

Spatial-Temporal Feedback Diffusion Guidance for Controlled Traffic Imputation

Xiaowei Mao^{1*}, Huihu Ding^{1*}, Yan Lin⁴, Tingrui Wu¹,
Shengnan Guo^{1,2}, Dazhuo Qiu⁴, Feiling Fang⁵, Jilin Hu⁶, Huaiyu Wan^{1,3†}

¹School of Computer Science and Technology, Beijing Jiaotong University, China

²Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, China

³Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, China

⁴Department of Computer Science, Aalborg University, Denmark

⁵School of Artificial Intelligence, China University of Geoscience, Beijing, China

⁶School of Data Science and Engineering, East China Normal University, China

{maoxiaowei, 22231044, guoshn, hywan}@bjtu.edu.cn, liyan@cs.aau.dk, jlhu@dase.ecnu.edu.cn

Abstract

Imputing missing values in spatial-temporal traffic data is essential for intelligent transportation systems. Among advanced imputation methods, score-based diffusion models have demonstrated competitive performance. These models generate data by reversing a noising process, using observed values as conditional guidance. However, existing diffusion models typically apply a uniform guidance scale across both spatial and temporal dimensions, which is inadequate for nodes with high missing data rates. Sparse observations provide insufficient conditional guidance, causing the generative process to drift toward the learned prior distribution rather than closely following the conditional observations, resulting in suboptimal imputation performance.

To address this, we propose FENCE, a spatial-temporal feedback diffusion guidance method designed to adaptively control guidance scales during imputation. First, FENCE introduces a dynamic feedback mechanism that adjusts the guidance scale based on the posterior likelihood approximations. The guidance scale is increased when generated values diverge from observations and reduced when alignment improves, preventing overcorrection. Second, because alignment to observations varies across nodes and denoising steps, a global guidance scale for all nodes is suboptimal. FENCE computes guidance scales at the cluster level by grouping nodes based on their attention scores, leveraging spatial-temporal correlations to provide more accurate guidance. Experimental results on real-world traffic datasets show that FENCE significantly enhances imputation accuracy.

Code — <https://github.com/maoxiaowei97/FENCE>

Introduction

Spatial-temporal traffic data are often represented as a graph of spatial-temporal time series, where each node is a traffic sensor continuously collecting observations and edges describe sensor relationships (Guo et al. 2024). Traffic data is essential for Intelligent Transportation Systems (ITS), supporting key services in ITS, such as real-time traffic display,

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[†]Corresponding author.

*These authors contributed equally.

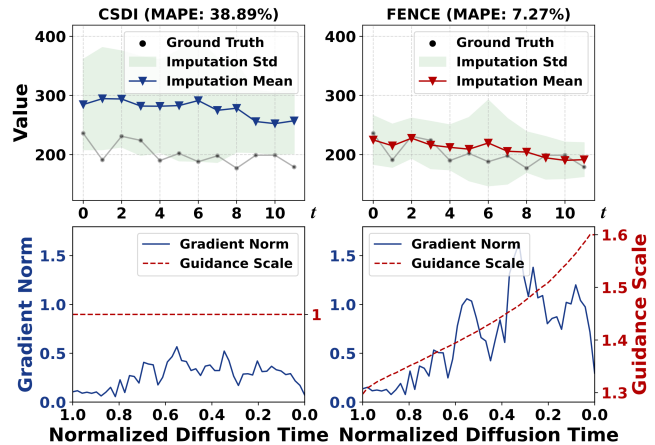


Figure 1: Motivation for FENCE. Unlike CSDI, which uses a fixed guidance scale, FENCE dynamically adjusts guidance scale based on consistency with observed data.

traffic prediction, and traffic signal control. However, this data is frequently incomplete due to equipment malfunctions, and network failures. These missing values degrade the performance of dependent applications. Consequently, imputing missing values is essential to ensure data quality and reliability (Wang et al. 2024; Miao et al. 2022).

Deep learning paradigms for spatial-temporal imputation can be broadly categorized into two main approaches: discriminative and generative models. Discriminative models directly learn a mapping function from observed to missing values using architectures like Recurrent Neural Networks (RNNs) (Miao et al. 2021), Graph Neural Networks (GNNs) (Lao et al. 2022), and Transformers (Nie et al. 2024). While often straightforward to train, their focus on direct prediction limits their ability to capture data distributions and the uncertainty associated with missing values.

In contrast, generative models aim to learn the underlying probability distribution of the data, enabling high-fidelity imputations (Luo et al. 2019; Yoon, Jordon, and Schaar 2018). Imputation is formulated as a conditional generation task, where missing values are sampled from this learned distribution based on the observed data (Zhou et al. 2024).

Among these, score-based diffusion models (Song et al. 2020) have emerged as a competitive method for imputation. These models learn the score function, defined as the gradient of the data’s log-likelihood, and utilize the conditional score to guide the generation process.

However, these models often yield suboptimal performance, particularly for nodes with high missing data rates. As illustrated in Fig. 1, a node with no observations during a time period is imputed inaccurately by CSDI (Tashiro et al. 2021), where even the estimated lower and upper bounds may fail to encompass the ground truth. This issue can be quantitatively assessed by examining the generative process. The degree to which the generated data \mathbf{x}_k satisfies condition \mathbf{c} is measured by the posterior likelihood, $p_{\theta,k}(\mathbf{c}|\mathbf{x}_k)$. To improve this likelihood, the diffusion model is guided by the gradient of the log-posterior likelihood. This guidance term is approximated by the difference between the conditional and unconditional score functions: $\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k|\mathbf{c}) - \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k)$. The L2-norm of this gradient vector quantifies the guidance strength. As shown in Fig. 1, during the generation process of CSDI, the node without observations exhibits a consistently low gradient norm. This indicates that the learned conditional distribution has collapsed to the unconditional prior. Consequently, the generative process is biased towards sampling from the marginal distribution $p_{\theta,k}(\mathbf{x}_k)$ instead of the conditional distribution $p_{\theta,k}(\mathbf{x}_k|\mathbf{c})$. Existing diffusion models for imputation lack mechanisms to control the guidance strength, leading to insufficient adherence to specific conditional observations and suboptimal performance.

To address this issue, we propose FENCE (Spatial-Temporal FEedback Diffusion Guidance), a novel method for controlled traffic imputation that dynamically adjusts the guidance scales throughout the generative process. FENCE introduces a feedback mechanism that adjusts the guidance scale based on an approximation of the posterior likelihood. Specifically, when the posterior likelihood decreases, indicating that the generated values do not sufficiently adhere to the conditional observations, the guidance scale is increased to enhance alignment with the observations. In contrast, when the posterior likelihood is high, indicating good alignment between the generated values and the observations, the guidance scale is reduced to avoid overcorrection. Furthermore, to account for varying degrees of alignment with conditional observations across different nodes, FENCE computes the guidance scale at the cluster level. By leveraging spatial-temporal correlations, FENCE ensures more accurate guidance scale adjustments for nodes with limited data availability, thereby improving the imputation quality.

Our contributions are summarized as follows:

- We propose FENCE, a spatial-temporal feedback diffusion guidance method that dynamically controls the guidance scales during the generative process, ensuring high-fidelity imputation of missing traffic data.
- We propose a cluster-aware guidance mechanism that leverages spatial-temporal correlations to compute accurate guidance scales tailored to each node.

- Extensive experiments show that FENCE significantly enhances the imputation accuracy in real-world spatial-temporal traffic datasets.

Related Work

Spatial-Temporal Imputation. Spatial-temporal imputation methods can be broadly classified into discriminative and generative paradigms. Discriminative models (Cao et al. 2018; Che et al. 2018; Weng et al. 2025), such as SAITS (Du, Côté, and Liu 2023) and ImputeFormer (Nie et al. 2024), learn deterministic mappings from observed data but fail to explicitly model data distributions.

In contrast, generative models aim to learn the underlying data distribution and treat imputation as conditional sampling, generating plausible values for the missing entries given the observed data (Yoon, Jordon, and Schaar 2018; Fortuin et al. 2020; Ipsen, Mattei, and Frellsen 2022).

Score-based diffusion models are powerful generative models for imputation. These models learn the score function, which is the gradient of the log-likelihood of the data distribution. During imputation, they leverage the score of the conditional distribution to estimate missing values. Models such as CSDI (Tashiro et al. 2021) and MIDM (Wang et al. 2023) condition the diffusion process on available observations. Several extensions further enhance conditioning: LSCD (Fons et al. 2025) incorporates spectral information; and PriSTI (Liu et al. 2023) integrates geographic context. To improve imputation consistency and inference speed, CSBI (Chen et al. 2023) leverages a Schrödinger bridge formulation; MTSCI (Zhou et al. 2024) generates multiple masks and auxiliary conditions during training; DSDI (Xiao et al. 2025) incorporates the predicted values into the denoising process, and CoSTI (Solís-García et al. 2025) employs consistency training to reduce inference times.

Despite their effectiveness in modeling complex distributions, diffusion models face challenges in spatial-temporal imputation, especially for nodes with high missing rates. In such cases, the limited observed data may be insufficient to effectively guide the model from its learned prior to converge to the true conditional distribution. Consequently, the imputed values often reflect general data patterns rather than adhering to the available observations. A key limitation is that current models lack mechanisms to dynamically adjust the scales of guidance strength based on the observed data.

Preliminaries

Definition 1. Traffic Network. We define the traffic network as a graph, i.e., $G = (V, E, \mathbf{A})$, where V represents the set of $|V| = N$ nodes (e.g., traffic sensors). E represents the set of edges. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix.

Definition 2. Traffic Data. Let $x_{v,t} \in \mathbb{R}$ denote the traffic data observed at node $v \in V$ at time slice t . The traffic data at time t across all nodes is $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \dots, x_{N,t}) \in \mathbb{R}^N$, and the traffic data over T time slices is $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T) \in \mathbb{R}^{N \times T}$.

Definition 3. Mask Matrix. To indicate the missing position in the observed traffic data, we introduce an observation masking matrix $\mathbf{M} \in \mathbb{R}^{N \times T}$, where $m_{v,t} = 0$ when $x_{v,t}$ is

missing, and $m_{v,t} = 1$ when $x_{v,t}$ is observed. The observed values in \mathbf{x} are denoted as $\mathbf{x}^o = \mathbf{x} \odot \mathbf{M}$, and the missing values in \mathbf{x} are denoted as $\mathbf{x}^m = \mathbf{x} \odot (\mathbf{1} - \mathbf{M})$.

Spatial-Temporal Traffic Imputation. Given the incomplete traffic observations \mathbf{x} , the mask matrix $\mathbf{M} \in \mathbb{R}^{N \times T}$ over T time slices, and a network graph G , the objective is to estimate the missing values in \mathbf{x} such that the estimation error at the missing positions is minimized.

Methods

This section presents our controlled traffic imputation method, beginning with guidance in diffusion models for imputation. We then propose spatial-temporal feedback diffusion guidance, followed by the theoretical foundations, key mechanisms, and procedures for training and inference.

Guidance in Diffusion Models for Traffic Imputation

In spatial-temporal diffusion models for traffic imputation, guidance is achieved by conditioning the reverse process on observed data and structural priors. This conditional information \mathbf{c} , is produced by a conditioning network \mathcal{F}_{cond} , which captures temporal and spatial dependencies. Given the observed data \mathbf{x}^o , the network first employs temporal attention to capture dependencies across time for each node, followed by spatial attention to aggregate information across nodes for each time slice. The conditioning also incorporates structural priors, such as node embeddings $\mathbf{E}_{node} \in \mathbb{R}^{N \times d}$ and learnable time slice embeddings $\mathbf{E}_{time} \in \mathbb{R}^{T \times d}$. The resulting vector, $\mathbf{c} = \mathcal{F}_{cond}(\mathbf{x}^o, \mathbf{E}_{node}, \mathbf{E}_{time})$, then guides the reverse process. Starting from Gaussian noise $\mathbf{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the model iteratively denoises the sample, with each step of the reverse process conditioned on \mathbf{c} :

$$p_\theta(\mathbf{x}_{k-1} | \mathbf{x}_k, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{k-1}; \mu_\theta(\mathbf{x}_k, k, \mathbf{c}), \sigma_k^2 \mathbf{I}), \quad (1)$$

where the mean μ_θ is parameterized using a denoising network ϵ_θ that predicts the added noise at step k :

$$\mu_\theta(\mathbf{x}_k, k, \mathbf{c}) = \frac{1}{\sqrt{\alpha_k}} \left(\mathbf{x}_k - \frac{1 - \alpha_k}{\sqrt{1 - \bar{\alpha}_k}} \epsilon_\theta(\mathbf{x}_k, k, \mathbf{c}) \right) \quad (2)$$

The denoising network ϵ_θ is trained to learn the conditional score function, where the score is proportional to the predicted noise: $\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k | \mathbf{c}) = -\frac{1}{\sqrt{1 - \bar{\alpha}_k}} \epsilon_\theta(\mathbf{x}_k, k, \mathbf{c})$. This score function provides the gradient direction to iteratively guide the sample \mathbf{x}_k to maximize its likelihood under the learned conditional distribution.

While the objective of a diffusion model is to maximize the data likelihood, this does not ensure the maximization of the posterior likelihood, $p_{\theta,k}(\mathbf{c} | \mathbf{x}_k)$, which quantifies how well a sample satisfies the observations. To better align the generated data with the observations, the reverse process can be guided by the gradient of the log-posterior likelihood, $\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{c} | \mathbf{x}_k)$, which can be formulated in terms of learnable components using Bayes' theorem:

$$\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{c} | \mathbf{x}_k) = \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k | \mathbf{c}) - \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k) \quad (3)$$

This equation reveals that the guidance can be achieved by subtracting the unconditional score, $\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k)$, from the conditional score. Classifier Free Guidance (CFG) (Ho and Salimans 2022) provides an efficient implementation by training a single denoising network to learn both scores. This is achieved by randomly providing the network during training with either the conditioning vector or an unconditional vector. The latter is constructed to model the prior distribution by feeding the conditioning network empty observations while retaining the structural priors. Using these scores, CFG constructs a guidance score by leveraging the gradient of the log-posterior likelihood. Specifically, the guidance score, $\nabla \log \tilde{p}_{\theta,k}(\mathbf{x}_k | \mathbf{c})$, is constructed by scaling this gradient (as derived in Eq. 3) by a guidance scale λ and adding it to the unconditional score:

$$\begin{aligned} \nabla_{\mathbf{x}_k} \log \tilde{p}_{\theta,k}(\mathbf{x}_k | \mathbf{c}) &= \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k) \\ &+ \lambda (\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k | \mathbf{c}) \\ &- \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k)) \end{aligned} \quad (4)$$

The guidance scale λ is a hyperparameter that adjusts the strength of the conditioning signal.

Spatial-Temporal Feedback Diffusion Guidance

CFG applies a uniform guidance scale λ to all nodes across all denoising steps. This approach has two limitations for traffic imputation. First, λ is a fixed hyperparameter that is difficult to optimize. Second, it fails to account for the varying degrees to which imputed values for different nodes satisfy the conditional observations at different times.

To address this, we introduce a feedback guidance mechanism that adaptively adjusts the guidance scales based on the posterior likelihood at each denoising step k . The posterior likelihood quantifies the alignment between the generated sample \mathbf{x}_k and the condition \mathbf{c} . When the posterior likelihood is high, indicating good alignment with the observed data, then the guidance scale remains low to avoid overcorrection. In contrast, when the posterior likelihood decreases, the guidance scale is increased to ensure adherence to the conditional information.

Posterior-Driven Dynamic Guidance Scaling To enable controlled imputation, we begin by introducing a global guidance mechanism. This approach treats the traffic data matrix comprising all nodes over T time slices as a sample. At each denoising step k , a unified guidance scale, denoted as $\lambda(\mathbf{x}_k, k)$, is dynamically adjusted for the corresponding noised sample \mathbf{x}_k based on the posterior likelihood. This scale controls the guidance vector, which is the difference between the conditional and unconditional score estimates. We define the unconditional score as $s_\theta(\mathbf{x}_k) := \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k)$ and the conditional score as $s_\theta(\mathbf{x}_k, \mathbf{c}) := \nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k | \mathbf{c})$. The resulting guidance score is:

$$\begin{aligned} \nabla_{\mathbf{x}_k} \log \tilde{p}_{\theta,k}(\mathbf{x}_k | \mathbf{c}) &= s_\theta(\mathbf{x}_k) \\ &+ \lambda(\mathbf{x}_k, k) (s_\theta(\mathbf{x}_k, \mathbf{c}) - s_\theta(\mathbf{x}_k)) \end{aligned} \quad (5)$$

To compute the guidance scale, we adopt the additive error formulation (Koulischer et al. 2025). This formulation assumes that the learned conditional distribution $p_{\theta,k}(\mathbf{x}_k | \mathbf{c})$

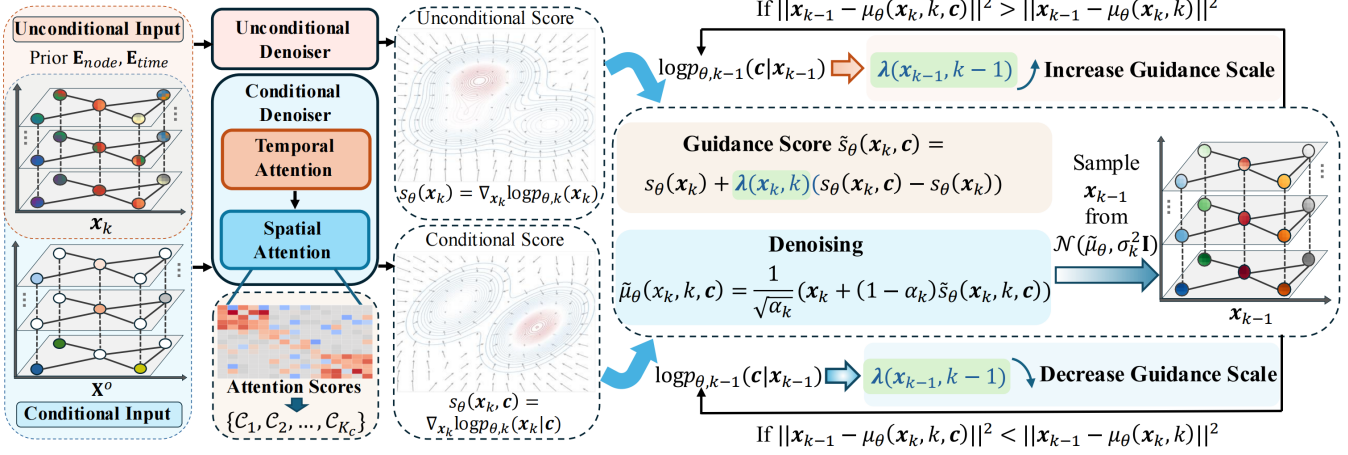


Figure 2: FENCE performs imputation by estimating both conditional and unconditional scores. It dynamically adjusts the guidance scale at each step by evaluating the posterior likelihood, controlling the scales of the conditional guidance strength to ensure consistency with observed data.

is a linear combination of the true conditional distribution $p_k(\mathbf{x}_k|\mathbf{c})$ and the unconditional distribution $p_k(\mathbf{x}_k)$:

$$p_{\theta,k}(\mathbf{x}_k|\mathbf{c}) = (1 - \pi)p_k(\mathbf{x}_k) + \pi p_k(\mathbf{x}_k|\mathbf{c}), \quad (6)$$

where $\pi \in [0, 1]$ is a hyperparameter that indicates the prior confidence in how well the conditional generative model has learned to adhere to the condition \mathbf{c} . Next, we compute the score function of this target distribution. Applying the chain rule for logarithmic derivatives yields (see Appendix for the full derivation):

$$\nabla_{\mathbf{x}_k} \log p_{\theta,k}(\mathbf{x}_k|\mathbf{c}) = \frac{\nabla_{\mathbf{x}_k} (p_{\theta,k}(\mathbf{x}_k|\mathbf{c}) - (1 - \pi)p_{\theta,k}(\mathbf{x}_k))}{p_{\theta,k}(\mathbf{x}_k|\mathbf{c}) - (1 - \pi)p_{\theta,k}(\mathbf{x}_k)} \quad (7)$$

By formulating the probability densities in the numerator and denominator using Bayes' theorem and applying the score identity ($\nabla_{\mathbf{x}_k} p(\mathbf{x}_k) = p(\mathbf{x}_k) \nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k)$), this expression can be arranged into the guidance score formulation of Eq. 5 (see Appendix). This process yields the formulation for the guidance scale:

$$\lambda(\mathbf{x}_k, k) = \frac{p_{\theta,k}(\mathbf{c}|\mathbf{x}_k)/p_{\theta,k}(\mathbf{c})}{p_{\theta,k}(\mathbf{c}|\mathbf{x}_k)/p_{\theta,k}(\mathbf{c}) - (1 - \pi)} \quad (8)$$

In practice, assuming the prior $p_{\theta,k}(\mathbf{c})$ is constant yields:

$$\lambda(\mathbf{x}_k, k) \approx \frac{p_{\theta,k}(\mathbf{c}|\mathbf{x}_k)}{p_{\theta,k}(\mathbf{c}|\mathbf{x}_k) - (1 - \pi)} \quad (9)$$

This formulation indicates that the guidance scale is a function of the posterior likelihood $p_{\theta,k}(\mathbf{c}|\mathbf{x}_k)$, and achieves our objective: when the posterior is high, indicating high consistency with the condition \mathbf{c} , the guidance scale approaches 1. As the posterior likelihood decreases toward the threshold $(1 - \pi)$, the guidance scale increases, applying stronger guidance to ensure adherence to the conditional distribution.

Posterior Likelihood Estimation The formulation for $\lambda(\mathbf{x}_k, k)$ in Eq. 9 depends on the posterior likelihood, which is not directly accessible. Inspired by (Koulishcher et al.

2025), we can estimate this value by tracking the diffusion's reverse Markov chain. The derivation begins with the definition of the posterior, $p_{\theta,k-1}(\mathbf{c}|\mathbf{x}_{k-1:K})$. By applying the chain rule of probability and leveraging the Markov property of the reverse diffusion process (i.e., $p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_{k:K}, \mathbf{c}) = p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{c})$), the posterior at step $k - 1$ is:

$$p_{\theta,k-1}(\mathbf{c}|\mathbf{x}_{k-1}) = p_{\theta,k}(\mathbf{c}|\mathbf{x}_k) \cdot \frac{p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{c})}{p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k)} \quad (10)$$

Taking the logarithm of Eq. 10, we can obtain the update function of the posterior likelihood from step k to $k - 1$:

$$\log p_{\theta,k-1}(\mathbf{c}|\mathbf{x}_{k-1}) = \log p_{\theta,k}(\mathbf{c}|\mathbf{x}_k) + \log p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{c}) - \log p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k) \quad (11)$$

As the reverse transition distributions are Gaussian, the log-likelihood difference becomes:

$$\begin{aligned} \log p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k, \mathbf{c}) - \log p_{\theta}(\mathbf{x}_{k-1}|\mathbf{x}_k) \\ = \frac{1}{2\sigma_k^2} (\|\mathbf{x}_{k-1} - \mu_{\theta}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k-1} - \mu_{\theta}(\mathbf{x}_k|\mathbf{c})\|^2) \end{aligned} \quad (12)$$

This formulation enables updating the posterior likelihood by comparing the outputs of the conditional and unconditional models at each step. Additionally, we introduce two hyperparameters: a temperature τ to scale the update strength and an offset δ to ensure guidance activates properly in early diffusion stages. This leads to the parameterized update equation for posterior:

$$\begin{aligned} \log p_{\theta,k-1}(\mathbf{c}|\mathbf{x}_{k-1}) &= \log p_{\theta,k}(\mathbf{c}|\mathbf{x}_k) \\ &- \frac{\tau}{2\sigma_k^2} (\|\mathbf{x}_{k-1} - \mu_{\theta}(\mathbf{x}_k|\mathbf{c})\|^2 - \|\mathbf{x}_{k-1} - \mu_{\theta}(\mathbf{x}_k)\|^2) - \delta \end{aligned} \quad (13)$$

By initializing $\log p_{\theta,K}(\mathbf{c}|\mathbf{x}_T)$ (e.g., to 0, assuming a uniform prior distribution) and applying this update rule iteratively from $k = K$ to 1, we can estimate the log-posterior at each denoising step. The feedback guidance loop is thus complete: at each step k , we use the current guidance scale $\lambda(\mathbf{x}_k, k)$ to sample \mathbf{x}_{k-1} . We then use this new sample \mathbf{x}_{k-1} in Eq. 13 to update the posterior $p_{\theta,k-1}(\mathbf{c}|\mathbf{x}_{k-1})$. Finally, this new posterior is fed into Eq. 9 to determine the guidance scale $\lambda(\mathbf{x}_{k-1}, k-1)$ for the next step.

Cluster-Aware Feedback Guidance While the global guidance mechanism adapts the scale across denoising steps, it applies this scale uniformly to all nodes, which is suboptimal because nodes differ in their alignment to observations. Fully per-node scaling, however, can be statistically unstable under sparse observations. To address this, we introduce a cluster-aware feedback guidance strategy, which aggregates information from a group of correlated nodes to compute the guidance scale for each node. To group the nodes, we leverage the spatial attention scores, $\mathbf{A}_{attn} \in \mathbb{R}^{N \times N}$, from the conditional denoising network. Since the attention scores quantify dynamic correlations that evolve during the reverse process, we employ k-means clustering at each denoising step to partition the set of nodes V into K_c disjoint clusters, $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{K_c}\}$.

During each step of the reverse diffusion process, for any node i belonging to the current cluster \mathcal{C}_j , we compute a cluster-level log-posterior. The aggregation rule for the cluster-level log-posterior is defined as:

$$\log p_{\theta,k-1,\mathcal{C}_j}(\mathbf{c}|\mathbf{x}_{k-1}) = \frac{1}{|\mathcal{C}_j|} \sum_{l \in \mathcal{C}_j} \log p_{\theta,k-1,l}(\mathbf{c}|\mathbf{x}_{k-1}), \quad (14)$$

where $\log p_{\theta,k-1,l}(\mathbf{c}|\mathbf{x}_{k-1})$ is the individually updated log-posterior for node l using Eq. 13. By averaging over all nodes in the cluster, we obtain a more stable estimate that is less susceptible to the high variance from any single node.

The cluster-level posterior, $p_{\theta,k,\mathcal{C}_j}(\mathbf{c}|\mathbf{x}_k)$, is then used to compute a shared guidance scale for all nodes within that cluster, using the formulation from Eq. 9:

$$\lambda_{\mathcal{C}_j}(\mathbf{x}_k, k) = \frac{p_{\theta,k,\mathcal{C}_j}(\mathbf{c}|\mathbf{x}_k)}{p_{\theta,k,\mathcal{C}_j}(\mathbf{c}|\mathbf{x}_k) - (1 - \pi)} \quad (15)$$

Training and Inference

Training. FENCE requires both unconditional and conditional predictions to compute the guidance scales. To prevent the learning of the unconditional prior from interfering with the conditional imputation, we adopt a two-stage training procedure. First, we train an unconditional generative model to learn the prior distribution $p_{\theta}(\mathbf{x})$. In this stage, the denoising network ϵ_{θ} is trained using only the unconditional vector which is generated from structural priors without any observations. After convergence, the weights of this unconditional model are saved. Next, we fine-tune this pre-trained model for the conditional imputation. The network weights are initialized from the saved unconditional model. The model is then trained using the conditional observations.

Algorithm 1: Inference of FENCE

- 1: **Input:** Conditional network ϵ_{θ} , unconditional network $\epsilon_{\theta}^{\text{uncond}}$, observed data \mathbf{x}^o and mask \mathbf{M} , total denoising steps K , hyperparameters π, τ, δ, K_c .
 - 2: **Initialize:**
 - 3: Sample $\mathbf{x}_K \sim \mathcal{N}(0, \mathbf{I})$.
 - 4: Initialize $\log p_{\theta,K,i}(\mathbf{c}|\mathbf{x}_K) \leftarrow 0$ for all nodes.
 - 5: **for** $k = K, \dots, 1$ **do**
 - 6: $\epsilon_{\text{cond}} \leftarrow \epsilon_{\theta}(\mathbf{x}_k, k, \mathbf{c})$
 - 7: $\epsilon_{\text{uncond}} \leftarrow \epsilon_{\theta}^{\text{uncond}}(\mathbf{x}_k, k, \mathbf{c}_{\text{uncond}})$
 - 8: Extract \mathbf{A}_{attn} and update clusters $\{\mathcal{C}_j\}_{j=1}^{K_c}$.
 - 9: Compute cluster-level scales $\lambda_{\mathcal{C}_j}$ by Eq. 14, Eq. 15.
 - 10: Update $\lambda_i \leftarrow \lambda_{\mathcal{C}_j}$ for each node.
 - 11: $\boldsymbol{\lambda}_k \leftarrow (\lambda_1, \dots, \lambda_N)$
 - 12: $\tilde{\epsilon}_{\theta} \leftarrow \epsilon_{\theta}^{\text{uncond}} + \boldsymbol{\lambda}_k \odot (\epsilon_{\theta}^{\text{cond}} - \epsilon_{\theta}^{\text{uncond}})$
 - 13: Compute $\tilde{\mu}_{\theta}$ by $\tilde{\epsilon}_{\theta}$ and sample $\mathbf{x}_{k-1} \sim \mathcal{N}(\tilde{\mu}_{\theta}, \sigma_k^2 \mathbf{I})$
 - 14: // Update posteriors for the next step
 - 15: Update $\log p_{\theta,k-1,i}(\mathbf{c}|\mathbf{x}_{k-1})$ using Eq. 13, Eq. 14.
 - 16: **end for**
 - 17: **return** \mathbf{x}_0
-

Inference. During inference, the denoising network utilizes a conditional context \mathbf{c} and an unconditional context $\mathbf{c}_{\text{uncond}}$ as inputs. Instead of applying a fixed guidance scale, FENCE dynamically adjusts the guidance scale at each step of the denoising process. This adjustment is driven by a feedback loop that continuously estimates the posterior likelihood to assess the alignment between the current sample and the conditional observations. Furthermore, to account for the varying degrees of alignment with conditional observations across different nodes, the feedback is computed at a cluster level, leveraging spatial-temporal correlations. The inference procedure is presented in Algorithm 1.

Experiments

Experimental Settings

Dateset. We conduct experiments on the PEMS04, PEMS07, and PEMS08 datasets (Chen et al. 2001). The datasets are split chronologically into training, validation, and test sets (60%/20%/20%), and input samples are generated by segmenting these sets into overlapping sequences using a sliding window.

Baselines. We evaluate the performance of FENCE against eight methods, covering machine learning baselines, discriminative models, and generative models. The discriminative models include: 1) **ASTGNN** (Guo et al. 2021), an attention-based graph neural network adapted for imputation via a reconstruction-based self-supervised learning objective (Cao et al. 2018); 2) **IGNNK** (Wu et al. 2021), an inductive GNN for kriging; 3) **GCASTN** (Peng et al. 2023), a contrastive self-supervised learning framework for imputation; and 4) **ImputeFormer** (Nie et al. 2024), which combines the Transformer with low-rank induction. The machine learning method is: 5) **LCR** (Chen et al. 2024), which leverages laplacian convolutional representations for time series imputation. The generative models include: 6) **mTAN** (Shukla

and Marlin 2021), employing a VAE for irregularly sampled time series; 7) **CSDI** (Tashiro et al. 2021), a conditional score-based diffusion model for imputation; and 8) **PriSTI** (Liu et al. 2023), a conditional diffusion framework that integrates geographic context.

Missing Patterns. We introduce two challenging missingness patterns: Spatially Random, Temporally Contiguous (SR-TC) and Spatially Clustered, Temporally Contiguous (SC-TC). 1) **SR-TC**: The total length of the series at each node is L , and the time series is divided into $\frac{L}{T}$ non-overlapping temporal patches of length T . For each of the N nodes, each temporal patch is independently masked with a probability of α , resulting in missing blocks that are contiguous in time but randomly distributed across nodes. 2) **SC-TC**: The N nodes are first partitioned into N_c distinct communities. A missing block is defined by a temporal patch of length T and a node community. Each of these $\frac{L}{T} \times N_c$ blocks is independently masked with a probability of α , causing entire communities of sensors to drop out simultaneously for continuous time periods. For our experiments, we set the missing rate $\alpha = 80\%$ and $T = 12$.

Performance Comparison

The overall performance is presented in Tab. 1. The key observations are as follows: (1) FENCE achieves state-of-the-art performance across all three datasets, both for the SR-TC and SC-TC missing patterns. Notably, FENCE outperforms the second-best method by an average of 6.26% in MAPE across all the datasets and missing patterns. (2) Among discriminative models, ImputeFormer demonstrates superior performance, benefiting from its integration of low-rank inductive bias combined with Transformers. (3) Score-based diffusion models, such as CSDI and PriSTI, perform competitively compared to machine learning and discriminative models. (4) FENCE significantly outperforms existing diffusion models, including CSDI and PriSTI, demonstrating the effectiveness of the dynamic feedback mechanism that adjusts the guidance scale during imputation. (5) Under the challenging SC-TC scenario, FENCE consistently outperforms all baselines across all metrics, demonstrating its effectiveness in handling highly sparse missing patterns.

Ablation Study

We conduct an ablation study to evaluate the effectiveness of the spatial-temporal feedback guidance mechanism and the cluster-aware guidance strategy. Specifically, we compare FENCE with three variants: (1) **wo-U**, which removes the modeling of unconditional scores and only models the conditional scores. (2) **wo-F**, which removes the spatial-temporal feedback guidance from the denoising process; (3) **wo-C**, which removes the cluster-aware guidance strategy and instead applies a uniform global guidance scale to all nodes at each denoising step.

The results are shown in Fig. 3. First, the **wo-U** variant, which does not model the unconditional prior distribution, results in suboptimal performance. This indicates that the two-stage training procedure, which first models the prior distribution, facilitates a more accurate modeling of

the conditional distribution. Second, we compare FENCE with the **wo-F** variant. The results show that incorporating spatial-temporal feedback guidance yields substantial performance gains across all metrics and missing patterns, showing the importance of dynamically adjusting the guidance scale based on the alignment between generated values and conditional observations. Finally, FENCE outperforms the **wo-C** variant, showing the effectiveness of the cluster-aware guidance strategy. This result demonstrates that computing guidance scales based on clustered information leads to more accurate imputations, compared to applying a uniform global scale to all nodes at each denoising step.

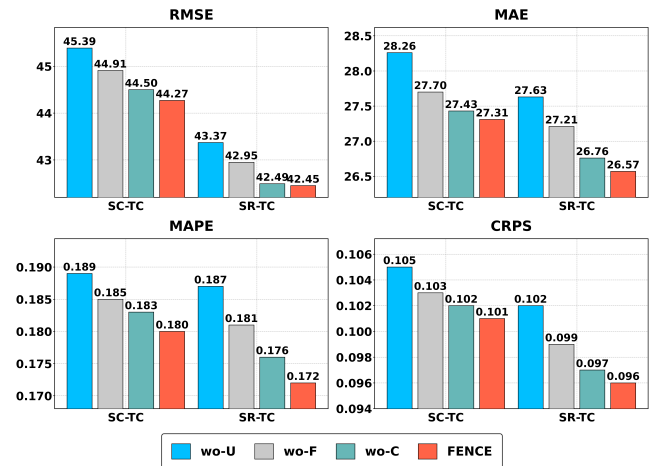


Figure 3: Ablation study.

Hyperparameter Analysis

We evaluate the key hyperparameters of FENCE on the PEMS04 dataset, as shown in Fig. 4. These include the prior confidence π , the guidance timing parameters (t_0, t_1) , and the number of clusters. The parameters (t_0, t_1) control the guidance offset δ and temperature τ , with their relationships illustrated in the appendix. As shown in Fig. 4, FENCE’s performance is relatively stable across different settings of (t_0, t_1) . Regarding the prior confidence in the conditional model, the best results are obtained when $\pi = 0.5$. A higher value of π indicates high confidence in the conditional model, which necessitates a very low posterior likelihood for applying guidance. In contrast, a lower value, such as $\pi = 0.5$, provides a broader operational range for FENCE, lowering the threshold for guidance activation. Next, we evaluate different settings of the ratio of node number to cluster number: $1, N/20, N/10, N/8, N$. The best performance is achieved at $N/20$, while setting the ratio to 1 or N results in degraded performance. This indicates that using either a global uniform guidance scale or a node-specific guidance scale is suboptimal compared to employing a cluster-level guidance scale.

Case Study

We compare FENCE with CFG, where the guidance scale is fixed at 1. In Fig. 5, for a node with no observations

Datasets	Miss Type	Metrics	ASTGNN	IGNNK	GCASTN	LCR	ImputeFormer	mTAN	CSDI	PriSTI	FENCE
PEMS04	SR-TC	MAE	31.47	32.69	30.54	28.75	<u>27.30</u>	31.20	27.63	27.51	26.57
		RMSE	45.94	47.27	44.93	44.10	43.81	45.36	<u>43.37</u>	43.46	42.45
		MAPE	0.192	0.201	0.191	0.189	<u>0.178</u>	0.209	0.187	0.184	0.172
	SC-TC	MAE	31.70	33.32	30.07	28.98	29.35	30.59	<u>28.26</u>	28.47	27.31
		RMSE	47.76	49.12	47.97	46.96	46.71	48.19	45.39	<u>45.31</u>	44.28
		MAPE	0.212	0.214	0.205	0.197	0.206	0.217	<u>0.189</u>	0.190	0.180
PEMS07	SR-TC	MAE	46.16	52.64	50.02	47.24	45.07	45.60	<u>44.36</u>	46.84	42.51
		RMSE	65.60	71.16	66.31	65.88	65.86	67.07	<u>64.37</u>	65.23	63.48
		MAPE	0.206	0.234	0.310	0.207	<u>0.195</u>	0.225	0.208	0.213	0.178
	SC-TC	MAE	47.63	55.12	45.95	44.97	<u>44.59</u>	45.31	44.78	45.18	43.12
		RMSE	73.85	79.54	74.29	74.11	73.75	75.06	74.33	<u>73.54</u>	73.06
		MAPE	0.265	0.332	0.260	0.248	0.232	0.247	0.228	<u>0.224</u>	0.215
PEMS08	SR-TC	MAE	26.72	27.17	26.82	25.52	25.27	27.09	24.32	<u>24.06</u>	22.77
		RMSE	41.22	42.76	41.87	40.90	<u>40.64</u>	44.92	41.09	42.01	40.26
		MAPE	0.180	0.194	0.186	0.166	<u>0.160</u>	0.187	0.167	0.164	0.147
	SC-TC	MAE	31.82	32.25	30.78	30.51	29.64	30.29	<u>28.23</u>	35.90	27.29
		RMSE	50.76	51.35	49.39	50.03	<u>48.37</u>	51.82	48.54	48.80	47.78
		MAPE	0.220	0.231	0.213	0.218	0.216	0.229	<u>0.190</u>	0.193	0.175

Table 1: Overall Imputation performance comparison. Bold and underlined fonts indicate the best and second-best results.

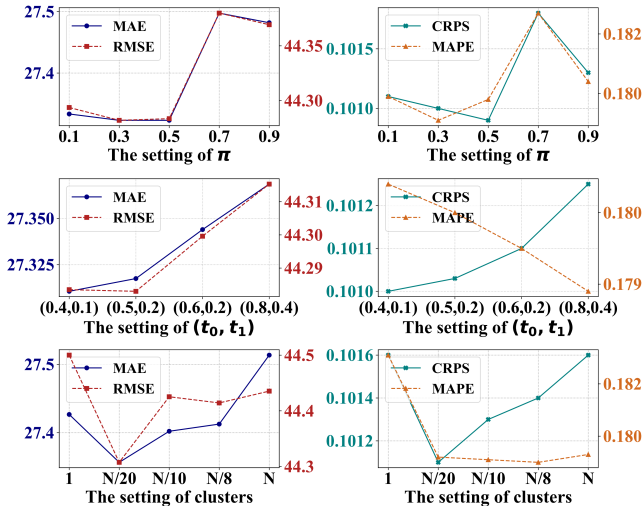


Figure 4: Effect of hyperparameters.

across 12 time slices, CFG’s fixed scale provides insufficient correction strength, causing the imputation to revert to the learned average and deviate from the ground truth. In contrast, FENCE’s dynamic guidance mechanism actively adjusts the guidance scale based on the posterior likelihood. When the imputation diverges from the observations, the guidance scale increases to strengthen the correction. This adaptive process results in an imputation that more accurately reflects the true conditional data distribution.

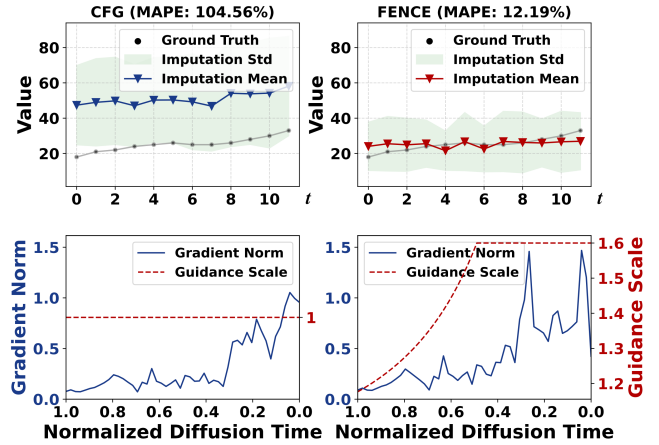


Figure 5: Case Study

Conclusion

This paper proposes FENCE, a spatial-temporal feedback diffusion guidance method that tackles the limitations of existing imputation methods based on diffusion models, which rely on a fixed guidance scale. FENCE dynamically adjusts the guidance scale based on posterior likelihood approximations, ensuring the generated values consistently align with observed data throughout the denoising process. Furthermore, the cluster-aware guidance mechanism leverages spatial-temporal correlations to tailor the guidance for different nodes, improving imputation accuracy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62372031) and the Beijing Natural Science Foundation (Grant No. 4242029).

References

- Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; and Li, Y. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31.
- Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 6085.
- Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; and Jia, Z. 2001. Freeway performance measurement system: mining loop detector data. *Transportation research record*, 1748(1): 96–102.
- Chen, X.; Cheng, Z.; Cai, H.; Saunier, N.; and Sun, L. 2024. Laplacian convolutional representation for traffic time series imputation. *IEEE Transactions on Knowledge and Data Engineering*, 36(11): 6490–6502.
- Chen, Y.; Deng, W.; Fang, S.; Li, F.; Yang, N. T.; Zhang, Y.; Rasul, K.; Zhe, S.; Schneider, A.; and Nevmyvaka, Y. 2023. Provably convergent schrödinger bridge with applications to probabilistic time series imputation. In *International Conference on Machine Learning*, 4485–4513. PMLR.
- Du, W.; Côté, D.; and Liu, Y. 2023. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219: 119619.
- Fons, E.; Sztrajman, A.; El-Laham, Y.; Ferrer, L.; Vyetrenko, S.; and Veloso, M. 2025. LSCD: Lomb-Scargle Conditioned Diffusion for Time series Imputation. *arXiv preprint arXiv:2506.17039*.
- Fortuin, V.; Baranchuk, D.; Rätsch, G.; and Mandt, S. 2020. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, 1651–1661. PMLR.
- Guo, S.; Lin, Y.; Wan, H.; Li, X.; and Cong, G. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5415–5428.
- Guo, S.; Wei, T.; Huang, Y.; Zhao, M.; Chen, R.; Lin, Y.; Lin, Y.; and Wan, H. 2024. An Experimental Evaluation of Imputation Models for Spatial-Temporal Traffic Data. *arXiv preprint arXiv:2412.04733*.
- Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.
- Ipsen, N. B.; Mattei, P.-A.; and Frelsen, J. 2022. How to deal with missing data in supervised deep learning? In *10th International Conference on Learning Representations*.
- Koulishcher, F.; Handke, F.; Deleu, J.; Demeester, T.; and Ambrogioni, L. 2025. Feedback Guidance of Diffusion Models. *arXiv preprint arXiv:2506.06085*.
- Lao, D.; Yang, X.; Wu, Q.; and Yan, J. 2022. Variational inference for training graph neural networks in low-data regime through joint structure-label estimation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 824–834.
- Liu, M.; Huang, H.; Feng, H.; Sun, L.; Du, B.; and Fu, Y. 2023. Pristi: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1927–1939. IEEE.
- Luo, Y.; Zhang, Y.; Cai, X.; and Yuan, X. 2019. E²GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 3094–3100. International Joint Conferences on Artificial Intelligence Organization.
- Miao, X.; Wu, Y.; Chen, L.; Gao, Y.; and Yin, J. 2022. An experimental survey of missing data imputation algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 35(7): 6630–6650.
- Miao, X.; Wu, Y.; Wang, J.; Gao, Y.; Mao, X.; and Yin, J. 2021. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 8983–8991.
- Nie, T.; Qin, G.; Ma, W.; Mei, Y.; and Sun, J. 2024. ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2260–2271.
- Peng, W.; Lin, Y.; Guo, S.; Tang, W.; Liu, L.; and Wan, H. 2023. Generative-contrastive-attentive spatial-temporal network for traffic data imputation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 45–56. Springer.
- Shukla, S. N.; and Marlin, B. M. 2021. Multi-time attention networks for irregularly sampled time series. *arXiv preprint arXiv:2101.10318*.
- Solís-García, J.; Vega-Márquez, B.; Nepomuceno, J. A.; and Nepomuceno-Chamorro, I. A. 2025. CoSTI: Consistency Models for (a faster) Spatio-Temporal Imputation. *arXiv preprint arXiv:2501.19364*.
- Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.
- Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34: 24804–24816.
- Wang, J.; Du, W.; Yang, Y.; Qian, L.; Cao, W.; Zhang, K.; Wang, W.; Liang, Y.; and Wen, Q. 2024. Deep learning for multivariate time series imputation: A survey. *arXiv preprint arXiv:2402.04059*.
- Wang, X.; Zhang, H.; Wang, P.; Zhang, Y.; Wang, B.; Zhou, Z.; and Wang, Y. 2023. An observed value consistent diffusion model for imputing missing values in multivariate time series. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2409–2418.

Weng, W.; Jiang, H.; Wu, M.; Han, X.; Gao, H.; Shen, G.; and Kong, X. 2025. Let's Group: A Plug-and-Play SubGraph Learning Method for Memory-Efficient Spatio-Temporal Graph Modeling. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 3471–3479. International Joint Conferences on Artificial Intelligence Organization.

Wu, Y.; Zhuang, D.; Labbe, A.; and Sun, L. 2021. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4478–4485.

Xiao, C.; Jiang, X.; Du, X.; Yang, W.; Lu, W.; Wang, X.; and Chetty, K. 2025. Boundary-enhanced time series data imputation with long-term dependency diffusion models. *Knowledge-Based Systems*, 310: 112917.

Yoon, J.; Jordon, J.; and Schaar, M. 2018. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, 5689–5698. PMLR.

Zhou, J.; Li, J.; Zheng, G.; Wang, X.; and Zhou, C. 2024. Mtsci: A conditional diffusion model for multivariate time series consistent imputation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 3474–3483.