JOINT VISUAL-TEMPORAL DIFFUSION MODEL FOR PROBABILISTIC TIME SERIES IMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Score-based diffusion models have emerged as powerful probabilistic tools for time series imputation, but their denoising process is still primarily conditioned on observed temporal values. When missing values are dense or form long contiguous gaps, such temporal context becomes sparse and fragmented, weakening the structural guidance available for generation. We revisit this problem from a complementary perspective: even when numerical observations are incomplete, the macroscopic shape of a time series often remains visually discernible in its line-graph rendering. Rather than treating imputation as an image-generation problem, we use this visual representation as a structural prior for temporal diffusion. We propose **VT-Diff**, a Joint Visual-Temporal Diffusion framework for probabilistic time series imputation. VT-Diff renders multivariate time series as variable-aware line graphs and encodes them with a Vision Transformer to obtain global and local structural tokens. The central component is a Cross-Attention Bottleneck with learnable temporal anchors, which translates spatial visual tokens into time-step-wise conditions for the temporal denoising network. This design allows visual cues to guide the global shape and local variations of the imputed trajectory, while the diffusion model performs generation in the original numerical time-series domain. Extensive experiments on diverse missing patterns show that VT-Diff achieves competitive or superior performance against strong baselines and produces more structurally coherent imputations, especially under extreme missingness.

1 INTRODUCTION

Time series data are ubiquitous across real-world applications (Johnson et al., 2016; Lv et al., 2015), yet they are frequently afflicted by missing values due to sensor failures, irregular sampling, or communication errors (Yi et al., 2016).

Such missingness severely disrupts temporal continuity, making time series imputation a fundamental preprocessing task. While early deterministic methods, such as recurrent neural networks (RNNs) (Che et al., 2018; Cao et al., 2018) and generative adversarial networks (GANs) (Yoon et al., 2018), are effective in certain settings, they generate a single imputed trajectory and struggle to explicitly model the uncertainty inherent in incomplete data.

Score-based diffusion models (Ho et al., 2020; Song et al., 2021) offer a natural probabilistic alternative and have recently shown strong performance in time series imputation. These methods generate missing values by iteratively denoising noisy trajectories conditioned on observed temporal values. However, when the missing rate is high or missing values form long contiguous gaps, the available temporal context becomes sparse and fragmented. The denoising network is then forced to infer extended missing regions from limited local evidence, which can lead to structurally inconsistent imputations even when point-wise errors remain competitive.

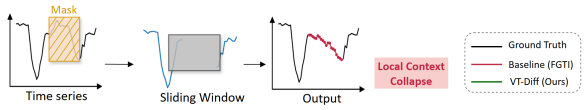
We address this limitation from a complementary perspective. Our key observation is that *structure can remain visually accessible even when temporal observations are incomplete*.

When incomplete sequences are rendered as 2D line graphs, their macroscopic trends and local variations become explicit spatial patterns, allowing vision models (Wang & Oates, 2015a; Li et al., 2023) to capture structural cues that are difficult to infer from fragmented temporal observations

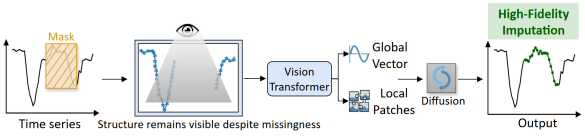
Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

alone. Importantly, however, we do not aim to perform imputation in the image domain. Rendered line graphs emphasize geometry and topology, but they do not by themselves preserve the exact numerical values required by time series reconstruction. This motivates a visual-temporal design: visual representations supply structural priors, while diffusion generation remains in the original temporal domain to maintain numerical fidelity. The key idea is to decouple where structure is perceived from where values are generated: structure is extracted from visual renderings, but missing values are generated by generative diffusion models in the temporal numerical space.

Motivated by this complementarity, we propose **VT-Diff**, a Joint Visual-Temporal Diffusion framework for probabilistic time series imputation. VT-Diff is built on a simple yet effective principle: visual structure and temporal diffusion offer complementary strengths for reconstructing incomplete time series. As illustrated in Figure 1, instead of conditioning solely on fragmented temporal observations (Figure 1a), VT-Diff further leverages structurally preserved cues from line-graph renderings as complementary guidance for generation (Figure 1b). In this framework, the visual branch captures macroscopic temporal dynamics, while the temporal diffusion process preserves precise numerical fidelity during reconstruction.



(a) Conventional temporal diffusion relies solely on observed temporal context.



(b) VT-Diff extracts visually preserved structural cues from line-graph renderings and converts them into denoising conditions for temporal generation.

Figure 1: Imputation mechanisms.

Specifically, VT-Diff encodes the rendered line graphs with a Vision Transformer (Dosovitskiy et al., 2021) and obtains two levels of structural representation. A *global embedding* summarizes macroscopic dynamics and is regularized by an auxiliary reconstruction objective, while local patch tokens retain fine-grained visual variations. The key challenge is to make these spatial tokens usable by a temporal denoising network. To this end, we introduce a Cross-Attention Bottleneck with learnable temporal anchors, which queries visual patch tokens from the perspective of individual time steps and produces step-wise structural conditions. These conditions are injected into the diffusion denoiser, allowing each temporal position to receive visual guidance relevant to its reconstruction. Thus, VT-Diff is not a direct concatenation of visual and temporal features, but a visual-to-temporal conditioning mechanism tailored for probabilistic imputation.

Our main contributions are summarized as follows:

- We propose **VT-Diff**, a probabilistic time series imputation framework that uses line-graph visual representations as structural priors for temporal diffusion. Unlike visual time-series models that operate primarily in the image domain, VT-Diff performs generation in the original numerical domain and uses visual cues as denoising conditions.
- We design a global-local visual conditioning mechanism for imputation. Global visual embeddings capture macroscopic dynamics, while local patch tokens are converted into time-step-wise structural conditions through a Cross-Attention Bottleneck with learnable temporal anchors.
- Experiments on multiple real-world datasets and missing patterns show that VT-Diff achieves competitive or superior performance against strong deterministic and probabilistic baselines, with particularly robust and structurally coherent imputations under severe missingness.

2 Related Works

Time Series Imputation. Deep time series imputation has evolved from deterministic architectures, such as SAITS (Du et al., 2023) and Glocal-IB (Yang et al., 2025), to probabilistic generative models, such as CSDI (Tashiro et al., 2021), which better capture predictive uncertainty. To improve reconstruction consistency, recent methods have introduced additional structural priors, including spatial graphs in deterministic and diffusion-based frameworks (Cini et al., 2022; Liu et al., 2023a) and frequency-domain components (Yang et al., 2024; Shen et al., 2026). A feature comparison

Table 1: Comparison among different time series imputation methods.

Categories	Ours	FSDI 2026	FGTI 2024	PriSTI 2023a	CSDI 2021	Glocal-IB 2025	SAITS 2023	GRIN 2022
Probabilistic Generative	✓	✓	✓	✓	✓	×	×	×
2D Visual Representation	✓	×	×	×	×	×	×	×
Structural Prior	✓	✓	✓	✓	×	×	×	✓

of representative methods is summarized in Table 1. Despite their success, these methods predominantly construct priors within temporal, spatial, or spectral domains. When observed values are sparse or fragmented, their conditioning signals may still be insufficient to recover global shape patterns over long missing intervals.

Visual Modeling for Time Series. Transforming time series into visual representations has shown promise for capturing structural patterns. For example, GAF (Wang & Oates, 2015a) maps time series into image-like representations, and ViTST (Li et al., 2023) renders irregularly sampled time series as line graphs and applies vision transformers for classification. These works demonstrate that visual encodings can expose global and local structures useful for time series analysis. However, existing visual time-series models are mainly designed for discriminative tasks or operate in the visual domain itself. Directly generating or reconstructing time series through pixels introduces a modality gap: line-graph images preserve geometric shape but do not directly provide the numerical precision required for imputation.

Our Position. VT-Diff differs from these lines of work in both objective and mechanism. It does not use line graphs as a replacement for temporal modeling, nor does it perform image-domain imputation. Instead, it treats visual representations as structural priors and translates them into step-wise conditions for a temporal generative diffusion denoiser. To the best of our knowledge, VT-Diff is *the first probabilistic imputation framework to explicitly use visual line-graph structures as denoising priors for temporal diffusion* (as shown in Table 1). Here, "Structural Prior" refers to explicitly constructed spatial, spectral, or visual structures used as additional conditioning beyond raw temporal observations. A comprehensive discussion of existing literature is provided in Appendix A.

3 METHODOLOGY

To mitigate the vulnerability of purely temporal dependencies under severe missingness, we propose **VT-Diff**, a Joint Visual-Temporal Diffusion framework (Figure 2). Rather than relying solely on temporal observations, VT-Diff adopts a structural perspective by rendering time series into line-graph representations. A Vision Transformer (ViT) encodes these representations into a global representation and local patch tokens, capturing macroscopic dynamics and fine-grained structural variations, respectively. The global representation is regularized by a reconstruction objective, while the local patch tokens are dynamically aligned with temporal positions through a Cross-Attention Bottleneck equipped with learnable temporal anchors. The resulting step-wise visual cues guide the temporal diffusion process, allowing VT-Diff to combine structural awareness from visual cues with numerically precise probabilistic denoising. The overall framework is jointly optimized using the denoising objective ($\mathcal{L}_{\text{denoising}}$) and the global reconstruction loss ($\mathcal{L}_{\text{recon}}$).

Problem formulation. Let $\mathbf{X} \in \mathbb{R}^{L \times D}$ be a multivariate time series with an observation mask $\mathbf{M}^{\text{ori}} \in \{0,1\}^{L \times D}$ (where $\mathbf{M}^{\text{ori}}_{ij} = 1$ if observed, 0 otherwise). Our goal is to learn a probabilistic model that captures the conditional distribution of missing values: $p(\mathbf{X}^{\text{mis}} \mid \mathbf{X}^{\text{ob}}, \mathbf{M}^{\text{ori}})$.

For self-supervised training, we apply an artificial mask $\mathbf{M} \leq \mathbf{M}^{\text{ori}}$ (element-wise) to create conditional inputs $\tilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{M}$. The model learns a mapping $\hat{\mathbf{X}} = \mathcal{F}_\theta(\tilde{\mathbf{X}}, \mathbf{M})$, optimized by computing the reconstruction loss exclusively on the artificially masked targets ($\mathbf{M}^{\text{ori}} - \mathbf{M}$):

$$\mathcal{L}(\theta) = \left\| (\mathbf{M}^{\text{ori}} - \mathbf{M}) \odot (\mathbf{X} - \hat{\mathbf{X}}) \right\|_2^2. \tag{1}$$

During inference, following standard protocols (Tashiro et al., 2021; Yang et al., 2024), evaluation metrics are computed strictly on the unseen test targets indicated by $\mathbf{M}^{\text{ori}} - \mathbf{M}$.

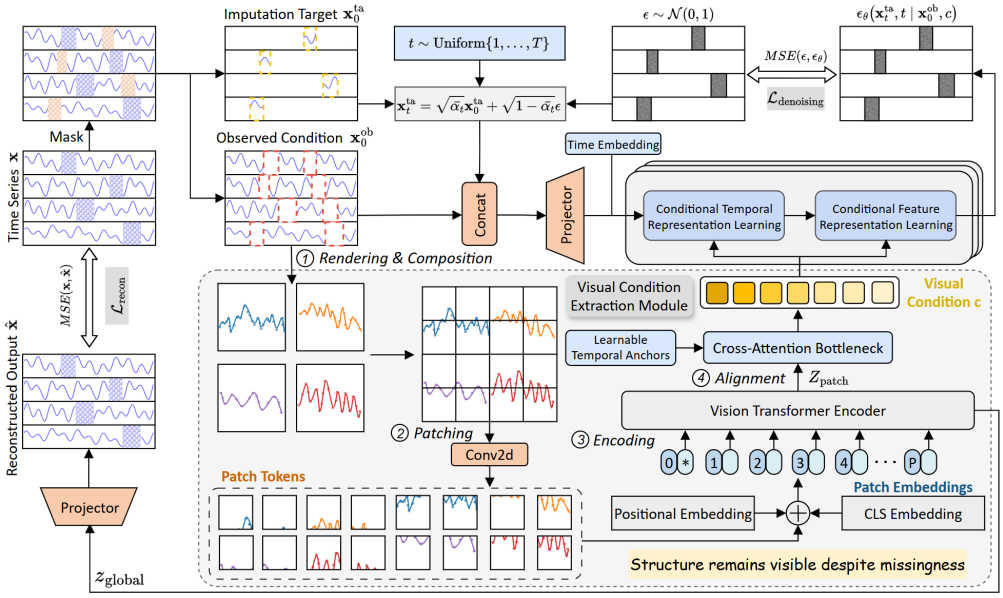


Figure 2: Overall architecture of VT-Diff. A Vision Transformer extracts structural tokens from line-graph renderings of incomplete time series. A Cross-Attention Bottleneck explicitly couples these visual cues with the temporal denoising process. The framework is jointly optimized via denoising ($\mathcal{L}_{\text{denoising}}$) and trend reconstruction ($\mathcal{L}_{\text{recon}}$) objectives.

3.1 Conditional Diffusion Model

VT-Diff employs a conditional score-based diffusion model for probabilistic imputation. Following DDPMs (Ho et al., 2020), we define the imputation target as $\mathbf{x}_0^{\text{ta}} = \mathbf{X} \odot (\mathbf{M}^{\text{ori}} - \mathbf{M})$. A fixed forward process then gradually corrupts $\mathbf{x}_0^{\text{ta}}$ with Gaussian noise:

$$\mathbf{x}_t^{\text{ta}} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{ta}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\bar{\alpha}_t$ denotes the cumulative noise schedule.

The core of VT-Diff lies in the conditional reverse denoising process. At each timestep t, the model predicts the denoising direction by conditioning not only on the observed values $\mathbf{x}_0^{\text{ob}}$ but also on the robust structural features extracted from our cross-modal architecture. We denote these 1D-aligned visual cues, seamlessly bridged by the Cross-Attention Bottleneck, as $\mathbf{c}$. The reverse process is parameterized as

$$p_\theta(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \mathbf{c}) = \mathcal{N}(\boldsymbol{\mu}_\theta, \tilde{\beta}_t \mathbf{I}), \tag{3}$$

where the mean $\boldsymbol{\mu}_\theta$ follows the standard DDPM formulation, with the noise predictor $\boldsymbol{\epsilon}_\theta$ explicitly conditioned on the visual cues $\mathbf{c}$. Here, the mean follows the standard DDPM formulation:

$$\boldsymbol{\mu}_\theta = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t^{\text{ta}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \mathbf{c}) \right). \tag{4}$$

The variance $\tilde{\beta}_t$ is fixed as

$$\tilde{\beta}_t = \begin{cases} \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, & t > 1, \\ \beta_1, & t = 1. \end{cases} \tag{5}$$

Structurally, $\boldsymbol{\epsilon}_\theta$ is parameterized by stacked residual layers that sequentially execute temporal and feature representation learning to predict the noise, explicitly conditioned on $\mathbf{c}$, t, and $\mathbf{x}_0^{\text{ob}}$(see Figure 9 in the Appendix). A concise review of standard diffusion preliminaries is in Appendix B.

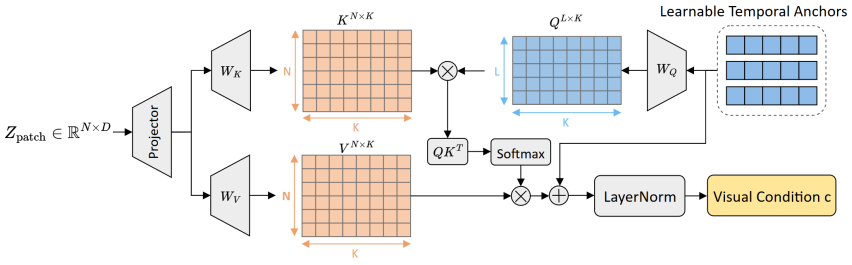


Figure 3: Illustration of the Cross-Attention module. Visual patches are mapped to keys and values ($\mathbf{K}$, $\mathbf{V}$), while learned temporal anchors form $\mathbf{Q}$. The module produces the visual condition $\mathbf{c}$.

3.2 Visual Rendering of Time Series

Line Graph Rendering. Temporal data are visualized as line graphs, where the horizontal and vertical axes capture timestamps and values, respectively. Observed data points are denoted by geometric markers (e.g., $*$) and connected with straight lines in chronological order, with missing segments interpolated seamlessly. The markers allow the Vision Transformer to distinguish true observations from interpolated segments while learning global structural patterns. Crucially, these interpolated values are strictly confined to the visual representation and bypass the temporal diffusion process to prevent information leakage. Since the scales of different variables vary significantly, we plot each variable as an individual line graph using a distinct color for channel differentiation. To maintain structural consistency, the geometric scales are kept identical across different time windows by establishing global boundaries based on the entire training set.

Image Composition. Given the set of individual line graphs for a multivariate time series with V variables, we organize them into a unified visual representation using a structured grid topology. Specifically, we arrange the corresponding line graphs in an $(l-1)\times l$ grid if $V \leq l(l-1)$, and an $l \times l$ grid otherwise, where $l = \lceil \sqrt{V} \rceil$. In cases where the number of variables does not perfectly fill the grid, the residual cells are simply left blank. To ensure an unobstructed visual receptive field for the subsequent Vision Transformer, all auxiliary graphical elements, such as axes, ticks, and margins, are deliberately discarded. Since the structural shape and spatial coordinates within the grid inherently encode the temporal sequence and magnitude variations, retaining these explicit labels would be redundant and potentially distract the feature extraction process.

3.3 Visual Encoding and Temporal Alignment

Rendered images are processed by a Vision Transformer (ViT)[1] to extract structural representations, which are subsequently aligned as temporal conditioning signals. To alleviate potential attention collapse during cross-modal alignment, we freeze the majority of the ViT parameters and only fine-tune the patch embeddings and the final two encoder layers using a distinct, smaller learning rate.

Patching and ViT-based Encoding. First, the line graph image is divided into a sequence of non-overlapping patches. These patches are flattened and linearly projected into visual tokens. By passing through the multi-head self-attention layers of the ViT encoder, the network extracts purely visual structural representations. Crucially, we decouple these representations into two distinct components: a global token $\mathbf{z}_{\text{global}}$ and local patch tokens $\mathbf{Z}_{\text{patch}}$. The global token $\mathbf{z}_{\text{global}}$ captures the overall macroscopic trajectory and is explicitly utilized to compute the trend reconstruction loss. Meanwhile, the local patch tokens $\mathbf{Z}_{\text{patch}}$ preserve fine-grained structural fluctuations and are forwarded to the cross-attention bottleneck to serve as the conditional input for the denoising network.

Cross-Attention Alignment. Since the extracted patch tokens $\mathbf{Z}_{\text{patch}} \in \mathbb{R}^{N\times D}$ (where N is the number of patches and D is the visual hidden dimension) reside in a 2D spatial domain, they are not temporally aligned with the L sequence positions of the diffusion model. To bridge this dimensional gap, we introduce a Cross-Attention Bottleneck, as illustrated in Figure 3. In this module, we instantiate a set of learnable parameters to act as temporal queries. These queries are subsequently projected via a weight matrix $\mathbf{W}_Q$ to form the query matrix $\mathbf{Q} \in \mathbb{R}^{L\times K}$, where L is the temporal

[1] https://huggingface.co/google/vit-base-patch16-224-in21k

sequence length and K is the targeted temporal condition dimension. Simultaneously, the visual patch tokens $\mathbf{Z}_{\text{patch}}$ first pass through a linear projector to transition the feature dimension from D to K, resulting in an aligned spatial representation in $\mathbb{R}^{N\times K}$. These aligned tokens are subsequently mapped via weights $\mathbf{W}_K$ and $\mathbf{W}_V$ to generate the key matrix $\mathbf{K} \in \mathbb{R}^{N\times K}$ and value matrix $\mathbf{V} \in \mathbb{R}^{N\times K}$. This mechanism dynamically aligns the spatial features to the temporal sequence via standard attention computation. Following a residual addition with the original learned queries and a Layer Normalization step, the module outputs the final visual condition $\mathbf{c} \in \mathbb{R}^{L\times K}$. By effectively transforming the representation space from $\mathbb{R}^{N\times D}$ to $\mathbb{R}^{L\times K}$, these explicit sequential visual cues strictly guide the subsequent score-based denoising process.

3.4 Training Objective

VT-Diff is trained end-to-end to jointly optimize a diffusion denoising process and global trend reconstruction.

Denoising Loss. The denoising network $\boldsymbol{\epsilon}_\theta$ is trained to predict the noise added in the forward diffusion process. Following the simplified DDPM objective, the loss is defined as

$$\mathcal{L}_{\text{denoising}} = \mathbb{E}_{\mathbf{x}_0^{\text{ta}},\boldsymbol{\epsilon},t}\left[\left\|(\mathbf{M}^{\text{ori}} - \mathbf{M}) \odot \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \mathbf{c})\right)\right\|^2\right]. \tag{6}$$

Here, $\mathbf{x}_t^{\text{ta}}$ denotes the noisy target obtained from the ground-truth imputation target $\mathbf{x}_0^{\text{ta}}$ at timestep t. Additionally, $\mathbf{M}^{\text{ori}}$ and $\mathbf{M}$ represent the original observation mask and the conditional observation mask, respectively. The predictor $\boldsymbol{\epsilon}_\theta$ is conditioned on the observed values $\mathbf{x}_0^{\text{ob}}$ and the visual conditioning signal $\mathbf{c}$ produced by the cross-attention module.

Global Trend Reconstruction Loss. To better alleviate unstable reconstructions under high missing ratios, we further introduce an auxiliary reconstruction objective to regularize global trajectory learning. The global token $\mathbf{z}_{\text{global}}$ captures coarse structural information of the rendered time series. We project $\mathbf{z}_{\text{global}}$ back to the temporal domain to obtain a deterministic trend estimate $\hat{\mathbf{x}}$. Since the original time series may contain inherent missing values, the reconstruction loss is computed only over entries observed in the original data:

$$\mathcal{L}_{\text{recon}} = \frac{\sum_{l=1}^{L}\sum_{d=1}^{D} M_{l,d}^{\text{ori}}\,(x_{l,d} - \hat{x}_{l,d})^2}{\sum_{l=1}^{L}\sum_{d=1}^{D} M_{l,d}^{\text{ori}}}, \tag{7}$$

where $\mathbf{M}_{\text{ori}} \in \{0,1\}^{L\times D}$ denotes the original observation mask, and L and D denote the sequence length and the number of variables. The reconstructed trend is given by $\hat{\mathbf{x}} = \text{Projector}(\mathbf{z}_{\text{global}})$. This objective regularizes the visual encoder to preserve global temporal structure.

Finally, the entire training objective is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoising}} + \lambda \cdot \mathcal{L}_{\text{recon}}, \tag{8}$$

where λ controls the trade-off between the denoising objective and the reconstruction regularization term. All model parameters are optimized jointly by minimizing $\mathcal{L}_{\text{total}}$. The detailed pseudo-code for the training and imputation procedures is provided in Appendix C.

4 Experiments

Datasets. We evaluate VT-Diff on seven real-world multivariate time series datasets across diverse domains: ETTh1 (Zhou et al., 2021) and Weather (Wu et al., 2021) for energy and climatology; Electricity (Trindade, 2015), Stock[2], and Illness (Wu et al., 2021) for economics and healthcare; as well as PEMS-Traffic (Lai et al., 2018) and METR-LA (Li et al., 2018) for high-frequency traffic monitoring. These datasets encompass various sampling intervals ranging from 5 minutes to a week. Following standard benchmarks (Tashiro et al., 2021; Yang et al., 2025), all datasets are systematically split into training, validation, and testing sets.

Baselines. We compare VT-Diff with 17 representative time series imputation methods spanning different modeling paradigms. These include: (i) recurrent and graph-based classic architectures,

[2]Historical daily stock market data comprising Open, High, Low, Close, Adjusted Close, and Volume features, collected from Yahoo Finance.

Table 2: CRPS on four multivariate datasets. Lower is better. **Bold** and underline denote the best and second-best results. The values are reported as mean ± std.

Dataset	Type	Setting	PRISTI	CSDI	SSSD	FGTI	FSDI	Ours
ETTh1	POINT	50%	0.213±0.001	0.281±0.002	0.231±0.004	0.297±0.011	0.167±0.000	**0.166±0.005**
		90%	0.436±0.004	0.484±0.021	0.606±0.006	0.610±0.002	0.350±0.009	**0.313±0.014**
	SEQ	20	0.411±0.013	0.187±0.026	0.252±0.003	0.593±0.007	0.169±0.002	**0.142±0.030**
		30	0.356±0.048	0.210±0.002	0.273±0.060	0.595±0.017	0.172±0.049	**0.161±0.027**
	BLOCK	50%	0.450±0.035	0.471±0.074	0.427±0.036	0.494±0.025	0.428±0.083	**0.286±0.020**
		90%	0.465±0.016	0.486±0.040	0.488±0.002	0.566±0.004	0.403±0.018	**0.297±0.038**
Weather	POINT	50%	0.082±0.008	0.080±0.001	0.184±0.004	0.048±0.001	0.057±0.000	**0.047±0.003**
		90%	0.310±0.021	0.178±0.004	0.643±0.006	**0.062±0.007**	0.115±0.000	0.104±0.003
	SEQ	20	0.183±0.007	0.144±0.005	0.284±0.000	0.101±0.000	0.083±0.002	**0.047±0.005**
		30	0.237±0.000	0.215±0.003	0.345±0.005	0.116±0.006	0.097±0.002	**0.050±0.008**
	BLOCK	50%	0.130±0.000	0.118±0.004	0.320±0.010	0.184±0.004	0.093±0.014	**0.073±0.006**
		90%	0.195±0.004	0.150±0.003	0.374±0.005	0.209±0.004	0.111±0.007	**0.088±0.002**
PEMS-Traffic	POINT	50%	0.240±0.012	0.205±0.036	0.332±0.003	0.255±0.047	0.159±0.008	**0.138±0.021**
		90%	0.371±0.034	0.367±0.004	0.587±0.054	0.407±0.005	0.272±0.007	**0.259±0.003**
	SEQ	20	0.376±0.026	0.233±0.018	0.503±0.042	0.219±0.064	0.193±0.003	**0.132±0.019**
		30	0.363±0.014	0.255±0.047	0.508±0.002	0.227±0.000	0.195±0.015	**0.153±0.028**
	BLOCK	50%	0.332±0.057	0.248±0.008	0.414±0.009	0.289±0.008	0.188±0.044	**0.141±0.005**
		90%	0.373±0.004	0.283±0.007	0.473±0.001	0.330±0.004	0.214±0.008	**0.168±0.007**
METR-LA	POINT	50%	0.262±0.001	0.241±0.001	0.402±0.000	0.358±0.002	0.220±0.003	**0.210±0.005**
		90%	0.337±0.004	0.354±0.019	0.550±0.004	0.424±0.006	0.311±0.018	**0.305±0.012**
	SEQ	20	0.434±0.003	0.323±0.003	0.703±0.005	0.347±0.019	0.307±0.034	**0.132±0.010**
		30	0.492±0.016	0.388±0.018	0.672±0.029	0.379±0.007	0.334±0.020	**0.153±0.034**
	BLOCK	50%	0.313±0.051	0.269±0.021	0.497±0.010	0.327±0.017	0.251±0.041	**0.216±0.016**
		90%	0.318±0.070	0.303±0.009	0.554±0.006	0.379±0.003	0.279±0.005	**0.245±0.006**

including BRITS (Cao et al., 2018) and GRIN (Cini et al., 2022); (ii) *Transformer*-based models for long-range temporal modeling, including Crossformer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), Non-stationary Transformer (Liu et al., 2022), and SAITS (Du et al., 2023); (iii) deep architectures focusing on multi-scale or frequency-aware representation learning, including TimesNet (Wu et al., 2023), ModernTCN (Luo & Wang, 2024), FreTS (Yi et al., 2023), Koopa (Liu et al., 2023b), and Glocal-IB (Yang et al., 2025); and (iv) generative models designed for time series imputation, including CSDI (Tashiro et al., 2021), SSSD (Alcaraz & Strodthoff, 2023), PRISTI (Liu et al., 2023a), FGTI (Yang et al., 2024), and FSDI (Shen et al., 2026). Most baselines are implemented using the unified Time-Series-Library (TSLib) (Wang et al., 2026), while the remaining methods are reproduced from their official repositories following the original configurations.

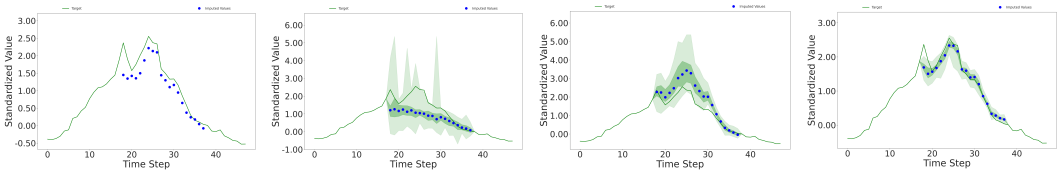
Metrics. Following standard protocols (Yang et al., 2024), we report Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Continuous Ranked Probability Score (CRPS), computed only on masked entries. RMSE and MAE measure deterministic reconstruction accuracy, while CRPS evaluates the quality of the predictive distribution.

Comprehensive experimental details are provided in Appendix D.

4.1 Overall Performance Comparison

As shown in Table 2, VT-Diff consistently achieves superior imputation quality across diverse missing patterns, evaluated via the Continuous Ranked Probability Score (CRPS). Our framework yields highly accurate empirical distributions, significantly outperforming strong baselines (e.g., FSDI, FGTI) even under extreme conditions like 90% or contiguous block missingness. This resilience highlights the core advantage of our joint visual-temporal paradigm. When severe data loss disrupts temporal continuity, visually preserved structural priors effectively anchor the generative process. By projecting these visual representations into the temporal space, VT-Diff seamlessly integrates macroscopic visual trends with precise numerical dynamics. Consequently, our framework not only ensures robust sequence estimation but also models inherent predictive uncertainty with high fidelity. All results are reported as the mean and standard deviation over three independent runs.

Exhaustive quantitative results, qualitative visualizations, and downstream forecasting evaluations are detailed in Appendix E.1, E.2, and E.4, respectively.



(a) Glocal-IB (b) PRISTI (c) FGTI (d) Ours

Figure 4: Visual comparison between our proposed VT-Diff framework and highly competitive baselines on the ILI dataset under a subsequence missingness of length 20.

4.2 Ablation Study

To assess each component's contribution under the 50% point-missing setting, Table 3 evaluates the full model against four variants. The **w/o Global Trend** variant removes the global token and reconstruction loss, while **w/o Local Patches** drops patch-level features. Additionally, the **w/o Visual Condition** variant simultaneously discards both the global token and local patches, and **w/o Cross-Attention** replaces cross-attention with mean pooling. The full model consistently achieves the best RMSE. Specifically, disabling the visual modality or patch-level guidance causes the most severe degradation, with the ILI RMSE rising sharply from 0.397 to 0.659, highlighting the necessity of localized visual cues. Furthermore, omitting the global trend impairs structural regularization, as the ETTh1 RMSE increases from 0.302 to 0.380, while mean pooling fails to preserve fine-grained cross-modal alignment between numerical queries and visual patches. Ultimately, these results demonstrate that global structural regularization, fine-grained visual guidance, and cross-modal alignment provide complementary benefits for robust imputation.

Table 3: Ablation study on three multivariate datasets measured by RMSE.

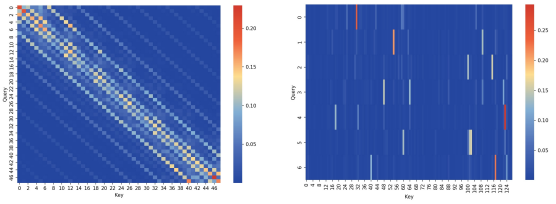
Variant	ETTh1	ILI	PEMS-Traffic
w/o Global Trend	0.380	0.467	0.399
w/o Local Patches	0.381	0.633	0.393
w/o Visual Condition	0.382	0.659	0.404
w/o Cross-Attention	0.381	0.470	0.417
VT-Diff (Ours)	**0.302**	**0.397**	**0.337**

4.3 Qualitative Analysis

Figure 4 visualizes imputation performance on the ILI dataset under a subsequence missingness of length 20, where the shaded regions denote the 50% and 95% empirical uncertainty intervals derived from the generated samples. While existing baselines struggle with peak underestimation, erratic variance, or excessively wide bounds, VT-Diff leverages visual cues to preserve complex non-linear dynamics, yielding concentrated uncertainty intervals that closely follow the ground-truth trajectory.

4.4 Interpretability Analysis

What does the visual encoder capture? To gain insights into the structural cues learned by the ViT backbone from the 2D line graphs, we analyze its self-attention heatmaps on the ETTh1 dataset (50% missing), as depicted in Figure 6. In our 2D rendering, valid observations appear as distinct markers, while missing intervals are connected by straight lines. The attention maps reveal that the model adaptively assigns the highest weights to dense observation regions (red circles), while down-weighting the featureless missing gaps (green circles) without pixel-level supervision. This demonstrates that the ViT preferentially extracts structural information from reliable observed trajectories while retaining awareness of the global visual shape, supporting the use of line-graph cues as complementary guidance under substantial missingness.



(a) Temporal Attention (b) Feature Attention

Figure 5: Visualization of cross-attention matrices in conditional representation learning modules.

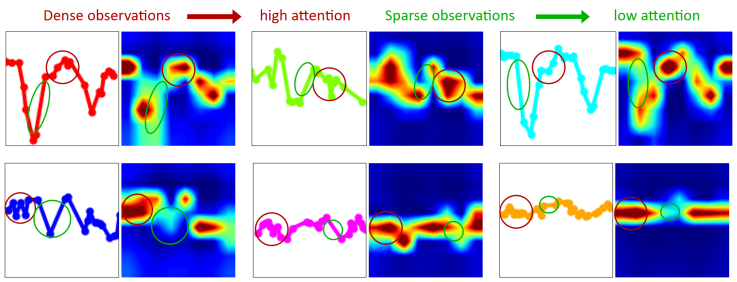


Figure 6: Visualization of ViT attention maps on ETTh1 (50% point missing) for selected variables. Left: Input images. Right: Corresponding heatmaps.

How do visual structural cues guide imputation? To examine how the extracted visual cues influence the denoising process, we further visualize the attention patterns within the conditional representation learning modules in Figure 5. The temporal attention map (Figure 5a) exhibits block-diagonal structures and recurring parallel stripes, indicating that the conditioned network attends to repeated temporal dependencies across different positions. Meanwhile, the feature attention map (Figure 5b) shows sparse vertical activations, suggesting that different variables selectively emphasize particular conditioning patterns rather than relying on uniform global information. Together, these observations illustrate how visual structural cues are transformed into position- and feature-dependent guidance for temporal denoising, helping the model exploit structural information while maintaining precise numerical reconstruction.

4.5 Computational Efficiency

We evaluate the computational efficiency of VT-Diff against representative diffusion baselines (CSDI, SSSD, and FGTI) in terms of training time per epoch and inference time per batch. As illustrated in Figure 7, while CSDI requires the least time, it yields suboptimal imputation performance. Conversely, advanced baselines like SSSD and FGTI achieve better performance but incur significantly higher computational costs. Among these robust frameworks, VT-Diff maintains competitive runtime relative to strong diffusion baselines. Specifically, our model maintains a training cost slightly lower than both FGTI and SSSD. More importantly, during inference, an inherent bottleneck for diffusion models, VT-Diff significantly outpaces these strong baselines, reducing the sampling time compared to FGTI. This demonstrates that rendering topological line graphs and explicitly coupling cross-modal features does not introduce prohibitive overhead. A comprehensive analysis of computational overhead, including parameter counts, FLOPs, and memory footprint, is detailed in Appendix E.5.

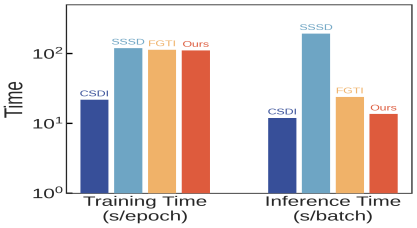


Figure 7: Comparison of training and inference time among representative diffusion baselines.

5 Conclusion

This work identifies a fundamental limitation of existing diffusion models: the vulnerability of purely temporal conditioning under severe missingness. To alleviate this issue, we propose **VT-Diff**, a joint visual-temporal diffusion framework that introduces visual structural guidance into temporal probabilistic generation. By rendering incomplete sequences as line graphs, VT-Diff employs a Vision Transformer to extract a global representation of macroscopic dynamics and local patch tokens of fine-grained variations. A Cross-Attention Bottleneck with learnable temporal anchors further aligns local visual cues with temporal positions, providing step-wise conditions for diffusion denoising. In this way, VT-Diff combines structural information from the visual branch with numerically precise temporal reconstruction. Extensive experiments demonstrate strong imputation performance across diverse missingness scenarios, with particularly pronounced gains under severe and contiguous missingness.

AI USE STATEMENT

In this work, we used generative AI tools only for limited editorial assistance, including grammar correction, wording refinement, and readability improvement. No AI tool was used to design the research method, conduct experiments, analyze results, generate claims, or draw conclusions. All technical contributions, experimental results, interpretations, and final statements were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the complete source code, including the VT-Diff framework, the 2D line-graph rendering pipeline, and all execution scripts, will be made publicly available upon acceptance of this paper. Furthermore, comprehensive experimental settings, such as dataset statistics, masking generation protocols, evaluation metrics, baseline and hyperparameter configurations, are meticulously documented in Appendix D. The detailed pseudo-code for both the joint training and imputation procedures is explicitly provided in Appendix C.

REFERENCES

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *TMLR*, 2023.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. In *NeurIPS*, volume 31, 2018.

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

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the gaps: Multivariate time series imputation by graph neural networks. In *ICLR*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 2023.

Wenjie Du, Jun Wang, Linglong Qian, Yiyuan Yang, Fanxing Liu, Zepu Wang, Zina M. Ibrahim, Haoxin Liu, Zhiyuan Zhao, Yingjie Zhou, Wenjia Wang, Kaize Ding, Yuxuan Liang, B. Aditya Prakash, and Qingsong Wen. Tsi-bench: Benchmarking time series imputation. *CoRR*, abs/2406.12747, 2024.

Jean-Pierre Eckmann, S Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *World Scientific Series on Nonlinear Science Series A*, 1995.

Elizabeth Fons, Alejandro Sztrajman, Yousef El-Laham, Luciana Ferrer, Svitlana Vyetrenko, and Manuela Veloso. Lscd: Lomb-scargle conditioned diffusion for time series imputation. In *ICML*, 2025.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *AISTATS*, pp. 1651–1661. PMLR, 2020.

Nima Hatami, Yann Gavet, and Johan Debayle. Classification of time-series images using deep convolutional neural networks. In *ICMV*. SPIE, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pp. 6840–6851, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 2016.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, pp. 95–104, 2018.

Lei Li, James McCann, Nancy S Pollard, and Christos Faloutsos. Dynammo: mining and summarization of coevolving sequences with missing values. In *KDD*, pp. 507–516, 2009.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*, 2018.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *NeurIPS*, 36:49187–49204, 2023.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2019.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. Pristi: A conditional diffusion framework for spatiotemporal imputation. In *ICDE*, pp. 1927–1939. IEEE, 2023a.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *NeurIPS*, volume 35, pp. 9881–9893, 2022.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *NeurIPS*, volume 36, 2023b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.

Yumeng Liu, Zheng Wang, Jikui Liu, Kaisa Zhang, Weidong Gao, and Xiaomao Fan. Time-frequency conditioned diffusion for multivariate time series imputation. In *ICDE*, pp. 2559–2572. IEEE, 2026.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, and Xiaojie Yuan. Multivariate time series imputation with generative adversarial networks. In *NeurIPS*, volume 31, 2018.

Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *Ieee transactions on intelligent transportation systems*, 16(2):865–873, 2015.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Wangmeng Shen, Hongfan Gao, Qingsong Zhong, Dingli Xu, and Jilin Hu. Fsdi: Frequency-shaped diffusion for time-series imputation. In *Proceedings of the ACM Web Conference 2026*, pp. 5310–5321, 2026.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, volume 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. Vits for sits: Vision transformers for satellite image time series. In *CVPR*, pp. 10418–10428, 2023.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, volume 34, pp. 24804–24816, 2021.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Zhiguang Wang and Tim Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015a.

Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *IJCAI*, pp. 3939–3945, 2015b.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In *IJCAI*, pp. 6778–6786, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

Jie Yang, Kexin Zhang, Guibin Zhang, Philip S. Yu, and Kaize Ding. Glocal information bottleneck for time series imputation. In *NeurIPS*, 2025.

Xinyu Yang, Yu Sun, Xiaojie Yuan, and Xinyang Chen. Frequency-aware generative models for multivariate time series imputation. In *NeurIPS*, volume 37, pp. 52595–52623, 2024.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. In *NeurIPS*, volume 36, 2023.

Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. St-mvl: filling missing values in geo-sensory time series data. In *IJCAI*, 2016.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets. In *ICML*, 2018.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NeurIPS*, volume 29, 2016.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *ICLR*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

A Related Works

A.1 Time Series Imputation

Time series imputation has evolved from classical statistical modeling to modern deep generative approaches (Wen et al., 2023). Early methods primarily exploit localized correlations, pattern search (Yi et al., 2016), or global matrix structures. Simple strategies such as mean imputation (Little & Rubin, 2019) and k-nearest neighbors (kNN) (Troyanskaya et al., 2001) do not explicitly model temporal dynamics. More structured approaches, including autoregressive models (e.g., ARIMA (Box et al., 2015)) and matrix completion techniques (Yu et al., 2016; Li et al., 2009), capture linear temporal dependencies and latent factors, yet often rely on stationarity assumptions that limit their flexibility in complex real-world scenarios.

With the advancement of deep learning, **deterministic neural architectures** have become prevalent for modeling non-linear temporal dependencies. Recurrent models such as GRU-D (Che et al., 2018) and BRITS (Cao et al., 2018) incorporate time-aware mechanisms to handle irregular sampling, while graph-based networks like GRIN (Cini et al., 2022) explicitly capture underlying spatial-temporal relations. Transformer-based models, including SAITS (Du et al., 2023) and iTransformer (Liu et al., 2024), further improve long-range dependency modeling through self-attention. Recent works also explore enhanced representation learning strategies under high missing rates. For instance, Glocal-IB (Yang et al., 2025) leverages an Information Bottleneck framework to balance global semantics and local variations. Despite strong empirical performance, these deterministic models typically optimize point-wise objectives (e.g., MSE), which may lead to over-smoothed predictions and limited uncertainty modeling.

To capture predictive uncertainty, **probabilistic generative models** have gained increasing attention. Early attempts based on Variational Autoencoders (VAEs) (Fortuin et al., 2020) and Generative Adversarial Networks (GANs) (Luo et al., 2018) demonstrated the feasibility of stochastic imputation, though training stability can be challenging. More recently, conditional score-based diffusion models, such as CSDI (Tashiro et al., 2021) and SSSD (Alcaraz & Strodthoff, 2023), have achieved state-of-the-art performance by formulating imputation as a conditional denoising process. To better capture complex dependencies, subsequent frameworks like PriSTI (Liu et al., 2023a), FSDI (Shen et al., 2026), and CDTI (Liu et al., 2026) have introduced specialized spatiotemporal conditioning and feature alignment mechanisms. Furthermore, other works enhance diffusion guidance using frequency-domain priors. For example, FGTI (Yang et al., 2024) incorporates spectral components, and LSCD (Fons et al., 2025) utilizes Lomb–Scargle periodograms to model periodic dependencies. These studies suggest that incorporating structured priors can improve generative consistency.

Nevertheless, most existing diffusion-based frameworks primarily rely on temporal-domain conditioning derived from observed sequential context (Tashiro et al., 2021). Under severe missingness, particularly when long contiguous gaps disrupt temporal continuity, this context becomes increasingly fragmented, making global structural patterns more difficult to recover. Motivated by the observation that the overall shape of an incomplete time series can remain visually discernible when rendered as a line graph, we propose **VT-Diff**, which introduces visual structural guidance into temporal probabilistic denoising. By combining structural cues extracted from line-graph renderings with numerically precise temporal generation, VT-Diff enables more reliable imputation under severe missingness.

A.2 Visual Modeling for Time Series

Transforming time series into two-dimensional visual representations has been explored in various contexts. Representative approaches include Gramian Angular Fields (GAF) (Wang & Oates, 2015a), Recurrence Plots (RP) (Eckmann et al., 1995; Hatami et al., 2018), and Markov Transition Fields (MTF) (Wang & Oates, 2015b). These transformations typically produce dense matrix representations that are processed using convolutional neural networks for discriminative tasks such as classification, and are generally developed under the assumption of fully available inputs. However, because they are based on deterministic algebraic or geometric transformations of the temporal sequence, their structural integrity severely degrades when applied to incomplete sequences.

More recently, Vision Transformers (ViTs) (Li et al., 2023; Tarasiou et al., 2023) have been applied to time-series image representations, demonstrating strong capability in modeling structural

patterns. However, they are predominantly designed for discriminative tasks (e.g., classification) under fully observed settings. More importantly, treating the visual domain as the primary space for generative tasks exposes a critical modality gap: pure pixel-based generation inherently lacks the precise numerical fidelity required for floating-point time series reconstruction.

To overcome these limitations, our approach utilizes simple yet topologically robust line-graph renderings, where macroscopic trends remain highly discernible despite extensive missingness. Rather than relying solely on visual generation, VT-Diff explicitly couples these visually preserved structural priors with probabilistic temporal denoising. This design allows the model to exploit visually accessible structural patterns while retaining numerical reconstruction in the temporal domain.

B PRELIMINARIES

Denoising Diffusion Probabilistic Models (DDPMs) are a class of generative models that consist of a predefined forward diffusion process and a learned reverse denoising process (Ho et al., 2020).

Forward Diffusion Process. The forward process, also known as the diffusion process, is a fixed Markov chain that progressively adds a small amount of Gaussian noise to a clean data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ over T discrete diffusion steps. The noise level at each step is controlled by a predefined variance schedule $\{\beta_t\}_{t=1}^T$ (Ho et al., 2020). The transition at any diffusion step t is defined as

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{9}$$

A key property of this process is the ability to sample $\mathbf{x}_t$ at any arbitrary step t in a closed form, conditioned on $\mathbf{x}_0$. By letting $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, this is given by

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{10}$$

This allows for a reparameterization of $\mathbf{x}_t$ as a linear combination of the initial data $\mathbf{x}_0$ and a standard Gaussian noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. As $T \to \infty$, $\bar{\alpha}_T \to 0$, and the distribution of $\mathbf{x}_T$ converges to an isotropic standard Gaussian distribution.

Reverse Denoising Process. The reverse process learns a Markov chain to invert the forward noising process. It starts from a pure noise sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoises it over T steps to generate a sample $\mathbf{x}_0$ that follows the original data distribution. This process is defined by a transition kernel $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ parameterized by a deep neural network:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_\theta^2(\mathbf{x}_t, t)\mathbf{I}). \tag{11}$$

When β_t is sufficiently small, the reverse transition kernel can also be approximated as a Gaussian distribution.

Training Objective. The model's training objective is derived by optimizing the variational lower bound (VLB) on the log-likelihood. In practice, a simplified objective is often used, which trains the model to predict the noise $\boldsymbol{\epsilon}$ added at any given diffusion step t (Ho et al., 2020). This noise prediction objective is formally equivalent to the score-matching objective used in score-based generative models (Song & Ermon, 2019), with both frameworks later unified under a common theory of stochastic differential equations (Song et al., 2021). This is accomplished using a noise prediction network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ with parameters θ. The training loss is given by

$$\mathcal{L}(\theta) := \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right], \tag{12}$$

where the expectation is taken over the data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the noise distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the uniform diffusion step $t \sim \mathcal{U}\{1, \ldots, T\}$. By minimizing the mean squared error between the predicted and true noise, the model implicitly learns the score of the data distribution, which guides the reverse denoising process. After training is complete, new data samples are generated by starting from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively applying the reverse transition for $t = T, \ldots, 1$.

C ALGORITHMIC DETAILS

Algorithm 1 Training procedure of VT-Diff

Require: Dataset $\mathcal{X}$ with inherent observation masks $\mathbf{M}^{ori}$, diffusion steps T, hyperparameter λ, learnable temporal anchors $\mathbf{Q}_{temp}$
Ensure: Optimized network ϵ_θ, Vision Transformer $\mathcal{V}$

repeat
  $\mathbf{X} \sim \mathcal{X}$
  Generate artificial mask $\mathbf{M} \leq \mathbf{M}^{ori}$
  $\mathbf{x}_0^{ob} \leftarrow \mathbf{X} \odot \mathbf{M}, \quad \mathbf{x}_0^{ta} \leftarrow \mathbf{X} \odot (\mathbf{M}^{ori} - \mathbf{M})$
  $\mathbf{z}_{global}, \mathbf{Z}_{patch} \leftarrow \mathcal{V}(\text{Render}(\text{Interpolate}(\mathbf{x}_0^{ob})))$
  $\hat{\mathbf{x}} \leftarrow \text{Project}(\mathbf{z}_{global})$
  $\mathbf{c} \leftarrow \text{CrossAttn}(\mathbf{Z}_{patch}, \mathbf{Q}_{temp})$
  $t \sim \text{Uniform}\{1, \ldots, T\}, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t^{ta} \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0^{ta} + \sqrt{1-\bar{\alpha}_t}\epsilon$
  $\mathcal{L}_{denoising} = \left\| (\mathbf{M}^{ori} - \mathbf{M}) \odot \left(\epsilon - \epsilon_\theta(\mathbf{x}_t^{ta}, t \mid \mathbf{x}_0^{ob}, \mathbf{c})\right) \right\|_2^2$
  $\mathcal{L}_{recon} = \|\mathbf{X} - \hat{\mathbf{x}}\|_2^2$
  Perform Gradient Descent on $\mathcal{L}_{total} = \mathcal{L}_{denoising} + \lambda \cdot \mathcal{L}_{recon}$
until converged

Algorithm 2 Imputation procedure of VT-Diff

Require: Observed data $\mathbf{X}^{ob}$ with inherent observation mask $\mathbf{M}^{ori}$, trained network ϵ_θ, fine-tuned ViT $\mathcal{V}$, optimized temporal anchors $\mathbf{Q}_{temp}$
Ensure: Imputed series $\hat{\mathbf{X}}$

$\mathbf{z}_{global}, \mathbf{Z}_{patch} \leftarrow \mathcal{V}(\text{Render}(\text{Interpolate}(\mathbf{X}^{ob})))$
$\mathbf{c} \leftarrow \text{CrossAttn}(\mathbf{Z}_{patch}, \mathbf{Q}_{temp})$
$\mathbf{x}_T^{ta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T$ **to** 1 **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} \leftarrow \mathbf{0}$
  $\mathbf{x}_{t-1}^{ta} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left[\mathbf{x}_t^{ta} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t^{ta}, t \mid \mathbf{X}^{ob}, \mathbf{c})\right] + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\mathbf{z}$
end for
Combine: $\hat{\mathbf{X}} \leftarrow \mathbf{x}_0^{ta} \odot (\mathbf{1} - \mathbf{M}^{ori}) + \mathbf{X}^{ob} \odot \mathbf{M}^{ori}$

This section provides the detailed pseudo-code for the proposed VT-Diff framework. Specifically, Algorithm 1 outlines the overall training procedure of VT-Diff, while Algorithm 2 describes the inference process for time series imputation.

D EXPERIMENTAL DETAILS

D.1 DATASETS

To comprehensively evaluate the proposed VT-Diff framework, we conduct experiments on seven widely adopted multivariate time series benchmarks. These datasets originate from various practical applications and exhibit distinct characteristics in terms of temporal granularity and domain-specific dynamics, thereby providing a robust testbed for our imputation methodology.

Energy and Climatology. The **ETTh1** (Zhou et al., 2021) dataset records hourly parameters from electricity transformers, challenging the model to recover data from complex seasonal missing patterns. The **Weather** (Wu et al., 2021) dataset includes 21 meteorological indicators (e.g., air temperature, humidity) recorded every 10 minutes, serving as a standard benchmark for fine-grained climatic changes.

Economics and Healthcare. The **Stock** dataset contains daily market fluctuations comprising 6 features, characterized by high volatility and complex market dependencies. The **Illness** (Wu et al., 2021) dataset tracks the weekly ratio of influenza-like illness (ILI) collected by the U.S. CDC, which challenges the model with smaller sample sizes and specific weekly periodicity.

Infrastructure and Transportation. The **Electricity** (Trindade, 2015) dataset records hourly electricity consumption across multiple clients, reflecting complex periodic consumption patterns. The **PEMS-Traffic** (Lai et al., 2018) and **METR-LA** (Li et al., 2018) datasets consist of high-frequency traffic monitoring readings from sensor networks, representing strong spatial-temporal dependencies and graph-structured correlations.

A detailed statistical summary of the evaluated datasets is provided in Table 4. Across all benchmarks, we uniformly fix the input sequence length to 48. For high-dimensional datasets, we use the first 32 variables, and all baseline methods are evaluated under the same variable selection and sequence-length settings to ensure a fair comparison. This configuration is determined empirically, offering a practical trade-off between capturing sufficient temporal context and controlling computational overhead.

Table 4: Summary of benchmark datasets. This table details the key statistics for each dataset, including feature dimensions, total time steps, dataset splits (Train/Val/Test), missing patterns, sampling frequencies, and application domains.

Dataset	Features	Seq Length	Dataset Size	Missing Patterns	Frequency	Domain
ETTh1	7	48	(8593, 2881, 2881)	POINT, SEQ, BLOCK	Hourly	Energy
Weather	21		(6953, 1001, 2001)		10 min	Weather
Electricity	32		(6953, 1001, 2001)		Hourly	Energy
Stock	6		(2532, 370, 738)		Daily	Finance
ILI	7		(629, 98, 194)		Weekly	Health
PEMS-Traffic	32		(6953, 1001, 2001)		5 min	Transportation
METR-LA	32		(6953, 1001, 2001)		5 min	Transportation

D.2 MASKING STRATEGIES

We note that the missing data literature distinguishes between *statistical missingness mechanisms* (MCAR/MAR/MNAR), which characterize how the probability of missingness depends on observed or unobserved variables (Little & Rubin, 2019), and *structural masking protocols*, which specify temporal patterns of missing entries to simulate realistic sparsity conditions. Following standard practice in time-series imputation (Du et al., 2024; Fons et al., 2025)[3], our evaluation adopts the latter perspective and does not explicitly model the missingness mechanism $P(M \mid X)$.

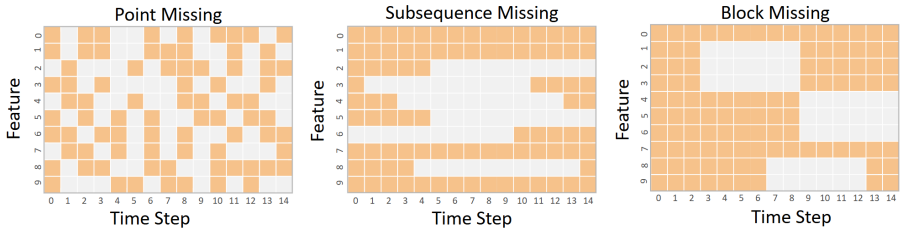


Figure 8: Illustration of the three masking strategies.

Guided by this perspective, we employ three distinct patterns of missing data to evaluate our framework: point, subsequence, and block. The missingness construction functions are directly available in the Python library PyGrinder[4]. To provide a vivid illustration of their structural differences, the visualizations of these scenarios are plotted in Figure 8.

- **Point Missing:** Individual observations are randomly dropped across temporal and feature dimensions via a Bernoulli process, simulating isolated sensor faults and unpredictable data loss.
- **Subsequence Missing:** Continuous temporal segments of a fixed length are masked on randomly selected features with stochastic starting points, reflecting uncoordinated, transient sensor outages in a multivariate system.
- **Block Missing:** Concurrent time steps across multiple variables are masked simultaneously. As the most challenging scenario, this emulates systemic failures like widespread power outages that severely restrict cross-channel context.

To evaluate robustness across varying sparsity conditions, we configure the missingness mechanisms as detailed in Table 5. Point and Block missing are tested at increasing missing rates (10%, 50%, and 90%), with block dimensions fixed at 3×3. For Subsequence missing, we maintain a 10% overall missing rate while varying the contiguous gap length ($seq_len \in \{10, 20, 30\}$) to assess interpolation across different temporal spans.

Table 5: Detailed hyperparameter configurations and corresponding target missing rates for the three evaluated missingness mechanisms.

Mechanism	**Hyperparameters**
Point Missing	$p = 0.1$ $p = 0.5$ $p = 0.9$
Subsequence Missing	$p = 0.1$, `seq_len` $= 10$ $p = 0.1$, `seq_len` $= 20$ $p = 0.1$, `seq_len` $= 30$
Block Missing	$p = 0.1$, `block_len` $= 3$, `block_width` $= 3$ $p = 0.5$, `block_len` $= 3$, `block_width` $= 3$ $p = 0.9$, `block_len` $= 3$, `block_width` $= 3$

[3]LSCD is excluded from our baselines, as its official repository currently provides only core code snippets.
[4]`https://github.com/WenjieDu/PyGrinder`

D.3 BASELINES

To ensure a comprehensive and fair evaluation, we benchmark our framework against 17 state-of-the-art baselines spanning four distinct modeling paradigms. Consistent with our experimental setup, all models are trained and evaluated with a fixed sequence length of 48.

Recurrent and Graph-based Architectures: We compare against BRITS (Cao et al., 2018), a bidirectional recurrent dynamics framework that directly models missing values using a specialized temporal feature decay mechanism, and GRIN (Cini et al., 2022), a graph recurrent network that reconstructs missing data by passing spatial-temporal messages across feature channels to capture multivariate correlations.

Transformer-based Architectures: Our baselines include Crossformer (Zhang & Yan, 2023), which utilizes a dimension-segment-wise structure to explicitly capture cross-variable dependencies in multivariate time series. We also evaluate PatchTST (Nie et al., 2023), which employs a patching mechanism combined with channel independence to enhance local semantic extraction and reduce computational complexity. iTransformer (Liu et al., 2024) inverts the standard Transformer architecture by treating the entire time series of individual variables as independent tokens. Non-stationary Transformer (Liu et al., 2022) incorporates series stationarization with a de-stationary attention mechanism to mitigate temporal distribution shifts. Finally, SAITS (Du et al., 2023) is specifically designed for missing data imputation, leveraging a joint-optimization objective and a diagonal-masked self-attention mechanism.

Multi-scale and Frequency-aware Architectures: This category includes TimesNet (Wu et al., 2023), which transforms 1D time series into 2D spaces based on multiple periodicities extracted via Fast Fourier Transform (FFT) to model complex variations. ModernTCN (Luo & Wang, 2024) is a modern temporal convolutional network optimized to efficiently capture long-term dependencies without attention mechanisms. FreTS (Yi et al., 2023) operates entirely in the frequency domain, utilizing MLPs to model both channel and temporal dependencies efficiently. Koopa (Liu et al., 2023b) leverages Koopman theory to disentangle complex temporal dynamics into stationary and non-stationary components. Glocal-IB (Yang et al., 2025) captures both global and local information bottlenecks to derive robust representations under noisy or sparse conditions.

Generative Architectures: We evaluate CSDI (Tashiro et al., 2021), a foundational conditional score-based diffusion model tailored for probabilistic time series imputation. SSSD (Alcaraz & Strodthoff, 2023) integrates structured state-space models within a diffusion framework to capture long-range temporal correlations more effectively. PRISTI (Liu et al., 2023a) enhances conditional diffusion with an auxiliary feature extraction module to improve noise estimation. Lastly, FGTI (Yang et al., 2024) and FSDI (Shen et al., 2026) represent advanced generative frameworks that incorporate targeted structural or frequency-domain priors to refine diffusion-based imputation trajectories.

D.4 Evaluation Metrics

To rigorously evaluate the imputation performance, we compute all metrics strictly on the unseen test targets as defined in our problem formulation. Let $\mathbf{X} \in \mathbb{R}^{L \times D}$ denote the ground truth multivariate time series, and $\hat{\mathbf{X}} \in \mathbb{R}^{L \times D}$ denote the imputed values. The original observation mask is given by $\mathbf{M}^{\text{ori}} \in \{0,1\}^{L \times D}$, and the artificially applied mask is $\mathbf{M} \leq \mathbf{M}^{\text{ori}}$. Consequently, the evaluation targets are accurately isolated by the indicator term $(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}$, which equals 1 only for the artificially masked entries meant for testing.

Root Mean Squared Error (RMSE): RMSE measures the average magnitude of the squared errors, penalizing larger deviations more heavily. For deterministic evaluation in our probabilistic framework, we utilize the median of the generated samples as the point estimate $\hat{\mathbf{X}}_{ij}$.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}\left(\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}\right)^2}{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}}} \tag{13}$$

Mean Absolute Error (MAE): MAE computes the average absolute difference between the predictions and the ground truth, providing a linear penalty for errors and a robust assessment of overall deterministic reconstruction accuracy.

$$\text{MAE} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}\left|\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}\right|}{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}} \tag{14}$$

Continuous Ranked Probability Score (CRPS): As a generative model, evaluating uncertainty quantification is critical. CRPS measures the compatibility of the estimated cumulative distribution function (CDF) F with the ground truth observation $\mathbf{X}_{ij}$.

$$\text{CRPS}(F, \mathbf{X}_{ij}) = \int_{-\infty}^{\infty}\left(F(z) - \mathbb{I}_{\{z \geq \mathbf{X}_{ij}\}}\right)^2 dz \tag{15}$$

where $\mathbb{I}$ is the indicator function. In practice, given a set of S generated samples $\{\hat{\mathbf{X}}_{ij}^{(s)}\}_{s=1}^{S}$ drawn from our diffusion model, the empirical CRPS for a single missing entry is approximated as:

$$\text{CRPS}_{\text{emp}}(\mathbf{X}_{ij}) = \frac{1}{S}\sum_{s=1}^{S}\left|\hat{\mathbf{X}}_{ij}^{(s)} - \mathbf{X}_{ij}\right| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left|\hat{\mathbf{X}}_{ij}^{(s)} - \hat{\mathbf{X}}_{ij}^{(s')}\right| \tag{16}$$

The overall model CRPS is then reported as the weighted double summation across all spatial-temporal dimensions:

$$\text{CRPS}_{\text{total}} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}\text{CRPS}_{\text{emp}}(\mathbf{X}_{ij})}{\sum_{i=1}^{L}\sum_{j=1}^{D}(\mathbf{M}^{\text{ori}} - \mathbf{M})_{ij}} \tag{17}$$

D.5 Implementation Details

Hardware and Software Environment All experiments and baseline models are implemented using Python and PyTorch. Model training and evaluation are conducted on a computing node equipped with eight NVIDIA H800 GPUs (with 94 GB memory each and a 400 W TDP limit), running NVIDIA Driver Version 590.48.01 and CUDA 13.1 under an Ubuntu Linux operating system.

Denoising Network Architecture. As illustrated in Figure 9, the denoising network is built upon a deep residual architecture comprising 6 stacked residual blocks. The noisy input is first mapped to a hidden channel dimension of 128 and added to the step-specific diffusion embedding. Within each of the 6 blocks, the representation is sequentially updated through a Temporal Learning module and a Feature Learning module. Crucially, both modules employ 1-layer Transformer-based cross-attention mechanisms (configured with 8 attention heads) to systematically fuse the noisy representations with the cross-modal visual information. Following these dual attention layers, the refined features are aggregated via residual connections. Ultimately, the network processes the accumulated representations through final projection layers to output the predicted noise ϵ_θ.

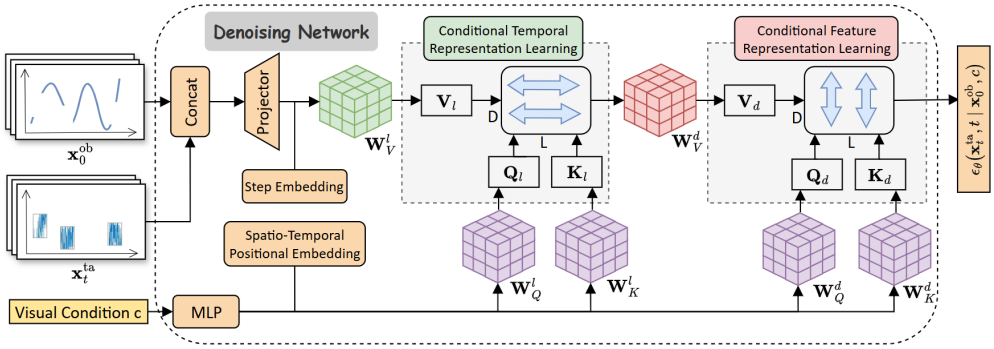


Figure 9: Architecture of the denoising network in VT-Diff.

Visual Backbone Configuration. We adopt a pre-trained ViT as the visual encoder. The backbone consists of 12 Transformer blocks with a hidden dimension of 768 and 12 attention heads, using a patch size of 16×16. The majority of the ViT parameters are frozen during training, while the patch embedding layer and the final two Transformer blocks are fine-tuned with a learning rate of 1×10^{-5}. For input resolutions different from the original pre-training resolution, the positional embeddings are adapted via bicubic interpolation. The class token is used as the global representation, whereas the patch tokens are forwarded to the Cross-Attention Bottleneck for temporal conditioning.

Hyperparameter Settings. The detailed hyperparameter configurations for model training, the diffusion process, and the network architecture are summarized in Table 6. The model is trained for 100 epochs with a batch size of 64. To ensure robustness and reproducibility, all experiments are conducted across three independent random seeds (2025, 2026, and 2027). To ensure stable cross-modal optimization and alleviate potential attention collapse, we apply a dual learning rate strategy: 1×10^{-3} for the main diffusion network and a reduced 1×10^{-5} for the visual backbone. The joint objective is balanced by setting the reconstruction loss weight (λ) to 0.05. The forward diffusion process is parameterized by $T = 200$ steps using a quadratic noise schedule (Quad, $\beta_{\text{start}} = 10^{-4}, \beta_{\text{end}} = 0.02$). For the denoising architecture, we deploy 6 residual layers with a base model dimension (d_{model}) of 128, a hidden channel dimension of 128, and 8 attention heads. Additionally, the 2D visual inputs are standardized to a resolution of 384×384.

Table 6: Detailed hyperparameter settings for model training and architecture.

Training	**Value**	**Architecture**	**Value**
Seed	{2025, 2026, 2027}	Diffusion Steps (T)	200
Batch Size	64	Noise Schedule	Quad ($\beta_{\text{start}} = 10^{-4}$, $\beta_{\text{end}} = 0.02$)
Training Epochs	100	Residual Layers	6
Diffusion LR	1×10^{-3}	d_{model}	128
ViT Backbone LR	1×10^{-5}	Hidden Channels	128
Loss Weight (λ)	0.05	Attention Heads	8
Sequence Length	48	Image Resolution	384×384

Table 7: Imputation performance (CRPS) on ILI, Stock, and Electricity datasets. **Bold** and underline denote the best and second-best results. The values are reported as mean ± std.

Dataset	Type	Setting	PRISTI	CSDI	SSSD	FGTI	FSDI	Ours
ILI	POINT	10%	0.304±0.003	0.655±0.012	0.496±0.006	0.288±0.009	0.507±0.005	**0.076±0.007**
		50%	0.420±0.004	0.959±0.004	0.670±0.003	0.386±0.004	0.568±0.000	**0.088±0.003**
		90%	1.709±0.001	1.166±0.007	0.733±0.020	0.654±0.000	0.698±0.003	**0.260±0.001**
	SEQ	10	0.344±0.011	1.041±0.018	0.600±0.025	0.403±0.042	1.281±0.013	**0.101±0.025**
		20	0.452±0.007	1.532±0.003	0.611±0.008	0.394±0.001	1.329±0.000	**0.109±0.005**
		30	0.509±0.004	1.945±0.010	0.641±0.007	0.393±0.006	1.757±0.001	**0.097±0.005**
	BLOCK	10%	0.466±0.005	0.659±0.005	0.667±0.003	0.414±0.008	1.342±0.007	**0.254±0.004**
		50%	0.627±0.030	0.823±0.009	0.783±0.020	0.508±0.013	1.476±0.038	**0.254±0.003**
		90%	0.718±0.015	0.966±0.006	0.734±0.002	0.549±0.008	1.115±0.063	**0.290±0.008**
Stock	POINT	10%	0.172±0.001	0.248±0.003	0.416±0.007	0.141±0.005	0.106±0.003	**0.003±0.007**
		50%	0.438±0.003	0.334±0.000	0.439±0.004	0.177±0.002	0.112±0.001	**0.006±0.004**
		90%	0.893±0.005	0.434±0.009	0.735±0.010	0.354±0.003	0.485±0.007	**0.017±0.002**
	SEQ	10	0.160±0.004	0.167±0.006	0.516±0.000	0.087±0.012	0.162±0.000	**0.003±0.005**
		20	0.169±0.030	0.873±0.007	0.629±0.009	0.080±0.004	0.112±0.006	**0.004±0.009**
		30	0.402±0.001	0.906±0.004	0.662±0.000	0.087±0.005	0.379±0.000	**0.003±0.006**
	BLOCK	10%	0.159±0.003	0.685±0.006	0.502±0.000	0.230±0.015	0.115±0.004	**0.013±0.007**
		50%	0.367±0.008	0.882±0.004	0.494±0.006	0.313±0.007	0.254±0.005	**0.020±0.004**
		90%	0.383±0.003	0.954±0.009	0.653±0.005	0.395±0.013	0.246±0.004	**0.023±0.005**
Electricity	POINT	10%	0.254±0.003	0.230±0.004	0.411±0.007	0.204±0.002	0.191±0.000	**0.153±0.005**
		50%	0.274±0.020	0.252±0.008	0.428±0.004	0.348±0.003	0.196±0.016	**0.179±0.005**
		90%	0.388±0.000	0.451±0.003	0.640±0.006	0.508±0.002	0.365±0.005	**0.285±0.002**
	SEQ	10	0.301±0.005	0.267±0.002	0.636±0.008	0.312±0.005	0.248±0.006	**0.150±0.002**
		20	0.375±0.006	0.274±0.007	0.648±0.004	0.435±0.008	0.253±0.003	**0.177±0.005**
		30	0.374±0.009	0.291±0.004	0.647±0.006	0.414±0.009	0.269±0.004	**0.202±0.006**
	BLOCK	10%	0.332±0.002	0.262±0.030	0.543±0.005	0.350±0.008	0.244±0.004	**0.154±0.009**
		50%	0.372±0.003	0.262±0.006	0.498±0.007	0.342±0.004	0.246±0.005	**0.201±0.009**
		90%	0.392±0.000	0.344±0.004	0.557±0.006	0.566±0.008	0.284±0.003	**0.222±0.006**

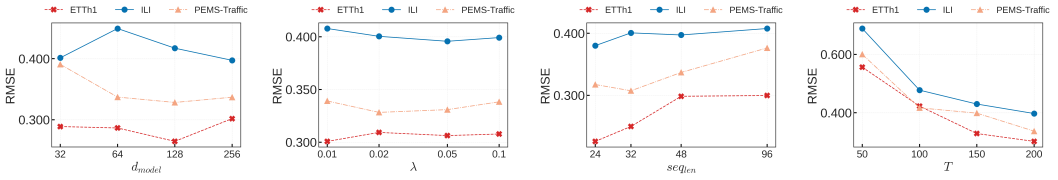
E Additional Experimental Results

E.1 Complete Quantitative Results

Due to space constraints in the main manuscript, this section provides the exhaustive deterministic and probabilistic evaluations across all datasets, missing patterns, and missing rates to supplement our main findings.

Table 10 and Table 11 present the comprehensive Root Mean Square Error (RMSE) results, while the complete Mean Absolute Error (MAE) evaluations are detailed in Table 12 and Table 13. Across this extensive benchmark, our visually-guided framework outperforms the baseline models in most datasets and masking settings.

Furthermore, Table 7 details the CRPS performance of our proposed method alongside state-of-the-art diffusion baselines on the ILI, Stock, and Electricity datasets. Consistent with the main text, our framework maintains robust probabilistic performance, outperforming the baselines in most point, sequence, and block missing scenarios.



(a) Model Dimension (b) Loss Weight (c) Sequence Length (d) Diffusion Steps

Figure 10: Sensitivity analysis of key hyperparameters on ETTh1, ILI, and PEMS-Traffic datasets.

E.2 Qualitative Visualizations

To intuitively demonstrate the imputation capabilities of VT-Diff, we present qualitative visualization results on the ETTh1, Weather, Electricity, and PEMS-Traffic datasets. Figure 11 illustrates a highly challenging scenario with a continuous missing length of 20 time steps.

As observed, early baselines like PRISTI and SSSD suffer from severe deviation and high uncertainty when facing extended missing windows. While strong baselines such as FGTI and FSDI manage to capture the general trend, they often fail to reconstruct local extremes and sharp fluctuations accurately. In contrast, VT-Diff leverages the global structural prior extracted from the 2D visual representation to effectively guide the denoising trajectory. Consequently, our method not only tightly aligns the imputed values (blue dots) with the ground truth (solid green lines) but also yields significantly narrower and more stable confidence intervals (light green shaded regions) in regions of high variance, demonstrating superior structural fidelity and robustness.

E.3 Sensitivity Analysis of Hyperparameters

To evaluate the robustness of our proposed VT-Diff, we conduct a sensitivity analysis on four hyperparameters: model dimension (d_{model}), reconstruction loss weight (λ), sequence length (seq_{len}), and diffusion steps (T). All experiments are performed under a 50% point missingness setting across the ETTh1, ILI, and PEMS-Traffic datasets, evaluated using the RMSE metric.

For model capacity (d_{model}), Figure 10(a) shows performance improves as d_{model} increases to 128, while scaling to 256 yields slight degradation due to potential overfitting. Thus, we set $d_{model} = 128$. Regarding the reconstruction loss weight (λ), the model exhibits high stability across the $[0.01, 0.1]$ range (Figure 10(b)). We adopt $\lambda = 0.05$, which achieves the lowest RMSE by optimally balancing the 2D visual prior and the temporal diffusion process. Figure 10(c) illustrates that imputation error naturally rises with longer sequences (seq_{len}), reflecting the inherent difficulty of maintaining global coherence. We select $seq_{len} = 48$ to balance long-term dependency capture with imputation fidelity. Finally, Figure 10(d) confirms that greater diffusion steps (T) consistently reduce error, with RMSE converging as T approaches 200. Consequently, we employ $T = 200$ to ensure optimal denoising quality.

Table 8: Downstream forecasting performance of PatchTST trained on sequences imputed by different models. **Bold** and underline denote the best and second-best results. The values are reported as mean ± std.

Dataset	Setting	Metric	FGTI	Glocal-IB	FSDI	Ours
ETTh1	POINT	RMSE	0.588±0.001	0.583±0.001	0.603±0.000	**0.578±0.000**
		MAE	0.381±0.001	0.376±0.001	0.391±0.000	**0.371±0.000**
	SEQ	RMSE	0.608±0.000	**0.577±0.001**	0.585±0.001	0.580±0.001
		MAE	0.396±0.001	**0.371±0.001**	0.375±0.001	0.372±0.001
	BLOCK	RMSE	0.601±0.001	0.582±0.001	0.607±0.003	**0.579±0.000**
		MAE	0.395±0.001	0.377±0.001	0.394±0.001	**0.370±0.001**
Weather	POINT	RMSE	1.042±0.006	1.046±0.001	1.055±0.007	**1.023±0.006**
		MAE	0.578±0.004	0.582±0.002	0.586±0.005	**0.577±0.005**
	SEQ	RMSE	**1.031±0.002**	1.049±0.010	1.033±0.008	1.036±0.008
		MAE	0.574±0.002	0.587±0.008	0.573±0.005	**0.571±0.007**
	BLOCK	RMSE	1.041±0.008	1.035±0.009	1.043±0.005	**1.014±0.016**
		MAE	0.577±0.009	0.574±0.007	0.578±0.004	**0.563±0.015**
PEMS-Traffic	POINT	RMSE	0.637±0.001	0.640±0.000	0.626±0.001	**0.624±0.000**
		MAE	0.332±0.001	0.331±0.000	0.323±0.001	**0.320±0.000**
	SEQ	RMSE	0.640±0.001	0.642±0.000	0.636±0.000	**0.635±0.000**
		MAE	0.332±0.001	0.339±0.001	0.331±0.001	**0.330±0.001**
	BLOCK	RMSE	0.644±0.000	0.640±0.000	0.631±0.001	**0.626±0.000**
		MAE	0.334±0.001	0.333±0.001	**0.326±0.001**	0.331±0.000
METR-LA	POINT	RMSE	0.684±0.006	0.672±0.003	0.679±0.004	**0.649±0.001**
		MAE	0.368±0.009	**0.361±0.002**	0.362±0.004	**0.361±0.002**
	SEQ	RMSE	0.665±0.001	0.680±0.002	0.666±0.001	**0.660±0.001**
		MAE	0.364±0.006	0.376±0.006	**0.361±0.005**	0.369±0.005
	BLOCK	RMSE	0.673±0.005	0.670±0.003	0.672±0.003	**0.654±0.001**
		MAE	0.363±0.006	0.365±0.001	0.363±0.002	**0.362±0.005**

E.4 Downstream Forecasting Tasks

To thoroughly assess the practical utility of the imputed sequences, we conduct downstream forecasting experiments utilizing a standard "impute-then-forecast" pipeline. Specifically, we train a forecasting model (PatchTST) on historical sequences imputed by different methods under POINT and BLOCK missing patterns (50% missing ratio) as well as the SEQ missing pattern (continuous missing length of 20). To ensure a fair evaluation, the downstream model uses the imputed sequences as input, and evaluates predictions strictly against true future values. All downstream experiments are independently repeated three times using different random seeds (2025, 2026, and 2027), with the final performance reported as the mean and standard deviation.

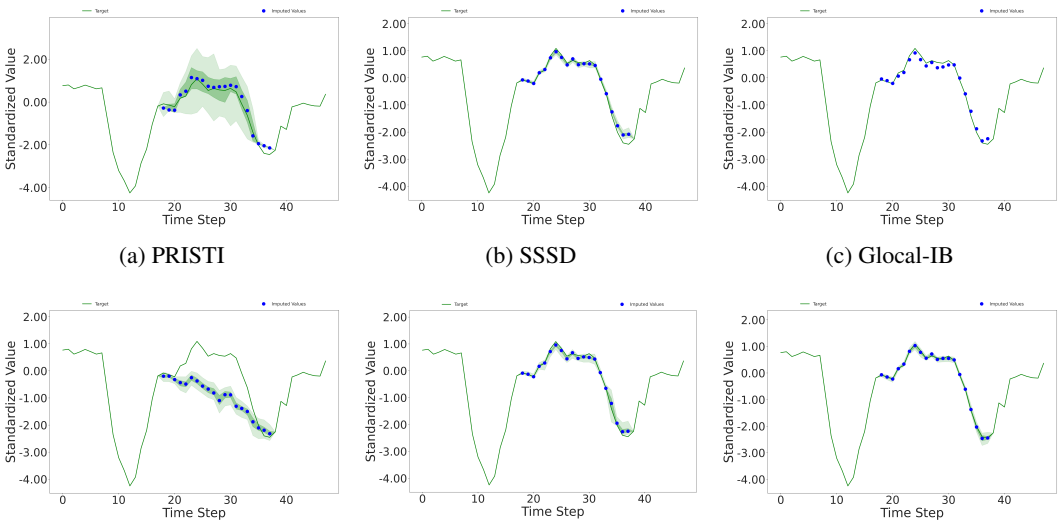
The comprehensive quantitative results, measured by both Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), are detailed in Table 8. Consistent across both metrics, the forecasting model trained on data imputed by our method achieves highly competitive predictions in the vast majority of scenarios. These favorable results demonstrate that our approach effectively preserves the essential temporal dependencies and global structural patterns of the original data. By providing high-quality historical contexts, our method enables the downstream predictor to extract more accurate features and capture the underlying temporal dynamics.

E.5 Comprehensive Computational Overhead Analysis

We evaluate the computational efficiency of VT-Diff against diffusion-based baselines on the ETTh1 dataset under a 50% point missing scenario, as shown in Table 9. VT-Diff contains 22.21 M trainable parameters, which is substantially fewer than SSSD but larger than CSDI and FGTI. Its denoising cost is 1.38 G FLOPs per step, remaining considerably lower than SSSD while introducing moderate overhead over FGTI due to the additional visual conditioning and cross-attention operations. In terms of runtime, VT-Diff requires 111.1 s per training epoch, slightly faster than both FGTI and SSSD, and achieves an inference time of 13.65 s/batch, substantially reducing the cost compared with FGTI and SSSD. The main overhead of VT-Diff lies in memory consumption: processing visual features increases the peak training and inference memory to 8488 MB and 1657 MB, respectively. Overall, these results indicate that VT-Diff trades additional memory and model capacity for improved modeling capability while maintaining competitive computational efficiency.

Table 9: Efficiency comparison of diffusion-based imputation models on ETTh1.

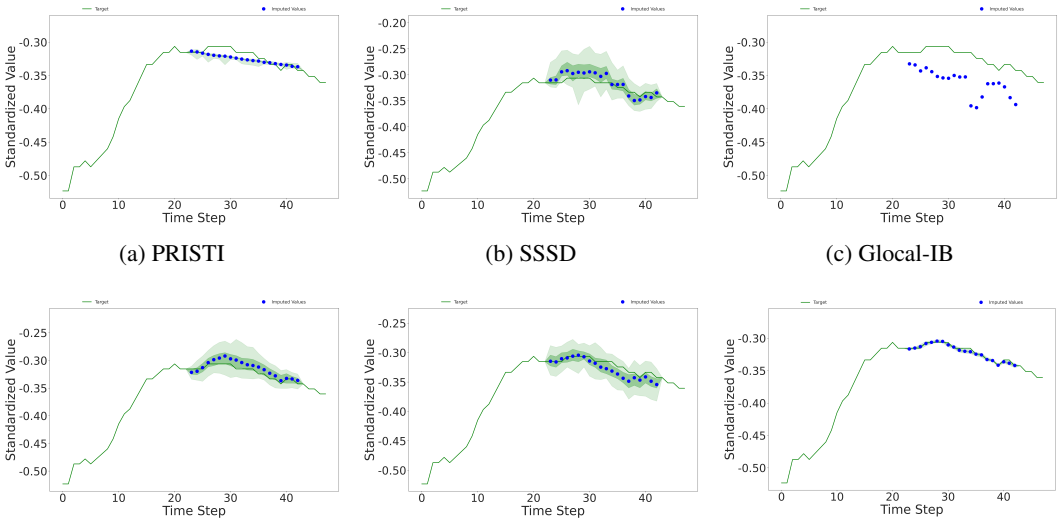
Metric	**CSDI**	**SSSD**	**FGTI**	**Ours**
Trainable Params (M)	0.41	48.11	1.95	22.21
Denoising FLOPs (G)	0.25	3.67	0.97	1.38
Training Memory (MB)	325	2175	1030	8488
Inference Memory (MB)	100	869	307	1657
Training Time (s/epoch)	21.80	119.68	113.59	111.13
Inference Time (s/batch)	11.95	193.46	23.99	13.65



(a) PRISTI (b) SSSD (c) Glocal-IB

(d) FGTI (e) FSDI (f) Ours

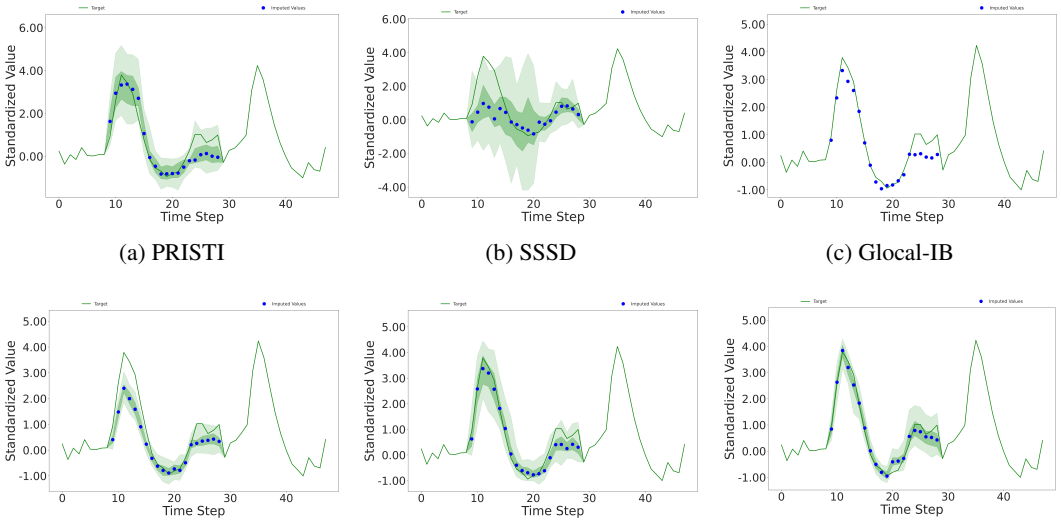
Figure 11: Qualitative imputation results on the ETTh1 dataset under a subsequence missing length of 20. The solid green lines represent the ground truth targets, and the blue dots denote the imputed values (median of the generated samples). The light green shaded regions indicate the confidence intervals of the probabilistic predictions.



(a) PRISTI (b) SSSD (c) Glocal-IB

(d) FGTI (e) FSDI (f) Ours

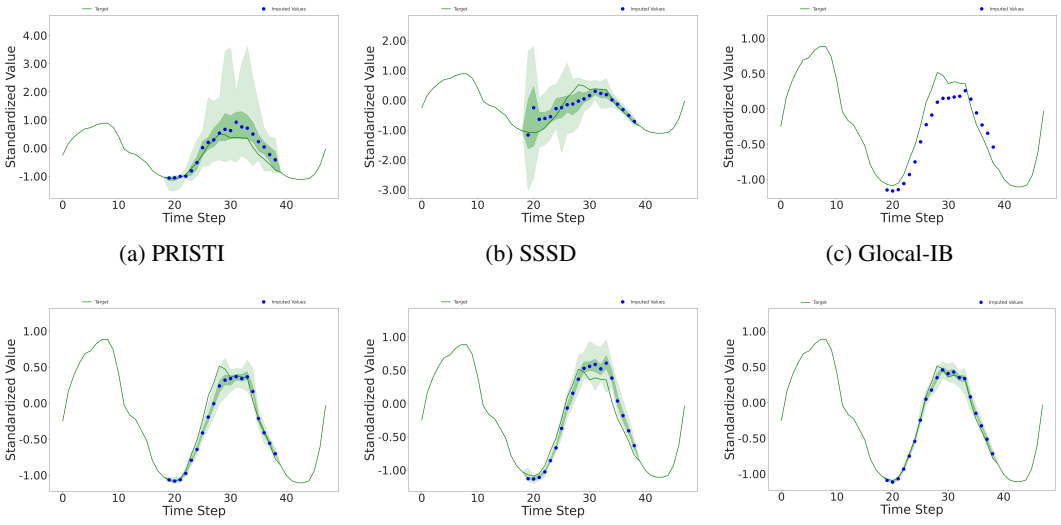
Figure 12: Qualitative imputation results on the Weather dataset under a subsequence missing length of 20. The solid green lines represent the ground truth targets, and the blue dots denote the imputed values (median of the generated samples). The light green shaded regions indicate the confidence intervals of the probabilistic predictions.



(a) PRISTI (b) SSSD (c) Glocal-IB

(d) FGTI (e) FSDI (f) Ours

Figure 13: Qualitative imputation results on the Electricity dataset under a subsequence missing length of 20. The solid green lines represent the ground truth targets, and the blue dots denote the imputed values (median of the generated samples). The light green shaded regions indicate the confidence intervals of the probabilistic predictions.



(a) PRISTI (b) SSSD (c) Glocal-IB

(d) FGTI (e) FSDI (f) Ours

Figure 14: Qualitative imputation results on the PEMS-Traffic dataset under a subsequence missing length of 20. The solid green lines represent the ground truth targets, and the blue dots denote the imputed values (median of the generated samples). The light green shaded regions indicate the confidence intervals of the probabilistic predictions.

Table 10: Imputation performance (RMSE) on ETTh1, ILI, and Stock datasets. **Bold** and underline denote the best and second-best results. Model abbreviations: **NonStat.**: Non-stationary Transformer, **Cross.**: Crossformer, **Patch.**: PatchTST, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB.

Dataset	Type	Setting	BRITS	GRIN	PRISTI	NonStat.	Cross.	FreTS	Koopa	Patch.	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
ETTh1	POINT	10%	0.413	0.234	0.291	0.240	0.359	0.314	0.335	0.315	0.236	0.303	0.398	0.250	0.245	0.221	0.211	0.246	0.248	**0.186**
		50%	0.587	0.340	0.399	0.385	0.473	0.445	0.452	0.432	0.357	0.468	0.806	0.373	0.394	0.442	0.316	0.415	**0.301**	0.302
		90%	1.020	0.877	1.447	0.686	0.728	0.773	0.767	0.777	0.706	0.768	0.890	0.849	0.725	0.905	0.689	0.743	0.588	**0.535**
	SEQ	10	0.606	0.450	0.749	0.449	0.514	0.590	0.586	0.570	0.399	0.496	0.522	0.378	0.370	0.610	0.353	0.369	0.304	**0.285**
		20	0.671	0.485	0.983	0.459	0.508	0.603	0.604	0.601	0.417	0.501	0.599	0.415	0.377	0.963	0.373	0.386	0.339	**0.248**
		30	0.730	0.507	0.952	0.478	0.546	0.682	0.697	0.676	0.470	0.573	0.590	0.497	0.404	0.827	0.419	0.411	0.329	**0.301**
	BLOCK	10%	0.719	0.556	0.581	0.551	0.547	0.519	0.551	0.501	0.435	0.513	0.742	0.587	0.577	0.512	0.413	0.576	0.714	**0.338**
		50%	0.820	0.640	0.708	0.554	0.642	0.630	0.649	0.605	0.534	0.627	0.799	0.687	0.626	0.645	0.487	0.631	0.684	**0.477**
		90%	0.896	0.708	0.783	0.599	0.689	0.706	0.713	0.686	0.604	0.683	0.868	0.736	0.664	0.732	0.562	0.665	0.638	**0.484**
ILI	POINT	10%	2.236	1.575	3.981	0.634	1.566	0.582	1.432	0.605	0.636	0.709	2.826	0.727	1.232	1.259	0.492	1.216	2.831	**0.350**
		50%	2.328	1.766	4.879	0.770	1.637	0.844	1.580	0.851	0.791	1.247	3.655	1.325	1.451	1.871	0.798	1.459	2.992	**0.397**
		90%	2.442	2.196	4.334	1.582	2.340	1.607	1.920	1.597	1.491	2.038	2.611	1.699	2.204	2.475	1.469	2.290	3.340	**1.019**
	SEQ	10	2.435	1.797	4.039	1.002	1.911	1.335	1.966	1.553	1.008	1.410	2.315	1.071	1.432	1.335	1.210	1.384	1.771	**0.404**
		20	2.544	1.859	4.311	1.347	1.992	1.878	2.278	1.927	1.320	1.807	2.180	1.388	1.435	1.893	1.575	1.421	2.007	**0.432**
		30	2.382	1.779	3.967	1.732	1.964	2.122	2.445	2.212	1.807	2.027	2.342	1.238	1.416	1.551	1.791	1.417	2.574	**0.395**
	BLOCK	10%	2.376	1.862	4.032	1.335	1.713	1.097	1.900	1.106	**0.996**	1.274	2.301	1.480	1.796	1.729	1.040	1.802	1.730	1.057
		50%	2.377	1.958	4.577	1.327	1.811	1.346	1.850	1.360	1.237	1.618	4.464	1.435	1.825	2.071	1.232	1.830	2.372	**0.995**
		90%	2.434	2.068	4.506	1.581	1.936	1.610	1.974	1.609	1.497	1.857	4.388	1.651	1.924	2.153	1.523	1.921	2.866	**1.098**
Stock	POINT	10%	3.321	2.551	1.318	0.087	2.762	0.141	0.177	0.124	0.091	0.510	0.215	1.798	2.578	0.664	0.101	2.214	1.112	**0.037**
		50%	3.703	2.882	4.380	0.113	2.845	0.156	0.182	0.149	0.113	0.957	0.251	1.986	2.893	1.623	0.134	2.337	1.165	**0.062**
		90%	4.404	3.193	3.847	0.460	3.397	0.461	0.473	0.474	0.459	3.231	0.503	3.101	2.927	1.713	0.450	2.504	1.825	**0.151**
	SEQ	10	3.334	3.208	1.206	0.119	2.695	0.219	0.233	0.208	0.118	0.328	0.226	2.234	2.609	0.512	0.162	2.223	1.424	**0.036**
		20	3.395	3.330	1.692	0.138	2.826	0.240	0.256	0.241	0.142	0.402	0.256	2.143	2.611	0.662	0.165	2.251	0.934	**0.040**
		30	3.394	3.388	5.245	0.182	2.828	0.274	0.285	0.274	0.188	0.363	0.276	2.561	2.716	1.118	0.184	2.211	2.093	**0.034**
	BLOCK	10%	4.138	3.021	1.306	0.228	2.658	0.195	0.208	0.179	0.168	0.251	0.249	2.463	3.255	0.375	0.163	2.208	0.871	**0.115**
		50%	4.262	3.152	6.405	0.242	3.097	0.207	0.221	0.203	0.200	0.960	0.268	2.548	3.338	1.636	0.196	2.345	2.111	**0.176**
		90%	4.342	3.292	10.641	0.257	3.226	0.225	0.241	0.222	0.227	1.145	0.276	2.494	3.345	2.617	0.218	2.391	2.304	**0.197**

Table 11: Imputation performance (RMSE) on Weather, Electricity, PEMS-Traffic, and METR-LA datasets. **Bold** and underline denote the best and second-best results. Model abbreviations: **NonStat.**: Non-stationary Transformer, **Cross.**: Crossformer, **Patch.**: PatchTST, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB.

Dataset	Type	Setting	BRITS	GRIN	PRISTI	NonStat.	Cross.	FreTS	Koopa	Patch.	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
Weather	POINT	10%	0.496	0.371	0.384	0.368	0.444	0.396	0.400	0.381	0.362	0.408	0.405	0.425	0.381	0.377	**0.350**	0.384	0.369	0.351
		50%	0.549	0.415	0.409	0.435	0.502	0.443	0.438	0.428	0.409	0.534	0.616	0.540	0.432	0.792	0.407	0.431	0.376	**0.369**
		90%	0.868	0.663	0.734	0.640	0.663	0.675	0.661	0.669	0.648	0.784	0.806	1.066	0.586	1.776	0.624	0.593	0.481	**0.443**
	SEQ	10	0.538	0.462	0.463	0.460	0.506	0.488	0.484	0.474	0.457	0.494	0.463	0.556	0.437	0.403	0.422	0.437	0.446	**0.335**
		20	0.566	0.515	0.593	0.532	0.540	0.565	0.567	0.562	0.532	0.560	0.531	0.566	0.454	0.432	0.476	0.457	0.422	**0.357**
		30	0.602	0.569	0.673	0.645	0.580	0.665	0.666	0.657	0.635	0.624	0.651	0.588	0.482	0.483	0.549	0.482	0.445	**0.350**
	BLOCK	10%	0.538	0.480	0.541	0.476	0.533	0.462	0.455	0.445	0.437	0.494	0.479	0.626	0.443	0.447	0.420	0.445	0.450	**0.392**
		50%	0.565	0.509	0.514	0.530	0.590	0.530	0.528	0.514	0.511	0.610	0.521	0.688	0.487	0.459	0.489	0.491	0.469	**0.403**
		90%	0.618	0.561	0.573	0.605	0.630	0.611	0.604	0.607	0.583	0.727	0.630	0.811	0.515	0.486	0.560	0.515	0.523	**0.428**
Electricity	POINT	10%	0.553	0.315	0.372	0.356	0.392	0.367	0.423	0.371	0.356	0.357	0.458	0.643	0.410	0.334	0.310	0.451	0.338	**0.261**
		50%	0.603	0.416	0.541	0.450	0.499	0.548	0.561	0.531	0.429	0.522	0.914	0.840	0.489	0.476	0.415	0.515	0.347	**0.310**
		90%	0.767	0.697	0.603	0.839	0.668	0.999	1.003	1.000	0.814	0.843	1.176	1.502	0.680	0.725	0.815	0.703	0.648	**0.488**
	SEQ	10	0.596	0.543	1.024	0.466	0.511	0.562	0.604	0.554	0.443	0.492	0.622	0.494	0.497	0.427	0.420	0.516	0.437	**0.264**
		20	0.596	0.603	1.243	0.487	0.508	0.588	0.610	0.591	0.461	0.502	0.692	0.558	0.509	0.453	0.438	0.526	0.448	**0.310**
		30	0.599	0.615	1.148	0.576	0.566	0.785	0.805	0.782	0.560	0.638	0.670	0.594	0.538	0.522	0.545	0.549	0.469	**0.351**
	BLOCK	10%	0.536	0.474	0.870	0.483	0.529	0.608	0.639	0.591	0.461	0.541	0.530	0.707	0.512	0.439	0.446	0.525	0.432	**0.272**
		50%	0.557	0.514	0.727	0.574	0.567	0.778	0.791	0.743	0.544	0.669	0.790	0.748	0.530	0.472	0.519	0.541	0.436	**0.347**
		90%	0.589	0.566	0.740	0.706	0.617	0.940	0.950	0.906	0.679	0.783	0.921	0.943	0.565	0.515	0.638	0.575	0.515	**0.379**
PEMS-Traffic	POINT	10%	0.528	0.365	0.428	0.384	0.427	0.417	0.474	0.421	0.390	0.400	0.494	0.376	0.396	**0.217**	0.342	0.435	0.379	0.219
		50%	0.562	0.437	0.569	0.485	0.499	0.584	0.590	0.572	0.468	0.521	0.917	0.495	0.466	0.412	0.440	0.483	0.403	**0.337**
		90%	0.716	0.697	0.864	0.848	0.610	0.955	0.970	0.958	0.823	0.746	1.118	0.663	0.581	0.653	0.812	0.593	0.575	**0.519**
	SEQ	10	0.523	0.502	0.915	0.487	0.499	0.625	0.630	0.623	0.474	0.520	0.619	0.572	0.452	0.444	0.439	0.455	0.443	**0.259**
		20	0.534	0.521	0.940	0.504	0.513	0.643	0.636	0.646	0.487	0.524	0.594	0.549	0.447	0.523	0.448	0.464	0.444	**0.311**
		30	0.527	0.522	0.841	0.578	0.522	0.746	0.773	0.741	0.569	0.600	0.625	0.524	0.452	0.751	0.534	0.464	0.444	**0.356**
	BLOCK	10%	0.488	0.475	0.847	0.503	0.517	0.656	0.673	0.649	0.492	0.564	0.552	0.556	0.451	0.362	0.454	0.469	0.451	**0.265**
		50%	0.509	0.495	0.703	0.598	0.530	0.752	0.765	0.732	0.563	0.623	0.815	0.646	0.479	0.417	0.520	0.491	0.453	**0.337**
		90%	0.524	0.513	0.674	0.719	0.555	0.876	0.892	0.860	0.684	0.685	0.895	0.629	0.513	0.448	0.639	0.518	0.494	**0.383**
METR-LA	POINT	10%	0.575	0.277	0.308	0.303	0.301	0.309	0.323	0.308	0.294	0.295	0.378	0.387	0.299	0.282	0.270	0.333	0.268	**0.207**
		50%	0.609	0.336	0.335	0.362	0.365	0.347	0.353	0.342	0.338	0.369	0.470	0.408	0.378	0.336	0.326	0.408	0.289	**0.276**
		90%	0.669	0.628	0.773	0.503	0.509	0.479	0.482	0.477	0.504	0.670	0.536	0.686	0.615	0.556	0.481	0.632	0.448	**0.422**
	SEQ	10	0.608	0.394	0.421	0.438	0.392	0.397	0.411	0.394	0.421	0.394	0.409	0.454	0.408	0.418	0.386	0.424	0.395	**0.259**
		20	0.616	0.473	0.520	0.470	0.456	0.461	0.473	0.460	0.465	0.444	0.463	0.482	0.445	0.464	0.441	0.466	0.448	**0.311**
		30	0.615	0.631	0.530	0.505	0.486	0.506	0.513	0.507	0.502	0.471	0.524	0.533	0.456	0.553	0.476	0.468	0.472	**0.356**
	BLOCK	10%	0.594	0.383	0.384	0.423	0.389	0.376	0.394	0.375	0.392	0.388	0.390	0.445	0.396	0.390	0.360	0.414	0.362	**0.233**
		50%	0.611	0.402	0.374	0.446	0.432	0.408	0.422	0.404	0.420	0.443	0.469	0.511	0.437	0.404	0.399	0.446	0.363	**0.302**
		90%	0.626	0.453	0.417	0.481	0.486	0.453	0.466	0.447	0.469	0.509	0.533	0.671	0.480	0.428	0.447	0.506	0.406	**0.353**

Table 12: Imputation performance (MAE) on ETTh1, ILI, and Stock datasets. **Bold** and underline denote the best and second-best results. Model abbreviations: **NonStat.**: Non-stationary Transformer, **Cross.**: Crossformer, **Patch.**: PatchTST, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB.

Dataset	Type	Setting	BRITS	GRIN	PRISTI	NonStat.	Cross.	FreTS	Koopa	Patch.	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
ETTh1	POINT	10%	0.281	0.159	0.184	0.159	0.244	0.207	0.227	0.203	0.156	0.212	0.243	0.170	0.171	0.150	0.143	0.172	0.123	**0.115**
		50%	0.413	0.235	0.245	0.251	0.332	0.286	0.292	0.277	0.232	0.327	0.524	0.252	0.276	0.308	0.210	0.293	0.175	**0.174**
		90%	0.766	0.633	0.896	0.431	0.522	0.473	0.469	0.475	0.436	0.542	0.567	0.609	0.519	0.549	0.423	0.532	0.363	**0.334**
	SEQ	10	0.426	0.310	0.455	0.287	0.355	0.398	0.400	0.380	0.260	0.333	0.329	0.249	0.238	0.369	0.228	0.237	**0.167**	0.170
		20	0.483	0.325	0.602	0.296	0.354	0.406	0.409	0.402	0.272	0.334	0.407	0.272	0.244	0.584	0.239	0.248	0.176	**0.149**
		30	0.533	0.338	0.590	0.312	0.384	0.458	0.465	0.454	0.313	0.389	0.392	0.322	0.254	0.525	0.273	0.259	0.179	**0.168**
	BLOCK	10%	0.509	0.392	0.369	0.373	0.373	0.350	0.370	0.335	0.291	0.348	0.483	0.402	0.380	0.333	0.287	0.381	0.444	**0.220**
		50%	0.596	0.450	0.441	0.372	0.440	0.408	0.420	0.392	0.349	0.426	0.521	0.474	0.426	0.408	0.328	0.421	0.424	**0.283**
		90%	0.661	0.500	0.500	0.401	0.478	0.451	0.455	0.439	0.393	0.468	0.561	0.510	0.450	0.467	0.371	0.453	0.400	**0.301**
ILI	POINT	10%	1.474	1.071	2.261	0.416	0.900	0.382	0.991	0.401	0.417	0.489	0.919	0.333	0.772	0.879	0.293	0.776	1.432	**0.169**
		50%	1.578	1.225	3.239	0.500	0.976	0.522	1.100	0.538	0.518	0.877	1.308	0.666	0.953	1.355	0.479	1.004	1.176	**0.196**
		90%	1.685	1.545	3.154	0.990	1.623	1.021	1.303	1.004	0.927	1.443	1.382	1.049	1.531	1.810	0.899	1.591	1.920	**0.555**
	SEQ	10	1.579	1.211	2.373	0.608	1.170	0.821	1.347	1.018	0.623	0.931	1.237	0.558	0.893	0.833	0.723	0.875	0.761	**0.213**
		20	1.653	1.244	2.868	0.864	1.290	1.231	1.576	1.264	0.841	1.186	1.203	0.767	0.858	1.240	0.951	0.878	1.372	**0.236**
		30	1.558	1.206	2.737	1.154	1.271	1.420	1.731	1.502	1.197	1.374	2.394	0.667	0.853	1.083	1.141	0.878	1.789	**0.210**
	BLOCK	10%	1.595	1.277	2.390	0.892	1.038	0.722	1.359	0.702	0.633	0.876	1.149	0.843	1.209	1.127	0.640	1.235	1.250	**0.573**
		50%	1.614	1.349	3.059	0.852	1.137	0.871	1.285	0.863	0.795	1.097	3.738	0.834	1.231	1.347	0.768	1.256	1.197	**0.539**
		90%	1.649	1.424	3.137	1.001	1.254	1.034	1.364	1.019	0.941	1.279	3.645	0.972	1.283	1.422	0.957	1.313	1.315	**0.608**
Stock	POINT	10%	2.809	2.376	1.106	0.060	2.311	0.100	0.126	0.087	0.064	0.366	0.151	1.228	2.304	0.461	0.071	1.533	0.580	**0.020**
		50%	3.198	2.708	3.358	0.078	2.410	0.108	0.129	0.103	0.079	0.706	0.183	1.530	2.562	0.831	0.093	1.649	0.683	**0.038**
		90%	3.903	3.003	2.287	0.195	2.992	0.196	0.216	0.216	0.193	2.909	0.230	2.664	2.548	1.272	0.181	1.768	1.404	**0.102**
	SEQ	10	2.813	3.044	0.991	0.087	2.254	0.157	0.168	0.148	0.085	0.244	0.162	1.635	2.325	0.384	0.115	1.557	1.132	**0.020**
		20	2.859	3.169	1.144	0.099	2.379	0.175	0.186	0.174	0.101	0.288	0.185	1.624	2.316	0.470	0.118	1.569	0.695	**0.023**
		30	2.857	3.227	4.318	0.132	2.387	0.198	0.206	0.200	0.136	0.269	0.203	2.030	2.403	0.833	0.130	1.542	1.329	**0.018**
	BLOCK	10%	3.641	2.844	1.156	0.155	2.226	0.139	0.149	0.127	0.117	0.185	0.180	2.118	2.885	0.278	0.116	1.518	0.656	**0.082**
		50%	3.760	2.969	4.836	0.165	2.660	0.146	0.157	0.143	0.139	0.686	0.194	2.181	2.944	1.367	0.137	1.660	1.489	**0.125**
		90%	3.837	3.110	8.892	0.177	2.792	0.158	0.171	0.155	0.153	0.812	0.199	2.120	2.969	2.075	0.156	1.705	1.445	**0.140**

Table 13: Imputation performance (MAE) on Weather, Electricity, PEMS-Traffic, and METR-LA datasets. **Bold** and underline denote the best and second-best results. Model abbreviations: **NonStat.**: Non-stationary Transformer, **Cross.**: Crossformer, **Patch.**: PatchTST, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB.

Dataset	Type	Setting	BRITS	GRIN	PRISTI	NonStat.	Cross.	FreTS	Koopa	Patch.	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
Weather	POINT	10%	0.209	0.114	0.093	0.111	0.190	0.136	0.139	0.124	0.109	0.163	0.102	0.114	0.142	0.078	0.104	0.144	0.067	**0.054**
		50%	0.248	0.152	0.100	0.164	0.240	0.172	0.168	0.159	0.144	0.265	0.303	0.198	0.189	0.353	0.137	0.187	0.077	**0.064**
		90%	0.483	0.357	0.310	0.285	0.363	0.302	0.292	0.297	0.292	0.424	0.418	0.594	0.322	0.996	0.269	0.327	0.153	**0.141**
	SEQ	10	0.240	0.199	0.143	0.161	0.230	0.188	0.191	0.184	0.167	0.198	0.151	0.202	0.181	0.092	0.140	0.182	0.097	**0.058**
		20	0.258	0.260	0.195	0.226	0.269	0.262	0.267	0.260	0.237	0.262	0.213	0.219	0.203	0.126	0.183	0.206	0.110	**0.065**
		30	0.273	0.307	0.256	0.309	0.290	0.327	0.335	0.322	0.305	0.301	0.300	0.234	0.213	0.149	0.220	0.212	0.129	**0.069**
	BLOCK	10%	0.228	0.192	0.136	0.201	0.254	0.192	0.192	0.182	0.176	0.229	0.143	0.239	0.197	0.135	0.163	0.198	0.126	**0.097**
		50%	0.250	0.220	0.150	0.230	0.288	0.226	0.228	0.212	0.215	0.311	0.198	0.289	0.213	0.140	0.194	0.215	0.125	**0.101**
		90%	0.295	0.272	0.195	0.286	0.329	0.277	0.283	0.274	0.266	0.386	0.270	0.387	0.239	0.159	0.245	0.240	0.149	**0.123**
Electricity	POINT	10%	0.412	0.224	0.251	0.258	0.273	0.259	0.308	0.260	0.258	0.253	0.317	0.452	0.302	0.235	0.222	0.336	0.232	**0.186**
		50%	0.450	0.299	0.358	0.327	0.365	0.391	0.406	0.377	0.312	0.380	0.708	0.573	0.367	0.372	0.301	0.391	0.237	**0.217**
		90%	0.558	0.522	0.423	0.590	0.499	0.705	0.710	0.705	0.568	0.628	0.893	1.088	0.520	0.497	0.571	0.542	0.442	**0.344**
	SEQ	10	0.445	0.405	0.737	0.340	0.368	0.393	0.432	0.389	0.320	0.346	0.436	0.353	0.373	0.307	0.303	0.391	0.299	**0.183**
		20	0.446	0.451	0.916	0.348	0.365	0.400	0.428	0.408	0.326	0.346	0.488	0.402	0.383	0.318	0.308	0.399	0.306	**0.215**
		30	0.447	0.458	0.843	0.412	0.414	0.541	0.568	0.542	0.398	0.446	0.460	0.429	0.408	0.374	0.387	0.418	0.324	**0.245**
	BLOCK	10%	0.399	0.351	0.623	0.356	0.386	0.435	0.465	0.420	0.338	0.383	0.366	0.501	0.386	0.318	0.327	0.395	0.305	**0.193**
		50%	0.413	0.382	0.527	0.421	0.418	0.558	0.574	0.532	0.398	0.488	0.569	0.527	0.400	0.346	0.380	0.410	0.307	**0.249**
		90%	0.437	0.419	0.740	0.510	0.454	0.673	0.687	0.647	0.490	0.574	0.667	0.657	0.428	0.377	0.458	0.437	0.352	**0.272**
PEMS-Traffic	POINT	10%	0.256	0.166	0.187	0.206	0.214	0.220	0.271	0.212	0.204	0.208	0.217	0.166	0.201	0.123	0.178	0.225	0.135	**0.090**
		50%	0.281	0.224	0.269	0.273	0.271	0.337	0.348	0.321	0.257	0.304	0.585	0.221	0.240	0.267	0.244	0.250	0.145	**0.126**
		90%	0.368	0.421	0.528	0.524	0.357	0.595	0.609	0.597	0.504	0.466	0.787	0.319	0.326	0.271	0.501	0.337	0.247	**0.238**
	SEQ	10	0.261	0.290	0.551	0.271	0.271	0.331	0.368	0.332	0.257	0.279	0.298	0.292	0.238	0.218	0.232	0.238	0.173	**0.102**
		20	0.266	0.304	0.600	0.277	0.276	0.334	0.365	0.343	0.259	0.270	0.288	0.274	0.232	0.260	0.232	0.245	0.173	**0.121**
		30	0.262	0.306	0.508	0.340	0.290	0.427	0.472	0.421	0.330	0.343	0.312	0.253	0.235	0.318	0.307	0.244	0.175	**0.139**
	BLOCK	10%	0.235	0.253	0.487	0.278	0.284	0.371	0.399	0.364	0.265	0.321	0.246	0.266	0.234	0.191	0.249	0.248	0.170	**0.099**
		50%	0.246	0.263	0.381	0.356	0.292	0.454	0.479	0.437	0.327	0.370	0.451	0.323	0.249	0.219	0.300	0.258	0.172	**0.131**
		90%	0.258	0.278	0.373	0.444	0.311	0.547	0.572	0.531	0.417	0.418	0.540	0.315	0.273	0.231	0.383	0.275	0.195	**0.155**
METR-LA	POINT	10%	0.300	0.162	0.163	0.173	0.184	0.167	0.177	0.167	0.166	0.176	0.185	0.243	0.191	0.159	0.154	0.210	0.141	**0.116**
		50%	0.316	0.189	0.173	0.197	0.217	0.183	0.190	0.180	0.185	0.225	0.256	0.226	0.233	0.182	0.178	0.248	0.149	**0.143**
		90%	0.355	0.381	0.388	0.257	0.296	0.243	0.245	0.242	0.256	0.404	0.284	0.382	0.337	0.259	0.244	0.359	**0.208**	0.209
	SEQ	10	0.318	0.222	0.210	0.230	0.225	0.204	0.214	0.205	0.217	0.219	0.195	0.261	0.246	0.217	0.203	0.255	0.184	**0.102**
		20	0.327	0.273	0.246	0.241	0.262	0.236	0.242	0.238	0.236	0.247	0.225	0.254	0.265	0.242	0.231	0.275	0.208	**0.121**
		30	0.329	0.383	0.269	0.260	0.283	0.262	0.266	0.263	0.257	0.262	0.259	0.293	0.271	0.276	0.251	0.279	0.221	**0.139**
	BLOCK	10%	0.297	0.212	0.196	0.225	0.226	0.198	0.210	0.200	0.210	0.226	0.190	0.237	0.239	0.207	0.196	0.250	0.175	**0.128**
		50%	0.302	0.225	0.201	0.235	0.251	0.408	0.223	0.214	0.223	0.265	0.237	0.267	0.261	0.216	0.215	0.270	0.176	**0.153**
		90%	0.313	0.255	0.217	0.250	0.279	0.235	0.243	0.233	0.243	0.313	0.274	0.362	0.282	0.227	0.236	0.295	0.194	**0.174**