'}}\right)\mathbf{Z}^{i,(\ell)}\mathbf{W}_{\text{V}}, \end{aligned} \tag{7}$$

where $\widetilde{\mathbf{Z}}^{i,(\ell)}_{\text{proj}} \in \mathbb{R}^{C\times D'}$ is the projected value, $\mathbf{W}_{\text{Q}}, \mathbf{W}_{\text{K}}, \mathbf{W}_{\text{V}} \in \mathbb{R}^{D'\times D'}$ are linear weights. In particular, since the projector $\mathbf{P}_{\text{proj}}$ is decoupled from the spatial dimension, the resulted attention map $\texttt{Softmax}(\mathbf{P}^{(\ell)}_{\text{proj}}\mathbf{W}_{\text{Q}}\mathbf{W}^{\text{T}}_{\text{K}}\mathbf{Z}^{i,(\ell),\text{T}}/\sqrt{D'}) \in \mathbb{R}^{C\times T}$ can be interpreted as an indicator of how the incomplete information flow can be compressed into a compact representation with smaller dimension, that is, an aggregation of available messages. More expositions on the projector will be provided in Section 5.6.

$\widetilde{\mathbf{Z}}^{i,(\ell)}_{\text{proj}}$ stores the principal temporal patterns within the input data. Then, we can recover the complete series with this compact representation by dispersing the projected information to all other full-length series by using the projector as a key dictionary:

$$\begin{aligned} \mathbf{Z}^{i,(\ell)}_{\text{hat}} &= \texttt{SelfAtten}(\mathbf{Z}^{i,(\ell)}, \mathbf{P}^{(\ell)}_{\text{proj}}, \widetilde{\mathbf{Z}}^{i,(\ell)}_{\text{proj}})), \\ &= \texttt{Softmax}\left(\frac{\mathbf{Z}^{i,(\ell)}\mathbf{W}_{\text{Q}}\mathbf{W}^{\text{T}}_{\text{K}}\mathbf{P}^{(\ell),\text{T}}_{\text{proj}}}{\sqrt{D'}}\right)\widetilde{\mathbf{Z}}^{i,(\ell)}_{\text{proj}}\mathbf{W}_{\text{V}}, \end{aligned} \tag{8}$$

then the above process is integrated into the Transformer encoder:

$$\begin{aligned} \widehat{\mathbf{Z}}^{i,(\ell)} &= \texttt{LayerNorm}(\mathbf{Z}^{i,(\ell)} + \mathbf{Z}^{i,(\ell)}_{\text{hat}}), \\ \mathbf{Z}^{i,(\ell+1)} &= \texttt{LayerNorm}(\widehat{\mathbf{Z}}^{i,(\ell)} + \texttt{FeedForward}(\widehat{\mathbf{Z}}^{i,(\ell)})), \end{aligned} \tag{9}$$

where $\mathbf{Z}^{i,(\ell+1)} \in \mathbb{R}^{T\times D'}$ is the imputation by the ℓ-th temporal interaction layer. Since the projector in Eqs. (7) and (8) can be obtained by end-to-end learning and is independent of the order in series, it has the property of data-dependent model in Eq. (4). To indicate how the above process learns the low-rank representation of temporal attention, we develop the following remark.

Remark (Difference between projected attention and canonical self-attention). *Given the query-key-value matrix*

$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T \times D'}$, *the canonical self-attention [38] in the temporal axis can be expressed compactly as:* $\texttt{SelfAtten}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \sigma(\mathbf{Q}\mathbf{K}^\top)\mathbf{V}$ *with the rank* $r \leq \min\{T, D'\}$. *For comparison, the two-step attentive process in Eqs.* (7) *and* (8) *are equivalent to the expanded form:*

$$\begin{aligned}\widehat{\mathbf{Z}} &= \texttt{SelfAtten}(\mathbf{Q}, \mathbf{P}, \texttt{SelfAtten}(\mathbf{P}, \mathbf{K}, \mathbf{V})),\\ &= \sigma(\mathbf{Q}\mathbf{P}^\top)\texttt{SelfAtten}(\mathbf{P}, \mathbf{K}, \mathbf{V}),\\ &= \sigma(\mathbf{Q}\mathbf{P}^\top)\sigma(\mathbf{P}\mathbf{K}^\top)\mathbf{V} \approx \frac{1}{N^2}\mathbf{Q}(\mathbf{P}^\top\mathbf{P})\mathbf{K}^\top\mathbf{V}.\end{aligned}$$

Recall that the projector $\mathbf{P} \in \mathbb{R}^{C \times D'}$ *can have a small projection dimension* C, *it can be viewed as a channel-wise matrix factorization to reduce redundancy within each time series. The rank of the projected attention matrix is* $r \leq \min\{C, D'\}$, *which is theoretically lower than the original rank. The projected attention guarantees expressivity by maintaining a large hidden dimension* D', *while at the same time admitting a low-rank solution using a small projection dimension* C. *The rank-reduced temporal attention matrix exploits the low-rankness of data in the temporal dimension, which is different from the low-rank adaptation of model parameters developed recently [34].*

The "projection-reconstruction" process in Eqs. (7) and (8) resemble the low-rank factorization process $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$. Inflow in Eq. (7) controls the amount of message used to form a dense representation in a lower-dimensional space. Outflow in Eq. (8) determines how hidden states can be reconstructed using only a few projected coordinates. This mechanism also brings about efficiency benefits. The canonical self-attention costs $O(T^2)$ time complexity. The complexity of the projected attention is $O(TC)$, which scales linearly (see Section A.1.3) and is efficient for longer sequences. In addition, low-rank attention preserves the dominating correlational structures and eliminates spurious correlations. The cleaned correlations allow the model to focus on the most relevant data as a reference.

4.4 Spatial Embedded Attention

The availability of observed temporal information is not sufficient for fine-grained imputation. In some cases, specific spatial events, such as traffic congestion, can lead to non-local patterns and unusual records. Therefore, it is reasonable to exploit the multivariate relationships between series as a complement. A straightforward way to address this problem is to apply a Transformer block in spatial dimension [17, 44]. Nevertheless, the three concerns discussed in Section 4.1 prevent the direct use of this technique.

Consequently, we design an embedded attention as an alternative to spatial attention. We highlight that the node embedding in Eq. (6) signifies not only the identity of series, but also a dense abstract of each individual. We then establish a correlation map using this *low-dimensional agent*. Formally, we assume that the message passing happens on a fully connected dense graph, and the edge weights are estimated by the pairwise correlating of node embedding:

$$\begin{aligned}\mathbf{Q}_e^{(\ell)} &= \texttt{Linear}(\mathbf{E}),\ \mathbf{K}_e^{(\ell)} = \texttt{Linear}(\mathbf{E}),\\ \mathbf{A}^{(\ell)} &= \texttt{Softmax}\left(\frac{\mathbf{Q}_e^{(\ell)}\mathbf{K}_e^{(\ell)\top}}{\sqrt{D'}}\right),\end{aligned} \tag{10}$$

where $\mathbf{A}^{(\ell)} \in \mathbb{R}^{N \times N}$ denotes the pairwise correlation score of all sensors, and $\mathbf{Q}_e^{(\ell)}, \mathbf{K}_e^{(\ell)} \in \mathbb{R}^{N \times D_{\text{emb}}}$ are linearly projected from the spatiotemporal embedding set $\mathbf{E} = [\bar{\mathbf{e}}^1 \| \bar{\mathbf{e}}^2 \| \cdots \| \bar{\mathbf{e}}^N] \in \mathbb{R}^{N \times D_s/T}$, with $\bar{\mathbf{e}}^i$ being the static node embedding averaging over the temporal heads $\{t : t+T\}$ from Eq. (6).

Given the graph representation over a period $\mathcal{Z} \in \mathbb{R}^{N \times T \times D'}$, the complexity of obtaining a full spatial attention matrix costs $O(N^2TD')$. To alleviate scalability concerns on large graphs, we adopt the normalization trick in [35] to reparameterize Eq. (10). Observe that the main bottleneck in Eq. (10) happens in the multiplication of two large matrix $\mathbf{Q}_e^{(\ell)}$ and $\mathbf{K}_e^{(\ell)}$, we can reduce the complexity using the associative property of matrix multiplication if we can decouple the `softmax` function. To this end, we apply the `softmax` on separate side of the Q-K matrix and approximate $\mathbf{A}$ as:

$$\mathbf{A}^{(\ell)} \approx \sigma_2(\widetilde{\mathbf{Q}}_e^{(\ell)})\sigma_1(\widetilde{\mathbf{K}}_e^{(\ell)})^\top, \tag{11}$$

where $\sigma(\cdot)$ is the abbreviation for `softmax`, and the subscript denotes the dimension that we perform normalization. The scaled Q-K functions $\widetilde{\mathbf{Q}}_e^{(\ell)} = \mathbf{Q}_e^{(\ell)}/\|\mathbf{Q}_e^{(\ell)}\|_F$, $\widetilde{\mathbf{K}}_e^{(\ell)} = \mathbf{K}_e^{(\ell)}/\|\mathbf{K}_e^{(\ell)}\|_F$ are used to ensure numerical stability. On top of Eq. (11), the complete embedded attention can be reformulated as:

$$\begin{aligned}\widetilde{\mathbf{Z}}_t^{(\ell)} &= \texttt{LayerNorm}(\mathbf{Z}_t^{(\ell)} + \sigma_2(\widetilde{\mathbf{Q}}_e^{(\ell)})\sigma_1(\widetilde{\mathbf{K}}_e^{(\ell)})^\top\mathbf{Z}_t^{(\ell)}),\\ &= \texttt{LayerNorm}(\mathbf{Z}_t^{(\ell)} + \sigma_2(\widetilde{\mathbf{Q}}_e^{(\ell)})\left(\sigma_1(\widetilde{\mathbf{K}}_e^{(\ell)})^\top\mathbf{Z}_t^{(\ell)}\right)),\\ \mathbf{Z}_t^{(\ell+1)} &= \texttt{LayerNorm}(\widetilde{\mathbf{Z}}_t^{(\ell)} + \texttt{FeedForward}(\widetilde{\mathbf{Z}}_t^{(\ell)})).\end{aligned} \tag{12}$$

By computing the multiplication of $\sigma_1(\widetilde{\mathbf{K}}_e^{(\ell)})^\top$ and $\mathbf{Z}_t^{(\ell)}$ at first, the above process admits a $O(ND_{\text{emb}})$ time complexity, which scales linearly with respect to the number of sensors. Since $\mathbf{E}$ is decoupled from temporal information, it is robust to missing values and reliable to infer a correlation map for global imputation. The full attention has the size $\mathbb{R}^{N \times (T \times D')} \times \mathbb{R}^{(T \times D') \times N} \to \mathbb{R}^{N \times N}$, while the embedded attention has $\mathbb{R}^{N \times D_{\text{emb}}} \times \mathbb{R}^{D_{\text{emb}} \times N} \to \mathbb{R}^{N \times N}$. In this sense, the attention map in Eq. (10) act as a factorized low-rank approximation of full attention. We highlight this property by comparing the two formulations in the following analysis.

Remark (Difference between embedded attention and canonical self-attention). *Given a hidden state* $\mathcal{Z} \in \mathbb{R}^{N \times T \times D'}$, *the self-attention can be computed on the folded matrix* $\mathbf{Z} \in \mathbb{R}^{N \times (TD')}$ *as* $\texttt{SelfAtten}(\mathbf{Z}, \mathbf{Z}, \mathbf{Z}) = \sigma(\mathbf{Z}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{Z}^\top)\mathbf{Z}\mathbf{W}_V$. *The resulting rank of the attention matrix obeys* $r \leq \min\{N, TD'\}$. *While the embedded attention has* $\texttt{SelfAtten}(\mathbf{E}, \mathbf{E}, \mathbf{Z}) = \sigma(\mathbf{E}\mathbf{W}_E)\sigma(\mathbf{E}\mathbf{W}_E)^\top\mathbf{Z}\mathbf{W}_V$. *If we ignore the possible rank-increasing effect of* `softmax`, *the above calculation generates the output with rank* $r \leq \min\{N, D_{emb}\}$. *Since the dimension of the node embedding* D_{emb} *is much smaller than the model dimension* D', *the rank of the embedded attention map has a lower bound of rank than the full attention. In addition, the model still has a large feedforward dimension* D' *to ensure capacity.*

4.5 Fourier Imputation Loss

As imputation model is typically optimized in a self-supervised manner, previous studies proposed adopting accumulated and hierarchical loss [7, 10, 26] to improve the supervision of layerwise imputation. However, we argue that such designs are not necessary and may generate overfitting. Instead, we propose a novel Fourier imputation loss (FIL) combined with a simple supervision loss as task-specific biases to achieve effective training and generalization.

Self-Supervised Masked Learning. To create supervised samples, we randomly whiten a proportion of incomplete observations (p_{whiten}) during model training. This operation ensures the generalizability of the imputation model (see Section 5.5). We use a masking indicator $\mathbf{M}_{\text{whiten}}$ to denote these locations where the masked values are marked as ones and others as zeros. Note that the supervision loss is only calculated on these manually whitened points, and models are forbidden to have access to the masked missing points used for evaluation. Therefore, the reconstruction loss of our model is a ℓ_1 loss on the final imputation $\widehat{\mathbf{X}}$:

$$\mathcal{L}_{\text{recon}} = \frac{1}{NT} \sum \|\mathbf{M}_{\text{whiten}} \odot (\widehat{\mathbf{X}} - \mathbf{Y})\|_1, \tag{13}$$

where $\mathbf{Y}$ is the label used for training.

Fourier Sparsity Regularization. As discussed above, we can obtain a reasonable imputation by constraining the rank of the estimated spatiotemporal tensor in the time domain. However, directly optimizing the rank of the tensor or matrix is challenging [18], as it includes some non-trivial or non-differentiable computations, such as truncated singular value decomposition (SVD). And the SVD of a $N \times T$ matrix costs $O(\min\{N^2T, NT^2\})$ complexity, which can become a bottleneck when integrated with deep models. Fortunately, we can simplify this process using the following lemma.

Lemma (Equivalence between convolution nuclear norm and Fourier ℓ_1 norm [3, 16]). *Given a smooth or periodic time series $\mathbf{x} \in \mathbb{R}^T$, its circulant (convolution) matrix $C(\mathbf{x}) \in \mathbb{R}^{T\times T}$ reflects the Tucker low-rankness, depicted by the convolutional nuclear norm. This property can be revealed by using the Discrete Fourier Transform (DFT). Let the DFT matrix be $\mathbf{U} \in \mathbb{C}^{T\times T}$, then the DFT is achieved by:*

$$\texttt{DFT}(\mathbf{x}) = \mathbf{U}\mathbf{x} = \mathbf{U}(C(\mathbf{x})[:, 0]) = (\mathbf{U}C(\mathbf{x}))[:, 0].$$

As DFT diagonalizes the circulant matrix by $C(\mathbf{x}) = \mathbf{U}^{\mathsf{H}} diag(\sigma_1, \ldots, \sigma_T)\mathbf{U}$ and $\mathbf{U}$ is a unitary matrix with the first column being ones, we have:

$$\begin{aligned}(\mathbf{U}C(\mathbf{x}))[:, 0] &= (diag(\sigma_1, \sigma_2, \ldots, \sigma_T)\mathbf{U})[:, 0], \\ &= [\sigma_1, \sigma_2, \ldots, \sigma_T]^{\mathsf{T}}.\end{aligned}$$

Therefore, we have $\|\texttt{DFT}(\mathbf{x})\|_0 = \|[\sigma_1, \sigma_2, \ldots, \sigma_T]^{\mathsf{T}}\|_0 = rank(C(\mathbf{x}))$, and $\|\texttt{DFT}(\mathbf{x})\|_1 = \|C(\mathbf{x})\|_*$.

This lemma means that we can efficiently obtain the singular values through DFT. The ℓ_0 norm is exactly the matrix rank and the ℓ_1 norm is equal to the nuclear norm $\|C(\mathbf{x})\|_*$. Since $\|C(\mathbf{x})\|_*$ serves as a convex surrogate of the rank of $\mathbf{x}$ in an approximate sense [16], we can equivalently achieve this goal by optimizing the Fourier ℓ_1 norm. Considering the above equivalence, we can develop a sparsity-constrained loss function in the frequency domain:

$$\begin{aligned}\bar{\mathbf{X}} &= \mathbf{M}_{\text{missing}} \odot \widehat{\mathbf{X}} + (1 - \mathbf{M}_{\text{missing}}) \odot \mathbf{Y}, \\ \mathcal{L}_{\text{FIL}} &= \frac{1}{NT} \sum \|\texttt{Flatten}(\texttt{FFT}(\bar{\mathbf{X}}, \text{dim} = [0, 1]))\|_1,\end{aligned} \tag{14}$$

where $\texttt{FFT}(\cdot)$ is the Fast Fourier Transform (FFT), $\texttt{Flatten}(\cdot) : \mathbb{R}^{N\times T} \to \mathbb{R}^{NT}$ rearranges the tensor form and $\|\cdot\|_1$ is the vector ℓ_1 norm. Since the spatiotemporal matrix can be regarded as a special RGB image from a global viewpoint, it also features a sparse Fourier spectrum in the space dimension [16]. We apply the FFT on both the space and time axes and then flatten it into a long vector. $\mathcal{L}_{\text{FIL}}$ is in fact a unsupervised loss that encourages the imputed values to be naturally compatible with the observed values globally.

Finally, the total loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{FIL}}, \tag{15}$$

where λ is a weight hyperparameter. It is worth commenting that the two loss functions complement each other: $\mathcal{L}_{\text{recon}}$ prompts the model to reconstruct the masked observations as precisely as possible in the space-time domain and $\mathcal{L}_{\text{FIL}}$ generalizes on unobserved points with regularization on the spectrum. This makes `ImputeFormer` work effectively in highly sparse observations.

5 EMPIRICAL EVALUATIONS

In this section, we evaluate our model on several well-known spatiotemporal benchmarks, comparing it with state-of-the-art baselines, and testing its generality on different scenarios. Then comprehensive analysis and case studies are provided. A brief summary of the adopted datasets is shown in Tab. 1. Detailed descriptions of experimental settings are provided in Section A.2. PyTorch implementations are available at https://github.com/tongnie/ImputeFormer.

Table 1: Statistics of benchmark datasets.

Datasets	Type	Steps	Nodes	Interval
METR-LA	Traffic speed	34,272	207	5 min
PEMS-BAY	Traffic speed	52,128	325	5 min
PEMS03	Traffic volume	26,208	358	5 min
PEMS04	Traffic volume	16,992	307	5 min
PEMS07	Traffic volume	28,224	883	5 min
PEMS08	Traffic volume	17,856	170	5 min
SOLAR	Power production	52,560	137	10 min
CER-EN	Energy consumption	8,868	435	30 min
AQI	Air pollutant	8,760	437	60 min
AQI36	Air pollutant	8,760	36	60 min

5.1 Results on Traffic Benchmarks

The imputation results on traffic speed and volume data are given in Tab. 2. As can be seen, `ImputeFormer` consistently achieves the best performance in all traffic benchmarks. Two strong competitors `GRIN` and `SPIN` show promising results in traffic speed data, which align with the results of their respective papers [7, 26]. However, their performance is inferior on volume datasets and is surpassed by simple baselines such as `ST-Transformer` and `Bi-MPGRU`. Compared to deep models, pure low-rank methods, such as matrix factorization and tensor completion, are less effective due to limited capacity. As for missing patterns, the structured block missing is more challenging than the point missing pattern. For instance, the vanilla Transformer is competitive in the point missing case, while it is ineffective in block missing case. Generally, `ImputeFormer` outperforms others by a large margin in this tricky scenario.

5.2 Results on Environmental and Energy Data

By exploiting the underlying low-rank structures, `ImputeFormer` can serve as a general imputer in a variety of spatiotemporal data. To demonstrate its versatility, we perform experiments on other spatiotemporal data, including energy and environmental data. Results are given in Tab. 3. It is observed that `ImputeFormer` exhibits superiority in other spatiotemporal datasets beyond traffic data. In particular, the correlation of solar stations cannot be described

Table 2: Results (in terms of MAE) on METR-LA, PEMS-BAY, PEMS03, PEMS04, PEMS07 and PEMS08 traffic benchmarks.

	Point missing						Block missing					
Models	PEMS-BAY	METR-LA	PEMS03	PEMS04	PEMS07	PEMS08	PEMS-BAY	METR-LA	PEMS03	PEMS04	PEMS07	PEMS08
Average	5.45	7.52	85.30	103.61	122.35	89.51	5.48	7.43	85.56	103.82	123.05	89.42
MICE [37]	2.82	2.89	20.07	28.60	37.11	30.26	2.36	2.73	21.90	32.45	37.20	26.66
TRMF [46]	2.10	3.51	18.80	24.34	29.06	20.27	2.09	3.36	18.71	24.47	29.42	19.80
LRTC-AR [4]	0.94	2.14	15.52	22.11	27.60	19.33	4.05	5.35	17.59	24.08	27.82	19.95
Bi-MPGRU	0.72	2.00	11.23	15.84	15.66	11.90	1.41	2.33	13.87	19.81	21.12	15.89
rGAIN [45]	1.90	2.81	13.32	22.86	24.41	16.33	2.21	2.95	14.85	23.26	26.69	27.12
BRITS [1]	1.84	2.42	12.74	20.00	23.97	15.78	1.91	2.40	12.93	19.80	23.26	16.37
SAITS [10]	1.33	2.25	12.40	20.23	22.81	15.12	1.58	2.32	12.43	20.35	22.82	16.80
Transformer [38]	0.76	2.18	12.04	16.76	16.86	12.58	1.69	3.58	24.07	29.63	33.14	25.61
ST-Transformer	0.75	2.19	11.44	16.22	15.84	12.10	1.71	3.58	23.55	29.17	32.14	24.67
TIDER [20]	1.43	2.68	15.02	22.17	21.38	18.46	2.46	4.95	21.12	23.74	28.66	21.00
TimesNet [41]	1.47	2.93	14.99	20.40	22.00	16.53	2.73	4.79	44.85	51.05	60.90	45.78
GRIN [7]	0.68	1.91	10.31	16.25	11.90	12.33	1.20	2.08	12.28	23.23	16.04	19.69
SPIN [26]	0.79	1.93	12.85	18.96	17.61	15.02	1.13	2.02	14.68	19.85	16.99	16.81
ImputeFormer	**0.64**	**1.80**	**8.23**	**14.92**	**11.38**	**11.01**	**0.95**	**1.86**	**9.02**	**16.83**	**13.82**	**12.50**
	5.9% ↓	**5.8% ↓**	**20.2% ↓**	**5.8% ↓**	**4.4% ↓**	**7.5% ↓**	**15.9% ↓**	**7.9% ↓**	**26.5% ↓**	**15.0% ↓**	**13.8% ↓**	**21.3% ↓**

by physical distance and can be inferred from the data. After comparing the performance of SAITS, Transformer, ST-Transformer, and ImputeFormer, it can be concluded that direct attention computations on both temporal and spatial dimensions are less beneficial than the low-rank attention. Furthermore, the spatial correlation of energy production is less pronounced. Canonical attention on the spatial axis can be redundant and generate spurious correlations. The use of embedded attention in our model can alleviate this issue.

Table 3: Results (in terms of MAE) on AQI, Solar, and CER-EN benchmarks. For Solar data, we compare the performances of baselines that are independent of the predefined graphs.

	SOLAR		CER-EN		Simulated faults	
Models	Point missing	Block missing	Point missing	Block missing	AQI36	AQI
Average	7.60	7.56	0.583	0.596	61.81	43.78
MICE	1.59	1.58	0.535	0.555	38.90	29.12
TRMF	2.44	2.35	0.557	0.559	41.91	27.67
Bi-MPGRU	N.A.	N.A.	0.247	0.349	12.02	15.41
rGAIN	1.52	1.64	0.418	0.440	15.69	22.13
BRITS	1.28	1.34	0.351	0.366	14.74	20.72
SAITS	0.98	1.25	0.341	0.368	19.79	21.09
Transformer	2.19	3.58	0.254	0.353	14.99	17.04
ST-Transformer	2.17	3.57	0.251	0.351	13.27	18.55
TIDER	2.84	3.87	0.336	0.377	32.85	18.11
TimesNet	2.93	4.73	0.328	0.460	32.30	28.99
GRIN	N.A.	N.A.	**0.235**	0.341	12.08	14.51
SPIN	N.A.	N.A.	OOM	OOM	11.89	14.31
ImputeFormer	**0.51**	**0.89**	0.236	**0.296**	**11.58**	**13.40**
	48.0% ↓	**28.8% ↓**	**0.4% ↑**	**13.2% ↓**	**2.6% ↓**	**6.4% ↓**

5.3 Ablation Study

To justify the rationale of model designs, we conduct ablation studies on the model structure. Results are shown in Tab. 4. Several intriguing findings can be observed: (1) After removing any of the temporal and spatial attention modules, the performance degenerates substantially; especially, the *spatial interaction contributes to the inference of block missing patterns significantly*, while the *temporal modules are crucial for point missing scenarios*. (2) *The incorporation of* MLP *benefits little for the imputation*, which validates our argument in Section A.1.2. (3) Compared to hierarchical loss on supervised points, FIL generalizes on the unobserved points and effectively reduces the estimation errors.

Table 4: Ablations studies on ImputeFormer.

Variation	Component		PEMS08		METR-LA	
	Spatial	Temporal	Point	Block	Point	Block
ImputeFormer	**Attention**	**Attention**	**11.24**	**12.86**	**1.80**	**1.88**
Replace	Attention	MLP	16.95	17.11	2.39	2.28
	MLP	Attention	12.84	17.42	2.20	2.92
	MLP	MLP	34.72	34.41	5.80	5.79
w/o	Attention	w/o	17.06	17.13	2.39	2.28
	w/o	Attention	12.87	17.44	2.21	2.93
Loss function	w/o FIL		11.63	13.35	1.85	1.93
	Hierarchical loss		11.35	13.07	1.84	1.92
Architecture	Order	T-S	11.26	13.16	1.80	1.87
		S-T	11.30	13.13	1.80	1.88
	Joint ST		17.50	20.00	1.94	2.58

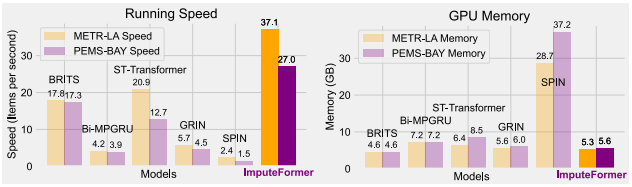


Figure 3: Comparison of computational efficiency.

5.4 Model Efficiency

We evaluate the computational efficiency of different architectures in Fig. 3. Intuitively, ImputeFormer exhibits high training efficiency. Due to the low-rank design philosophy, ImputeFormer is approximately 15 times faster than the state-of-the-art Transformer baseline (SPIN). It is also cost-effective in GPU memory consumption.

5.5 Robustness and Versatility Analysis

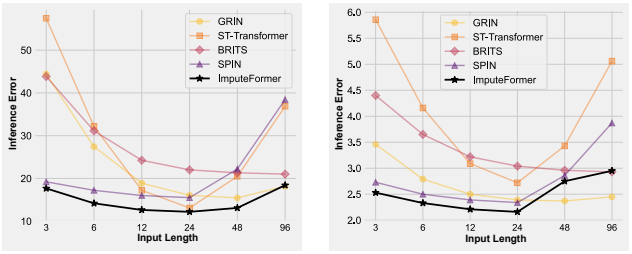
Inference under Different Missing Rates. Deep imputation models are subject to the distribution shift problem between training

and testing datasets. A desirable characteristic is that a model can deal with different missing patterns during inference. Therefore, we consider a challenging scenario in which a model is trained with a fixed missing rate but evaluated on different scenarios with varying missing rates. This constructs a zero-shot transfer evaluation. It is noteworthy in Tab. 5 that both `ImputeFormer` and `SPIN` are more robust than other baselines in these scenarios. RNNs and vanilla Transformers can overfit the training data with a fixed data missing pattern, thereby showing inferior generalization ability.

Table 5: Inference under varying missing rate with a single trained model (Zero-shot).

Models	PEMS08			METR-LA		
	Missing rate			Missing rate		
	50%	75%	95%	50%	75%	95%
BRITS	17.21	22.01	52.78	2.61	3.04	5.11
SAITS	16.03	31.32	83.79	2.44	3.37	6.80
ST-Transformer	11.65	13.11	39.95	2.32	2.72	5.16
GRIN	13.25	16.06	42.61	2.06	2.39	4.07
SPIN	15.13	15.51	18.30	2.11	2.34	3.03
ImputeFormer	**11.52**	**12.18**	**17.35**	**1.96**	**2.17**	**2.79**

Inference with Varying Sequence Length. In reality, the imputation model can face time series with different lengths and sampling frequencies. We can adopt a well-trained model to perform inference on varying sequence length. Results are shown in Fig. 4. It is obvious that `ImputeFormer` can readily generalize to sequences with different lengths and more robust than other models.



(a) PEMS08 (b) METR-LA

Figure 4: Inference under different lengths of input sequence with a single trained model (zero-shot).

Dealing with Highly Sparse Observation. To evaluate the performance on highly sparse data, we further train and test models with lower observation rates. Results are shown in Tab. 6. Generally speaking, both Transformer- and RNN-based models are susceptible to sparse training data. Due to the low-rank constraints on the attention matrix and loss function, our model is more robust with highly sparse data. Since the attention map in `SPIN` is calculated only at the observed points, it is more stable than other baselines. But our model consistently achieves lower imputation errors.

Random Masking in Training. Random masking strategy is used to create supervised samples for model training. Therefore, the distribution of masking samples of training data and missing observations of testing data should be close to ensure good performance [31]. However, it can be difficult to know exactly the missing patterns or missing rates in advance in many scenarios. Therefore, a proper masking strategy is of vital importance. To

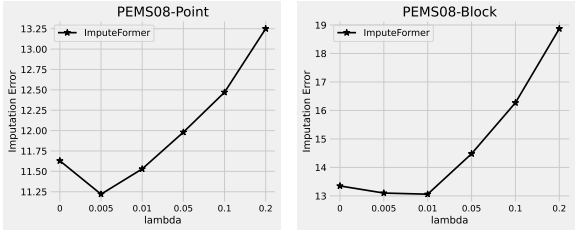
Table 6: Results on `PEMS08` data with sparse observations (Training from scratch).

Models	Missing rate			
	60%	70%	80%	90%
BRITS	18.60	19.75	21.44	24.17
SAITS	17.53	18.24	19.39	21.27
ST-Transformer	13.67	14.32	15.86	23.98
GRIN	14.04	15.01	17.26	25.47
SPIN	13.64	14.30	15.19	17.13
ImputeFormer	**12.57**	**13.17**	**13.98**	**15.94**

Table 7: Results on `PEMS03` with various masking strategies.

Point missing	Masking Probability			
Models	25%	50%	75%	Combined
Bi-MPGRU	11.30	11.52	12.20	11.48
BRITS	13.06	13.86	16.06	13.70
SAITS	12.42	16.13	21.63	12.61
ST-Transformer	11.19	11.43	12.22	11.39
GRIN	9.55	9.74	10.39	9.72
SPIN	11.08	11.21	13.85	12.04
ImputeFormer	**7.66**	**8.16**	**11.44**	**8.45**
Block missing	Masking Probability			
Models	25%	50%	75%	Combined
Bi-MPGRU	13.32	13.36	13.96	13.33
BRITS	12.26	13.01	15.58	12.63
SAITS	12.35	15.73	20.14	12.32
ST-Transformer	23.51	23.76	23.90	23.26
GRIN	11.94	12.05	12.68	11.99
SPIN	13.10	13.68	13.84	13.97
ImputeFormer	**8.89**	**9.23**	**16.96**	**8.80**

evaluate the impact of the masking rate in training data, we further consider four different masking strategies during model training: the masking rates are set to 0.25, 0.5, 0.75, and a combination of them [0.25, 0.5, 0.75] respectively. As shown in Tab. 7, most models perform best when the masking rate is close to the missing rate in the point missing scenario (e.g., 25%). However, when the missing rate is unclear due to randomness within the failure generation process, such as the block missing, the combination strategy is more advantageous. For example, Transformers including `ST-Transformer`, `SAITS`, and `ImputeFormer` can benefit from this strategy. More importantly, such a hybrid masking method enables the model to work successfully on varying observation rates during inference.



(a) PEMS08, point missing (b) PEMS08, block missing

Figure 5: Impact of λ in FIL.

Impact of Fourier Imputation Loss. We study the impact of hyperparameter λ in Eq. (15). Fig. 5 displays the imputation error under different λ values. The imputation model can benefit from the FIL design, and a too large penalty on the sparsity of the spectrum can lead to a smooth and inaccurate reconstruction.

5.6 Case Studies: Interpretability

This section studies the interpretability using data examples.

Spectrum Analysis. To corroborate the hypothesis that our model has the merits of both deep learning and low-rank methods, we analyze the singular value (SV) spectrum of the imputations. Fig. 6 shows the cumulative SV distribution of different competing models. `ImputeFormer` has a close SV cumulative distribution to complete data, and the first 85 SVs can account for 80% of the energy. There exist two additional interesting observations: (1) deep learning models without explicit low-rank modeling such as canonical Transformers downplay the role of the first few dominant SVs; (2) pure low-rank models such as MF generate an oversmoothing result that too much energy is constrained to the first part of spectrum. Thus, we can ascribe the desirable performance of our model to the *good balance of significant signals and high-frequency noise.*

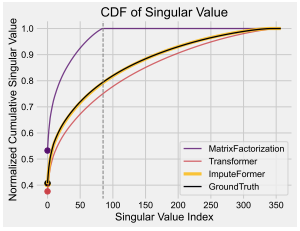


Figure 6: Cumulative distribution of singular values.

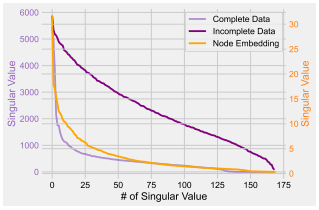


Figure 7: Singular spectrum of data and node embedding.

Interpretations on Spatial Embedding. To illustrate the role of node embedding, we analyze the SV spectrum of the `PEMS08` data in Fig. 7. The complete data show a prominent low-rank property, but the SVs of incomplete data dramatically expand. In contrast, the node embedding also displays a similar low-rank distribution, which can act as a dense surrogate for each sensor. Furthermore, we analyze the multivariate attention map obtained by correlating the node embedding in Fig. 8. It is evident that as the embedded attention layers become deeper, the learned attention maps approach the actual ones. However, *incomplete data produce noisy correlations with little informative pattern.* Fig. 9 displays the t-SNE visualization of each node embedding with two projected coordinates in `PEMS08`. The embeddings tend to form clusters, and different clusters are apart from others. This phenomenon is in accordance with highway traffic sensor systems that proximal sensors share similar readings.

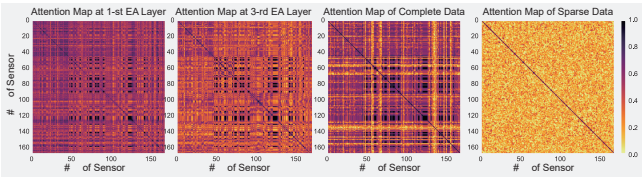


Figure 8: Multivariate attention maps of `PEMS08` data.

Interpretations on Temporal Projector. To illustrate the mechanism of the temporal projected attention in Eqs. (7) and (8), inflow and outflow attention maps are shown in Figs. 11. It can be seen that these matrices quantify how the information of incomplete states flows into compact representations and then is recovered to complete states. Inflows show that only a fraction of the message is directed towards the projector, while different attention heads can provide varying levels of information density. Meanwhile, outflows indicate that a small number of temporal modes can reconstruct useful neural representations for imputation. *This can be analogous to the low-rank reconstruction process, which serves as an inductive bias for time series with low information density.* We further examine the SV distribution of different hidden states in the last temporal attention layer. As evidenced by Fig. 10, after flow through the projected attention layer, the hidden states have lower SVs than the incomplete inputs and are closer to the complete representations.

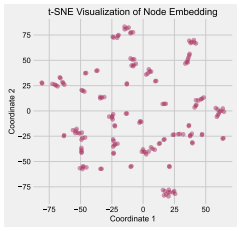


Figure 9: The t-SNE visualization of node embedding.

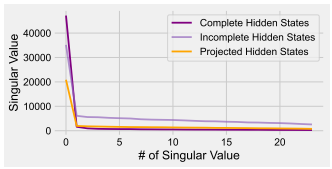


Figure 10: Singular values of different hidden states in the temporal attention layer.

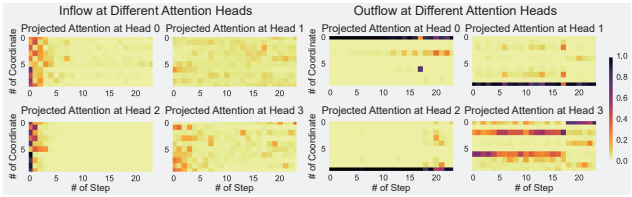


Figure 11: Inflow and outflow in the projected attention layer.

6 CONCLUSION

This paper demonstrates a low rankness-induced Transformer model termed `ImputeFormer` to address the missing spatiotemporal data imputation problem. Taking advantage of the low-rank factorization, we design projected temporal attention and embedded spatial attention to incorporate structural priors into the Transformer model. Furthermore, a Fourier sparsity loss is developed to regularize the solution's spectrum. The evaluation results on various benchmarks indicate that `ImputeFormer` not only consistently achieves state-of-the-art imputation accuracy, but also exhibits high computational efficiency, generalizability across various datasets, versatility for different scenarios, and interpretability. Therefore, we believe that it has the potential to advance research on spatiotemporal data for general imputation tasks. Future work can adopt it to achieve time series representation learning task and explore the multipurpose pretraining problem for time series.

ACKNOWLEDGMENTS

The work was supported by research grants from the National Natural Science Foundation of China (52125208), National Key R&D Programs of China (2022YFB2602100), the China National Postdoctoral Program for Innovative Talents (BX20220231), the China Postdoctoral Science Foundation (2022M712409), and the Science and Technology Commission of Shanghai Municipality (22dz1203200).

REFERENCES

[1] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems* 31 (2018).

[2] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 6085.

[3] Xinyu Chen, Zhanhong Cheng, Nicolas Saunier, and Lijun Sun. 2022. Laplacian convolutional representation for traffic time series imputation. *arXiv preprint arXiv:2212.01529* (2022).

[4] Xinyu Chen, Mengying Lei, Nicolas Saunier, and Lijun Sun. 2021. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 12301–12310.

[5] Xinyu Chen and Lijun Sun. 2021. Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 4659–4673.

[6] Xinyu Chen, Jinming Yang, and Lijun Sun. 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies* 117 (2020), 102673.

[7] Andrea Cini, Ivan Marisca, and Cesare Alippi. 2021. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. *arXiv preprint arXiv:2108.00298* (2021).

[8] Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. 2024. Taming local effects in graph-based spatiotemporal forecasting. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Commission for Energy Regulation (CER). 2012. CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010. Dataset. https://www.ucd.ie/issda/data/commissionforenergyregulationcer/ SN: 0012-00.

[10] Wenjie Du, David Côté, and Yan Liu. 2023. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications* 219 (2023), 119619.

[11] Azul Garza and Max Mergenthaler-Canseco. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589* (2023).

[12] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. 2023. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759* (2023).

[13] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[14] Wei Liang, Yuhui Li, Kun Xie, Dafang Zhang, Kuan-Ching Li, Alireza Souri, and Keqin Li. 2023. Spatial-Temporal Aware Inductive Graph Neural Network for C-ITS Data Recovery. *IEEE Transactions on Intelligent Transportation Systems* 24, 8 (2023), 8431–8442.

[15] Yuebing Liang, Zhan Zhao, and Lijun Sun. 2022. Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns. *Transportation Research Part C: Emerging Technologies* 143 (2022), 103826.

[16] Guangcan Liu and Wayne Zhang. 2022. Recovery of future data via convolution nuclear norm minimization. *IEEE Transactions on Information Theory* 69, 1 (2022), 650–665.

[17] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4125–4129.

[18] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2012. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 208–220.

[19] Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. 2023. PriSTI: A Conditional Diffusion Framework for Spatiotemporal Imputation. *arXiv preprint arXiv:2302.09746* (2023).

[20] Shuai Liu, Xiucheng Li, Gao Cong, Yile Chen, and Yue Jiang. 2022. Multivariate Time-series Imputation with Disentangled Temporal Representations. In *The Eleventh International Conference on Learning Representations*.

[21] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. 2019. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems* 32 (2019).

[22] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. 2018. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems* 31 (2018).

[23] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. 2019. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*. AAAI Press Palo Alto, CA, USA, 3094–3100.

[24] Jiawei Ma, Zheng Shou, Alireza Zareian, Hassan Mansour, Anthony Vetro, and Shih-Fu Chang. 2019. CDSA: cross-dimensional self-attention for multivariate, geo-tagged time series imputation. *arXiv preprint arXiv:1905.09904* (2019).

[25] Wei Ma and George H Chen. 2019. Missing not at random in matrix completion: The effectiveness of estimating missingness probabilities under a low nuclear norm assumption. *Advances in neural information processing systems* 32 (2019).

[26] Ivan Marisca, Andrea Cini, and Cesare Alippi. 2022. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. *Advances in Neural Information Processing Systems* 35 (2022), 32069–32082.

[27] Nazeer Muhammad, Nargis Bibi, Adnan Jahangir, and Zahid Mahmood. 2018. Image denoising with norm weighted fusion estimators. *Pattern Analysis and Applications* 21 (2018), 1013–1022.

[28] Tong Nie, Guoyang Qin, and Jian Sun. 2022. Truncated tensor Schatten p-norm based approach for spatiotemporal traffic data imputation with complicated missing patterns. *Transportation research part C: emerging technologies* 141 (2022), 103737.

[29] Tong Nie, Guoyang Qin, Lijun Sun, Wei Ma, Yu Mei, and Jian Sun. 2023. Contextualizing MLP-Mixers Spatiotemporally for Urban Data Forecast at Scale. *arXiv preprint arXiv:2307.01482* (2023).

[30] Tong Nie, Guoyang Qin, Yunpeng Wang, and Jian Sun. 2023. Correlating sparse sensing for large-scale traffic speed estimation: A Laplacian-enhanced low-rank tensor kriging approach. *Transportation Research Part C: Emerging Technologies* 152 (2023), 104190.

[31] Tong Nie, Guoyang Qin, Yunpeng Wang, and Jian Sun. 2023. Towards better traffic volume estimation: Jointly addressing the underdetermination and nonequilibrium problems with correlation-adaptive GNNs. *Transportation Research Part C: Emerging Technologies* 157 (2023), 104402.

[32] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).

[33] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4454–4458.

[34] Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. 2023. The Truth is in There: Improving Reasoning in Language Models with Layer-Selective Rank Reduction. *arXiv preprint arXiv:2312.13558* (2023).

[35] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. 2021. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 3531–3539.

[36] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* 34 (2021).

[37] Stef Van Buuren and Karin Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software* 45 (2011), 1–67.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*.

[39] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768* (2020).

[40] Xudong Wang, Yuankai Wu, Dingyi Zhuang, and Lijun Sun. 2023. Low-rank Hankel tensor completion for traffic speed estimation. *IEEE Transactions on Intelligent Transportation Systems* 24, 5 (2023), 4862–4871.

[41] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186* (2022).

[42] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. 2021. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4478–4485.

[43] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 753–763.

[44] Yongchao Ye, Shiyao Zhang, and James JQ Yu. 2021. Spatial-temporal traffic data imputation via graph attention convolutional network. In *International Conference on Artificial Neural Networks*. Springer, 241–252.

[45] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*. PMLR, 5689–5698.

[46] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems* 29 (2016).

[47] Ruiyang Zhang, Yang Liu, and Hao Sun. 2020. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures* 215 (2020), 110704.

[48] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2267–2276.

A APPENDIX

A.1 Additional Discussions

A.1.1 Architectural Comparison. We provide an illustration to show the structural difference between different paradigms in Fig. 12.

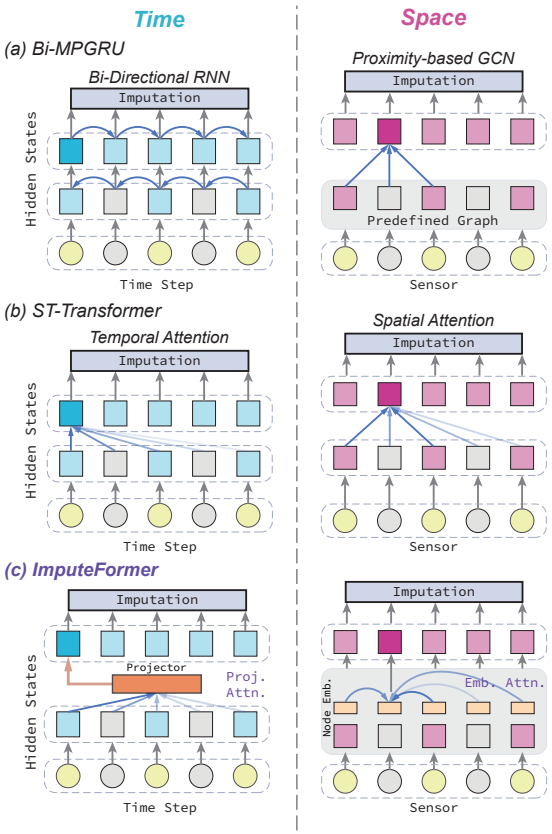


Figure 12: (a) MPGRU adopts Bi-RNNs to gather available readings from consecutive time points and GCNs to collect neighborhood data on predefined graphs. (b) Transformers compute all pairwise correlations of the raw data in both spatial and temporal axes. (c) `ImputeFormer` utilizes projected attention along the temporal axis and embedded attention based on the node representation in the spatial axis.

A.1.2 Effective Input Embedding. Unlike images or languages, time series have low semantic densities [32]. Therefore, many forecasting models flatten and abstract the input series to reduce information redundancy [29, 33]. Specifically, given the input $\mathbf{X}_{t:t+T} \in \mathbb{R}^{N\times T}$, each series can be processed by a MLP shared by all series: $\mathbf{z}^{i,(0)} = \texttt{MLP}(\mathbf{x}^i)$, where $\texttt{MLP}(\cdot) : \mathbb{R}^T \to \mathbb{R}^D$. However, we claim that this technique is not suitable for imputation. If we express it as follows:

$$x_{t+h} = \sigma\left(\sum_{k=0}^{T} w_{k,h} x_{t+k} + b_{k,h}\right),\ h \in \{0, \dots, T\}, \tag{16}$$

it is evident that the linear weights only depends on the relative position in the sequence and are agnostic to the data flow. Since missing data points and intervals can occur at arbitrary locations in the series, fixed weights can learn spurious relationships between each time step, thus overfitting the missing patterns in the training data. Therefore, we suggest not to use linear mappings on the time axis to account for the varying missing time points.

A.1.3 Low-Rankness in Self-Attention. Wang et al. [39] studied the observation that the self-attention matrix in Transformer is low-rank and proposed a linear attention. Our proposed temporal projected attention model shares a similar idea as this work but has different mechanisms and formulations. We indicate the differences in the following exposition. Given $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T\times D}$, and the projector $\mathbf{P} \in \mathbb{R}^{C\times D}$, temporal projected attention is formulated as:

$$\begin{aligned}&\texttt{SelfAtten}(\mathbf{Q}, \mathbf{P}, \texttt{SelfAtten}(\mathbf{P}, \mathbf{K}, \mathbf{V})),\\ &= \sigma(\mathbf{Q}\mathbf{P}^\mathsf{T})\texttt{SelfAtten}(\mathbf{P}, \mathbf{K}, \mathbf{V}) = \underbrace{\sigma(\mathbf{Q}\mathbf{P}^\mathsf{T})}_{T\times C}\underbrace{\sigma(\mathbf{P}\mathbf{K}^\mathsf{T})}_{C\times T}\mathbf{V},\end{aligned} \tag{17}$$

while the linear attention [39] assigns two learnable matrices $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{C\times T}$ and has the following form:

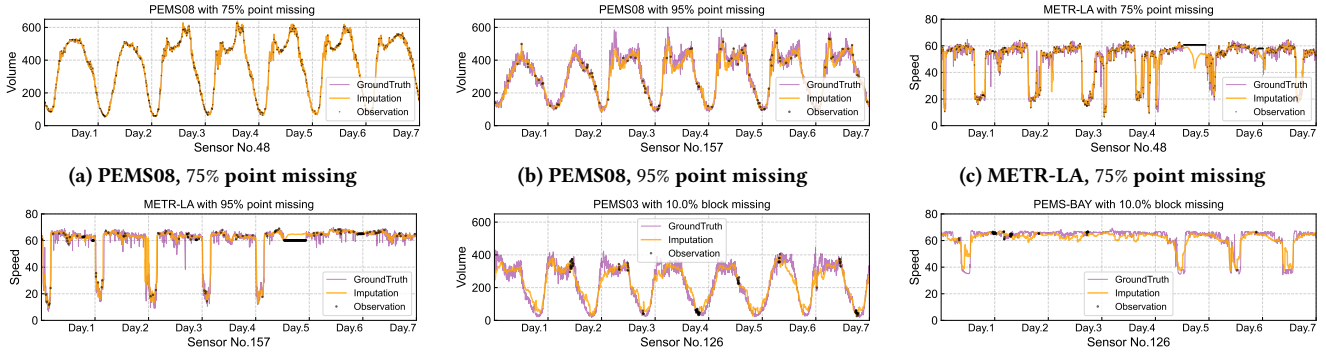
$$\texttt{SelfAtten}(\mathbf{Q}, \mathbf{E}\mathbf{K}, \mathbf{F}\mathbf{V})),\ = \sigma(\underbrace{\mathbf{Q}\mathbf{K}^\mathsf{T}}_{T\times T}\underbrace{\mathbf{E}^\mathsf{T}}_{T\times C})\mathbf{F}\mathbf{V}. \tag{18}$$

As for complexity, we can compute Eq. (17) in the order: $\sigma(\mathbf{Q}\mathbf{P}^\mathsf{T}) > \sigma(\mathbf{P}\mathbf{K}^\mathsf{T}) > \sigma(\mathbf{P}\mathbf{K}^\mathsf{T})\mathbf{V} > \sigma(\mathbf{Q}\mathbf{P}^\mathsf{T})\sigma(\mathbf{P}\mathbf{K}^\mathsf{T})\mathbf{V}$, which admits $O(4TDC)$ complexity. Similarly, Eq. (18) has the same complexity. Although of the same complexity, our model has an explicit and symmetric formulation that brings improved model expressivity. The advantages are threefold: (1) *explicit low-rank factorization*: Linformer [39] does not directly achieve a low-rank factorization of the attention matrix. It first computes the full attention matrix $\mathbf{Q}\mathbf{K}^\mathsf{T}$ with size $T \times T$ and then compresses it to $T \times C$. Instead, `ImputeFormer` directly factorizes the full attention matrix from $T\times T$ to $T\times C, C\times T$, which is beneficial for dealing with redundancy and missingness in the attention matrix; (2) *pattern adaptation*: Linformer sets the compression matrix $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{C\times T}$ completely learnable, which is agnostic to the missing patterns of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. These static parameters cannot account for the varying missing patterns. Instead, we obtain the $T \times C$ factor matrix through the query $\mathbf{Q}\mathbf{P}^\mathsf{T}$, which is pattern-adaptive; (3) *increased capacity*: Linformer has $T \times C$ learnable parameters, while ours has $C \times D$ parameters, which has a larger model capacity while having the same time complexity.

A.2 Reproducibility

A.2.1 Implementations. We build our model and baselines based on the `SPIN` repository (https://github.com/Graph-Machine-Learning-Group/spin). All experiments were performed on a single NVIDIA RTX A6000 GPU (48 GB). For the hyperparameters of `ImputeFormer`, we set the hidden size to 256, the input projection size to 32, the node embedding size to 64, the projected size to 6, the number of attention layers to 3, and the sequence length to 24 for all data. We also keep the same training, validation and evaluation split as [7, 26] and report the metrics on the masked evaluation points.

A.2.2 Dataset Descriptions. We adopt heterogeneous spatiotemporal benchmark datasets to evaluate the imputation performance.
Traffic Speed Data. Our experiments include two commonly used traffic speed datasets, named `METR-LA` and `PEMS-BAY`. `METR-LA` contains spot speed data from 207 loop sensors over a period of 4



(a) PEMS08, 75% **point missing** **(b) PEMS08,** 95% **point missing** **(c) METR-LA,** 75% **point missing**

(d) METR-LA, 95% **point missing** **(e) PEMS03,** 10% **block missing** **(f) PEMS-BAY,** 10% **block missing**

Figure 13: Visualization examples of imputation for traffic speed and volume data under different missing patterns.

months from Mar 2012 to Jun 2012, located at the Los Angeles County highway network. PEMS-BAY records 6 months of speed data from 325 static sensors in the San Francisco South Bay Area.

Traffic Volume Data. We adopt four traffic volume data, including PEMS03, PEMS04, PEMS07, and PEMS08. They contain the highway traffic volume record collected by the Caltrans Performance Measurement System (PeMS) and aggregated into 5-minute intervals.

Energy and Environmental Data. Four energy and environmental data are selected to evaluate the generality of models, including: (1) Solar: solar power production records from 137 synthetic PV farms in Alabama state in 2006, which are sampled every 10 minutes; (2) CER-EN: smart meters measuring energy consumption from the Irish Commission for Energy Regulation Smart Metering Project [9]. Following the setting in [7], we select 435 time series aggregated at 30 minutes for evaluation. (3) AQI: PM2.5 pollutant records collected by 437 air quality monitoring stations in 43 Chinese cities from May 2014 to April 2015 with the aggregation interval of 1 hour. Note that AQI data contains nearly 26% missing data. (4) AQI36: a subset of AQI data which contains 36 sensors in Beijing distinct.

A.2.3 Experimental Settings and Baseline Methods. This section describes the detailed information on experimental setups.

Missing patterns. For traffic, Solar and CER-EN, we consider two scenarios discussed in [7, 26]: (1) Point missing: randomly remove observed points with 25% probability; (2) Block missing: randomly drop 5% of the available data and at the same time simulate a sensor failure lasting for $\mathcal{L} \sim \mathcal{U}(12, 48)$ steps with 0.15% probability. We keep the above missing rates the same as in the previous work [7, 26]. In addition, we also evaluated the performance under sparser conditions. For example, the block missing with 10% probability corresponds to a total missing rate of $\approx 90 \sim 95\%$. Note that matrix or tensor models can only handle in-sample imputation, where the observed training data and the test data are in the same time period. However, deep models can work in out-of-sample scenarios [7] where the training and test sequences are disjoint. We adopt the out-of-sample tests for deep models and in-sample tests for others.

Baseline Methods. We compare our model with SOTA deep-learning and low-rank imputation methods. For statistical and optimization models, we consider: (1) Observation average (Average); (2) Temporal regularized matrix factorization (TRMF) [46]; (3) Low-rank autoregressive tensor completion (LRTC-AR) [4]; (4) MICE [37]. For deep imputation models, we select several competitive baselines: (1) SPIN [26]: sparse spatiotemporal attention model with state-of-the-art imputation performance; (2) GRIN [7]: message-passing-based bidirectional RNN model with competitive performance; (3) SAITS [10]: Temporal Transformer model with diagonally masked attention; (4) BRITS [1]: bidirectional RNN model for imputation; (5) rGAIN [45]: GAIN model with bidirectional recurrent encoder and decoder; (6) Transformer/ST-Transformer [38]: canonical Transformer with self-attention in temporal or spatial-temporal dimensions; (7) TiDER [20]: matrix factorization with disentangled neural representations; (8) TimesNet [41]: 2D convolution-based general time series analysis model; (9) BiMPGRU: a bidirectional RNN-based GCN model, which is similar to DCRNN [13].

Ablation Studies. Particularly, we examine the following variations: (a) Temporal blocks: we replace the temporal interaction module with MLP or directly remove it; (b) Spatial blocks: we replace the spatial interaction module with MLP or directly remove it; (c) Loss function: We remove the FIL or replace it with a hierarchical loss used in [10, 26]; (d) Architecture: we evaluate the impacts of the order of spatial-temporal blocks and the joint attention strategy. We adopt two data to evaluate and other settings remain the same.

Evaluation Metrics. To evaluate the model, we simulate different observation conditions by removing parts of the raw data to construct incomplete samples based on different missing rates (p_{missing}). Evaluation metrics are then calculated for these simulated missing points. We use a masking indicator $\mathbf{M}_{\text{missing}}$ to denote these locations in which the unobserved (missing) values are marked as ones, observed as zeros. Note that the masked points for evaluation are not available for the models during all stages. The mean absolute error (MAE) is adopted to report the results.

A.3 Imputation Visualization

We provide several visualization examples in Fig. 13. As evidenced, ImputeFormer can generate reasonable imputations by learning the inherent structures of spatiotemporal data. Previous studies have discovered that low-rank models can cause oversmoothing estimation [4, 30]. Due to the representation power of deep architectures, our model can provide a detailed reconstruction. In particular, although only limited temporal information is available in the block missing case, it can resort to the node embedding as the query to spatial relations, thereby generating an effective imputation.