TIME-FREQUENCY MEMORY-GUIDED DIFFUSION FOR GENERATIVE TIME SERIES IMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Score-based diffusion models have shown strong performance in probabilistic time series imputation. However, their performance can degrade under extreme sparsity, where limited observations provide insufficient context to reliably guide the reverse diffusion process. Consequently, while the generated imputations may exhibit plausible local patterns, they often fail to adequately preserve the broader temporal structures aligned with the observed context. To address this issue, we propose *Time-Frequency Memory-Guided Diffusion* (TFMemDiff), a unified framework that augments conditional diffusion with complementary memory priors in both the temporal and frequency domains. Rather than treating memory vectors as auxiliary features to be simply concatenated with diffusion inputs, TFMemDiff uses memory as an external structural reference that guides conditional score estimation throughout the reverse diffusion process. Specifically, TFMemDiff incorporates a *Frequency Memory Module* (FMM) to retrieve recurring global spectral patterns and a *Temporal Memory Module* (TMM) to capture localized temporal dependencies. These memories distill reusable structural knowledge from historical training sequences and make it accessible to each incomplete instance during denoising. Extensive experiments on multiple real-world datasets demonstrate the effectiveness of TFMemDiff across diverse missingness settings, showing that memory-guided structural priors can substantially improve generative imputation under severe sparsity.

1 INTRODUCTION

While time series data is critical for diverse applications (Johnson et al., 2016; Li et al., 2018; Zhou et al., 2021), real-world deployments frequently suffer from missing values due to sensor or transmission failures (Cao et al., 2018; Yoon et al., 2018). The resulting incompleteness, which manifests as either sporadic point missing or severe continuous block missing, poses substantial challenges for downstream modeling and decision-making (Tashiro et al., 2021). Consequently, time series imputation has become a fundamental yet challenging research problem (Wu et al., 2023; Alcaraz & Strodthoff, 2023).

Existing time series imputation methods can be broadly categorized into *deterministic* and *generative* approaches. Deterministic models, typically based on RNNs (e.g., BRITS (Cao et al., 2018)) or Transformers (e.g., SAITS (Du et al., 2023)), or advanced information bottleneck paradigms (e.g., Glocal IB (Yang et al., 2025)), optimize point-wise reconstruction objectives and offer computational efficiency. However, these methods typically produce point estimates and do not explicitly model the uncertainty associated with missing values. In contrast, generative models, ranging from GANs (Yoon et al., 2018) to score-based diffusion models (e.g., CSDI (Tashiro et al., 2021), SSSD (Alcaraz & Strodthoff, 2023), FGTI (Yang et al., 2024)), explicitly model the data distribution and iteratively generate imputations via denoising. This paradigm has demonstrated strong empirical performance by producing diverse and realistic samples that better reflect the stochastic nature of time series data.

Despite their strong empirical performance (Tashiro et al., 2021; Alcaraz & Strodthoff, 2023), current diffusion-based imputation models can become less reliable under extreme sparsity (Liu et al., 2023). When the missing rate is high, the remaining observations may provide only limited contextual information for conditional generation, making it difficult for the denoising process to accurately recover severely missing regions. This limitation suggests that conditioning solely on the current

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.

incomplete instance may be insufficient, especially when the observed context no longer contains enough evidence to infer global dynamics or recurring temporal patterns.

At a fundamental level, this limitation arises because existing diffusion-based imputation models rely primarily on information extracted from the current incomplete sequence itself. When observations are extremely sparse, this instance-level context may be too weak to recover long-range temporal regularities or recurring patterns. This motivates us to introduce structural guidance beyond sparse local observations. In this work, we propose to use memory priors learned from historical training sequences as external structural references for diffusion-based imputation. Importantly, the memory is not used as a simple additional feature attached to the input. Instead, it serves as a reusable structural knowledge base that is dynamically retrieved and injected into conditional score estimation. To the best of our knowledge, memory-augmented *generative* time series imputation remains underexplored, and TFMemDiff provides an early attempt to formulate it within a diffusion framework.

To bridge this gap, we propose *Time-Frequency Memory-Guided Diffusion Imputation* (TFMemDiff), a unified framework that anchors the generative process with memory priors learned from historical observations, providing structural references beyond the sparse context of the current sequence. The key idea is to shift conditional generation from being purely instance-conditioned to being memory-guided: the reverse diffusion process is conditioned not only on the observed values of the current sequence, but also on retrieved structural prototypes accumulated from historical data. Unlike conventional approaches that rely solely on sparse contextual observations and are thus more susceptible to structurally inconsistent imputations, TFMemDiff introduces a *dual-domain memory mechanism* to provide complementary structural guidance. Specifically, TFMemDiff organizes historical structural knowledge into two complementary memory spaces. The *Frequency Memory Module (FMM)* stores and retrieves stable global periodic prototypes, while the *Temporal Memory Module (TMM)* retrieves instance-specific local temporal patterns. These retrieved memories act as conditional structural priors for score estimation rather than as direct substitutes for missing values. By guiding the denoising trajectory with both global spectral and local temporal references, TFMemDiff better constrains the diffusion process and promotes structurally coherent imputations under extreme missing conditions.

Our contributions are summarized as follows:

i) We identify an underexplored failure mode of diffusion-based time series imputation under extreme sparsity, where the observed context of the current incomplete sequence becomes too weak to provide reliable structural guidance for reverse diffusion.

ii) We introduce a memory-guided generative imputation perspective. Instead of conditioning diffusion solely on the sparse observations of each incomplete instance, TFMemDiff retrieves reusable structural prototypes learned from historical training data and uses them as external priors for conditional score estimation.

iii) We design a dual-domain memory architecture that organizes structural knowledge into complementary frequency and temporal memories. The Frequency Memory Module (FMM) retrieves global spectral prototypes, while the Temporal Memory Module (TMM) retrieves instance-specific local temporal patterns, jointly providing multi-scale guidance for denoising.

iv) Experiments on multiple real-world datasets validate the effectiveness of TFMemDiff across diverse missingness settings, especially under severe sparsity.

2 Related Works

Diffusion-based Time Series Imputation. Diffusion-based imputation methods have progressively improved conditional generation from different perspectives. CSDI (Tashiro et al., 2021) establishes conditional score-based diffusion for time series imputation, while SSSD (Alcaraz & Strodthoff, 2023) strengthens temporal dependency modeling with structured state-space layers. More recent methods further explore frequency information: FGTI (Yang et al., 2024) incorporates frequency-aware representations into conditional generation, whereas FSDI (Shen et al., 2026) exploits spectral characteristics to reshape the diffusion process.

As summarized in Table 1, these approaches mainly enhance temporal representations, frequency-aware representations, or the diffusion dynamics of the current incomplete sequence. In contrast,

Table 1: Conceptual comparison with diffusion-based time series imputation methods.

Categories	CSDI	SSSD	FGTI	FSDI	Ours
Frequency-aware Modeling	×	×	✓	✓	✓
Historical Memory Prior	×	×	×	×	✓
External Structural Guidance	×	×	×	×	✓
Time-Frequency Memory Retrieval	×	×	×	×	✓

TFMemDiff introduces *historical global memory* as an explicit complementary source of guidance, allowing reusable structural knowledge learned from historical training data to supplement sparse instance-level observations. Moreover, TFMemDiff organizes such knowledge through complementary time- and frequency-domain memories, yielding dual-domain memory guidance for conditional diffusion under severe missingness.

The key distinction of TFMemDiff is not the use of memory as an additional neural component, but the formulation of memory as an external structural prior for generative imputation. Existing diffusion-based imputers mainly construct conditioning signals from the current incomplete sequence, possibly enhanced by temporal, state-space, or spectral representations. In contrast, TFMemDiff explicitly retrieves reusable time-frequency prototypes from historical training data and uses them to guide conditional score estimation. This makes TFMemDiff a memory-prior guided diffusion framework rather than a simple combination of diffusion and memory modules.

For completeness, we provide a broader review of statistical, deterministic, diffusion-based, and memory-augmented imputation methods in Appendix A.

3 METHODOLOGY

We propose **TFMemDiff**, a Time-Frequency Memory-Guided Diffusion framework for generative time series imputation. As illustrated in Figure 1, TFMemDiff consists of a conditional diffusion backbone and a dual-domain memory enhancement module. The backbone performs conditional denoising for generative imputation, while the memory module provides explicit structural priors retrieved from historical data.

The key idea of TFMemDiff is to stabilize diffusion-based imputation under severe context sparsity by introducing *dual-domain memory anchors*. Specifically, we maintain a Time Memory Bank to capture representative temporal patterns and a Frequency Memory Bank to model global spectral patterns. The retrieved prototypes from both domains are jointly injected into the reverse diffusion process, providing additional structural cues that help guide the denoising process across diverse missing patterns.

Problem formulation. We study the task of multivariate time series imputation. Let $\mathbf{X} \in \mathbb{R}^{L \times D}$ denote the complete time series and $\mathbf{M} \in \{0,1\}^{L \times D}$ be a binary mask indicating observed entries. The model receives the partially observed input $\tilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{M}$, where $\odot$ denotes element-wise multiplication. The objective is to learn a function $\mathcal{F}_\theta$ that reconstructs the complete sequence from the masked input and the observation mask:

$$\hat{\mathbf{X}} = \mathcal{F}_\theta(\tilde{\mathbf{X}}, \mathbf{M}), \tag{1}$$

where $\hat{\mathbf{X}}$ denotes the imputed result. Usually, the model is trained by minimizing the reconstruction error on missing entries: $\mathcal{L}(\theta) = \left\| (1 - \mathbf{M}) \odot (\mathbf{X} - \hat{\mathbf{X}}) \right\|$.

3.1 CONDITIONAL IMPUTATION BACKBONE

TFMemDiff adopts a conditional score-based diffusion model as the generative backbone for probabilistic imputation. Following DDPMs (Ho et al., 2020), a fixed forward process gradually corrupts the imputation target $\mathbf{x}_0^{\text{ta}}$ with Gaussian noise:

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

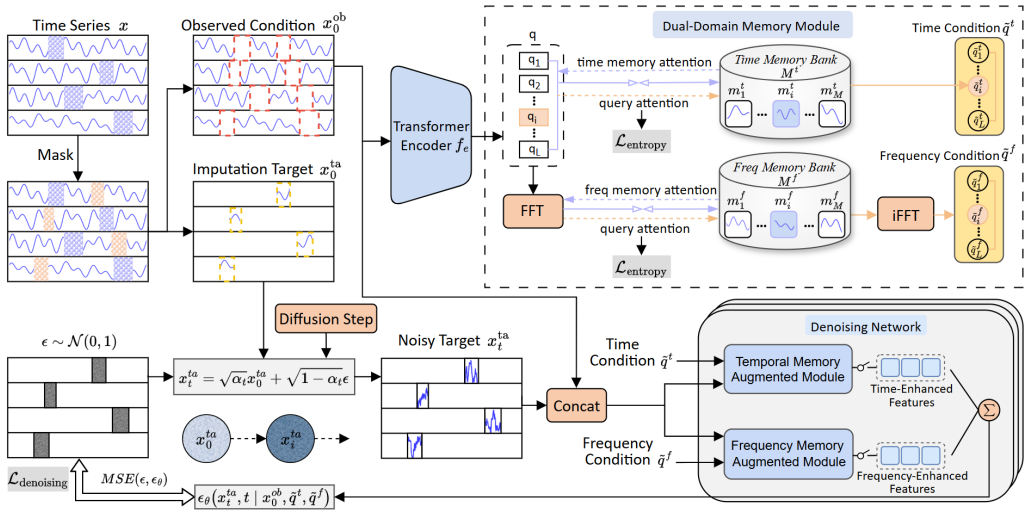


Figure 1: An overview of the proposed TFMemDiff framework. The model utilizes a Transformer encoder to extract latent queries, which interact with both time and frequency memory banks to retrieve domain-specific prototypes. These retrieved features serve as dual-domain structural priors to condition the denoising process for probabilistic time-series imputation.

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

The core of TFMemDiff 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 dual-domain memory priors retrieved from the memory module. We denote the retrieved time- and frequency-domain prototypes as $\tilde{\mathbf{q}} = \{\tilde{\mathbf{q}}^{\text{t}}, \tilde{\mathbf{q}}^{\text{f}}\}$. The reverse process is parameterized as

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

where the mean $\boldsymbol{\mu}_\theta$ is defined following standard DDPM parameterization, with the noise predictor $\boldsymbol{\epsilon}_\theta$ explicitly conditioned on $\tilde{\mathbf{q}}$. Here,

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

where the variance is set to a non-learnable, predefined constant $\tilde{\beta}_t$. The constant is given by

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

For efficient inference, we employ DPM-Solver++ with 20 sampling steps to accelerate the reverse denoising process. Additional notations, diffusion preliminaries, mathematical details of TFMemDiff, and the architecture of $\boldsymbol{\epsilon}_\theta$ (Figure 6) are provided in Appendices B, D, and F.5.

3.2 Memory Enhancement Module

To stabilize diffusion-based imputation under sparse observations, we introduce a Memory Enhancement Module (MEM) that provides explicit structural priors retrieved from historical data. MEM consists of two complementary memory banks: a Time Memory Bank $\mathcal{M}^{\text{t}}$ that captures representative temporal waveforms, and a Frequency Memory Bank $\mathcal{M}^{\text{f}}$ that models global spectral patterns. The retrieved prototypes from both domains are jointly used to guide the reverse diffusion process.

Prototype initialization. To provide informative inductive bias and avoid random memory initialization, we initialize both memory banks using a dual-domain clustering strategy. Specifically, to extract latent representations without target leakage, we employ a pre-trained Transformer encoder on a 10% subset of the training sequences, strictly applying the same masking protocol used during the training phase to ensure prototypes are derived solely from observable context. K-means clustering

is then applied to the latent features to initialize the Time Memory Bank $\mathcal{M}^t = \{m_1^t, \ldots, m_M^t\}$, where M denotes the predefined number of memory prototypes. For the Frequency Memory Bank $\mathcal{M}^{\mathrm{f}} = \{m_1^f, \ldots, m_M^f\}$, clustering is performed on the corresponding FFT-transformed representations. This procedure ensures that both banks contain diverse and highly distinct prototypical patterns prior to training.

Memory update. During training, both memory banks are incrementally refined using a Top-K sparse update strategy to mitigate memory collapse (Rae et al., 2016). As the update procedure is identical for both domains, we describe it using the Time Memory Bank $\mathcal{M}^{\mathrm{t}}$.

Let $\mathbf{q}$ denote the latent query representations. We define the valid query set $\mathcal{J}$ by retaining query positions with available observations. For each $\mathbf{q}_j$ ($j \in \mathcal{J}$), the Top-K most similar memory slots $\mathcal{N}_j^K$ are selected based on cosine similarity, yielding the sparse attention map:

$$\mathbf{v}_{i,j}^t = \begin{cases} \frac{\exp\left(\frac{\langle \mathbf{m}_i^t, \mathbf{q}_j \rangle}{\|\mathbf{m}_i^t\| \|\mathbf{q}_j\|} / \tau\right)}{\sum_{k \in \mathcal{N}_j^K} \exp\left(\frac{\langle \mathbf{m}_k^t, \mathbf{q}_j \rangle}{\|\mathbf{m}_k^t\| \|\mathbf{q}_j\|} / \tau\right)}, & \text{if } j \in \mathcal{J} \text{ and } i \in \mathcal{N}_j^K \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where τ is a temperature parameter.

The aggregated update content is computed as $\tilde{\mathbf{m}}_i^t = \sum_{j \in \mathcal{J}} \mathbf{v}_{i,j}^t \mathbf{q}_j$. A memory slot is activated only when selected by at least one valid query, indicated by $\mathbb{I}_i$, say $\mathbb{I}_i = \mathbb{1}[\sum_{j \in \mathcal{J}} \mathbf{v}_{i,j}^t > 0]$. Then, the gated update is performed by

$$\mathbf{m}_i^t \leftarrow \mathbb{I}_i \cdot \left[(1 - \psi_i^t) \circ \mathbf{m}_i^t + \psi_i^t \circ \tilde{\mathbf{m}}_i^t\right] + (1 - \mathbb{I}_i) \cdot \mathbf{m}_i^t, \tag{7}$$

where $\psi_i^t = \sigma\left(\mathbf{m}_i^t U_\psi + \tilde{\mathbf{m}}_i^t W_\psi\right)$. U_ψ and W_ψ are learnable projection matrices, σ is the Sigmoid function, and $\circ$ denotes element-wise multiplication. The Frequency Memory Bank $\mathbf{M}^{\mathrm{f}}$ is updated in the same manner.

Dual-domain retrieval. During the forward pass, the model retrieves relevant prototypes from both memory banks to construct complementary structural guidance. Let $u \in \{\mathrm{t}, \mathrm{f}\}$ denote the time and frequency domains. The domain-specific queries are defined as

$$\mathbf{q}^{\mathrm{t}} = \mathbf{q}, \qquad \mathbf{q}^{\mathrm{f}} = \phi_{\mathrm{f}}(\mathbf{q}), \tag{8}$$

where $\phi_{\mathrm{f}}(\cdot)$ denotes an rFFT-based frequency transformation that encodes the real and imaginary spectral components into the latent feature space through a learnable projection.

For each domain, similarity-based attention weights over all memory slots are computed as

$$\mathbf{w}_{j,i}^u = \frac{\exp\left(\frac{\langle \mathbf{m}_i^u, \mathbf{q}_j^u \rangle}{\|\mathbf{m}_i^u\| \|\mathbf{q}_j^u\|} / \tau'\right)}{\sum_{k=1}^M \exp\left(\frac{\langle \mathbf{m}_k^u, \mathbf{q}_j^u \rangle}{\|\mathbf{m}_k^u\| \|\mathbf{q}_j^u\|} / \tau'\right)}, \tag{9}$$

where j indexes the query sequence and i indexes the memory slots, τ' here is a temperature parameter that controls the sharpness of the attention distribution.

Using these weights, we aggregate the prototypes to construct the final conditions:

$$\tilde{\mathbf{q}}_j^{\mathrm{t}} = \sum_{i=1}^M \mathbf{w}_{j,i}^{\mathrm{t}} \mathbf{m}_i^{\mathrm{t}}, \qquad \tilde{\mathbf{q}}_j^{\mathrm{f}} = \sum_{i=1}^M \mathbf{w}_{j,i}^{\mathrm{f}} \mathbf{m}_i^{\mathrm{f}}. \tag{10}$$

The resulting representations provide complementary temporal and frequency-domain guidance for the denoising backbone.

3.3 Training Objective

The model is trained end-to-end using a composite objective consisting of a diffusion reconstruction loss and an auxiliary entropy regularization. The diffusion loss drives the generative imputation process, while the entropy term encourages sparse memory retrieval.

Table 2: RMSE on four multivariate datasets (lower is better). **Bold** and underlined denote the best and second-best results. (**NonStat.**: Non-stationary Transformer, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB).

Dataset	Type	Setting	NonStat.	FreTS	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
Stock	SEQ	10	0.136	0.225	0.131	0.448	0.248	2.488	2.364	0.889	0.165	2.223	1.424	**0.040**
		20	0.161	0.250	0.159	0.603	0.259	2.568	2.343	0.999	0.155	2.251	0.934	**0.042**
		30	0.193	0.308	0.345	0.366	0.276	2.452	2.394	0.997	0.176	2.221	2.093	**0.043**
	BLOCK	10%	0.116	0.215	0.129	0.560	0.259	2.463	2.627	0.863	0.183	2.208	0.871	**0.074**
		50%	0.154	0.209	0.150	0.970	0.252	2.548	2.748	1.636	0.165	2.345	2.111	**0.107**
		90%	0.183	0.231	0.175	1.285	0.437	2.494	2.809	2.617	0.180	2.391	2.304	**0.113**
Electricity	SEQ	10	0.454	0.529	0.435	0.473	0.499	0.494	0.456	0.714	0.427	0.516	0.437	**0.409**
		20	0.475	0.575	0.454	0.494	0.546	0.558	0.471	0.784	0.453	0.526	0.448	**0.406**
		30	0.557	0.731	0.548	0.614	0.566	0.594	0.500	0.655	0.522	0.549	0.469	**0.426**
	BLOCK	10%	0.467	0.579	0.451	0.533	0.460	0.673	0.512	0.693	0.446	0.525	0.495	**0.437**
		50%	0.520	0.704	0.495	0.628	0.580	0.669	0.530	0.747	0.519	0.541	0.541	**0.492**
		90%	0.569	0.807	0.543	0.705	0.800	0.882	0.565	0.739	0.638	0.575	0.632	**0.513**
PEMS-BAY	SEQ	10	0.472	0.427	0.455	0.419	0.393	0.561	0.477	0.463	0.444	0.461	0.622	**0.378**
		20	0.555	0.575	0.541	0.536	0.582	0.614	0.515	0.694	0.523	0.477	0.680	**0.379**
		30	0.662	0.715	0.653	0.626	0.698	0.700	0.534	0.846	0.751	0.496	0.634	**0.379**
	BLOCK	10%	0.434	0.365	0.409	0.379	**0.325**	0.567	0.387	0.357	0.362	0.379	0.390	0.371
		50%	0.458	0.414	0.448	0.445	0.436	0.794	0.420	0.476	0.417	0.435	0.455	**0.408**
		90%	0.488	0.751	0.480	0.486	0.474	0.821	0.452	0.534	0.448	0.474	0.510	**0.437**
METR-LA	SEQ	10	0.429	0.412	0.426	0.395	0.393	0.454	0.385	0.396	0.418	0.424	0.395	**0.377**
		20	0.474	0.460	0.475	0.442	0.454	0.481	0.407	0.514	0.464	0.466	0.448	**0.390**
		30	0.516	0.508	0.512	0.475	0.506	0.532	0.429	0.569	0.552	0.468	0.472	**0.394**
	BLOCK	10%	0.435	0.395	0.422	0.404	0.371	0.697	0.355	**0.351**	0.390	0.414	0.395	0.397
		50%	0.446	0.408	0.431	0.440	0.443	0.595	0.403	0.470	0.404	0.446	0.423	**0.402**
		90%	0.451	0.524	0.448	0.488	0.439	0.669	0.478	0.535	0.428	0.506	0.449	**0.426**

Diffusion reconstruction loss. The primary objective is to train the denoising network $\boldsymbol{\epsilon}_\theta$ to predict the noise injected during the forward diffusion process. Following the simplified DDPM objective, the loss is defined as

$$\mathcal{L}_{\text{denoising}} = \mathbb{E}_{\mathbf{x}_0^{\text{ob}},\boldsymbol{\epsilon},t}\left[\left\|(1-\mathbf{M}) \odot (\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}^t, \tilde{\mathbf{q}}^f))\right\|^2\right]. \tag{11}$$

Here, $\mathbf{M}$ denotes the binary observation mask, $\mathbf{x}_t^{\text{ta}}$ denotes the noised target derived from $\mathbf{x}_0^{\text{ta}}$. The noise predictor $\boldsymbol{\epsilon}_\theta$ is conditioned on the observed data $\mathbf{x}_0^{\text{ob}}$ and the time- and frequency-domain prototypes $\tilde{\mathbf{q}}^{\text{t}}$ and $\tilde{\mathbf{q}}^{\text{f}}$.

Memory entropy loss. Memory retrieval produces attention weight distributions for both time and frequency domains. For a sample s, let $\mathbf{w}_j^{u,s} \in \mathbb{R}^M$ denote the attention weights at step j for domain $u \in \{\text{t}, \text{f}\}$. To encourage sparse retrieval, we minimize the entropy of these distributions:

$$\mathcal{L}_{\text{entropy}} = \frac{1}{2SN} \sum_{s=1}^{N} \sum_{u\in\{\text{t,f}\}} \sum_{j=1}^{L} \sum_{i=1}^{M} -w_{j,i}^{u,s} \log(w_{j,i}^{u,s}). \tag{12}$$

Finally, the whole training objective is given by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoising}} + \gamma \cdot \mathcal{L}_{\text{entropy}}, \tag{13}$$

where γ balances reconstruction accuracy and retrieval sparsity. All model parameters are jointly optimized by minimizing $\mathcal{L}_{\text{total}}$.

The complete algorithmic procedures, including memory initialization, training, and inference, are provided in Appendix E.

4 Experiments

Datasets. We comprehensively evaluate TFMemDiff on eight widely used real-world time series datasets, covering both multivariate and univariate settings. Specifically, the multivariate benchmarks include `Stock`[1], `Weather` (Wu et al., 2021), `ILI`, `Electricity` (Trindade, 2015), `PEMS-BAY`,

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

Table 3: Ablation results of TFMemDiff across different data scenarios. We evaluate the univariate and multivariate settings using the ETTh1 (OT) and Stock datasets, respectively.

Method	Univariate				Multivariate					
	POINT		**SEQ**		**POINT**		**SEQ**		**BLOCK**	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
w/o Memory	0.125	0.098	0.159	0.108	0.239	0.170	0.217	0.155	0.239	0.174
w/o TMM	0.076	0.058	0.102	0.080	0.092	0.054	0.054	0.043	0.206	0.165
w/o FMM	0.075	0.058	0.103	0.091	0.079	0.053	0.052	0.032	0.189	0.128
w/o Sparse Attention	0.068	0.047	0.099	0.078	0.082	0.043	0.048	0.023	0.113	0.067
w/o Memory update	0.067	0.044	0.100	0.079	0.083	0.044	0.045	0.024	0.112	0.068
TFMemDiff	**0.065**	**0.041**	**0.092**	**0.070**	**0.077**	**0.042**	**0.042**	**0.022**	**0.108**	**0.062**

and `METR-LA` (Li et al., 2018), which exhibit complex temporal dependencies, spatial-temporal patterns, and cross-variable correlations. The univariate datasets consist of `ETTh1` (Zhou et al., 2021), and `Metro`, representing diverse sampling frequencies and dataset scales. This diverse collection enables a thorough evaluation of the proposed model under different data characteristics.

Baselines. We compare TFMemDiff against 11 representative and competitive methods, covering a broad spectrum of modeling paradigms. These include: i) advanced *Transformer*-based models designed for long-range temporal modeling, such as iTransformer (Liu et al., 2024), Non-stationary Transformer (Liu et al., 2022), and SAITS (Du et al., 2023); ii) modern deep architectures that emphasize multi-scale or frequency-aware *representation learning*, including TimesNet (Wu et al., 2023), ModernTCN (Luo & Wang, 2024), FreTS (Yi et al., 2023), and Glocal-IB (Yang et al., 2025); and iii) popular *generative* models explicitly designed for time series imputation, including CSDI (Tashiro et al., 2021), SSSD (Alcaraz & Strodthoff, 2023), FGTI (Yang et al., 2024), and FSDI (Shen et al., 2026). This diverse set of baselines ensures a comprehensive and fair comparison.

Metrics. Following common practice in time series imputation (Yang et al., 2024), we report root mean squared error (RMSE), mean absolute error (MAE), and continuous ranked probability score (CRPS), computed exclusively on masked entries. RMSE and MAE evaluate point-wise reconstruction accuracy, while CRPS assesses the quality of the imputation distribution.

Experimental details, including dataset statistics, masking protocols, baselines, and implementation details, are provided in Appendix F to ensure full transparency and reproducibility.

4.1 Overall Performance Comparison

We evaluate TFMemDiff against 11 strong baselines on four real-world multivariate datasets: Stock, Electricity, PEMS-BAY, and METR-LA. Following the masking protocol in Section F.2, we consider two challenging structured missing settings: *Subsequence Masking* with segment lengths of 10, 20, and 30, and *Block Masking* with missing ratios of 10%, 50%, and 90%. These settings enable a controlled evaluation of imputation performance under increasing structured sparsity.

Results. Table 2 summarizes the results on four multivariate datasets, averaged over three independent runs. Overall, TFMemDiff achieves strong performance across a wide range of masking scenarios. For instance, under 90% BLOCK masking on Stock, TFMemDiff achieves an RMSE of 0.113, compared with over 2.30 for FSDI. Moreover, TFMemDiff demonstrates strong adaptability to different data characteristics. On the complex traffic dataset PEMS-BAY under SEQ 30 masking, it outperforms the second-best baseline (Glocal-IB) by reducing the RMSE from 0.496 to 0.379, corresponding to an improvement of over 23%. This demonstrates that TFMemDiff is capable of robustly recovering complex temporal patterns across diverse datasets, even under severe missing conditions. Comprehensive experimental results, including additional RMSE and MAE comparisons, CRPS evaluation, are provided in Appendix G.

4.2 Ablation and Memory Mechanism Analysis

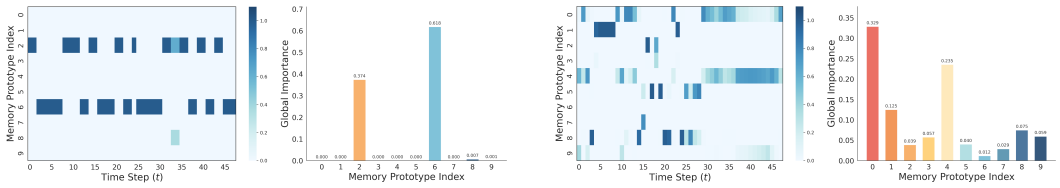
4.2.1 Ablation Study

We first perform an ablation study on the ETTh1 (OT) and Stock datasets to quantitatively assess the contribution of each component. The following model variants are considered: i) *w/o Memory*, where both memory modules are removed; ii)*w/o TMM*, where the time-domain memory is disabled; and iii)

w/o FMM, where the frequency-domain memory is disabled. iv) *w/o Sparse Attention*, which replaces the sparse memory update mechanism with standard full attention. v) *w/o Memory Update*, where the initialized memory prototypes are kept fixed during training while memory retrieval remains unchanged. The full TFMemDiff model serves as the reference.

As summarized in Table 3, removing both memory components causes the largest performance degradation, confirming the importance of memory guidance. Removing either the TMM or FMM also degrades performance, indicating their complementary roles. Replacing sparse memory interaction with full attention consistently increases the error, while freezing the memory prototypes performs worse than the full model, demonstrating the benefit of sparse and iterative memory updates.

Overall, these results demonstrate the complementary contributions of the time- and frequency-domain memories, as well as the effectiveness of the proposed memory update mechanism. The FMM provides global frequency-domain guidance, the TMM provides adaptive temporal guidance, and the sparse, iterative update strategy further improves the utilization of historical memory across diverse missing patterns.



(a) Frequency-domain Attention

(b) Time-domain Attention

Figure 2: Visualization of Memory Attention. The left panels illustrate the temporal evolution of attention weights, while the right panels report the aggregated importance of each memory prototype.

4.2.2 Visualization of Memory Attentions.

To further examine the dual-domain memory mechanism, we visualize the memory retrieval patterns on ETTh1 under subsequence masking with a length of 10, as shown in Figure 2. The visualization includes attention heatmaps over time and the aggregated importance of memory prototypes.

The frequency-domain memory exhibits sparse, stable retrieval patterns, with a few prototypes dominating across temporal positions. This indicates the FMM effectively captures global periodic characteristics, providing a consistent reference regardless of local missingness. Conversely, the time-domain memory demonstrates dynamic, context-dependent behavior. Its attention heatmaps reveal clear prototype transitions over time, enabling the TMM to adaptively retrieve relevant local patterns and refine fine-grained structures.

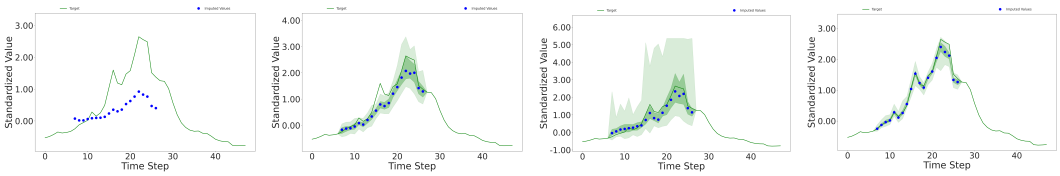
Table 4: Generality evaluation of the memory module.

Dataset	Setting	Metric	SSSD	+Memory
ILI	POINT 90%	RMSE	1.774	**1.201**
		MAE	1.070	**0.745**
	SEQ 30	RMSE	1.674	**1.660**
		MAE	0.932	**0.903**
	BLOCK 90%	RMSE	1.863	**1.481**
		MAE	1.079	**1.070**
Stock	POINT 90%	RMSE	2.188	**0.251**
		MAE	1.708	**0.181**
	SEQ 30	RMSE	2.452	**0.228**
		MAE	1.953	**0.163**
	BLOCK 90%	RMSE	2.494	**0.250**
		MAE	2.120	**0.179**

Taken together, these complementary retrieval patterns provide clear evidence that the proposed dual-domain memory design enhances both the interpretability and effectiveness of TFMemDiff, enabling it to simultaneously capture global regularities and local temporal variations.

4.3 Qualitative Imputation Results

We provide qualitative comparisons to assess the imputation quality of TFMemDiff under structured missing scenarios. Specifically, we visualize the ILI dataset under subsequence missingness of length 20, as shown in Figure 3. Additional qualitative results are provided in Appendix G.4.



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

Figure 3: Visual comparison between our proposed TFMemDiff framework and competitive baselines on the ILI dataset under the subsequence missing scenario (length=20) with a 10% missing rate.

TFMemDiff more closely follows the ground-truth trajectory within the missing region and better preserves the temporal pattern. These qualitative observations are consistent with the quantitative results, further supporting the effectiveness of TFMemDiff under structured missingness.

4.4 Generality of the Memory Mechanism

To assess whether the proposed memory mechanism depends on a specific denoising backbone, we replace the default Transformer with SSSD (Alcaraz & Strodthoff, 2023), an SSM-based diffusion architecture, and integrate the same dual-domain memory module (denoted as SSSD+Memory). All training objectives and masking protocols are kept identical for a fair comparison.

As shown in Table 4, across the evaluated datasets and extreme missing scenarios, SSSD+Memory consistently outperforms the vanilla SSSD baseline.

These empirical results indicate that our proposed memory retrieval mechanism is not tightly coupled to a specific backbone design. The memory module acts as a plug-and-play source of structural guidance by introducing retrievable historical information into the generative process, and can be effectively integrated into at least one alternative diffusion architecture with consistent improvements.

4.5 Computational Efficiency

Figure 4 compares the training and inference time of TFMemDiff with several diffusion-based baselines. TFMemDiff achieves the lowest runtime in both stages, requiring only 0.23 s/epoch for training and 30.87 s/batch for inference.

Despite introducing additional memory modules, TFMemDiff does not sacrifice computational efficiency. The memory operations introduce only moderate additional model complexity, while the overall training cost remains low. Together, these results show that TFMemDiff achieves a favorable balance between memory-enhanced modeling and computational efficiency. A more comprehensive comparison of model parameters, FLOPs, memory consumption, and runtime is provided in Appendix H.

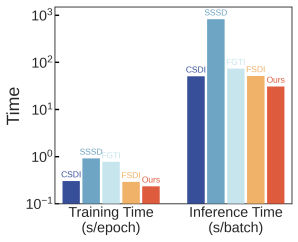


Figure 4: Efficiency comparison among representative diffusion baselines.

5 Conclusion

We propose TFMemDiff, a diffusion-based time series imputation framework that introduces historical global memory as an additional conditioning source. TFMemDiff retrieves complementary temporal and frequency-domain prototypes to provide richer structural guidance beyond sparse local observations. Extensive experiments on real-world datasets demonstrate the effectiveness of TFMemDiff across diverse missingness settings, particularly under severe and structured missingness. Ablation studies further verify that the two memory components play complementary roles: the time-domain memory captures evolving temporal patterns, while the frequency-domain memory preserves global periodic structures. Beyond time series imputation, the proposed memory-guided diffusion framework offers a general modeling paradigm for learning from irregular and sparse sequential data, suggesting promising directions for future research.

AI USE STATEMENT

In this work, we used generative AI tools solely for limited editorial assistance, including grammar correction, wording refinement, and readability improvement. We have not used generative AI tools for designing the research method, conducting experiments, analyzing results, generating scientific claims, or drawing conclusions. Furthermore, no AI tools were used to generate any figures, code, or technical artifacts. We have thoroughly reviewed all AI-assisted work. Specifically, we manually verified all AI-polished text to ensure it strictly aligns with our original scientific intent, introduces no fabricated information, and preserves the technical integrity of the manuscript. We take full responsibility for the final content of this work, including text, claims, and all artifacts presented in this submission.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our empirical results, comprehensive algorithmic details are provided in Appendix E. Detailed descriptions of the experimental setup, including the real-world datasets, the construction protocols for all missingness scenarios (Point, Subsequence, and Block masking), and the complete hyperparameter settings, are documented in Appendix F. The source code is also available anonymously for review at `https://anonymous.4open.science/r/TFMemDiff-33B3/`.

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.

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

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.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*, pp. 1705–1714, 2019.

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.

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

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, 2023.

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, 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.

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.

Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, 2020.

Jack Rae, Jonathan J Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Timothy Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. *NeurIPS*, 29, 2016.

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.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pp. 2256–2265, 2015.

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.

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.

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.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *ICLR*, 2015.

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.

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.

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 Work

Time series imputation. Time series imputation has evolved from classical statistical techniques to modern deep generative frameworks. Early statistical methods mainly exploit local correlations or global matrix structures. Simple strategies such as mean imputation and k-nearest neighbors (kNN) ignore temporal dependencies, while more advanced models, including autoregressive methods (e.g., ARIMA (Box et al., 2015)) and matrix factorization (Yu et al., 2016), attempt to model linear temporal dependencies and latent factors. However, their strong assumptions of linearity and stationarity limit their effectiveness on complex, non-linear, and multivariate time series.

To overcome these limitations, **deterministic deep learning models** have been widely adopted to learn non-linear mappings from observed values to missing entries. Early approaches based on recurrent neural networks, such as GRU-D (Che et al., 2018) and BRITS (Cao et al., 2018), explicitly handle irregular sampling through time-aware mechanisms. More recently, Transformer-based architectures (e.g., SAITS (Du et al., 2023), iTransformer (Liu et al., 2024)) leverage self-attention to capture long-range dependencies. Furthermore, to mitigate the risk of overfitting to local noise under severe missingness, advanced representation learning paradigms have been proposed. For example, Glocal-IB (Yang et al., 2025) introduces a Glocal Information Bottleneck framework that explicitly aligns global semantic structures with local details, yielding highly robust latent representations. Despite their effectiveness, these methods produce deterministic point estimates by optimizing pixel-wise objectives such as mean squared error, which often leads to over-smoothed imputations and fails to reflect the inherent uncertainty of real-world time series.

Recently, **generative probabilistic models** have emerged as a promising alternative by explicitly modeling the data distribution. Early attempts using variational autoencoders (VAEs) (Fortuin et al., 2020) and generative adversarial networks (GANs) (Luo et al., 2018) demonstrated the benefits of probabilistic imputation but often suffer from training instability or mode collapse. More recently, conditional score-based diffusion models, such as CSDI (Tashiro et al., 2021) and SSSD (Alcaraz & Strodthoff, 2023), have shown strong performance in time series imputation by modeling the data distribution and generating samples through conditional denoising. Building upon this paradigm, recent approaches further strengthen the guidance signal by incorporating frequency-domain information. For instance, FGTI (Yang et al., 2024) leverages dominant and high-frequency components, while LSCD (Fons et al., 2025) utilizes the Lomb-Scargle periodogram to capture spectral dependencies, both effectively steering the diffusion process.

Nevertheless, existing diffusion-based time series imputation methods rely heavily on the observed context of the current incomplete sequence for conditional guidance. Under extreme sparsity, such observations may provide insufficient structural information for recovering severely missing regions. Our work addresses this limitation by introducing memory-based guidance derived from historical training data. Unlike prior methods that primarily depend on sparse instance-level observations, TFMemDiff retrieves dataset-level prototypical patterns from external memory banks as an additional conditioning source. By organizing such historical knowledge in both the time and frequency domains, the proposed dual-domain memory provides complementary structural guidance and helps preserve temporal regularities under severe missingness.

External memory-augmented modeling. Memory networks introduce an explicit storage mechanism to capture long-term dependencies and recurring patterns (Weston et al., 2015). In time series forecasting, memory-augmented models retrieve relevant historical patterns from stored representations, improving long-horizon prediction accuracy. In anomaly detection, memory modules are commonly used to store prototypes of normal behavior, where deviations are identified by reconstruction failure (Gong et al., 2019; Park et al., 2020).

Despite their success in related tasks, memory mechanisms remain underexplored in generative time-series imputation. In this work, TFMemDiff introduces a dual-domain memory mechanism that retrieves temporal and frequency-domain prototypes from historical training data as explicit global conditioning, complementing the sparse local observations available to the diffusion model under severe missingness.

B PRELIMINARIES

B.1 NOTATIONS

Definition 1. Time Series. Let $\mathbf{X} \in \mathbb{R}^{L \times D}$ denote a time series sequence of length L with D channels. It can be represented as a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L]^\top$, where $\mathbf{x}_l \in \mathbb{R}^D$ is the vector at time step l. We refer to $\mathbf{X}$ as a *univariate* time series when $D = 1$, and *multivariate* when $D > 1$.

Definition 2. Incomplete Time Series. Given a time series where specific entries $x_{l,d}$ may be missing, we define a binary mask matrix $\mathbf{M} \in \{0, 1\}^{L \times D}$. The entry $M_{l,d} = 1$ indicates that $x_{l,d}$ is observed, while $M_{l,d} = 0$ indicates it is missing. Accordingly, we denote the observed part as $\mathbf{X}^{\mathrm{ob}}$ and the missing part as $\mathbf{X}^{\mathrm{ta}}$, such that $\mathbf{X} = \mathbf{X}^{\mathrm{ob}} \cup \mathbf{X}^{\mathrm{ta}}$. Here, the incomplete time series $\mathbf{X}^{\mathrm{ob}}$ is considered a realization of the observed random process $\mathcal{X}^{\mathrm{ob}}$.

B.2 DENOISING DIFFUSION PROBABILISTIC MODELS

Denoising Diffusion Probabilistic Models (DDPMs) are a powerful class of deep generative models that consist of a predefined forward diffusion process and a learned reverse denoising process (Ho et al., 2020; Sohl-Dickstein et al., 2015).

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:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{14}$$

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:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{15}$$

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 (typically a U-Net or Transformer):

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_\theta^2(\mathbf{x}_t, t)\mathbf{I}). \tag{16}$$

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:

$$\mathcal{L}(\theta) := \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right], \tag{17}$$

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 CONDITIONAL IMPUTATION BACKBONE IN TFMEMDIFF

We employ a conditional diffusion model as the underlying generative framework, following the paradigm of DDPMs (Ho et al., 2020; Sohl-Dickstein et al., 2015), consisting of a fixed forward noising process and a learnable reverse denoising process. Unlike standard implementations, our denoising network $\epsilon_\theta(\cdot)$ is meticulously designed with dual-domain memory augmentation to capture complex temporal dynamics. The detailed architecture and component interactions of this network are illustrated in Figure 6.

Forward Diffusion in Incomplete Data. For completeness, we explicitly detail the forward process adopted in our framework. Let $\mathbf{x}_0^{\text{ta}}$ denote the clean imputation target and β_t be a fixed variance schedule. The forward process is defined as a fixed Markov chain that gradually adds Gaussian noise to the data. A key property of this formulation is that it allows sampling $\mathbf{x}_t^{\text{ta}}$ at an arbitrary timestep t directly from $\mathbf{x}_0^{\text{ta}}$. Specifically, given a noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the closed-form transition kernel is expressed as (see D.1 for the detailed derivation):

$$\mathbf{x}_t^{\text{ta}} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{ta}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{18}$$

where $\bar{\alpha}_t$ denotes the cumulative noise schedule parameter. This noised target, representing the data distribution after t steps of diffusion, serves as a key input for the subsequent denoising network.

Reverse Denoising for Imputation. Here, we provide the detailed mathematical formulation of the conditional reverse process. At each step t, the model takes the noised target $\mathbf{x}_t^{\text{ta}}$ as input and leverages strong conditional information to guide the transition to $\mathbf{x}_{t-1}^{\text{ta}}$. Crucially, in addition to the observed condition $\mathbf{x}_0^{\text{ob}}$, this process is guided by the dual-domain priors retrieved from our memory module: the time-domain prototype $\tilde{\mathbf{q}}^t$ and the frequency-domain prototype $\tilde{\mathbf{q}}^f$.

For notational brevity, we denote the collection of these memory priors as $\tilde{\mathbf{q}} = \{\tilde{\mathbf{q}}^t, \tilde{\mathbf{q}}^f\}$. Consequently, the reverse Markov chain, parameterized by a neural network θ, is formalized as:

$$p_\theta(\mathbf{x}_{0:T}^{\text{ta}}) := p(\mathbf{x}_T^{\text{ta}}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}), \quad \mathbf{x}_T^{\text{ta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{19}$$

where the transition kernel at each step is parameterized as a Gaussian distribution:

$$p_\theta(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}) := \mathcal{N}\left(\mathbf{x}_{t-1}^{\text{ta}}; \boldsymbol{\mu}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}), \sigma_t^2 \mathbf{I}\right). \tag{20}$$

Formally, the conditioning information for the reverse process is defined as a tuple $(\mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})$, which incorporates both the observed data $\mathbf{x}_0^{\text{ob}}$ and the prototype patterns $\tilde{\mathbf{q}}$ retrieved from the memory module. This provides rich and complementary guidance for the denoising process. Following the parameterization of DDPMs (Ho et al., 2020) (see derivation in D.2), we define the mean $\boldsymbol{\mu}_\theta$ and variance σ_t^2 of the transition distribution as:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t^{\text{ta}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}) \right), \tag{21}$$

where the variance is set to a non-learnable, predefined constant $\tilde{\beta}_t$ such that $\sigma_t^2 = \tilde{\beta}_t$. The constant is given by:

$$\tilde{\beta}_t = \begin{cases} \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, & \text{if } t > 1, \\ \beta_1, & \text{if } t = 1. \end{cases} \tag{22}$$

During inference, rather than performing ancestral sampling over the full DDPM trajectory, we employ DPM-Solver++ with 20 sampling steps to accelerate the reverse denoising process.

D MATHEMATICAL DETAILS OF TFMEMDIFF

D.1 DERIVATION OF THE CLOSED-FORM FORWARD PROCESS

The forward diffusion process is defined as a fixed Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$. For a single step, the transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is given by:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \tag{23}$$

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Using the reparameterization trick, we can express $\mathbf{x}_t$ as:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}, \quad \text{where } \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{24}$$

To derive the closed-form expression for arbitrary timestep t, we can recursively expand the equation. Starting by substituting $\mathbf{x}_{t-1}$ with its dependence on $\mathbf{x}_{t-2}$:

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \underbrace{\sqrt{\alpha_t(1-\alpha_{t-1})}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}}_{\text{merged noise term}}.\end{aligned} \tag{25}$$

Since the noise terms $\boldsymbol{\epsilon}_{t-2}$ and $\boldsymbol{\epsilon}_{t-1}$ are independent Gaussian variables $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, their linear combination is also a Gaussian distribution. The variance of the merged noise term is:

$$\text{Var} = (\sqrt{\alpha_t(1-\alpha_{t-1})})^2 + (\sqrt{1-\alpha_t})^2 = \alpha_t(1-\alpha_{t-1}) + 1 - \alpha_t = 1 - \alpha_t\alpha_{t-1}. \tag{26}$$

Thus, we can merge the two noise terms into a single variable $\bar{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with a scaling factor of $\sqrt{1-\alpha_t\alpha_{t-1}}$:

$$\mathbf{x}_t = \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}. \tag{27}$$

By repeating this recursion back to $t = 0$, we obtain the final closed-form expression:

$$\mathbf{x}_t = \sqrt{\prod_{i=1}^{t}\alpha_i}\mathbf{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{28}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

D.2 DERIVATION OF THE CONDITIONAL REVERSE PROCESS

In this section, we provide the detailed derivation for the mean parameterization of the conditional reverse process. Our goal is to derive the transition kernel $p_\theta(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})$ used in the main paper.

D.2.1 POSTERIOR DISTRIBUTION OF THE TRANSITION

Although the reverse transition $q(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}})$ is intractable, the posterior conditioned on the starting point $\mathbf{x}_0^{\text{ta}}$ is tractable and follows a Gaussian distribution. Incorporating our specific conditions (observed values $\mathbf{x}_0^{\text{ob}}$ and memory priors $\tilde{\mathbf{q}}$), we apply Bayes' theorem:

$$q(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}) = \frac{q(\mathbf{x}_t^{\text{ta}} \mid \mathbf{x}_{t-1}^{\text{ta}}, \mathbf{x}_0^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})q(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_0^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})}{q(\mathbf{x}_t^{\text{ta}} \mid \mathbf{x}_0^{\text{ta}}, \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})}. \tag{29}$$

Since the forward diffusion process is a fixed Markov chain determined solely by the noise schedule β_t, the conditional terms $\mathbf{x}_0^{\text{ob}}$ and $\tilde{\mathbf{q}}$ do not alter the Gaussian structure of the diffusion kernels. We expand the probability density functions as follows:

$$\begin{aligned}q(\mathbf{x}_{t-1}^{\text{ta}} \mid \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ta}}) &\propto \exp\left(-\frac{1}{2}\left[\frac{(\mathbf{x}_t^{\text{ta}} - \sqrt{\alpha_t}\mathbf{x}_{t-1}^{\text{ta}})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1}^{\text{ta}} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0^{\text{ta}})^2}{1-\bar{\alpha}_{t-1}}\right]\right) \\ &= \exp\left(-\frac{1}{2}\left[\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)(\mathbf{x}_{t-1}^{\text{ta}})^2 - 2\left(\frac{\sqrt{\alpha_t}\mathbf{x}_t^{\text{ta}}}{\beta_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0^{\text{ta}}}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^{\text{ta}} + C\right]\right),\end{aligned} \tag{30}$$

where C denotes terms independent of $\mathbf{x}_{t-1}^{\text{ta}}$. By matching this expression with the standard Gaussian density function $\mathcal{N}(\mathbf{x}_{t-1}^{\text{ta}}; \tilde{\boldsymbol{\mu}}_t, \tilde{\beta}_t\mathbf{I})$, we can identify the posterior mean $\tilde{\boldsymbol{\mu}}_t$:

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{ta}}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t^{\text{ta}} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0^{\text{ta}}. \tag{31}$$

D.2.2 Parameterization via Noise Prediction

Equation (31) depends on the ground truth $\mathbf{x}_0^{\text{ta}}$, which is unknown during the reverse sampling process. To resolve this, we utilize the property of the closed-form forward process derived in the previous section:

$$\mathbf{x}_t^{\text{ta}} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{ta}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}. \tag{32}$$

Re-arranging this term allows us to express $\mathbf{x}_0^{\text{ta}}$ in terms of $\mathbf{x}_t^{\text{ta}}$ and the noise $\boldsymbol{\epsilon}$:

$$\mathbf{x}_0^{\text{ta}} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t^{\text{ta}} - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right). \tag{33}$$

Substituting Equation (33) back into Equation (31), we simplify the expression for the mean:

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t^{\text{ta}} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\left(\frac{\mathbf{x}_t^{\text{ta}} - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right) \\
&= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{\beta_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\right)\mathbf{x}_t^{\text{ta}} - \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon} \\
&= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t^{\text{ta}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right).
\end{aligned} \tag{34}$$

Finally, we approximate the true noise $\boldsymbol{\epsilon}$ using our proposed network $\boldsymbol{\epsilon}_\theta$, which is conditioned on the observed data $\mathbf{x}_0^{\text{ob}}$ and the retrieved memory priors $\tilde{\mathbf{q}}$. This yields the final mean parameterization presented in Eq. (4) of the main text:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t^{\text{ta}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}})\right). \tag{35}$$

E ALGORITHMIC DETAILS

Algorithm 1 Dual-Domain Prototype Initialization

Input: Training set $\mathcal{X}$, pre-trained encoder f_e, number of prototypes M, ratio $\alpha = 0.1$
Output: Time Memory Bank $\mathbf{M}^t$, Freq Memory Bank $\mathbf{M}^f$
Initialize latent set $\mathcal{Q} \leftarrow \varnothing$
for each $x \in \text{RandomSample}(\mathcal{X}, \alpha)$ **do**
  $\tilde{x} \leftarrow \text{LinearInterp}(\text{Mask}(x))$
  $\mathbf{q} \leftarrow f_e(\tilde{x})$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathbf{q}\}$
end for
$\{c_1, \ldots, c_M\} \leftarrow \text{KMeans}(\mathcal{Q}, M)$
$\mathbf{M}^t \leftarrow \{m_i^t = c_i\}_{i=1}^M$
$\mathcal{Q}^f \leftarrow \{\phi_f(\mathbf{q}) \mid \mathbf{q} \in \mathcal{Q}\}$
$\{s_1, \ldots, s_M\} \leftarrow \text{KMeans}(\mathcal{Q}^f, M)$
$\mathbf{M}^f \leftarrow \{m_i^f = s_i\}_{i=1}^M$
return $\mathbf{M}^t, \mathbf{M}^f$

This section provides pseudocode for TFMemDiff. Algorithm 1 describes dual-domain memory initialization, Algorithm 2 summarizes the training procedure, and Algorithm 3 presents the imputation process.

Algorithm 2 Training procedure of TFMemDiff

Input: Complete dataset $\mathcal{X}$, mask $\mathbf{M}$, diffusion steps T, hyperparameter γ, first-stage trained encoder f_e
Output: Optimized network $\boldsymbol{\epsilon}_\theta$, updated memory banks $\mathcal{M}^t, \mathcal{M}^f$
repeat
  $\mathbf{x}_0 \sim \mathcal{X}$
  $\mathbf{x}_0^{\text{ob}} \leftarrow \mathbf{x}_0 \odot \mathbf{M}, \quad \mathbf{x}_0^{\text{ta}} \leftarrow \mathbf{x}_0 \odot (1 - \mathbf{M})$
  Extract latent queries $\mathbf{q} \leftarrow f_e(\mathbf{x}_0^{\text{ob}})$
  Update and retrieve time-domain memory: $\tilde{\mathbf{q}}^t, \mathbf{W}^t \leftarrow \text{TMM}(\mathbf{q}, \mathbf{M})$
  Update and retrieve frequency-domain memory: $\tilde{\mathbf{q}}^f, \mathbf{W}^f \leftarrow \text{FMM}(\mathbf{q}, \mathbf{M})$
  $t \sim \text{Uniform}\{1, \ldots, T\}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t^{\text{ta}} \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{ta}} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
  Compute denoising loss: $\mathcal{L}_{\text{denoising}} = \left\|(\mathbf{1} - \mathbf{M}) \odot \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t^{\text{ta}}, t \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}^t, \tilde{\mathbf{q}}^f\right)\right)\right\|^2$
  Compute entropy loss $\mathcal{L}_{\text{entropy}}$ using $\mathbf{W}^t, \mathbf{W}^f$ via Eq. (12)
  Perform Gradient Descent on $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoising}} + \gamma \cdot \mathcal{L}_{\text{entropy}}$
until converged

Algorithm 3 Imputation procedure of TFMemDiff

Input: Observed data $\mathbf{x}_0^{\text{ob}}$, mask $\mathbf{M}$, trained network $\boldsymbol{\epsilon}_\theta$, trained encoder f_e, number of solver steps S
Output: Imputed series $\hat{\mathbf{x}}_0$
Extract latent queries: $\mathbf{q} \leftarrow f_e(\mathbf{x}_0^{\text{ob}})$
Retrieve time-domain memory guidance: $\tilde{\mathbf{q}}^t \leftarrow \text{TMM}(\mathbf{q}, \mathbf{M})$
Retrieve frequency-domain memory guidance: $\tilde{\mathbf{q}}^f \leftarrow \text{FMM}(\mathbf{q}, \mathbf{M})$
$\mathbf{x}_{t_S}^{\text{ta}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $s = S$ **to** 1 **do**
  $\hat{\boldsymbol{\epsilon}} \leftarrow \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_{t_s}^{\text{ta}}, t_s \mid \mathbf{x}_0^{\text{ob}}, \tilde{\mathbf{q}}^t, \tilde{\mathbf{q}}^f\right)$
  $\mathbf{x}_{t_{s-1}}^{\text{ta}} \leftarrow \text{DPM-Solver} + + \left(\mathbf{x}_{t_s}^{\text{ta}}, \hat{\boldsymbol{\epsilon}}, t_s\right)$
end for
Combine: $\hat{\mathbf{x}}_0 \leftarrow \mathbf{x}_0^{\text{ta}} \odot (1 - \mathbf{M}) + \mathbf{x}_0^{\text{ob}} \odot \mathbf{M}$

F EXPERIMENTAL DETAILS

F.1 DATASETS

In this section, we provide detailed descriptions of the real-world datasets utilized in our experiments. Spanning diverse domains, these datasets are categorized into multivariate and univariate groups as follows:

Multivariate Datasets. We use six multivariate datasets covering finance, weather, health, energy, and transportation domains. Stock contains daily stock prices from multiple corporations; Weather includes 21 meteorological variables recorded every 10 minutes (Wu et al., 2021); and ILI records weekly influenza-like illness ratios reported by the U.S. CDC (Wu et al., 2021). Electricity contains hourly electricity consumption records, while PEMS-BAY and METR-LA contain traffic speed measurements from sensor networks (Trindade, 2015; Li et al., 2018). For computational efficiency and consistent dimensionality, we use the first 32 clients/sensors for Electricity, PEMS-BAY, and METR-LA, with the same selected variables used for all compared methods.

Univariate Datasets. To verify the robustness of our model in reconstructing univariate time series without cross-variable cues, we employ two additional datasets. The ETTh1 dataset records hourly parameters from electricity transformers (Zhou et al., 2021); following standard protocols, we specifically select the Oil Temperature (OT) column to evaluate the model's robustness in recovering data from complex missing patterns. The Metro dataset records hourly traffic data on Interstate 94 Westbound, where we focus on imputing the traffic_volume feature to recover incomplete transportation profiles. These diverse subsets are used to assess the model's generalization ability in restoring missing values across widely varying sampling frequencies and data distributions.

The comprehensive statistics for all the aforementioned datasets are summarized in Table 5. We empirically set the sequence length to 48 for all experiments, which strikes a balance between capturing sufficient temporal dependencies and maintaining computational efficiency.

Table 5: 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
Weather	21	48	(6953, 1001, 2001)	POINT, SEQ, BLOCK	10 min	Weather
Electricity	32		(6953, 1001, 2001)		Hourly	Energy
PEMS-BAY	32		(6953, 1001, 2001)		5 min	Transportation
METR-LA	32		(6953, 1001, 2001)		5 min	Transportation
ILI	7		(629, 98, 194)		Weekly	Health
Stock	6		(2532, 370, 738)		Daily	Finance
ETTh1(OT)	1		(8593, 2881, 2881)	POINT, SEQ	Hourly	Energy
Metro	1		(26219, 3754, 7505)		5 min	Transportation

F.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 (Liu et al., 2023; Fons et al., 2025)[2], our evaluation adopts the latter perspective and does not explicitly model the missingness mechanism $P(M \mid X)$.

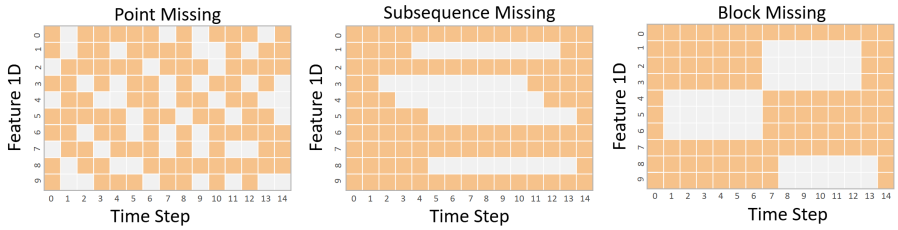


Figure 5: Illustration of the three masking strategies.

Guided by this perspective, we provide detailed definitions and implementation settings for the three masking strategies used in our experiments. As visualized in Figure 5, these strategies are designed to:

- **Point Missing:** Data points are masked independently across time and dimensions following a Bernoulli distribution. This setting simulates sporadic sensor noise, bit errors during transmission, or random data loss, testing the model's ability to aggregate information from scattered observations.
- **Subsequence Missing:** Continuous segments of time points are masked independently for each feature. This scenario simulates independent malfunctions of individual sensors within a network, where one sensor may fail while others remain operational.
- **Block Missing:** Large rectangular regions covering multiple variates and continuous time steps are masked simultaneously. This implies that a subset of features loses values for the same duration.

Implementation Details. To ensure a consistent evaluation, we standardize masking configurations across datasets whenever applicable, as summarized in Table 6. For **Point Missing**, we vary the missing ratio as $p \in \{0.1, 0.5, 0.9\}$. For **Subsequence Missing**, we fix the missing ratio at 0.1 and vary the segment length $l \in \{10, 20, 30\}$. For **Block Missing**, we use masking factors $\{0.1, 0.5, 0.9\}$ and a fixed temporal length of 6, while adapting the block width to the feature dimensionality of each dataset.

Table 6: Detailed masking configurations. p denotes the missing ratio, l the subsequence length, and factor, len, and width specify the missing ratio and block dimensions for Block Missing. "All" indicates a setting shared across all datasets.

Missing Patterns	**Dataset**	**Hyperparameters**
Point Missing	All	$p \in \{0.1, 0.5, 0.9\}$
Subsequence Missing	All	fix $p = 0.1$, $l \in \{10, 20, 30\}$
Block Missing	Weather	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=6
	ILI	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=6
	Stock	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=6
	Electricity	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=10
	PEMS-BAY	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=10
	METR-LA	factor $\in \{0.1, 0.5, 0.9\}$, len=6, width=10

[2]Although LSCD is a highly relevant and insightful work, we omit it from our baselines as its released implementation does not provide a complete end-to-end training and evaluation pipeline.

F.3 BASELINES

To comprehensively evaluate the performance of TFMemDiff, we conduct a comparative study against a diverse set of 11 state-of-the-art baselines, ranging from advanced Transformer variants to specialized generative models. Specifically, Transformer-based methods include **iTransformer** (Liu et al., 2024), which innovates on tokenization through inverted dimension embedding, **Non-stationary Transformer** (Liu et al., 2022) designed for non-stationary time series modeling, and the self-attention-based **SAITS** (Du et al., 2023). Beyond Transformers, we evaluate modern deep learning architectures such as **TimesNet** (Wu et al., 2023) for 2D temporal modeling, **ModernTCN** (Luo & Wang, 2024) utilizing large kernel convolutions, **FreTS** (Yi et al., 2023) operating in the frequency domain, and **Glocal-IB** (Yang et al., 2025), which leverages the Information Bottleneck principle to preserve global semantic structures against local noise. Furthermore, considering the specific requirements of missing value recovery, we include specialized generative and imputation models such as the score-based diffusion model **CSDI** (Tashiro et al., 2021), the structured state space model **SSSD** (Alcaraz & Strodthoff, 2023), the frequency-aware generative model **FGTI** (Yang et al., 2024), and **FSDI** (Shen et al., 2026), which exploits spectral characteristics to improve the diffusion process. Most baselines are implemented based on the unified Time-Series-Library (TSLib) (Wang et al., 2026), while others are adopted from their official repositories to ensure fair comparison.

F.4 EVALUATION METRICS

Following standard protocols in time series imputation (Cao et al., 2018; Tashiro et al., 2021), we quantitatively evaluate imputation performance using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Continuous Ranked Probability Score (CRPS). Let $\mathbf{M} \in \{0,1\}^{L\times D}$ denote the binary observation mask, where $m_{t,d} = 0$ indicates a missing entry and $m_{t,d} = 1$ an observed entry. All metrics are computed exclusively on masked entries (Du et al., 2023).

The **MAE** measures the average absolute reconstruction error:

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{L}\sum_{d=1}^{D}(1-m_{t,d})\left|x_{t,d}-\hat{x}_{t,d}\right|, \tag{36}$$

where $N = \sum_{t,d}(1-m_{t,d})$ is the total number of masked entries.

The **RMSE** measures the quadratic reconstruction error:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{L}\sum_{d=1}^{D}(1-m_{t,d})\left(x_{t,d}-\hat{x}_{t,d}\right)^2}. \tag{37}$$

Compared with MAE, RMSE assigns greater penalties to large reconstruction errors. For RMSE and MAE, we use the mean of the generated imputation samples as the point estimate.

For probabilistic generative models, we additionally report the **Continuous Ranked Probability Score (CRPS)** to evaluate the quality of the imputation distribution. Given S independently generated imputation samples $\{\hat{x}_{t,d}^{(s)}\}_{s=1}^{S}$ for a masked entry $x_{t,d}$, the empirical CRPS is computed as

$$\text{CRPS}_{t,d} = \frac{1}{S}\sum_{s=1}^{S}\left|\hat{x}_{t,d}^{(s)}-x_{t,d}\right| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left|\hat{x}_{t,d}^{(s)}-\hat{x}_{t,d}^{(s')}\right|. \tag{38}$$

The final CRPS is averaged over all masked entries:

$$\text{CRPS} = \frac{1}{N}\sum_{t=1}^{L}\sum_{d=1}^{D}(1-m_{t,d})\,\text{CRPS}_{t,d}. \tag{39}$$

Lower values of MAE, RMSE, and CRPS indicate better imputation performance. While MAE and RMSE evaluate point-wise reconstruction accuracy, CRPS evaluates the quality of the probabilistic imputation distribution.

F.5 Implementation Details

Hardware and Software Environment. The proposed model and all baselines are implemented using the PyTorch framework. All experiments are conducted on a high-performance server equipped with **NVIDIA RTX A6000 GPUs** (48GB memory per GPU) and **CUDA 12.2**. To ensure reproducibility and fair comparison, we execute each experiment on a single GPU.

Detailed Architecture of the Denoising Network. As illustrated in Figure 6, the denoising network $\epsilon_\theta(\cdot)$ comprises input/output projectors and a deep residual backbone. The **input projector** maps the concatenated noisy target x_t^{ta} and observed condition x_0^{ob} into latent space via a 1×1 convolution. The backbone consists of n_{layers} residual blocks, where diffusion steps are injected via sinusoidal embeddings. Crucially, dual-domain memory conditions are processed by Transformer-based encoders and dynamically integrated through dual-condition and fusion gates, alongside projected side information. Finally, the **output projector**, implemented as a 2-layer MLP with ReLU, projects latent features back to the original space for noise estimation.

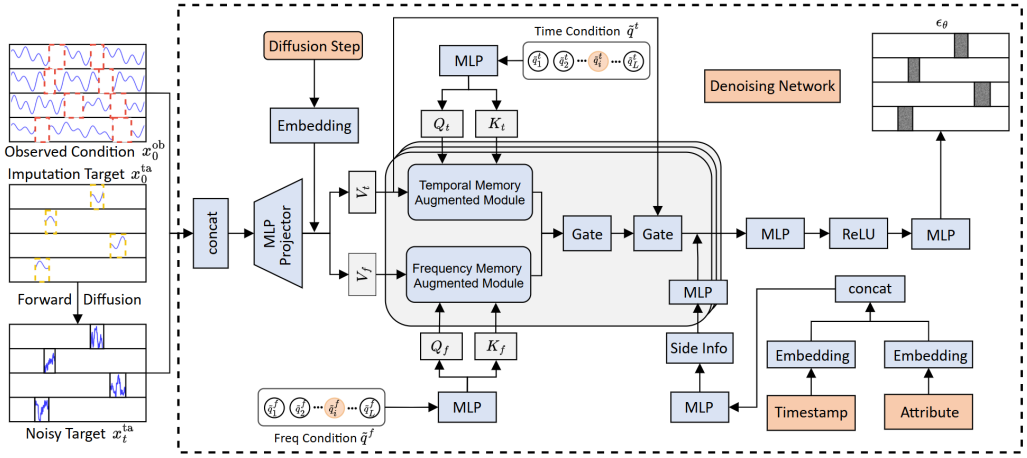


Figure 6: Architecture of the Denoising Network $\epsilon_\theta(\cdot)$ in TFMemDiff.

Hyperparameter Settings. Detailed hyperparameter configurations are summarized in Table 7. TFMemDiff uses an input sequence length of $L = 48$, a hidden dimension of $d_{\text{model}} = 128$, six residual layers, and eight attention heads. The diffusion process adopts a quadratic noise schedule ranging from $\beta_{\text{start}} = 10^{-4}$ to $\beta_{\text{end}} = 0.02$. For efficient inference, we employ DPM-Solver++ with 20 sampling steps and generate 100 stochastic imputation samples for each input.

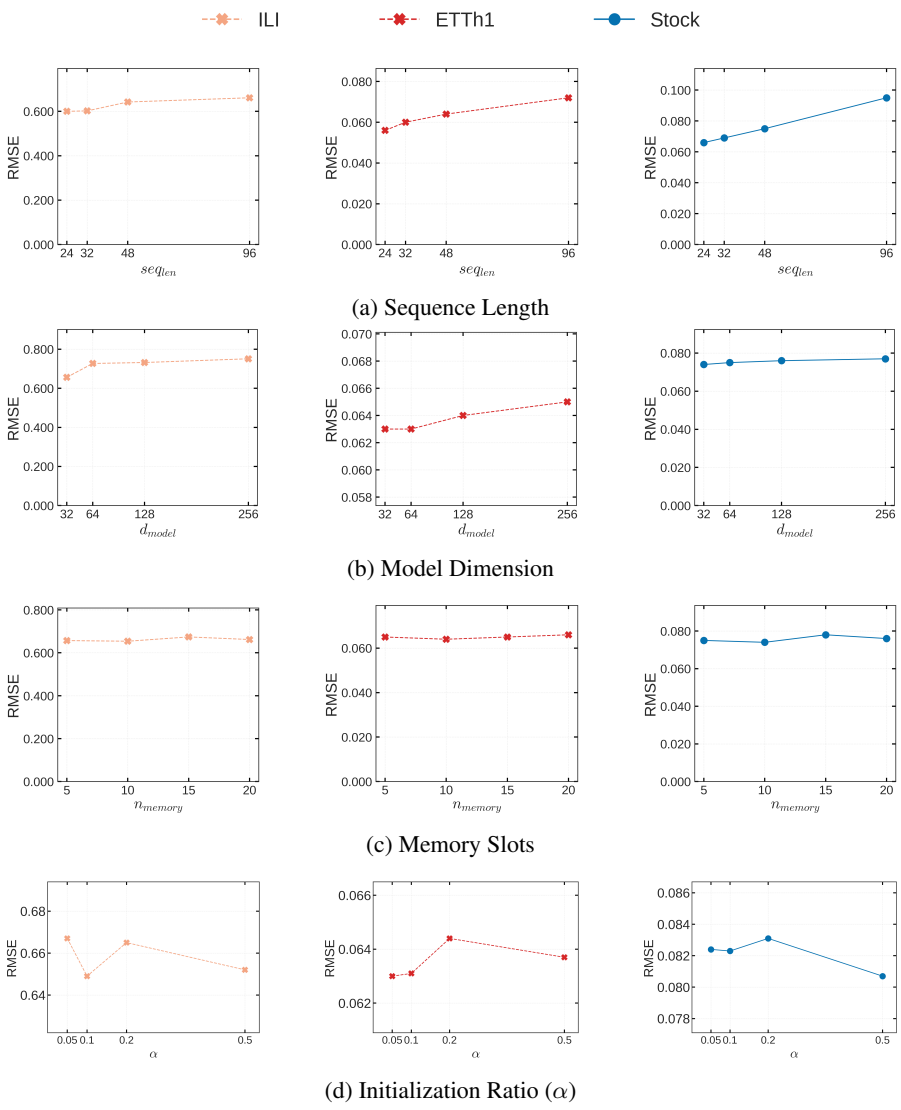
The dual-domain memory module contains $N_{\text{memory}} = 10$ slots, with temperature parameters $\tau = 0.05$ for memory updating and $\tau' = 0.05$ for memory retrieval. We use a memory initialization ratio of $\alpha = 0.1$, i.e., 10% of the training data are used to construct the initial memory prototypes. During training, we employ the Adam optimizer with an initial learning rate of 1×10^{-3}, a weight decay of 1×10^{-4}, and an entropy-loss weight of $\gamma = 0.01$. A dynamic batch size strategy is adopted: 256 for ILI, Stock, and ETTh1, and 64 for the remaining datasets. Training runs for at most 100 epochs with early stopping using a patience of 10 epochs.

Table 7: Detailed hyperparameter settings used in the experiments.

Parameter	**Value**	**Description**
Training Strategy		
optimizer	Adam	Optimization algorithm
learning_rate	1×10^{-3}	Initial learning rate
weight_decay	1×10^{-4}	Weight decay coefficient
γ	0.01	Entropy loss weight
train_epochs	100	Maximum training epochs
patience	10	Early stopping patience
seed	{2025, 2026, 2027}	Random seed
Model Architecture		
seq_len (L)	48	Input sequence length
d_model	128	Hidden feature dimension
residual_layers	6	Number of residual layers
n_heads	8	Number of attention heads
diff_schedule	Quad ($\beta_{\text{start}} = 10^{-4}, \beta_{\text{end}} = 0.02$)	Noise schedule
Memory Module		
n_memory (N)	10	Number of memory slots
τ	0.05	Temperature parameter for memory updating
τ'	0.05	Temperature parameter for memory retrieval
α	0.1	Memory initialization ratio
Inference Strategy		
sampler	DPM-Solver++	Accelerated diffusion sampler
sampling_steps	20	Number of inference steps per sample
n_samples	100	Number of stochastic imputation samples
Dataset-Specific Configuration		
batch_size	256 (ILI/Stock/ETTh1) 64 (remaining datasets)	Training batch size

Sensitivity Analysis of Hyperparameters. To evaluate the sensitivity of TFMemDiff, we analyze seven key hyperparameters under 50% point missing: memory size n_{memory}, hidden dimension d_{model}, sequence length seq_{len}, entropy-loss weight γ, temperature τ, encoder pre-training epochs and memory initialization ratio α. Results on three datasets are shown in Figure 7, with independently scaled y-axes for readability.

The memory size $n_{\text{memory}} \approx 10$ generally yields favorable performance, while TFMemDiff remains relatively stable across different hidden dimensions. The best performance is generally obtained around $\gamma = 0.01$ and $\tau = 0.05$, and encoder pre-training largely saturates after 50 epochs. For the memory initialization ratio α, moderate ratios perform well overall, with $\alpha = 0.1$ showing favorable performance across the evaluated datasets, while larger ratios do not yield consistent improvements. Longer input sequences provide limited additional benefits and may moderately degrade performance on more complex datasets such as Stock and ETTh1.



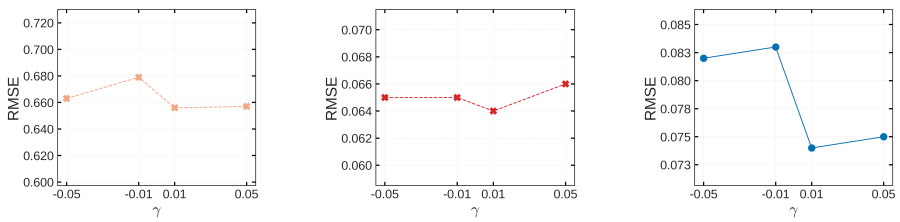
(a) Sequence Length

(b) Model Dimension

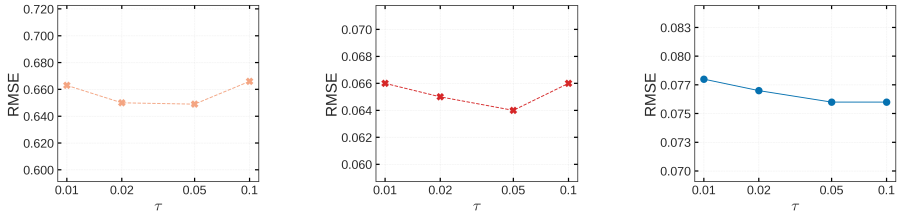
(c) Memory Slots

(d) Initialization Ratio (α)

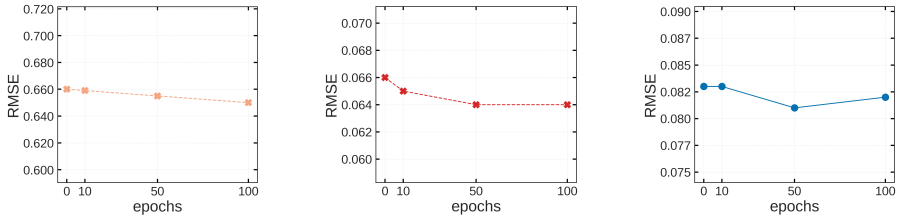
Figure 7: Hyperparameter sensitivity analysis on sequence length, model dimension, number of memory slots, memory initialization ratio α, the entropy loss weight γ, the temperature parameter τ, and the encoder pre-training epochs, reflected by RMSE.



(e) Entropy Loss Weight (γ)



(f) Temperature (τ)



(g) Pre-training Epochs

Figure 7: (continued)

G Additional Experiments

G.1 Performance on Multivariate Datasets

We first provide additional evaluation results on the multivariate datasets. The supplementary RMSE results are reported in Table 10. To complement the RMSE evaluation, we further report Mean Absolute Error (MAE) results in Tables 11 and 12. These results provide additional comparisons across different missingness patterns and missing ratios.

Notably, under point masking, TFMemDiff is not uniformly superior across all datasets, as randomly scattered missing values retain relatively rich local context. Its advantage is more pronounced under subsequence and block masking, where historical memory guidance provides additional structural information.

G.2 Performance on Univariate Datasets

To further evaluate the generalization capability of TFMemDiff beyond multivariate settings, we conduct experiments on two univariate datasets: ETTh1 (Oil Temperature) and Metro. We evaluate performance under Point Masking (10%, 50%, and 90%) and Subsequence Masking (lengths of 10, 20, and 30). The corresponding RMSE and MAE results are reported in Table 13.

Statistical Stability. For the additional experiments, each setting is repeated three times using distinct random seeds $\{2025, 2026, 2027\}$. The results are reported as Mean $\pm$ Standard Deviation to quantify the variability across different random initializations.

Table 8: CRPS comparison on Stock, Electricity, and METR-LA under different missing patterns and settings. **Bold** indicate the best performance.

Dataset	Type	Setting	CSDI	SSSD	FGTI	FSDI	Ours
Stock	POINT	50%	0.333	0.438	0.177	0.111	**0.006**
		90%	0.483	0.606	0.609	0.350	**0.018**
	SEQ	20	0.873	0.629	0.080	0.112	**0.004**
		30	0.210	0.272	0.594	0.171	**0.003**
	BLOCK	50%	0.882	0.493	0.313	0.253	**0.020**
		90%	0.485	0.487	0.566	0.402	**0.023**
Electricity	POINT	50%	0.261	0.428	0.347	0.448	**0.260**
		90%	0.450	0.640	0.508	**0.365**	0.482
	SEQ	20	0.274	0.648	0.434	0.252	**0.237**
		30	0.291	0.646	0.413	0.268	**0.252**
	BLOCK	50%	0.311	0.497	0.342	0.297	**0.288**
		90%	0.369	0.556	0.566	0.345	**0.328**
METR-LA	POINT	50%	0.341	0.401	0.357	0.341	**0.287**
		90%	0.353	0.549	0.424	**0.311**	0.354
	SEQ	20	0.322	0.702	0.346	0.306	**0.275**
		30	0.388	0.672	0.379	0.333	**0.285**
	BLOCK	50%	**0.272**	0.497	0.327	0.275	0.287
		90%	0.310	0.553	0.378	0.306	**0.300**

G.3 CRPS Performance Analysis

To further evaluate the probabilistic imputation quality of TFMemDiff, we report the Continuous Ranked Probability Score (CRPS) on three representative datasets, namely Stock, Electricity, and METR-LA, covering finance, energy, and transportation domains, respectively. CRPS evaluates the quality of the conditional distribution over missing values, with lower values indicating better probabilistic imputation performance.

As shown in Table 8, TFMemDiff achieves the best CRPS in 15 out of 18 evaluated settings across Point, Subsequence, and Block Missing patterns. The improvements are particularly pronounced on Stock, where TFMemDiff consistently obtains the lowest CRPS across all six settings. On Electricity and METR-LA, TFMemDiff also achieves the best performance in most settings, demonstrating its effectiveness in modeling uncertainty under diverse missing patterns and missingness levels.

G.4 Qualitative Analysis

To provide additional qualitative comparisons beyond the ILI visualization presented in the main text, we further compare TFMemDiff with five competitive baselines, including SAITS, CSDI, FGTI, Glocal-IB, and FSDI. We visualize the imputation results on the Electricity and METR-LA datasets under subsequence masking with a missing length of 20 and a 10% missing rate. As shown in Figures 8 and 9, TFMemDiff achieves better reconstruction quality on both datasets, with smaller deviations in the missing regions compared with most baselines. The results further demonstrate the effectiveness of TFMemDiff on different multivariate time series.

H COMPUTATIONAL EFFICIENCY

Table 9 reports the computational efficiency of TFMemDiff and representative diffusion-based baselines on ILI under 50% missingness. TFMemDiff contains 3.26M parameters and requires 0.32G FLOPs, remaining substantially lighter than SSSD and FGTI in computational complexity. Although the dual-domain memory module introduces additional memory overhead, the overall GPU memory consumption remains moderate, requiring 5.92GB during training and 0.57GB during inference.

In terms of runtime, TFMemDiff achieves the lowest training and inference time among the compared methods, requiring 0.23 seconds per training epoch and 30.87 seconds per inference batch. TFMemDiff also maintains a moderate model size and computational cost, despite incorporating the dual-domain memory modules. During the inference phase, we report the runtimes of all baseline methods under their canonical default configurations, ensuring they achieve their optimally imputation accuracy without being compromised by arbitrary solver modifications. Notably, the efficiency gains are not solely due to the accelerated sampler, as TFMemDiff also exhibits low FLOPs and training runtime independent of the inference solver.

Table 9: Computational efficiency comparison on ILI with 50% missingness.

Metric	**CSDI**	**SSSD**	**FGTI**	**FSDI**	**TFMemDiff**
Params (M)	0.41	48.11	1.95	0.41	3.26
FLOPs (G)	0.25	3.67	0.97	0.25	0.32
Training Memory (GB)	4.00	17.00	14.17	4.08	5.92
Inference Memory (GB)	0.33	0.59	0.82	0.44	0.57
Training Time (s/epoch)	0.30	0.92	0.77	0.29	0.23
Inference Time (s/batch)	50.66	825.72	51.57	73.87	30.87

Table 10: RMSE performance comparison on multivariate datasets. Lower is better. **Bold** and underlined indicate the best and second-best performance. Abbreviations: **NonStat.**: Non-stationary Transformer, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB. The values are reported as mean ± std.

Dataset	Type	Setting	NonStat.	FreTS	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
ILI	POINT	10%	0.657±0.004	0.760±0.013	0.699±0.005	0.885±0.003	2.826±0.029	1.778±0.022	1.245±0.009	1.111±0.015	0.475±0.006	1.384±0.048	2.831±0.003	**0.457±0.006**
		50%	0.794±0.008	0.849±0.004	0.753±0.009	1.360±0.005	3.655±0.022	1.782±0.009	1.786±0.005	1.680±0.010	0.735±0.005	1.494±0.000	2.992±0.003	**0.657±0.008**
		90%	1.652±0.022	1.566±0.016	1.481±0.022	2.131±0.011	2.611±0.021	1.774±0.004	2.375±0.005	2.451±0.007	1.454±0.022	2.227±0.013	3.334±0.004	**1.310±0.012**
	SEQ	10	0.984±0.059	1.301±0.040	0.509±0.012	1.433±0.044	2.315±0.090	1.764±0.012	1.308±0.015	1.477±0.071	0.982±0.049	1.390±0.011	1.771±0.003	**0.442±0.042**
		20	1.360±0.054	1.835±0.082	0.545±0.003	1.880±0.096	2.180±0.002	1.707±0.001	1.450±0.001	1.617±0.000	1.228±0.003	1.800±0.000	2.001±0.002	**0.413±0.002**
		30	2.367±0.078	1.991±0.092	1.619±0.010	2.010±0.092	2.342±0.078	1.674±0.096	1.605±0.091	1.383±0.046	1.411±0.013	1.935±0.044	2.574±0.002	**0.423±0.018**
	BLOCK	10%	1.032±0.075	1.216±0.087	0.948±0.072	1.418±0.012	2.301±0.043	2.003±0.013	1.962±0.010	1.942±0.016	1.039±0.008	1.639±0.021	1.730±0.005	**0.587±0.001**
		50%	1.203±0.015	1.343±0.029	1.130±0.002	1.610±0.041	4.464±0.020	1.839±0.043	1.916±0.042	1.728±0.034	1.150±0.032	1.626±0.014	2.372±0.002	**0.799±0.041**
		90%	1.429±0.048	1.561±0.041	1.315±0.054	1.822±0.041	4.388±0.097	1.863±0.025	1.993±0.025	1.792±0.020	1.309±0.045	1.863±0.011	2.866±0.009	**0.855±0.035**
Weather	POINT	10%	0.381±0.008	0.422±0.005	0.382±0.010	0.417±0.008	0.444±0.010	0.708±0.001	0.377±0.011	0.360±0.009	0.347±0.009	0.288±0.000	0.369±0.002	**0.224±0.017**
		50%	0.434±0.004	0.449±0.006	0.416±0.005	0.547±0.005	0.515±0.037	0.686±0.001	0.461±0.007	0.792±0.004	0.413±0.005	**0.344±0.000**	0.376±0.005	0.428±0.013
		90%	0.547±0.100	0.680±0.100	0.642±0.003	0.822±0.002	0.744±0.000	0.630±0.000	0.609±0.001	1.776±0.001	0.626±0.001	0.479±0.000	0.481±0.003	**0.436±0.002**
	SEQ	10	0.456±0.013	0.485±0.015	0.454±0.014	0.496±0.012	0.476±0.011	0.736±0.000	0.427±0.005	0.403±0.013	0.409±0.019	0.387±0.016	0.446±0.004	**0.382±0.012**
		20	0.543±0.018	0.576±0.006	0.548±0.016	0.583±0.008	0.582±0.007	0.745±0.010	0.455±0.004	0.432±0.006	0.474±0.013	0.388±0.000	0.422±0.003	**0.383±0.004**
		30	0.645±0.010	0.870±0.009	1.121±0.013	0.653±0.003	0.624±0.008	0.698±0.019	0.496±0.015	0.483±0.009	0.562±0.0005	**0.331±0.000**	0.445±0.007	0.467±0.002
	BLOCK	10%	0.524±0.014	0.493±0.013	0.481±0.014	0.522±0.011	0.485±0.013	0.631±0.017	0.504±0.013	0.447±0.005	0.447±0.014	**0.382±0.003**	0.472±0.002	0.401±0.012
		50%	0.516±0.006	0.508±0.005	0.488±0.005	0.587±0.005	0.523±0.006	0.648±0.005	0.497±0.009	0.459±0.009	0.472±0.005	0.448±0.000	0.553±0.007	**0.439±0.004**
		90%	0.547±0.001	0.525±0.001	0.505±0.002	0.627±0.001	0.552±0.000	0.653±0.001	0.500±0.001	0.486±0.001	0.475±0.001	0.480±0.000	0.559±0.005	**0.462±0.002**
Stock	POINT	10%	0.100±0.001	0.248±0.002	0.108±0.001	0.513±0.008	0.235±0.003	2.608±0.007	2.250±0.005	0.570±0.003	0.108±0.001	2.214±0.002	1.112±0.003	**0.042±0.000**
		50%	0.120±0.003	0.155±0.000	0.114±0.000	1.322±0.002	0.316±0.001	2.425±0.000	2.648±0.002	2.948±0.001	0.126±0.005	2.337±0.007	1.165±0.002	**0.077±0.000**
		90%	0.496±0.034	0.493±0.034	0.484±0.035	3.166±0.001	0.511±0.034	2.188±0.033	2.761±0.005	2.909±0.001	0.482±0.035	2.504±0.021	1.825±0.002	**0.156±0.004**
Electricity	POINT	10%	0.340±0.001	0.358±0.007	0.336±0.012	0.353±0.004	0.768±0.005	0.353±0.000	0.362±0.004	0.334±0.007	**0.305±0.002**	0.451±0.004	0.338±0.002	0.414±0.001
		50%	0.453±0.008	0.501±0.034	0.450±0.004	0.500±0.009	0.898±0.005	0.563±0.007	0.440±0.003	0.513±0.009	0.476±0.056	0.515±0.030	**0.347±0.001**	0.436±0.008
		90%	0.803±0.001	0.982±0.005	0.791±0.004	0.815±0.007	1.176±0.034	0.996±0.040	**0.564±0.012**	0.768±0.040	0.725±0.064	0.703±0.003	0.648±0.003	0.793±0.005
PEMS-BAY	POINT	10%	0.256±0.001	0.688±0.005	0.671±0.004	0.235±0.007	0.257±0.004	0.315±0.006	0.272±0.003	0.244±0.001	**0.217±0.047**	0.272±0.050	0.258±0.002	0.387±0.015
		50%	0.352±0.056	0.695±0.050	0.697±0.003	0.342±0.006	0.630±0.005	0.461±0.004	0.364±0.007	0.433±0.004	0.412±0.002	0.344±0.007	**0.233±0.007**	0.408±0.005
		90%	0.584±0.006	0.582±0.004	0.782±0.006	0.568±0.003	0.765±0.006	0.867±0.009	0.520±0.005	0.647±0.005	0.653±0.002	0.532±0.000	**0.490±0.001**	0.656±0.007
METR-LA	POINT	10%	0.291±0.001	0.478±0.005	0.629±0.005	0.292±0.004	0.336±0.008	0.309±0.005	**0.266±0.005**	0.316±0.006	0.282±0.004	0.333±0.012	0.268±0.002	0.401±0.004
		50%	0.342±0.006	0.485±0.007	0.643±0.004	0.359±0.005	0.468±0.007	0.419±0.001	0.314±0.004	0.420±0.045	0.336±0.005	0.408±0.001	**0.289±0.009**	0.413±0.008
		90%	0.487±0.006	0.473±0.005	0.554±0.008	0.501±0.003	0.534±0.006	0.680±0.050	0.479±0.005	0.504±0.001	0.556±0.006	0.632±0.007	**0.448±0.003**	0.465±0.004

Table 11: MAE performance comparison on multivariate datasets. Lower is better. **Bold** and underlined indicate the best and second-best performance. Abbreviations: **NonStat.**: Non-stationary Transformer, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB. The values are reported as mean ± std.

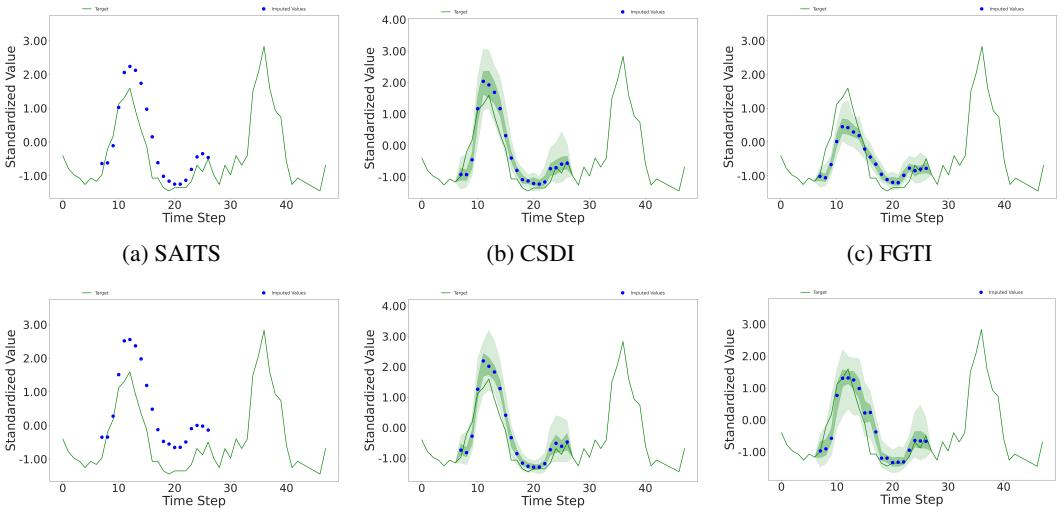
Dataset	Type	Setting	NonStat.	FreTS	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
Weather	POINT	10%	0.122±0.001	0.186±0.000	0.123±0.000	0.170±0.000	0.157±0.000	0.153±0.001	0.135±0.000	0.070±0.000	0.097±0.000	0.128±0.000	**0.067±0.003**	0.070±0.001
		50%	0.159±0.000	0.173±0.000	0.147±0.000	0.272±0.000	0.218±0.001	0.159±0.001	0.210±0.000	0.353±0.000	0.135±0.000	0.167±0.000	0.077±0.002	**0.074±0.001**
		90%	0.250±0.001	0.305±0.001	0.290±0.001	0.450±0.001	0.361±0.001	0.116±0.000	0.343±0.001	0.996±0.001	0.270±0.001	0.259±0.000	0.153±0.008	**0.077±0.000**
	SEQ	10	0.170±0.001	0.193±0.001	0.173±0.001	0.211±0.001	0.178±0.001	0.177±0.003	0.178±0.001	0.092±0.000	0.139±0.002	0.138±0.000	0.097±0.002	**0.076±0.002**
		20	0.230±0.003	0.267±0.002	0.240±0.002	0.283±0.002	0.263±0.002	0.196±0.006	0.191±0.001	0.126±0.001	0.176±0.002	0.146±0.000	0.110±0.003	**0.075±0.001**
		30	0.304±0.004	0.475±0.005	0.588±0.007	0.322±0.004	0.317±0.001	0.167±0.012	0.234±0.005	0.149±0.003	0.217±0.004	0.150±0.000	**0.129±0.003**	0.161±0.001
	BLOCK	10%	0.223±0.002	0.196±0.000	0.191±0.001	0.229±0.001	0.178±0.000	0.394±0.010	0.227±0.001	0.135±0.001	0.161±0.000	0.140±0.000	0.146±0.002	**0.127±0.000**
		50%	0.218±0.000	0.208±0.001	0.197±0.000	0.291±0.000	0.209±0.001	0.404±0.002	0.221±0.001	**0.140±0.001**	0.180±0.000	0.154±0.000	0.157±0.002	0.152±0.001
		90%	0.249±0.001	0.229±0.001	0.218±0.001	0.329±0.001	0.244±0.000	0.408±0.000	0.232±0.000	**0.159±0.001**	0.190±0.001	0.181±0.000	0.193±0.006	0.162±0.000
ILI	POINT	10%	0.428±0.004	0.476±0.010	0.454±0.002	0.606±0.010	0.919±0.025	1.009±0.011	0.767±0.008	0.715±0.010	0.281±0.001	0.861±0.008	1.432±0.002	**0.232±0.003**
		50%	0.516±0.002	0.532±0.002	0.484±0.003	0.948±0.010	1.308±0.021	1.058±0.006	1.172±0.002	1.235±0.005	0.446±0.002	0.990±0.001	1.176±0.009	**0.324±0.003**
		90%	1.034±0.012	0.977±0.012	0.918±0.013	1.487±0.007	1.382±0.172	1.070±0.003	1.657±0.004	1.805±0.005	0.884±0.014	1.588±0.016	1.920±0.003	**0.769±0.005**
	SEQ	10	0.615±0.003	0.823±0.013	0.248±0.002	0.959±0.001	1.237±0.009	1.016±0.006	0.807±0.004	0.642±0.045	0.599±0.005	1.003±0.011	0.761±0.002	**0.214±0.001**
		20	0.886±0.002	1.251±0.002	0.273±0.001	1.299±0.004	1.203±0.032	0.961±0.008	0.906±0.001	0.755±0.039	0.750±0.002	1.191±0.060	1.372±0.006	**0.206±0.001**
		30	1.763±0.003	1.378±0.003	1.082±0.005	1.388±0.004	2.394±0.054	0.932±0.006	1.017±0.039	0.586±0.018	0.895±0.003	1.406±0.283	1.789±0.002	**0.203±0.000**
	BLOCK	10%	0.670±0.005	0.780±0.005	0.610±0.004	0.923±0.007	1.149±0.028	1.185±0.012	1.284±0.007	1.024±0.103	0.609±0.004	1.078±0.055	1.250±0.002	**0.291±0.002**
		50%	0.767±0.001	0.860±0.001	0.730±0.001	1.071±0.002	3.738±0.081	1.044±0.002	1.270±0.002	0.873±0.028	0.711±0.001	1.062±0.003	1.200±0.002	**0.488±0.003**
		90%	0.921±0.002	0.995±0.002	0.836±0.003	1.204±0.002	3.645±0.049	1.079±0.002	1.315±0.001	0.924±0.014	0.811±0.002	1.254±0.003	1.315±0.002	**0.530±0.003**
Stock	POINT	10%	0.069±0.001	0.181±0.002	0.074±0.001	0.368±0.006	0.169±0.002	2.104±0.005	1.941±0.006	0.400±0.001	0.076±0.001	1.533±0.000	0.580±0.003	**0.023±0.000**
		50%	0.083±0.000	0.108±0.000	0.080±0.000	1.000±0.003	0.233±0.001	1.916±0.002	2.217±0.002	2.582±0.001	0.088±0.000	1.649±0.005	0.683±0.002	**0.042±0.000**
		90%	0.205±0.005	0.199±0.005	0.185±0.005	2.891±0.009	0.230±0.005	1.708±0.001	2.340±0.004	2.373±0.001	0.182±0.005	1.768±0.015	1.404±0.002	**0.104±0.000**
	SEQ	10	0.097±0.001	0.162±0.001	0.092±0.001	0.326±0.001	0.178±0.001	1.973±0.011	2.066±0.020	0.645±0.005	0.116±0.001	1.557±0.003	1.132±0.009	**0.021±0.000**
		20	0.114±0.002	0.179±0.002	0.112±0.001	0.431±0.003	0.187±0.001	2.048±0.004	2.077±0.006	0.715±0.007	0.110±0.001	1.569±0.001	0.695±0.007	**0.022±0.001**
		30	0.136±0.001	0.223±0.002	0.251±0.003	0.264±0.002	0.198±0.002	1.953±0.040	2.027±0.044	0.769±0.012	0.123±0.000	1.542±0.004	1.329±0.007	**0.023±0.000**
	BLOCK	10%	0.079±0.001	0.156±0.002	0.089±0.001	0.384±0.016	0.191±0.003	2.118±0.020	2.298±0.035	0.686±0.005	0.131±0.001	1.518±0.001	0.656±0.007	**0.042±0.002**
		50%	0.106±0.001	0.151±0.001	0.106±0.001	0.715±0.011	0.184±0.002	2.181±0.008	2.389±0.003	1.367±0.001	0.118±0.000	1.660±0.006	1.489±0.006	**0.062±0.000**
		90%	0.126±0.001	0.165±0.001	0.123±0.001	0.951±0.019	0.265±0.004	2.120±0.002	2.436±0.001	2.075±0.010	0.128±0.001	1.705±0.005	1.445±0.004	**0.069±0.000**

Table 12: MAE performance comparison on multivariate datasets. Lower is better. **Bold** and underlined indicate the best and second-best performance. Abbreviations: **NonStat.**: Non-stationary Transformer, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB. The values are reported as mean ± std.

Dataset	Type	Setting	NonStat.	FreTS	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
Electricity	POINT	10%	0.245±0.000	0.252±0.000	0.243±0.000	0.248±0.000	0.531±0.000	0.252±0.000	0.265±0.001	0.235±0.000	**0.219±0.000**	0.336±0.000	0.232±0.002	0.414±0.001
		50%	0.313±0.000	0.355±0.000	0.404±0.001	0.365±0.000	0.698±0.000	0.391±0.000	0.400±0.000	0.363±0.000	0.371±0.000	0.391±0.000	**0.237±0.003**	0.307±0.002
		90%	0.559±0.000	0.687±0.000	0.549±0.000	0.599±0.001	0.892±0.001	0.709±0.001	**0.422±0.002**	0.570±0.001	0.496±0.001	0.542±0.000	0.442±0.003	0.590±0.001
	SEQ	10	0.327±0.000	0.368±0.001	0.312±0.001	0.331±0.000	0.352±0.002	0.353±0.003	0.337±0.002	0.499±0.001	0.307±0.003	0.391±0.001	0.299±0.006	**0.287±0.003**
		20	0.336±0.002	0.391±0.001	0.319±0.001	0.340±0.000	0.378±0.002	0.401±0.001	0.347±0.001	0.558±0.000	0.317±0.003	0.399±0.000	0.306±0.003	**0.283±0.002**
		30	0.398±0.002	0.504±0.001	0.390±0.001	0.432±0.006	0.386±0.003	0.428±0.001	0.371±0.001	0.455±0.001	0.374±0.002	0.418±0.003	0.324±0.004	**0.297±0.003**
	BLOCK	10%	0.343±0.003	0.412±0.001	0.329±0.001	0.374±0.000	0.325±0.000	0.485±0.000	0.347±0.000	0.490±0.002	0.317±0.001	0.396±0.001	0.351±0.008	**0.314±0.000**
		50%	0.380±0.001	0.501±0.000	0.361±0.001	0.454±0.001	0.406±0.002	0.469±0.001	0.352±0.001	0.536±0.002	0.356±0.001	0.410±0.001	0.377±0.004	**0.351±0.002**
		90%	0.415±0.000	0.578±0.002	0.404±0.000	0.517±0.003	0.579±0.000	0.610±0.000	0.414±0.000	0.532±0.009	0.416±0.000	0.437±0.000	0.432±0.004	**0.400±0.001**
PEMS-BAY	POINT	10%	0.142±0.001	0.357±0.000	0.354±0.004	0.141±0.003	0.135±0.003	0.182±0.005	0.149±0.001	0.116±0.000	0.123±0.003	0.144±0.005	**0.107±0.005**	0.182±0.002
		50%	0.184±0.001	0.356±0.004	0.361±0.005	0.199±0.001	0.322±0.000	0.255±0.001	0.201±0.006	0.259±0.004	0.266±0.001	0.193±0.001	**0.122±0.008**	0.200±0.000
		90%	0.281±0.002	0.275±0.002	0.384±0.003	0.306±0.004	0.385±0.002	0.484±0.001	0.287±0.005	0.362±0.003	0.270±0.004	0.296±0.001	**0.216±0.003**	0.336±0.006
	SEQ	10	0.231±0.001	0.210±0.003	0.223±0.002	0.213±0.004	0.190±0.003	0.297±0.008	0.250±0.003	0.214±0.001	0.217±0.015	0.240±0.013	0.216±0.002	**0.180±0.001**
		20	0.273±0.001	0.281±0.007	0.268±0.003	0.271±0.001	0.271±0.002	0.328±0.007	0.281±0.003	0.325±0.003	0.259±0.001	0.264±0.004	0.234±0.004	**0.183±0.002**
		30	0.321±0.001	0.340±0.003	0.319±0.001	0.307±0.001	0.325±0.003	0.378±0.001	0.278±0.026	0.428±0.010	0.318±0.000	0.271±0.001	0.273±0.003	**0.182±0.021**
	BLOCK	10%	0.223±0.012	0.192±0.005	0.212±0.002	0.208±0.006	**0.169±0.008**	0.333±0.005	0.213±0.001	0.179±0.005	0.190±0.008	0.212±0.002	0.193±0.003	0.184±0.001
		50%	0.237±0.001	0.216±0.002	0.232±0.003	0.253±0.002	0.209±0.004	0.439±0.001	0.231±0.002	0.205±0.003	0.218±0.003	0.237±0.001	0.204±0.009	**0.203±0.002**
		90%	0.251±0.003	0.387±0.008	0.248±0.002	0.273±0.003	0.234±0.003	0.458±0.001	0.247±0.002	0.288±0.003	0.230±0.001	0.259±0.001	0.226±0.003	**0.221±0.001**
METR-LA	POINT	10%	0.165±0.002	0.257±0.004	0.349±0.003	0.172±0.001	0.167±0.001	0.191±0.004	0.164±0.002	0.187±0.006	0.158±0.002	0.210±0.001	**0.141±0.002**	0.201±0.001
		50%	0.186±0.001	0.260±0.002	0.349±0.001	0.218±0.001	0.256±0.001	0.237±0.001	0.187±0.003	0.232±0.001	0.181±0.001	0.248±0.003	**0.149±0.002**	0.205±0.001
		90%	0.246±0.004	0.238±0.003	0.282±0.000	0.284±0.000	0.282±0.001	0.376±0.002	0.273±0.003	0.256±0.002	0.258±0.001	0.359±0.003	**0.208±0.004**	0.243±0.001
	SEQ	10	0.223±0.002	0.210±0.006	0.218±0.002	0.217±0.005	0.189±0.002	0.260±0.002	0.222±0.000	0.189±0.001	0.216±0.002	0.255±0.005	0.184±0.001	**0.183±0.010**
		20	0.241±0.001	0.234±0.003	0.239±0.004	0.243±0.001	0.221±0.001	0.254±0.002	0.238±0.003	0.236±0.004	0.241±0.004	0.275±0.003	0.208±0.004	**0.196±0.002**
		30	0.263±0.002	0.262±0.001	0.261±0.005	0.257±0.001	0.251±0.001	0.292±0.006	0.247±0.002	0.269±0.003	0.275±0.001	0.279±0.003	0.221±0.009	**0.197±0.001**
	BLOCK	10%	0.228±0.002	0.204±0.001	0.218±0.001	0.229±0.004	0.181±0.001	0.441±0.000	0.206±0.002	**0.177±0.001**	0.206±0.003	0.250±0.004	0.191±0.003	0.200±0.001
		50%	0.228±0.002	0.215±0.004	0.225±0.002	0.259±0.002	0.213±0.001	0.305±0.001	0.214±0.001	0.275±0.002	0.216±0.001	0.270±0.003	0.199±0.003	**0.197±0.001**
		90%	0.235±0.002	0.281±0.002	0.232±0.002	0.287±0.004	0.225±0.001	0.370±0.005	0.226±0.002	0.321±0.003	0.227±0.003	0.295±0.002	0.215±0.008	**0.214±0.001**

Table 13: RMSE and MAE performance comparison on univariate datasets. Lower is better. **Bold** and underlined indicate the best and second-best performance. Abbreviations: **NonStat.**: Non-stationary Transformer, **iTrans.**: iTransformer, **M.TCN**: ModernTCN, **Glocal.**: Glocal-IB. The values are reported as mean ± std.

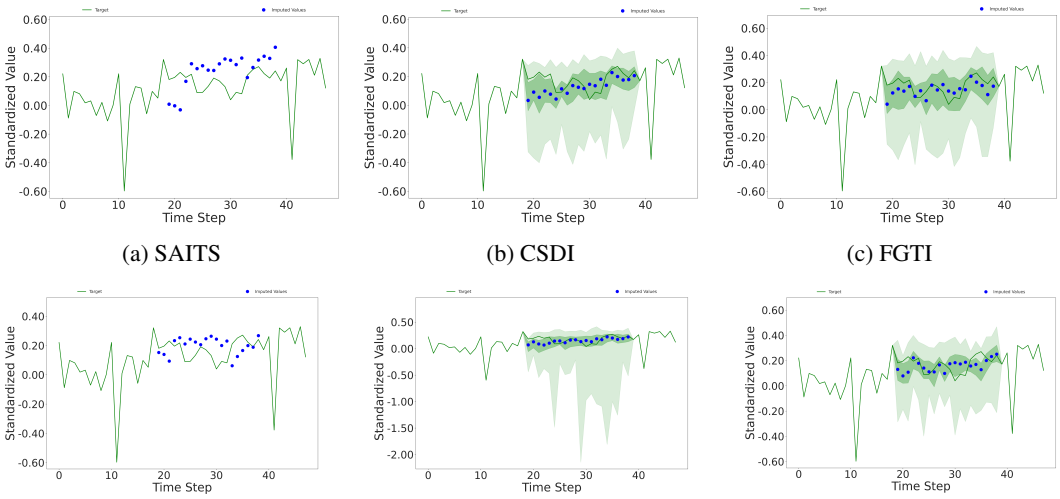
Dataset	Type	Setting	Metric	NonStat.	FreTS	TimesNet	iTrans.	CSDI	SSSD	SAITS	FGTI	M.TCN	Glocal.	FSDI	Ours
ETTh1 (OT)	POINT	10%	RMSE	0.060±0.001	0.059±0.001	0.053±0.001	0.160±0.003	0.142±0.000	0.376±0.003	0.286±0.002	0.059±0.000	0.052±0.001	0.132±0.000	0.162±0.003	**0.047±0.001**
			MAE	0.043±0.000	0.044±0.000	**0.038±0.000**	0.124±0.001	0.112±0.000	0.331±0.003	0.241±0.001	0.045±0.000	**0.038±0.000**	0.099±0.001	0.128±0.001	**0.038±0.000**
		50%	RMSE	0.078±0.000	0.081±0.000	0.067±0.000	0.334±0.001	0.144±0.000	0.377±0.001	0.280±0.001	0.105±0.000	0.066±0.000	0.129±0.000	0.165±0.007	**0.065±0.000**
			MAE	0.057±0.000	0.060±0.000	0.048±0.000	0.258±0.001	0.112±0.000	0.332±0.001	0.238±0.001	0.084±0.000	0.048±0.000	0.098±0.000	0.130±0.008	**0.041±0.000**
		90%	RMSE	0.169±0.007	0.172±0.006	0.170±0.006	0.652±0.002	0.190±0.006	0.376±0.000	0.345±0.002	0.283±0.005	0.169±0.006	0.218±0.000	0.275±0.003	**0.116±0.000**
			MAE	0.104±0.001	0.106±0.001	0.103±0.001	0.452±0.001	0.129±0.001	0.376±0.000	0.345±0.002	0.219±0.001	0.102±0.001	0.169±0.000	0.201±0.004	**0.089±0.000**
	SEQ	10	RMSE	0.126±0.002	0.116±0.002	0.105±0.002	0.142±0.002	0.161±0.001	0.376±0.003	0.291±0.003	0.110±0.000	0.100±0.001	0.201±0.000	0.170±0.003	**0.062±0.000**
			MAE	0.098±0.001	0.090±0.001	0.079±0.001	0.111±0.001	0.126±0.001	0.331±0.004	0.246±0.002	0.083±0.001	0.076±0.000	0.155±0.000	0.136±0.006	**0.046±0.000**
		20	RMSE	0.145±0.001	0.138±0.001	0.133±0.002	0.165±0.002	0.164±0.001	0.373±0.004	0.306±0.005	0.149±0.003	0.140±0.001	0.285±0.005	0.176±0.009	**0.092±0.001**
			MAE	0.113±0.001	0.107±0.001	0.103±0.002	0.130±0.001	0.129±0.001	0.330±0.004	0.253±0.004	0.114±0.002	0.109±0.001	0.228±0.004	0.139±0.006	**0.070±0.001**
		30	RMSE	0.156±0.002	0.153±0.002	0.150±0.002	0.183±0.002	0.170±0.002	0.368±0.005	0.296±0.004	0.171±0.003	0.148±0.002	0.244±0.001	0.215±0.008	**0.110±0.001**
			MAE	0.122±0.001	0.119±0.001	0.116±0.001	0.142±0.001	0.135±0.001	0.324±0.004	0.247±0.004	0.132±0.003	0.114±0.001	0.202±0.000	0.168±0.004	**0.085±0.001**
Metro	POINT	10%	RMSE	0.357±0.002	0.383±0.001	0.304±0.002	0.405±0.001	0.989±0.001	0.994±0.000	0.398±0.001	0.335±0.002	0.293±0.004	0.334±0.000	0.153±0.002	**0.125±0.001**
			MAE	0.223±0.002	0.267±0.001	0.187±0.001	0.289±0.000	0.849±0.002	0.878±0.001	0.281±0.001	0.214±0.001	0.180±0.002	0.237±0.000	0.094±0.003	**0.072±0.000**
		50%	RMSE	0.394±0.000	0.568±0.002	0.353±0.001	0.519±0.000	0.988±0.001	0.993±0.001	0.496±0.001	0.474±0.002	0.431±0.002	0.463±0.000	0.189±0.009	**0.143±0.001**
			MAE	0.270±0.000	0.420±0.001	0.234±0.001	0.388±0.000	0.860±0.002	0.877±0.002	0.343±0.000	0.304±0.001	0.278±0.001	0.316±0.000	0.106±0.003	**0.079±0.000**
		90%	RMSE	0.595±0.002	0.974±0.003	0.643±0.001	0.707±0.002	1.090±0.003	0.993±0.000	0.812±0.002	1.034±0.001	0.936±0.003	0.979±0.000	0.548±0.004	**0.472±0.001**
			MAE	0.426±0.001	0.752±0.003	0.448±0.001	0.552±0.001	0.914±0.002	0.877±0.000	0.639±0.002	0.784±0.001	0.704±0.002	0.850±0.000	0.332±0.005	**0.298±0.000**
	SEQ	10	RMSE	0.431±0.003	0.830±0.001	0.466±0.002	0.585±0.005	0.743±0.007	0.990±0.001	0.800±0.005	0.886±0.007	0.648±0.010	0.991±0.000	0.228±0.003	**0.134±0.001**
			MAE	0.330±0.002	0.636±0.001	0.334±0.002	0.436±0.003	0.573±0.005	0.874±0.001	0.636±0.004	0.656±0.006	0.470±0.006	0.867±0.000	0.142±0.006	**0.077±0.001**
		20	RMSE	0.476±0.007	0.932±0.006	0.539±0.008	0.606±0.004	0.866±0.003	0.997±0.002	0.864±0.002	1.208±0.001	0.834±0.004	1.004±0.000	0.357±0.001	**0.190±0.005**
			MAE	0.364±0.004	0.730±0.005	0.398±0.004	0.450±0.003	0.672±0.003	0.882±0.003	0.709±0.002	0.956±0.002	0.634±0.003	0.879±0.001	0.226±0.007	**0.108±0.002**
		30	RMSE	0.506±0.009	0.972±0.013	0.574±0.017	0.637±0.009	0.983±0.004	0.993±0.004	0.905±0.004	1.230±0.004	0.904±0.008	1.001±0.000	0.451±0.004	**0.270±0.005**
			MAE	0.393±0.006	0.762±0.011	0.425±0.011	0.483±0.007	0.783±0.004	0.876±0.004	0.750±0.006	0.989±0.005	0.694±0.007	0.878±0.000	0.284±0.001	**0.156±0.002**



(a) SAITS (b) CSDI (c) FGTI

(d) Glocal-IB (e) FSDI (f) Ours

Figure 8: Imputation results on Electricity under the subsequence missing scenario (length=20) with a 10% missing rate.



(a) SAITS (b) CSDI (c) FGTI

(d) Glocal-IB (e) FSDI (f) Ours

Figure 9: Imputation results on METR_LA under the subsequence missing scenario (length=20) with a 10% missing rate.