Graph Structure Learning for Spatial-Temporal Imputation: Adapting to Node and Feature Scales

Xinyu Yang[1], Yu Sun[1*], Xinyang Chen[2*], Ying Zhang[1], Xiaojie Yuan[1]

[1]College of Computer Science, DISSec, Nankai University, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
{yangxinyu@dbis., sunyu@, yingzhang@, yuanxj@}nankai.edu.cn, chenxinyang@hit.edu.cn

Abstract

Spatial-temporal data collected across different geographic locations often suffer from missing values, posing challenges to data analysis. Existing methods primarily leverage fixed spatial graphs to impute missing values, which implicitly assume that the spatial relationship is roughly the same for all features across different locations. However, they may overlook the different spatial relationships of diverse features recorded by sensors in different locations. To address this, we introduce the multi-scale **G**raph **S**tructure **L**earning framework for spatial-temporal **I**mputation (**GSLI**) that dynamically adapts to the heterogeneous spatial correlations. Our framework encompasses node-scale graph structure learning to cater to the distinct global spatial correlations of different features, and feature-scale graph structure learning to unveil common spatial correlation across features within all stations. Integrated with prominence modeling, our framework emphasizes nodes and features with greater significance in the imputation process. Furthermore, GSLI incorporates cross-feature and cross-temporal representation learning to capture spatial-temporal dependencies. Evaluated on six real incomplete spatial-temporal datasets, GSLI showcases the improvement in data imputation.

Introduction

The stations at different geographic locations may occur missing values when recording spatial-temporal data through multiple kinds of sensors (Zhao et al. 2020b; Fan et al. 2023). Figure 1(a) presents an example of spatial-temporal data with four features recorded by seven stations located in the Netherlands. As shown in Figure 1(b), the readings of feature FH in station AMS at t_3 and feature DD in station DBT at t_{17} are missing. This may lead to anomalies in the patterns discovered by analysis models, thus creating challenges in mining spatial-temporal data (Ren et al. 2023).

The spatial correlation of spatial-temporal data is usually given as a graph where stations are nodes and edges indicate geographic distance or connectivity between stations (Li et al. 2018; Wu et al. 2019; Fan et al. 2023). Existing works typically utilize the given graph and graph neural networks (GNN) to capture spatial dependencies (Cini, Marisca, and Alippi 2022; Liu et al. 2023a). Since AMS and DBT stations are geographically closest to each other, their corresponding edge weights in the graph are larger. Therefore, as shown in Figure 1(b), we can get an accurate imputation result of the feature DD in DBT, according to the DD value in AMS station. Unfortunately, this is not always the case, and we can observe that using the FH value in DBT station will mislead the existing methods for imputing the FH value in AMS station. The reason is that features DD and FH are recorded by sensors from different domains, where DD captures wind direction data and FH is related to wind speed. There is a significant correlation between wind direction information from AMS and DBT stations as they are located close to each other. However, since AMS is situated in an airport with a relatively sparse environment and DBT is located in a municipality with many buildings, the relationship between wind speed values of two stations is not clear.

Figure 1(c) illustrates the spatial relationships of different features across all stations, where we extract the attention map from the cross-feature self-attention mechanism. As shown, the spatial relationships for different features are varied, unlike the implicit assumption by existing methods that spatial relationships across stations are similar for different features. In addition, in Figure 1(d), we also find that there is a relatively fixed spatial relationship between the features within each station. Specifically, the attention maps between features within the two stations ELD and ELL are generally similar. This suggests that there also exist correlations between different features across all stations. However, this correlation cannot be reflected in the given graph and thus cannot contribute to the imputation of existing methods.

Enlightened by the aforesaid analysis, we consider the multi-scale **G**raph **S**tructure **L**earning framework for spatial-temporal **I**mputation (**GSLI**). Our main contributions can be summarized as follows.

(1) We present the node-scale graph structure learning to model the fine-grained global spatial correlations of different features. By adaptively learning independent global graph structures for different features, our framework can mitigate the negative effects between features in different domains and improve imputation performance.

(2) We design feature-scale graph structure learning to

*Corresponding authors.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

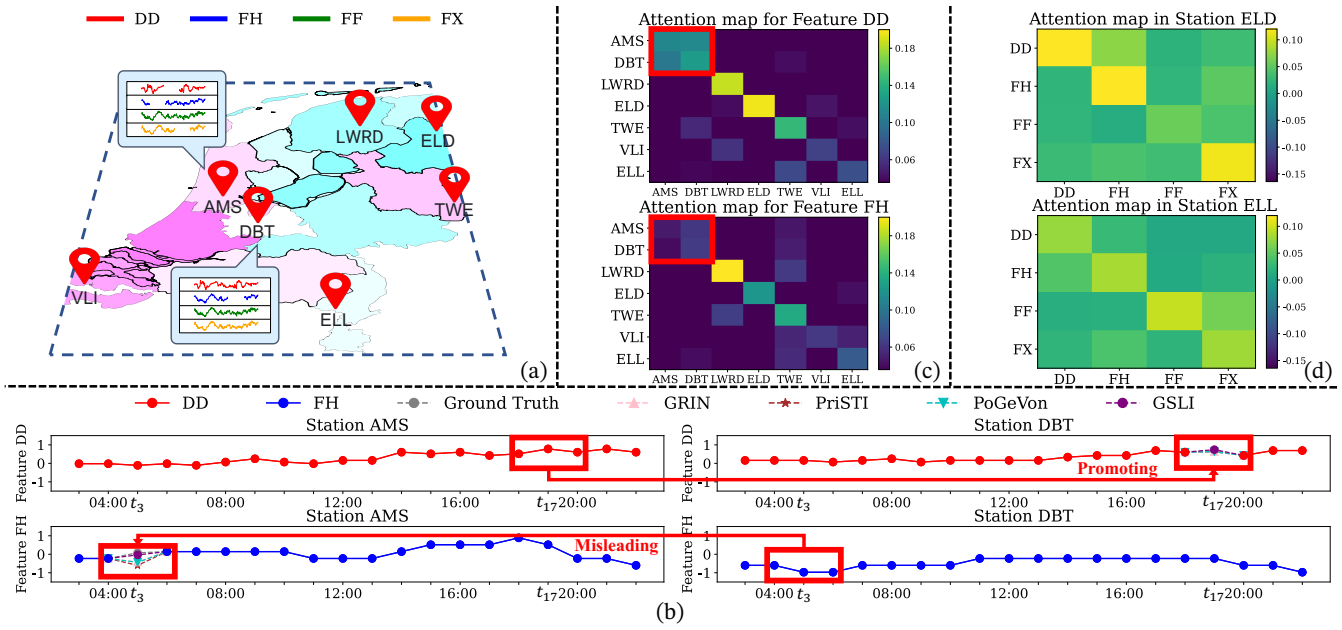

Figure 1: (a) Incomplete spatial-temporal data with four features recorded in different stations in the Netherlands. (b) Imputation examples at timestamps t_3 and t_{17}. (c) The extracted attention maps for features DD and FH. (d) The extracted attention maps for the four features in stations ELD and ELL.

learn the common spatial correlation of different features over all nodes. Our framework can capture the spatial dependencies between features across each station with the help of the learned feature-scale graph structure.

(3) We incorporate prominence modeling into the graph structure learning processes to account for the varying influence of different nodes and different features. Thus, the nodes and features that contribute more to imputation can get stronger weights in the graph structures.

Experimental evaluations over real-world incomplete datasets demonstrate the superiority of our GSLI, by utilizing cross-feature representation learning and cross-temporal representation learning.

Related Work

Spatial-Temporal Imputation Spatial-temporal data, when viewed as multivariate time series by disregarding spatial correlation, often undergoes imputation using time series imputation methods. Time series imputation utilizes various methods, including statistical approaches like mean imputation (Kantardzic 2011), last observation carried forward (Amiri and Jensen 2016), and local interpolation (Acuna and Rodriguez 2004), alongside techniques such as TRMF (Yu, Rao, and Dhillon 2016), BTMF (Chen and Sun 2022), and TIDER (Liu et al. 2023b) which employ low-rank matrix factorization. RNN-based methods GRU-D (Che et al. 2018) and BRITS (Cao et al. 2018), along with GAN-integrated methods GAN-2-Stage (Luo et al. 2018), E^{2}GAN (Luo et al. 2019), SSGAN (Miao et al. 2021b), as well as self-attention and convolutional approaches in STCPA (Xu et al. 2022), SAITS (Du, Côté, and Liu 2023), and TimesNet (Wu et al. 2023) emphasize temporal dependencies. Moreover, VAEs in MIWAE (Mattei and Frellsen 2019), GP-VAE (Fortuin et al. 2020), TimeCIB (Choi and Lee 2024), diffusion models in CSDI (Tashiro et al. 2021), MIDM (Wang et al. 2023b), SSSD (Alcaraz and Strodthoff 2023), and GPT4TS (Zhou et al. 2023) using large language models are explored. These methods, however, typically ignore the spatial adjacency crucial for spatial-temporal data, indicating potential improvements in the spatial dependency modeling.

For spatial-temporal imputation, LRTC-TNN (Xinyu, Jinming, and Lijun 2020) uses low-rank tensor completion, GRIN (Cini, Marisca, and Alippi 2022) pioneers GNNs, and SPIN (Marisca, Cini, and Alippi 2022) targets error accumulation of GRIN for highly sparse data. STD-GAE (Fan et al. 2023) focuses on denoising graph autoencoders, while DAMR (Ren et al. 2023) dynamically extracts spatial correlations. PriSTI (Liu et al. 2023a) combines diffusion models with GNNs using GWN (Wu et al. 2019), learning rough graph structures. PriSTI (Liu et al. 2023a) combines diffusion models with GNNs using GWN (Wu et al. 2019), learning rough graph structures. Moreover, PoGeVon (Wang et al. 2023a) predict missing values over both node time series features and graph structures, ImputeFormer (Nie et al. 2024) and CASPER (Jing et al. 2024) utilize Transformer to capture spatial dependencies. However, these methods often overlook the heterogeneity and common spatial dependencies among different features within nodes (Chen et al. 2023; Chen, Wang, and Xu 2023; Chen et al. 2024). On the contrary, our GSLI addresses feature heterogeneity via node-scale graph structure learning and prominence modeling, while also capturing spatial dependencies between features through feature-scale graph structure learning.

Spatial-Temporal Graph Structure Learning In the early stages of modeling spatial-temporal data, researchers commonly use the inherent graph structure and GNN to learn spatial dependencies(Li et al. 2018; Zhao et al. 2020b). Pioneering the enhancement of spatial information within the given graph structure, GWN (Wu et al. 2019) introduces graph structure learning by assigning two learnable embedding vectors to each node. While methods like MTGNN (Wu et al. 2020) and GTS (Shang, Chen, and Bi 2021) design frameworks for learning discrete graph structures, AGCRN (Bai et al. 2020) and CCRNN (Ye et al. 2021) further these advancements by incorporating node-specific convolutions and learning independent graph structures for each convolution layer, respectively. SLCNN (Zhang et al. 2020) aimed to understand both global and local structural information in spatial-temporal data, whereas MegaCRN (Jiang et al. 2023) adapted graph structures based on input signals. CrossGNN (Huang et al. 2023) utilizes graph structure learning to adapt to multiple scales of temporal periods, and heterogeneity between all variables in the forecasting tasks. Unfortunately, these methods often focus on forecasting tasks, ignoring the crucial heterogeneity and correlation between features within nodes and differences in influence between nodes for the imputation task. Compared to the forecasting task, imputing missing values is more difficult to capture temporal dependencies with incomplete observations, thus requiring learning accurate fine-grained spatial dependencies. In contrast, our method captures feature-independent global spatial dependencies and spatial dependencies between features through the node and feature scales of graph structure learning, and reflects differences in node and feature influence by modeling prominence.

Methodology

In this section, we present the multi-scale Graph Structure Learning framework for spatial-temporal Imputation (GSLI). The framework is built upon node-scale spatial learning, feature-scale spatial learning, cross-feature representation learning, and cross-temporal representation learning.

Problem Definition

Spatial-temporal data $\langle \mathcal{G}, \mathbf{X} \rangle$ can be separated into two components: spatial correlation and temporal signal. The spatial correlation is represented by a static graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of N nodes and $\mathcal{E}$ is the set of edges reflect the inherent relationships between nodes. The adjacency matrix from $\mathcal{G}$ is denoted by $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $A_{ij} \in \mathbf{A}$ reflects the weight of the edge $\langle v_i, v_j \rangle \in \mathcal{E}$, $v_i, v_j \in \mathcal{V}$. The temporal signal $\mathbf{X} \in \mathbb{R}^{N \times T \times F}$ is the graph signal obtained by recording T consecutive timestamps of F features for each node over $\mathcal{G}$. The missing status of $\mathbf{X}$ can be expressed by the mask matrix $\mathbf{M} \in \{0,1\}^{N \times T \times F}$. If $m_{i,j,k} \in \mathbf{M}$ is equal to 0, it indicates that the observation $x_{i,j,k} \in \mathbf{X}$ is missing. Given the incomplete spatial-temporal data, we aim to estimate all missing values $\mathbf{X} \odot (1 - \mathbf{M})$ in the temporal signal $\mathbf{X}$, where $\odot$ denotes the Hadamard product.

Node-scale Spatial Learning

To address the challenge of feature heterogeneity, i.e. the objects recorded by the sensors in the station are from different domains, we learn the node-scale graph structure for each feature independently to capture the global spatial dependencies between nodes, as shown in Figure 2. We first split the input representation $\mathbf{R} \in \mathbb{R}^{N \times T \times F \times C} = \left\{ \mathbf{R}_f \in \mathbb{R}^{N \times T \times C} \right\}_{f=1}^{F}$ into F parts based on features, where C is the channel number in the deep space for each feature. Then, we adopt the canonical approach (Wu et al. 2019) to assign two learnable meta node embeddings $\mathbf{\Omega_f^1}, \mathbf{\Omega_f^2} \in \mathbb{R}^{N \times d}$ to each feature. We denote $\mathbf{\Omega_f^1}$ as the source node embedding and $\mathbf{\Omega_f^2}$ as the target node embedding. Since the average attention scores for each station obtained through the cross-feature self-attention mechanism are different, it inspires us that different nodes influence the overall imputation differently[1]. To account for the varying influence of different nodes on feature f, we use the source embedding to learn the prominence vector $\mathbf{P_f^\Omega} \in \mathbb{R}^{N \times d}$ for each node,

$$\mathbf{P_f^\Omega} = \mathrm{MLP}\left(\mathbf{\Omega_f^1}\right). \tag{1}$$

To keep the resulting shape not changing, we then utilize the Hadamard product to obtain the refined source embedding $\widehat{\mathbf{\Omega}}_f^1$ with $\mathbf{P_f^\Omega}$,

$$\widehat{\mathbf{\Omega}}_f^1 = \mathbf{\Omega_f^1} \odot \mathbf{P_f^\Omega}. \tag{2}$$

This means that edges sourced from highly influential nodes will carry stronger weight in the learned graph structure. The meta-graph $\dot{\mathcal{G}}_f^\Omega$ which represents the global spatial correlations specific to the feature f can be obtained by:

$$\dot{\mathbf{A}}_f^\Omega = \mathrm{SoftMax}\left[\mathrm{ReLU}\left(\widehat{\mathbf{\Omega}}_f^1 \mathbf{\Omega_f^2}^\top\right)\right], \tag{3}$$

where $\dot{\mathbf{A}}_f^\Omega$ is the adjacency matrix of $\dot{\mathcal{G}}_f^\Omega$, $\mathrm{ReLU}(\cdot)$ is applied to eliminate the weakly correlated edges of meta-graphs, the $\mathrm{SoftMax}(\cdot)$ function is used to normalize the adjacency matrices of meta-graphs.

By using the adjacency matrices of both the input graph $\mathcal{G}$ and the meta-graph $\dot{\mathcal{G}}_f^\Omega$, we can use graph diffusion convolution (Li et al. 2018) to capture the node-scale spatial dependencies of signal $\mathbf{R}_f$:

$$\mathbf{R_f^{NL}} = \Sigma_{k=0}^{K} \Bigg[\dot{\mathbf{A}}_f^\Omega \mathbf{R}_f \mathbf{\Theta}_{k,f}^{\mathbf{\Omega 1}} + \left(\mathbf{D^O}^{-1}\mathbf{A}\right)^k \mathbf{R}_f \mathbf{\Theta}_{k,f}^{\mathbf{\Omega 2}} + \left(\mathbf{D^I}^{-1}\mathbf{A}^\top\right)^k \mathbf{R}_f \mathbf{\Theta}_{k,f}^{\mathbf{\Omega 3}} \Bigg], \tag{4}$$

where K is the step number of the graph diffusion process, $\mathbf{\Theta}_{k,f}^{\mathbf{\Omega 1}}, \mathbf{\Theta}_{k,f}^{\mathbf{\Omega 2}}, \mathbf{\Theta}_{k,f}^{\mathbf{\Omega 3}} \in \mathbb{R}^{C \times C}$ are graph convolution kernels, $\mathbf{D^O}$ and $\mathbf{D^I}$ are the out-degree and in-degree matrices of $\mathbf{A}$, respectively. The output $\mathbf{R^{NL}} \in \mathbb{R}^{N \times T \times F \times C}$ of node-scale spatial learning is obtained by concatenating the graph diffusion convolution output $\mathbf{R_f^{NL}}$ from each feature,

$$\mathbf{R^{NL}} = \mathrm{Concat}\left(\mathbf{R_1^{NL}} \| \mathbf{R_2^{NL}} \ldots \| \mathbf{R_F^{NL}}\right). \tag{5}$$

[1]Please see Prominence Modeling section in Appendix (Yang et al. 2025) for details.

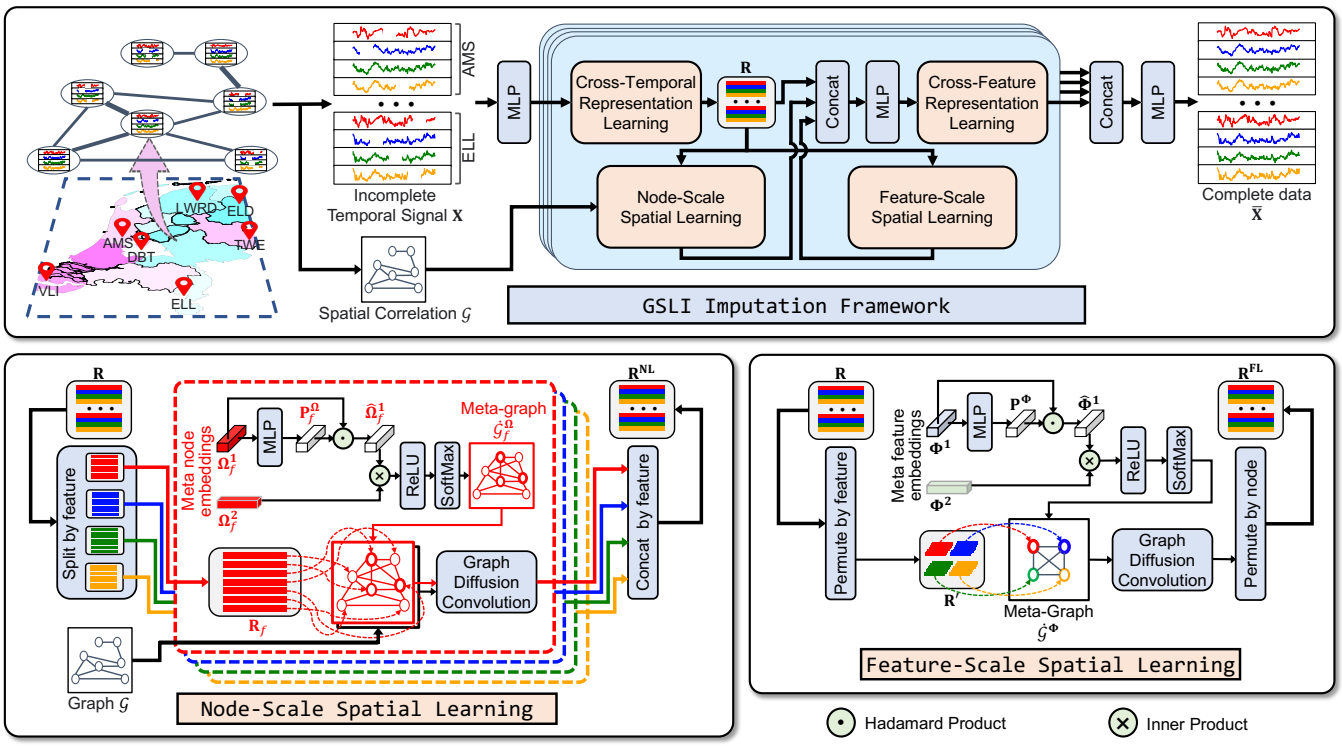

Figure 2: The overview of multi-scale Graph Structure Learning framework for spatial-temporal Imputation (GSLI). GSLI incorporates node-scale spatial learning, which can adapt to feature heterogeneity, and feature-scale spatial learning, which can exploit correlations between features. With cross-feature representation learning and cross-temporal representation learning, GSLI can effectively capture spatio-temporal dependencies for imputation.

This allows us to capture the spatial dependence of each feature independently and avoid the patterns corresponding to different features in this module interfering with each other. For the node-scale spatial learning, the time complexity is $\mathcal{O}(FN^2TC + FNTC^2 + FN^2d + FNd^2)$, the space complexity is $\mathcal{O}(FNTC + FC^2 + FN^2 + FNd + Fd^2)$. Please refer to the Complexity Analysis section in Appendix (Yang et al. 2025) for a detailed analysis.

If node i of the graph with feature heterogeneity satisfying the correlation weight of feature f_1 from node j to node i is x, and the correlation weight of feature f_2 from node j to node i is $y \neq x$, we find the canonical graph convolution cannot address the feature heterogeneity as follows.

Proposition 1. *The result of $\dot{\mathbf{A}}^{\mathbf{\Omega}}\mathbf{R}$ in the first term of the canonical graph diffusion convolution of the channel c for f_2 feature at timestamp t for the node i is*

$$a^{\mathbf{\Omega}}_{i1} r_{1,f_2,c} + \cdots + x r_{j,f_2,c} + \cdots + a^{\mathbf{\Omega}}_{iN} r_{N,f_2,c},$$

which is in conflict with the expected result $(a^{\mathbf{\Omega}}_{i1} r_{1,f_2,c} + \cdots + y r_{j,f_2,c} + \cdots + a^{\mathbf{\Omega}}_{iN} r_{N,f_2,c})$, where $\dot{\mathbf{A}}^{\mathbf{\Omega}} \in \mathbb{R}^{N\times N}$ is the learned global graph structure, $a^{\mathbf{\Omega}}_{ij} \in \dot{\mathbf{A}}^{\mathbf{\Omega}}$, $r_{1,f_2,c}, r_{N,f_2,c}, r_{j,f_2,c} \in \mathbf{R}$.

Under the same premise, our node-scale spatial learning can also adapt to feature heterogeneity, thus mitigating the misleading of heterogeneous features in neighboring nodes for imputation.

Proposition 2. *The result of $\dot{\mathbf{A}}^{\mathbf{\Omega}}_f \mathbf{R}_f$ in the first term of Equation 4 of the channel c for f_2 feature at timestamp t for the node i is capable to get the expected value:*

$$a^{\mathbf{\Omega}}_{f_2,i,1} r_{f_2,1} + \cdots + y r_{f_2,j} + \cdots + a^{\mathbf{\Omega}}_{f_2,i,N} r_{f_2,N}.$$

where $a^{\mathbf{\Omega}}_{f_2,i,j} \in \dot{\mathbf{A}}^{\mathbf{\Omega}}_{f_2}$, $r_{f_2,1}, r_{f_2,N}, r_{f_2,j} \in \mathbf{R}_f$.

The proofs are based on the information flow analysis, with details in Appendix.Proofs section (Yang et al. 2025).

Feature-scale Spatial Learning

For modeling the spatial correlation of different features over all nodes, we first define two learnable meta feature embeddings $\mathbf{\Phi^1}, \mathbf{\Phi^2} \in \mathbb{R}^{F\times d}$, where $\mathbf{\Phi^1}$ is the source feature embedding and $\mathbf{\Phi^2}$ is the target feature embedding. To reflect the influence and heterogeneity of different features, we model the prominence vector $\mathbf{P^\Phi} \in \mathbb{R}^{F\times d}$ for each feature,

$$\mathbf{P^\Phi} = \text{MLP}\left(\mathbf{\Phi^1}\right). \tag{6}$$

Next, we refine the source feature embedding $\mathbf{\Phi^1}$ by:

$$\widehat{\mathbf{\Phi}}^{\mathbf{1}} = \mathbf{\Phi^1} \odot \mathbf{P^\Phi}. \tag{7}$$

The source feature embeddings for features that are less heterogeneous from other features and can contribute to imputing missing values in other features will have stronger weights. With the inner product between the embeddings,

we can learn the meta-graph $\dot{\mathcal{G}}^{\mathbf{\Phi}}$ which represents the common spatial correlation of different features within each node,

$$\dot{\mathbf{A}}^{\mathbf{\Phi}} = \mathrm{SoftMax}\left[\mathrm{ReLU}\left(\widehat{\mathbf{\Phi}}^{\mathbf{1}}\mathbf{\Phi}^{\mathbf{2}\top}\right)\right], \tag{8}$$

where $\dot{\mathbf{A}}^{\mathbf{\Phi}}$ denotes the adjacency matrix of $\dot{\mathcal{G}}^{\mathbf{\Phi}}$.

To capture the spatial dependencies between different features, we first permute the input $\mathbf{R} \in \mathbb{R}^{N\times T\times F\times C}$ into $\mathbf{R}' \in \mathbb{R}^{F\times N\times T\times C}$ according to the features,

$$\mathbf{R}' = \mathrm{Permute}_{(2,0,1,3)}\left(\mathbf{R}\right). \tag{9}$$

Then we obtain $\mathbf{R'^{FL}}$ through the graph diffusion convolution layer based on $\dot{\mathcal{G}}^{\mathbf{\Phi}}$:

$$\mathbf{R'^{FL}} = \Sigma_{k=0}^{K}\dot{\mathbf{A}}^{\mathbf{\Phi}}\mathbf{R}'\mathbf{\Theta}_k^{\mathbf{\Phi}}, \tag{10}$$

where $\mathbf{\Theta}_k^{\mathbf{\Phi}} \in \mathbb{R}^{C\times C}$ are graph convolution kernels. The output of the feature-scale spatial learning $\mathbf{R^{FL}} \in \mathbb{R}^{N\times T\times F\times C}$ can obtain from permuting the output of the diffusion convolution layer according to the nodes of the input data,

$$\mathbf{R^{FL}} = \mathrm{Permute}_{(1,2,0,3)}\left(\mathbf{R'^{FL}}\right). \tag{11}$$

Cross-Feature Representation Learning

The goal of this phase is to self-adaptively obtain the representation that captures spatial dependencies between features across different nodes, which can be challenging to model a large number of spatial dependencies, i.e., $(N \times F)^2$. To overcome this challenge, we use the input representation $\mathbf{R}$, with the outputs $\mathbf{R^{NL}}$ and $\mathbf{R^{FL}}$ obtained from the two scales of spatial learning, as inputs for this stage.

We start by concatenating the three inputs based on features and fusing them using MLP to obtain $\mathbf{E} \in \mathbb{R}^{N\times T\times F\times C}$,

$$\mathbf{E} = \mathrm{MLP}\left[\mathrm{Concat}\left(\mathbf{R}\|\mathbf{R^{NL}}\|\mathbf{R^{FL}}\right)\right]. \tag{12}$$

Then we split $\mathbf{E} = \left\{\mathbf{E}_t \in \mathbb{R}^{(N\times F)\times C}\right\}_{t=1}^{T}$ into T segments according to timestamps and merge the node and feature dimensions of each segment. Taking advantage of the Transformer (Vaswani et al. 2017), the learning process to obtain cross-feature representation of each timestamp $\mathbf{Z}_t \in \mathbb{R}^{(N\times F)\times C}$is:

$$\mathbf{Z}_t = \mathrm{SoftMax}\left(\frac{\mathbf{Q}_t{\mathbf{K}_t}^{\top}}{\sqrt{C}}\right)\mathbf{V}_t, \tag{13}$$

where $\mathbf{Q}_t = \mathbf{E}_t\mathbf{W}_Q^{\mathbf{CF}}$, $\mathbf{K}_t = \mathbf{E}_t\mathbf{W}_K^{\mathbf{CF}}$, $\mathbf{V}_t = \mathbf{E}_t\mathbf{W}_V^{\mathbf{CF}}$, $\mathbf{W}_Q^{\mathbf{CF}}, \mathbf{W}_K^{\mathbf{CF}}, \mathbf{W}_V^{\mathbf{CF}} \in \mathbb{R}^{C\times C}$ are learnable parameters. Therefore, we can learn common spatial dependencies across different timestamps. The output cross-feature representation $\mathbf{Z} \in \mathbb{R}^{N\times T\times F\times C}$ is the result of flattening the node and feature dimensions and concatenating each timestamp,

$$\mathbf{Z} = \mathrm{Concat}\left[\left\{\mathrm{Flatten}_{(0,1)}\left(\mathbf{Z}_t\right)\right\}_{t=1}^{T}\right]. \tag{14}$$

The time complexity of this module is $\mathcal{O}(F^2NTC + FNTC^2 + F^2d + Fd^2)$, the space complexity is $\mathcal{O}(FNTC + C^2 + F^2 + Fd + d^2)$. For a detailed analysis, please see the Complexity Analysis section in Appendix (Yang et al. 2025).

Cross-Temporal Representation Learning

Our goal in this stage is to capture temporal dependencies that can improve the performance of imputation. Since capturing temporal dependencies on the original input signal is more reliable (Zhang, Zheng, and Qi 2017; Lim et al. 2021), we utilize the input temporal signal $\mathbf{X^I}$ of the spatial-temporal data as the input.

We first project $\mathbf{X^I}$ into deep space to obtain $\mathbf{H} \in \mathbb{R}^{N\times T\times F\times C}$,

$$\mathbf{H} = \mathrm{MLP}\left(\mathbf{X^I}\right). \tag{15}$$

For capturing dependencies between different timestamps, we split $\mathbf{H} = \left\{\mathbf{H}_y \in \mathbb{R}^{T\times C}\right\}_{y=1}^{(N\times K)}$ into $(N\times K)$ segments according all features across all nodes. Next, we can obtain the cross-temporal representation of a feature within a node $\mathbf{R}_y \in \mathbb{R}^{T\times C}$ by

$$\mathbf{R}_y = \mathrm{SoftMax}\left(\frac{\mathbf{Q}_y{\mathbf{K}_y}^{\top}}{\sqrt{C}}\right)\mathbf{V}_y, \tag{16}$$

where $\mathbf{Q}_y = \mathbf{H}_y\mathbf{W}_Q^{\mathbf{CT}}$, $\mathbf{K}_y = \mathbf{H}_y\mathbf{W}_K^{\mathbf{CT}}$, $\mathbf{V}_y = \mathbf{H}_y\mathbf{W}_V^{\mathbf{CT}}$. To get the cross-feature representation $\mathbf{R}$, we need to concatenate all $\mathbf{R}_y$ and flatten them based on the node to which the features belong,

$$\mathbf{R} = \mathrm{Flatten}_{(0,3)}\left[\mathrm{Concat}\left(\{\mathbf{R}_y\}_{y=1}^{N\times F}\right)\right]. \tag{17}$$

Therefore, we can learn common temporal dependencies across different features.

The Framework of GSLI

In this section, we introduce the multi-scale Graph Structure Learning framework for spatial-temporal data Imputation (GSLI). The framework mainly consists of multiple layers with the same architecture. Each layer incorporates our proposed node-scale spatial learning, feature-scale spatial learning, cross-feature representation learning, and cross-temporal representation learning. These components work together to capture the spatial-temporal dependencies required for accurate imputation. Following previous studies (Tashiro et al. 2021; Liu et al. 2023a; Nie et al. 2024), we first learn temporal dependencies and then learn spatial dependencies .

When training the framework, it's impossible to know the ground truth of real missing values. Therefore, we randomly selected some observations from $\mathbf{X}$ as the training label $\mathbf{X^B} \in \mathbf{X}$, $\mathbf{X^B} \in \mathbb{R}^{N\times T\times F\times C}$. We use $\mathbf{M^B}$ to represent the missing status of $\mathbf{X^B}$ and compose the remaining observations as the input signal $\mathbf{X^I} = \mathbf{X} \setminus \mathbf{X^B}$ to the framework. Then, we train GSLI by minimizing $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}\left\|\left(\mathbf{X^B} - \overline{\mathbf{X}}\right)\odot\mathbf{M^B}\right\|_2^2, \tag{18}$$

where $\overline{\mathbf{X}}$ is the output of the framework. Note that to encourage the framework to focus on more diverse temporal and spatial dependencies and enhance adaptability and flexibility, we select different $\mathbf{X^B}$ for each training step.

When imputing incomplete spatial-temporal data, we utilize the original temporal signal as the input signal $\mathbf{X^I} = \mathbf{X}$. The final estimated missing value we obtain is $\overline{\mathbf{X}}\odot(1-\mathbf{M})$.

Dataset	#Nodes	#Timestamps	#Features	Missing	Type
DutchWind (Institute 2023)	7	8688	4	0.92%	Wind
BeijingMEO (KDD-cup 2018a)	18	8784	5	0.81%	Meteo
LondonAQ (KDD-cup 2018b)	13	10897	3	13.81%	Air Quality
CN (Zheng et al. 2014)	140	2203	6	25.3%	Air Quality
Los (Zhao et al. 2020a)	207	2016	1	1.25%	Traffic
LuohuTaxi (Zhao et al. 2020a)	156	2976	1	24.76%	Traffic

Table 1: Dataset summary

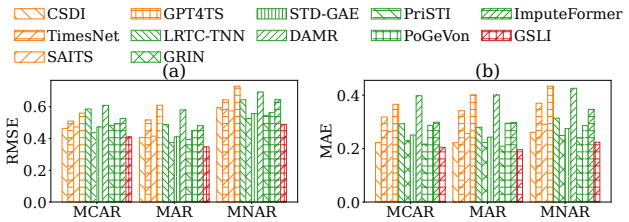

Figure 3: Varying the missing mechanism over DutchWind dataset with 10% missing values

Experiment

In this section, we evaluate the performance of our GSLI in imputation accuracy. The experiments are conduct on a machine equipped with an Intel Xeon Silver 4314 2.40GHz CPU and an NVIDIA GeForce RTX 4090 24GB GPU. The code and datasets are available online (Yang et al. 2025)[2].

Experimental Setup

Datasets In our experiments, we use six spatial-temporal datasets that have real-world missing values. Due to the unavailability of the ground truth of the missing values, we do not include them in the evaluation of imputation accuracy during comparative experiments, as noted in previous studies (Liu et al. 2023a; Ren et al. 2023). The main characteristics of these datasets are summarized in Table 1. For spatial information, since DutchWind, BeijingMEO, and LondonAQ do not explicitly provide adjacency matrices, we build adjacency matrices using the thresholded Gaussian kernel (Shuman et al. 2013) and the station coordinates following previous works (Liu et al. 2023a; Ren et al. 2023).

Baselines We compare with four state-of-the-art multivariate time series imputation methods: CSDI (Tashiro et al. 2021), TimesNet (Wu et al. 2023), SAITS (Du, Côté, and Liu 2023) and GPT4TS (Zhou et al. 2023), as well as seven spatial-temporal imputation methods: LRTC-TNN (Xinyu, Jinming, and Lijun 2020), GRIN (Cini, Marisca, and Alippi 2022), STD-GAE (Fan et al. 2023), DAMR (Ren et al. 2023), PriSTI (Liu et al. 2023a), PoGeVon (Wang et al. 2023a), and ImputeFormer (Nie et al. 2024).

Imputation Comparison

We first conduct experiments with various missing rates and missing mechanisms to evaluate the imputation of GSLI.

Since we can not access the ground truth of the missing values, we randomly remove different percentages of the observations as imputation labels for evaluation through the Missing Completely at Random (MCAR) mechanism (Bohannon et al. 2005). We evaluate the imputation performance using RMSE (Jeffery, Garofalakis, and Franklin 2006) and MAE (Chai and Draxler 2014) , as suggested by previous studies (Fan et al. 2023; Liu et al. 2023a). For both metrics, a smaller value indicates a more accurate imputation. We repeat each experiment five times and report the average results in Table 2. We can find that as the missing rate increases, the imputation performance of most methods decreases due to the reduction of information available to the imputation models Additionally, we observe that the current spatial-temporal imputation methods do not outperform the multivariate time series imputation methods in a significant manner. This can be attributed to the imprecision of the given graph structure affects the modeling of spatial dependencies. On the contrary, our GSLI achieves consistently superior performance to other methods over different datasets with various missing rates. This is because our GSLI accurately models spatial dependencies through graph structure learning at both the node and feature scales. Moreover, as we learn fine-grained spatial independent correlations for each feature, our method can outperform spatial-temporal imputation methods that only model inter-node dependencies.

Since the occurrence of missing data in real-world scenarios is usually related to the external environment or the sensors themselves, we additionally consider the missing at random (MAR) (Xia et al. 2017) and missing not at random (MNAR) (Twala 2009) mechanisms. Following existing works (Yoon, Jordon, and van der Schaar 2018; Miao et al. 2021a), we explore the imputation performance of different methods with various missing mechanisms. Figure 3 and Missing Mechanisms section in Appendix (Yang et al. 2025) illustrate the performance of different imputation methods with 10% missing values. We find that various imputation methods have similar performance levels with different missing mechanisms. As a result, we use MCAR by default in other experiments. In addition, our GSLI method consistently delivers optimal results across different missing mechanisms. This indicates that the GSLI can adapt well to different real-life missing data scenarios.

Ablation Study

To validate the efficiency of each component of GSLI, we set up the following ablation variants: **(1) TemporalGCN**: This variant utilizes cross-temporal representation learning to capture temporal dependencies, and the graph diffusion convolution to model spatial dependencies based on the existing adjacency matrix between nodes. **(2) TemporalFeatrueRL**: The temporal dependencies are captured using cross-temporal representation learning, and spatial dependencies are captured using cross-feature representation learning. **(3) w/o Cross-temporal**: We do not use cross-temporal representation learning to capture temporal dependencies in this situation. **(4) w/o Cross-feature**: We do not utilize cross-feature representation learning to model spatial dependencies between features across different nodes. **(5) w/o Feature-Split&Scale**: We replace our node-scale spatial learning and feature-scale spatial learning with the canonical Graph Diffusion Convolution. **(6) w/o Promi-**

[2]https://github.com/GSLI25/GSLI25/

Dataset	Missing rate	Metric	CSDI	TimesNet	SAITS	GPT4TS	LRTC-TNN	GRIN	STD-GAE	DAMR	PriSTI	PoGeVon	ImputeFormer	GSLI
DutchWind	10%	RMSE	0.464	0.510	0.473	0.560	0.586	0.437	0.473	0.609	0.483	0.493	0.526	**0.410**
		MAE	0.223	0.318	0.265	0.366	0.293	0.229	0.251	0.398	0.217	0.287	0.299	**0.205**
	20%	RMSE	0.490	0.620	0.482	0.669	0.611	0.441	0.490	0.611	0.489	0.472	0.542	**0.421**
		MAE	0.241	0.428	0.272	0.468	0.316	0.234	0.266	0.404	0.227	0.256	0.313	**0.213**
	30%	RMSE	0.505	0.717	0.498	0.755	0.659	0.450	0.518	0.615	0.507	0.497	0.551	**0.436**
		MAE	0.258	0.515	0.285	0.544	0.349	0.240	0.291	0.407	0.242	0.282	0.321	**0.223**
	40%	RMSE	0.526	0.787	0.504	0.817	0.705	0.455	0.564	0.621	0.537	0.569	0.565	**0.448**
		MAE	0.284	0.578	0.293	0.600	0.392	0.247	0.337	0.411	0.269	0.364	0.330	**0.234**
BeijingMEO	10%	RMSE	0.466	0.476	0.486	0.527	0.619	0.432	0.485	0.723	0.457	0.534	0.516	**0.399**
		MAE	0.208	0.306	0.290	0.358	0.322	0.242	0.291	0.510	0.213	0.353	0.283	**0.203**
	20%	RMSE	0.478	0.528	0.494	0.634	0.658	0.438	0.494	0.730	0.472	0.541	0.527	**0.407**
		MAE	0.217	0.365	0.303	0.483	0.343	0.248	0.297	0.517	0.229	0.347	0.295	**0.210**
	30%	RMSE	0.490	0.599	0.498	0.720	0.693	0.445	0.500	0.713	0.504	0.535	0.541	**0.415**
		MAE	0.227	0.440	0.306	0.571	0.365	0.254	0.301	0.503	0.262	0.333	0.307	**0.217**
	40%	RMSE	0.506	0.674	0.506	0.788	0.734	0.453	0.508	0.716	0.562	0.562	0.541	**0.423**
		MAE	0.240	0.514	0.316	0.636	0.390	0.262	0.306	0.509	0.301	0.364	0.311	**0.224**
LondonAQ	10%	RMSE	0.298	0.406	0.375	0.481	0.490	0.311	0.597	0.721	0.314	0.375	0.402	**0.272**
		MAE	0.182	0.264	0.249	0.321	0.310	0.198	0.471	0.493	0.192	0.232	0.262	**0.173**
	20%	RMSE	0.321	0.539	0.398	0.633	0.532	0.332	0.609	0.744	0.401	0.395	0.412	**0.305**
		MAE	0.190	0.358	0.259	0.429	0.334	0.204	0.475	0.499	0.206	0.240	0.266	**0.188**
	30%	RMSE	0.340	0.658	0.413	0.741	0.584	0.351	0.665	0.786	0.502	0.415	0.436	**0.320**
		MAE	0.199	0.449	0.265	0.508	0.367	0.213	0.510	0.522	0.234	0.250	0.275	**0.191**
	40%	RMSE	0.375	0.747	0.429	0.811	0.642	0.360	0.631	0.782	0.624	0.428	0.433	**0.335**
		MAE	0.212	0.520	0.278	0.564	0.405	0.223	0.495	0.522	0.280	0.263	0.279	**0.205**
CN	10%	RMSE	0.472	0.668	0.474	0.490	0.561	0.403	0.370	0.879	0.387	0.634	0.389	**0.253**
		MAE	0.182	0.458	0.285	0.315	0.357	0.241	0.205	0.612	0.179	0.394	0.204	**0.120**
	20%	RMSE	0.436	0.698	0.481	0.509	0.596	0.417	0.388	0.884	0.423	0.633	0.401	**0.267**
		MAE	0.194	0.485	0.289	0.333	0.387	0.251	0.216	0.622	0.197	0.394	0.210	**0.129**
	30%	RMSE	0.442	0.733	0.490	0.544	0.644	0.434	0.405	0.906	0.477	0.632	0.418	**0.282**
		MAE	0.211	0.515	0.297	0.365	0.426	0.264	0.228	0.629	0.226	0.399	0.223	**0.139**
	40%	RMSE	0.465	0.766	0.499	0.593	0.706	0.451	0.425	0.905	0.512	0.582	0.444	**0.299**
		MAE	0.233	0.545	0.304	0.409	0.471	0.277	0.243	0.629	0.262	0.349	0.237	**0.150**
Los	10%	RMSE	0.311	0.531	0.535	0.397	0.501	0.295	0.945	0.513	0.293	0.365	0.445	**0.263**
		MAE	0.177	0.339	0.292	0.257	0.332	0.188	0.541	0.363	0.186	0.209	0.227	**0.159**
	20%	RMSE	0.333	0.560	0.538	0.452	0.557	0.306	0.942	0.515	0.313	0.377	0.442	**0.273**
		MAE	0.185	0.365	0.296	0.297	0.372	0.194	0.539	0.362	0.200	0.216	0.223	**0.164**
	30%	RMSE	0.355	0.602	0.544	0.543	0.618	0.319	0.944	0.521	0.353	0.390	0.459	**0.282**
		MAE	0.193	0.401	0.299	0.356	0.415	0.201	0.541	0.368	0.225	0.223	0.230	**0.168**
	40%	RMSE	0.398	0.649	0.558	0.651	0.667	0.332	0.944	0.525	0.408	0.401	0.461	**0.294**
		MAE	0.210	0.440	0.306	0.422	0.454	0.208	0.541	0.363	0.267	0.230	0.231	**0.173**
LuohuTaxi	10%	RMSE	0.467	0.514	0.452	0.576	0.680	0.436	0.783	0.667	0.523	0.497	0.456	**0.410**
		MAE	0.313	0.388	0.308	0.429	0.459	0.301	0.612	0.523	0.367	0.340	0.309	**0.276**
	20%	RMSE	0.468	0.540	0.455	0.692	0.718	0.439	0.780	0.671	0.525	0.498	0.453	**0.414**
		MAE	0.315	0.415	0.312	0.503	0.489	0.304	0.612	0.527	0.363	0.342	0.308	**0.279**
	30%	RMSE	0.472	0.577	0.456	0.773	0.767	0.443	0.781	0.684	0.531	0.503	0.456	**0.419**
		MAE	0.319	0.450	0.312	0.563	0.530	0.307	0.612	0.538	0.370	0.346	0.311	**0.283**
	40%	RMSE	0.474	0.621	0.461	0.828	0.780	0.448	0.781	0.684	0.537	0.511	0.456	**0.424**
		MAE	0.323	0.488	0.317	0.608	0.548	0.311	0.613	0.539	0.374	0.351	0.311	**0.287**

Table 2: Imputation performance of GSLI compared to existing methods with various missing rates

Method	DutchWind		BeijingMEO		LondonAQ		CN	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
TemporalGCN	0.4453	0.2335	0.4175	0.2189	0.3133	0.2023	0.2989	0.1492
TemporalFeatrueRL	0.4223	0.2062	0.4111	0.2109	0.2926	0.1856	0.3053	0.1520
w/o Cross-temporal	0.4221	0.2124	0.4308	0.2333	0.3515	0.2302	0.3723	0.2104
w/o Cross-feature	0.4140	0.2070	0.4160	0.2214	0.3079	0.1999	0.2609	0.1255
w/o Feature-Split&Scale	0.4147	0.2103	0.4018	0.2051	0.2825	0.1812	0.3100	0.1443
w/o Prominence	0.4132	0.2076	0.4041	0.2080	0.2809	0.1799	0.2595	0.1240
w/o Node-scale	0.4130	0.2081	0.4015	0.2055	0.2845	0.1844	0.2631	0.1263
w/o Feature-scale	0.4213	0.2057	0.4083	0.2093	0.2966	0.1877	0.2932	0.1460
w/o GSL	0.4218	0.2060	0.4090	0.2095	0.2990	0.1907	0.2962	0.1475
GSLI	**0.4101**	**0.2051**	**0.3986**	**0.2034**	**0.2720**	**0.1730**	**0.2534**	**0.1202**

Table 3: Ablation analysis of GSLI with 10% missing values

nence: When modeling spatial dependencies, the graph structure learning for both scales does not model the prominence of nodes in meta-graphs. **(7) w/o Node-scale**: We only model the spatial correlation between different features for graph structure learning. **(8) w/o Feature-scale**: We learn different node-scale graph structures for different features, and ignore the spatial correlation between features. **(9) w/o GSL**: We do not perform any graph structure learning. Instead, we use the graph diffusion convolution that takes the given adjacency matrix as input and the cross-feature representation learning to learn the spatial dependencies.

As these variants involve verifying the role of learning graph structures between different features, we performed ablation experiments on four datasets that recorded multiple features, the results are shown in Table 3. The results indicate that each component of the GSLI is crucial, especially in learning the common spatial correlation between different features within nodes for graph structure learning, and performing cross-temporal representation learning. Notably, TemporalGCN consistently performs less than TemporalFeatureSA. This suggests that the given adjacency matrix between nodes cannot accurately reflect the spatial correlations in reality. This also confirms the necessity of adopting different scales for learning graph structures.

Conclusion

In this work, we design the multi-scale Graph Structure Learning framework for spatial-temporal Imputation (GSLI), addressing the challenges of imputing missing values in spatial-temporal data due to feature heterogeneity and latent common correlation between features among all nodes. By applying node-scale and feature-scale graph structure learning alongside prominence modeling, GSLI improves the imputation accuracy, as demonstrated across six diverse datasets with real missing values.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (62302241, 62372252, 72342017, 62306085, 62272250), the Natural Science Foundation of Tianjin (No. 22JCJQJC00150), Shenzhen College Stability Support Plan (GXWD20231130151329002).

References

Acuna, E.; and Rodriguez, C. 2004. The treatment of missing values and its effect on classifier accuracy. In *IFCS*.

Alcaraz, J. M. L.; and Strodthoff, N. 2023. Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models. *TMLR*.

Amiri, M.; and Jensen, R. 2016. Missing data imputation using fuzzy-rough methods. *Neurocomputing*, 205: 152–164.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *NeurIPS*.

Bohannon, P.; Flaster, M.; Fan, W.; and Rastogi, R. 2005. A Cost-Based Model and Effective Heuristic for Repairing Constraints by Value Modification. In *SIGMOD*.

Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; and Li, Y. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. In *NeurIPS*.

Chai, T.; and Draxler, R. R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3): 1247–1250.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D. A.; and Liu, Y. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific reports*, 8(1): 6085.

Chen, X.; and Sun, L. 2022. Bayesian Temporal Factorization for Multidimensional Time Series Prediction. *IEEE TPAMI*, 44(9): 4659–4673.

Chen, Y.; Shi, K.; Wang, X.; and Xu, G. 2023. MTSTI: A Multi-task Learning Framework for Spatiotemporal Imputation. In *ADMA*.

Chen, Y.; Shi, K.; Wu, Z.; Chen, J.; Wang, X.; McAuley, J. J.; Xu, G.; and Yu, S. 2024. A Temporally Disentangled Contrastive Diffusion Model for Spatiotemporal Imputation. *CoRR*, abs/2402.11558.

Chen, Y.; Wang, X.; and Xu, G. 2023. GATGPT: A Pre-trained Large Language Model with Graph Attention Network for Spatiotemporal Imputation. *CoRR*, abs/2311.14332.

Choi, M.; and Lee, C. 2024. Conditional Information Bottleneck Approach for Time Series Imputation. In *ICLR*.

Cini, A.; Marisca, I.; and Alippi, C. 2022. Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks. In *ICLR*.

Du, W.; Côté, D.; and Liu, Y. 2023. SAITS: Self-attention-based imputation for time series. *ESWA*, 219: 119619.

Fan, Y.; Yu, X.; Wieser, R.; Meakin, D.; Shaton, A.; Jaubert, J.; Flottemesch, R.; Howell, M.; Braid, J.; Bruckman, L. S.; French, R. H.; and Wu, Y. 2023. Spatio-Temporal Denoising Graph Autoencoders with Data Augmentation for Photovoltaic Data Imputation. In *SIGMOD*.

Fortuin, V.; Baranchuk, D.; Rätsch, G.; and Mandt, S. 2020. GP-VAE: Deep Probabilistic Time Series Imputation. In *AISTATS*.

Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; and Wang, Y. 2023. CrossGNN: Confronting Noisy Multivariate Time Series Via Cross Interaction Refinement. In *NeurIPS*.

Institute, R. N. M. 2023. Release page for the Dutch dataset. https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens/.

Jeffery, S. R.; Garofalakis, M. N.; and Franklin, M. J. 2006. Adaptive Cleaning for RFID Data Streams. In *VLDB*.

Jiang, R.; Wang, Z.; Yong, J.; Jeph, P.; Chen, Q.; Kobayashi, Y.; Song, X.; Fukushima, S.; and Suzumura, T. 2023. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. In *AAAI*.

Jing, B.; Zhou, D.; Ren, K.; and Yang, C. 2024. CASPER: Causality-Aware Spatiotemporal Graph Neural Networks for Spatiotemporal Time Series Imputation. *CoRR*, abs/2403.11960.

Kantardzic, M. 2011. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

KDD-cup. 2018a. Release page for the Beijing dataset. https://www.dropbox.com/s/jjta4addnyjndd8/.

KDD-cup. 2018b. Release page for the London dataset. https://www.dropbox.com/s/ht3yzx58orxw179/.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Lim, B.; Arık, S.; Loeff, N.; and Pfister, T. 2021. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

Liu, M.; Huang, H.; Feng, H.; Sun, L.; Du, B.; and Fu, Y. 2023a. PriSTI: A Conditional Diffusion Framework for Spatiotemporal Imputation. In *ICDE*.

Liu, S.; Li, X.; Cong, G.; Chen, Y.; and Jiang, Y. 2023b. Multivariate Time-series Imputation with Disentangled Temporal Representations. In *ICLR*.

Luo, Y.; Cai, X.; Zhang, Y.; Xu, J.; and Yuan, X. 2018. Multivariate Time Series Imputation with Generative Adversarial Networks. In *NeurIPS*.

Luo, Y.; Zhang, Y.; Cai, X.; and Yuan, X. 2019. E^2GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation. In *IJCAI*.

Marisca, I.; Cini, A.; and Alippi, C. 2022. Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations. In *NeurIPS*.

Mattei, P.; and Frellsen, J. 2019. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. In *ICML*.

Miao, X.; Wu, Y.; Chen, L.; Gao, Y.; Wang, J.; and Yin, J. 2021a. Efficient and Effective Data Imputation with Influence Functions. In *VLDB*.

Miao, X.; Wu, Y.; Wang, J.; Gao, Y.; Mao, X.; and Yin, J. 2021b. Generative Semi-supervised Learning for Multivariate Time Series Imputation. In *AAAI*.

Nie, T.; Qin, G.; Ma, W.; Mei, Y.; and Sun, J. 2024. ImputeFormer: Low Rankness-Induced Transformers for Generalizable Spatiotemporal Imputation. In *KDD*.

Ren, X.; Zhao, K.; Riddle, P. J.; Taskova, K.; Pan, Q.; and Li, L. 2023. DAMR: Dynamic Adjacency Matrix Representation Learning for Multivariate Time Series Imputation. In *SIGMOD*.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *ICLR*.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE SPMAG*, 30(3): 83–98.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *NeurIPS*.

Twala, B. 2009. An Empirical Comparison of Techniques for Handling Incomplete Data Using Decision Trees. *AAI*, 23(5): 373–405.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*.

Wang, D.; Yan, Y.; Qiu, R.; Zhu, Y.; Guan, K.; Margenot, A.; and Tong, H. 2023a. Networked Time Series Imputation via Position-aware Graph Enhanced Variational Autoencoders. In *KDD*.

Wang, X.; Zhang, H.; Wang, P.; Zhang, Y.; Wang, B.; Zhou, Z.; and Wang, Y. 2023b. An Observed Value Consistent Diffusion Model for Imputing Missing Values in Multivariate Time Series. In *KDD*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*.

Xia, J.; Zhang, S.; Cai, G.; Li, L.; Pan, Q.; Yan, J.; and Ning, G. 2017. Adjusted weight voting algorithm for random forests in handling missing values. *PR*, 69: 52–60.

Xinyu, C.; Jinming, Y.; and Lijun, S. 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *TRC*, 117: 102673.

Xu, Q.; Ruan, S.; Long, C.; Yu, L.; and Zhang, C. 2022. Traffic Speed Imputation with Spatio-Temporal Attentions and Cycle-Perceptual Training. In *CIKM*.

Yang, X.; Sun, Y.; Chen, X.; Zhang, Y.; and Yuan, X. 2025. Our code, data and appendix. https://github.com/GSLI25/GSLI25/.

Ye, J.; Sun, L.; Du, B.; Fu, Y.; and Xiong, H. 2021. Coupled Layer-wise Graph Convolution for Transportation Demand Prediction. In *AAAI*.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2018. GAIN: Missing Data Imputation using Generative Adversarial Nets. In *ICML*.

Yu, H.; Rao, N.; and Dhillon, I. S. 2016. Temporal Regularized Matrix Factorization for High-dimensional Time Series Prediction. In *NeurIPS*.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*, 1655–1661.

Zhang, Q.; Chang, J.; Meng, G.; Xiang, S.; and Pan, C. 2020. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. In *AAAI*.

Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; and Li, H. 2020a. Release page for the Los and LuohuTaxi datasets. https://github.com/lehaifeng/T-GCN/tree/master/T-GCN/T-GCN-PyTorch/data/.

Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; and Li, H. 2020b. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE TITS*, 21(9): 3848–3858.

Zheng, Y.; Yi, X.; Li, M.; Li, R.; Shan, Z.; Chang, E.; and Li, T. 2014. Release page for the CN dataset. http://research.microsoft.com/apps/pubs/?id=246398/.

Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *NeurIPS*.