Discretizing Continuous Time Series for Imputation with Masked Diffusion Training

Dongbin Kim[1], **Seungyun Lee**[1], **Geonwoo Shin**[1], **Jaewook Lee**[1] *

[1]Seoul National University

{dongbin413,rats96,shin0621,jaewook}@snu.ac.kr

Abstract

Time series imputation is a crucial area for reliable time series analysis, yet it remains challenging due to the complex temporal dynamics and noise of real-world data. Existing approaches, however, exhibit two limitations: missing and observed values are embedded within the same representation space without explicit structural separation, and continuous diffusion-based methods recover the missing values only through many iterative denoising steps rather than by predicting them directly. To address these, we propose the Masked Diffusion Time-series Imputation Model (MDTIM), which leverages the training paradigm of masked diffusion model for imputation tasks. The `[MASK]` token is structurally orthogonal to valid observations, and because the corrupted state is the partially observed series itself, the model is trained to predict the missing values directly from the observed ones, naturally aligning both the representation and the learning objective with the imputation task. To bridge the gap between discrete masked diffusion and the continuous, ordinal nature of time series, we further introduce Stochastic Discretization, which maps continuous values to ordinal-aware tokens while preserving continuous dynamics. Our experiments on diverse benchmarks confirm that MDTIM achieves superior robustness and scalability, consistently outperforming state-of-the-art deterministic and generative baselines across various missing scenarios. Code and project page are available at `https://dongbeank.github.io/MDTIM/`.

1 Introduction

Time series imputation is a fundamental task in time series analysis, as real-world data are often partially observed due to sensor failures or transmission errors [16]. At its core, imputation aims to recover the underlying clean signal from partial observations, which requires the representation and learning objective of the model to be well-aligned with the structure of partial observability.

From this perspective, we observe that existing approaches leave room for improvement along two axes. First, at the representation level, discriminative Transformer-based methods [6] and continuous diffusion models [33] inform the model of missing positions through a missingness indicator, an attention mask, or a conditioning mask, and the question is how that information enters the network. An indicator marks the position, but a placeholder value (e.g., zero) still occupies the same embedding space as valid observations, so the network has to learn to discount it [8]. An attention mask can remove an interaction, but it cannot announce anything to the rest of the sequence. Second, at the objective level, continuous diffusion models such as CSDI [33] are trained to denoise a Gaussian-corrupted copy of the missing values at a sampled noise level, so the quantity imputation actually needs, the posterior over the missing values given the observed ones, is obtained only after the full reverse chain is integrated.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

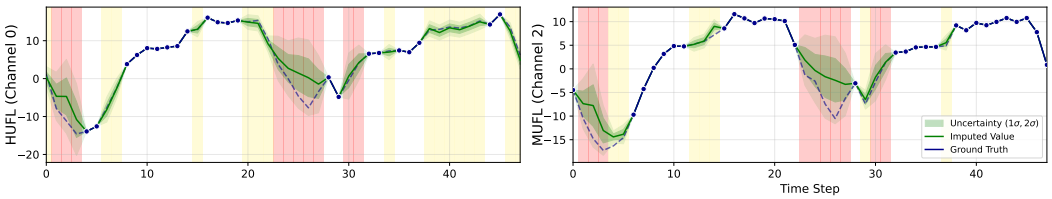


Figure 1: **Imputation results of MDTIM on the ETTh dataset.** Red regions indicate intervals where both channels are missing, while yellow regions represent single-channel missingness. The uncertainty (σ), derived from the output probability of MDTIM, is estimated to be higher in the red regions **(information scarcity)** compared to the yellow regions **(richer information)**.

Masked Diffusion Models (MDM), recently developed in Natural Language Processing [28], offer a natural fit for both aspects. Unlike continuous diffusion that corrupts data with Gaussian noise, MDM replaces tokens with a special `[MASK]` symbol and learns to recover the original values. The `[MASK]` token is structurally orthogonal to any valid observation, and it is a symbol that the rest of the sequence reads, so every observed position sees where the gaps are and conditions on their shape. Because the corrupted state is the partially observed series itself, the model is trained to predict the clean value at every masked position given the observed ones, and a single forward pass returns the full distribution over every missing value, which is exactly the quantity imputation asks for. We compare these mechanisms under controlled conditions in Section 5.5.

Despite this conceptual fit, directly applying MDM to time series imputation presents a fundamental challenge. Time series data are inherently continuous and ordinal, whereas language tokens are discrete and categorical. Naive quantization severs the ordinal relationships between adjacent values and loses information below the grid resolution, which is particularly problematic when fine-grained reconstruction is needed. This raises our central research question: *How can we leverage the training paradigm of masked diffusion while preserving the continuous and ordinal nature of time series data?*

To answer this, we propose the **Masked Diffusion Time-series Imputation Model (MDTIM)**, a framework that bridges discrete masked diffusion and continuous time series modeling. We introduce *Stochastic Discretization*, which injects noise during tokenization to preserve information in expectation, and replace the standard one-hot objective with *Ordinal-Aware Soft Labeling* to capture the ordinal relationships between tokens. We further incorporate a spectral consistency objective and an expectation-based decoder for precise continuous reconstruction.

Our main contributions are summarized as follows:

- We propose a Masked Diffusion Framework for time series imputation in which missingness enters the model as a symbol the sequence reads, giving a structural separation between missing and observed representations.
- We introduce a discretization pipeline adapted to masked diffusion imputation, combining *Stochastic Discretization* and *Ordinal-Aware Soft Labeling*, which preserves ordinal relationships and is unbiased in expectation.
- We employ an Expectation-based Unmasking strategy and Spectral Consistency Regularization, allowing MDTIM to reconstruct continuous time series precisely. Our method consistently outperforms state-of-the-art deterministic and generative baselines across diverse benchmarks.

2 Related Work

2.1 Time Series Imputation

Time series data are often partially observed due to sensor faults, irregular sampling, or data-collection constraints [35]. Naive deletion or mean/zero imputation can bias estimation and degrade downstream tasks [16], so time-series imputation aims to recover missing values by leveraging temporal dependencies, cross-variable correlations, and informative missingness patterns [4, 6].

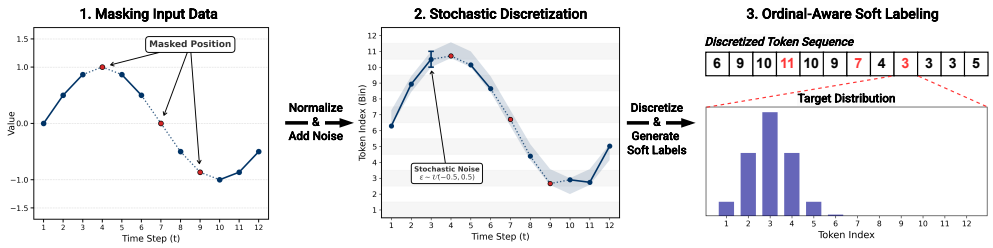


Figure 2: **Overview of the proposed Stochastic Discretization and Ordinal-Aware Soft Labeling framework.** The process consists of three stages: (1) *Masking Input Data*, where missing values are identified; (2) *Stochastic Discretization*, which applies instance-adaptive normalization and injects stochastic noise $\epsilon \sim \mathcal{U}(-0.5, 0.5)$ to bridge the continuous-discrete gap; and (3) *Ordinal-Aware Soft Labeling*, which maps continuous values to discrete tokens while generating soft target distributions to preserve ordinal semantic relationships.

Methodologically, deep learning approaches have progressed from RNN-based architectures with masking [4, 3] to Transformer-based models that leverage self-attention to capture long-range dependencies; SAITS [6], in particular, achieves strong performance via Diagonally-Masked Self-Attention. Beyond deterministic models, diffusion-based models such as CSDI [33] and SSSD [20] produce probabilistic imputations by conditioning on observed data and iterative denoising. SSSD further incorporates structured state space models to better capture long-range temporal dependencies. In this formulation an absent value cannot be represented by Gaussian corruption, so missingness has to be supplied from outside as a mask, and the missing values are recovered only after the full reverse chain. In masked diffusion, erasure is itself missingness, so a partially observed series is already a valid state of the forward process and the model fits $p_\theta(x_0 \mid x_t)$ directly.

2.2 Masked Diffusion Model

Discrete diffusion models extend the diffusion paradigm to categorical state spaces [13]. Among them, D3PM [2] introduced an absorbing-state formulation, where tokens are progressively replaced by a special `[MASK]` symbol and recovered through iterative denoising. MDLM [28] adapted this framework to language modeling via a continuous-time weighted cross-entropy objective, demonstrating its effectiveness as a non-autoregressive alternative for conditional text generation. Building on MDLM, subsequent works have improved robustness through unmasking-order-aware training [15] and conditional fidelity by treating observed tokens as fixed anchors during sampling [17, 27]. Despite these advances, masked diffusion has been studied almost exclusively in discrete modalities such as NLP, and its extension to time series remains underexplored due to the fundamental gap between discrete tokens and continuous, ordinal temporal signals.

3 Discrete Representation for Time-Series Modeling

Although the Masked Diffusion Model (MDM) offers a promising framework for imputation, directly applying it to time series presents a structural challenge: MDMs are inherently designed for discrete state spaces (e.g., text tokens), whereas time series data consist of continuous numerical values. To bridge this gap, we introduce a unified discretization framework illustrated in Figure 2, which combines *Stochastic Discretization* to mitigate quantization error and *Ordinal-Aware Soft Labeling* to preserve the temporal order of the original signals within the discrete vocabulary.

3.1 Stochastic Discretization

Instance-Adaptive Normalization. To stabilize local statistics against distribution shifts, we normalize each input window $x \in \mathbb{R}^{T \times C}$ into $\tilde{x}$. Specifically, the normalization statistics are computed solely using the observed values (excluding masked positions), ensuring that there is no data-leakage in the training and imputation process.

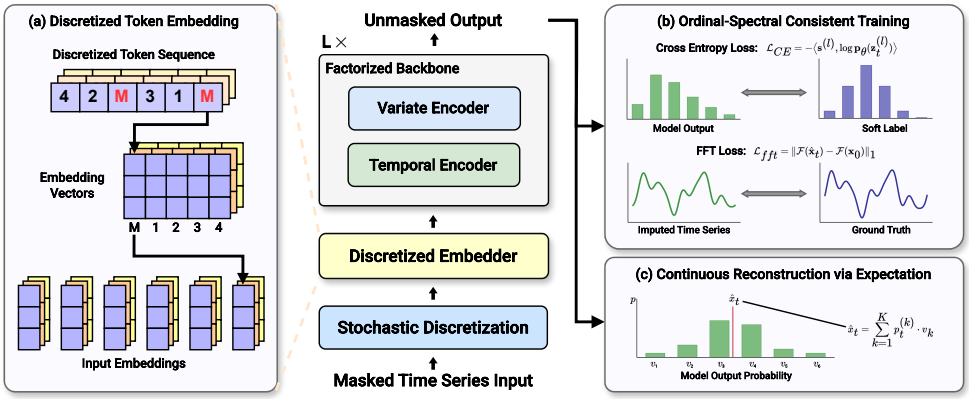


Figure 3: **Overview of the Masked Diffusion Time-series Imputation Model (MDTIM) framework.** The architecture operates on the discretized input through three key phases: **(a) Discretized Token Embedding:** The discrete tokens generated by the stochastic tokenizer are mapped into dense vector representations. **(b) Ordinal-Spectral Consistent Training:** The backbone is trained with a dual objective combining Ordinal-Aware Soft Cross-Entropy and Spectral Consistency Loss. This ensures spectral consistency across global frequencies while maintaining local ordinal accuracy. **(c) Continuous Reconstruction via Expectation:** Finally, continuous values are recovered by computing the probability-weighted expectation of the predicted token distribution, enabling precise dense reconstruction from discrete outputs.

Stochastic Token Generation. We map the normalized values to a discrete vocabulary of size K. Specifically, we first project the continuous value $\tilde{x}_{t,c}$ onto the grid coordinates and inject uniform noise $\epsilon \sim \mathcal{U}(-0.5, 0.5)$ before rounding:

$$c_{t,c} = \frac{\tilde{x}_{t,c} - v_{\min}}{v_{\max} - v_{\min}} \cdot (K-1) + 1, \quad z_{t,c} = \lfloor c_{t,c} + \epsilon \rceil, \tag{1}$$

where $\lfloor \cdot \rceil$ denotes the nearest integer function. This stochastic mechanism preserves information in expectation. For instance, a coordinate $c_{t,c} = 3.6$ is assigned to token 4 with 60% probability and token 3 with 40%, ensuring $\mathbb{E}[z_{t,c}] \approx c_{t,c}$. This allows the model to learn precise dynamics beyond the fixed grid resolution.

3.2 Ordinal-Aware Soft Labeling

Treating discretized tokens as independent classes ignores their ordinal nature (e.g., token k is semantically closer to $k+1$ than $k+10$). To enforce continuity, we employ Ordinal-Aware Soft Labeling with a truncated Gaussian kernel.

Given a ground-truth token index $y \in \{1, \ldots, K\}$, the target probability s_i for the i-th class is defined as:

$$s_i = \begin{cases} \frac{1}{Z} \exp\left(-\frac{(i-y)^2}{\sigma^2}\right) & \text{if } |i-y| \le w, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where w is the truncation window size, σ controls the smoothness in units of token bins, and Z is the normalization constant. We use $w = 2$ and $\sigma = 1$ for every dataset and missing rate, fixed a priori (Appendix C.2.1). Crucially, we assign zero probability to the `[MASK]` token ($s_0 = 0$) and distant bins. This concentrates probability mass on valid ordinal neighbors while preserving the structural separation of the mask state.

4 Masked Diffusion Framework

Building upon the discrete representation of time series data described in Section 3, we introduce the **Masked Diffusion Time-series Imputation Model (MDTIM)**, a framework designed to model the

joint distribution of multivariate time series and incorporate training techniques of masked diffusion models. Figure 3 illustrates the overall pipeline.

4.1 Factorized Temporal-Variate Backbone

To effectively model the joint distribution of multivariate time series, we employ a Factorized Temporal-Variate Transformer based on the Diffusion Transformer (DiT) proposed by Peebles and Xie [25]. This architecture processes the input through alternating factorization of the time (T) and feature (C) axes.

Discretized Embedding. Given the masked input indices $\mathbf{z}_t \in \{0, \dots, K\}^{B\times T\times C}$ at diffusion step t, we utilize a distinct embedding matrix $\mathbf{E}_c$ for each channel c to preserve semantic orthogonality. The input is projected to an initial hidden state $\mathbf{H}^0 \in \mathbb{R}^{B\times T\times C\times D}$, where D is the hidden dimension. The `[MASK]` token (index 0) receives its own embedding, so masked positions are present as keys in every attention block and observed positions can read where the gaps are, rather than merely being prevented from attending to them.

Interleaved DiT Blocks. The core network consists of L layers, each sequentially processing the temporal and feature axes via two DiT blocks conditioned on the diffusion timestep t, with Rotary Position Embeddings (RoPE) [32] encoding relative positions. The Temporal block reshapes $\mathbf{H}^{(l-1)}$ to $\mathbb{R}^{(B\cdot C)\times T\times D}$ and captures time-dependencies along the temporal axis:

$$\mathbf{H}'^{(l)} = \text{TemporalEncoder}(\mathbf{H}^{(l-1)}, t). \tag{3}$$

The Variate block then reshapes $\mathbf{H}'^{(l)}$ to $\mathbb{R}^{(B\cdot T)\times C\times D}$ and symmetrically models cross-channel correlations:

$$\mathbf{H}^{(l)} = \text{VariateEncoder}(\mathbf{H}'^{(l)}, t). \tag{4}$$

The final output is projected to vocabulary size $K_{\text{out}} = 1.5K$, covering the extended range $[-1.5, 1.5]$. Since normalization uses the observed entries only, a missing value can fall outside the range they span, and the wider output grid lets the model place it there. The soft label of Eq. 2 keeps its mass within w bins of the true token, so such extreme predictions stay extreme rather than drifting toward the center of the grid.

4.2 Ordinal-Spectral Consistent Training Objective

To simultaneously ensure local reconstruction fidelity and global temporal coherence, we formulate a dual-domain objective. This combines a discrete diffusion loss adapted for ordinal continuity with an auxiliary spectral regularization term.

Ordinal-Aware Masked Diffusion. We adopt the continuous-time training paradigm of Masked Diffusion Language Models (MDLM) [2, 28], optimizing a weighted variational lower bound (NELBO). Standard MDLM employs a simple cross-entropy loss against one-hot targets, which treats all incorrect tokens equally, failing to capture the ordinal magnitude inherent in time-series data.

To address this, we employ the *Ordinal-Aware Soft Labels* derived in Sec. 3.2 as the optimization targets. For a set of masked indices $\mathcal{M}_t$ at timestep t, the diffusion loss is formulated as:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_t \left[w(t) \sum_{l\in\mathcal{M}_t} -\left\langle \mathbf{s}^{(l)}, \log \mathbf{p}_\theta(\mathbf{z}_t^{(l)}) \right\rangle \right], \tag{5}$$

where $\log \mathbf{p}_\theta(\mathbf{z}_t^{(l)}) \in \mathbb{R}^K$ is the predicted log-probability vector, and $\mathbf{s}^{(l)} \in \mathbb{R}^K$ denotes the soft target distribution. The detailed formulation of time-dependent weight $w(t)$ and the corresponding noise schedule are detailed in Appendix C.2.2.

Replacing the one-hot target $\mathbf{e}_y$ with $\mathbf{s}^{(l)}$ can be viewed as a smooth relaxation of the standard MDLM objective. Writing $\mathbf{s}^{(l)} = (1-\epsilon)\mathbf{e}_y + \epsilon\, \mathbf{u}^{(l)}$ with $\epsilon = 1 - s_y^{(l)}$, the per-token loss decomposes as:

$$-\langle \mathbf{s}^{(l)}, \log \mathbf{p}_\theta \rangle = (1-\epsilon)\mathcal{L}_{\text{NELBO}}^{(l)} + \epsilon D_{\text{KL}}(\mathbf{u}^{(l)} \| \mathbf{p}_\theta) + \text{const}, \tag{6}$$

so that the original NELBO is recovered when $\mathbf{s}^{(l)}$ collapses to a one-hot ($\epsilon \to 0$) and otherwise encourages the predicted distribution to align with the ordinal neighborhood of y. The detailed derivation is provided in Appendix B.

Spectral Consistency Regularization. While the token-wise objective ensures local accuracy, it may neglect global temporal correlations and frequency structures. To strictly enforce global coherence, we incorporate a *Spectral Consistency Loss*, adapting the frequency-domain regularization proposed in DiffusionTS [41]. During training, we compute a continuous estimate of the full sequence, $\hat{\mathbf{x}}_t$, by applying expectation-based decoding (Sec. 4.3). We then minimize the L_1 distance between the Fourier representations of the reconstructed and ground-truth signals:

$$\mathcal{L}_{\text{FFT}} = \mathbb{E}_t \left[\|\mathcal{F}(\hat{\mathbf{x}}_t) - \mathcal{F}(\mathbf{x}_0)\|_1 \right], \tag{7}$$

where $\mathcal{F}(\cdot)$ denotes the Fast Fourier Transform (FFT).

Therefore, the total training objective is a weighted combination: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{FFT}}$.

4.3 Continuous Reconstruction via Expectation

Since *Stochastic Discretization* injects random noise into the input, the model output depends on the specific noise realization. To obtain a robust estimate, we run inference M times (e.g., $M = 10$) with independent noise samples and average the predicted probability distributions, $\bar{p}_{t,c} = \frac{1}{M} \sum_{m=1}^{M} p_{t,c}^{(m)}$. The continuous value is then recovered as the expectation over the averaged distribution:

$$\hat{x}_{t,c} = \sum_{k=1}^{K} \bar{p}_{t,c}^{(k)} \cdot v_k, \tag{8}$$

where v_k is the center value of the k-th bin within $[-1.5, 1.5]$. The result is then denormalized using the instance-wise statistics to recover the original scale.

5 Experiments

We empirically present a comprehensive evaluation of the proposed Masked Diffusion Time-series Imputation Model (MDTIM), comparing it against state-of-the-art deterministic and generative baselines. Our experiments focus on imputation accuracy, robustness to complex missing patterns, and computational efficiency.

5.1 Experimental Setup

We evaluate our method on widely used benchmarks: Energy, ETTh, Weather, and the synthetic Sine dataset. To rigorously test the models, we simulate data corruption using two distinct mechanisms: *Uniform* masking, where observations are dropped randomly to mimic independent failures, and *Geometric* masking, which removes consecutive time steps to simulate prolonged sensor malfunctions or transmission errors. For both scenarios, we assess performance across varying missing rates of 30%, 50%, and 70%.

We compare MDTIM against a comprehensive suite of 19 baselines spanning various architectural paradigms. These include RNN-based methods such as BRITS[3], MRNN[40], and GRU-D[4]; and Transformer-based models including the canonical Transformer[34], Informer[43], PatchTST[24], SAITS[6], and ImputeFormer[23]. Also, we include recent high-performance convolutional and linear architectures: TimesNet[37], SCINet[19], DLinear[42], FILM[44], and FreTS[39]. Finally, we evaluate against probabilistic frameworks including GP-VAE[9], US-GAN[22], and the diffusion-based CSDI[33] and FGTI[38]. We evaluate each trained model under three random mask seeds and report the averaged Mean Absolute Error (MAE). Variation across training seeds is reported in Appendix D.4.

5.2 Imputation Performance

Main Results. As shown in Table 1, MDTIM achieves the lowest MAE on Energy, ETTh, and Weather across both Uniform and Geometric missing patterns. FGTI [38], a frequency-aware diffusion model, attains the best results on the synthetic Sine dataset whose signal is dominated by a few well-defined frequencies, while MDTIM trails by only a marginal gap (e.g., 0.002 vs. 0.001 at 30% Uniform). On the more complex real-world ETTh dataset, however, FGTI degrades substantially (0.218 vs. 0.127 at 30% Uniform), suggesting that frequency-domain priors generalize less reliably

Table 1: Quantitative comparison of multivariate time-series imputation performance ($L = 48$). We report the MAE of MDTIM and baselines averaged over 3 random mask seeds at missing ratios of 30% and 70% (50% in Appendix D.1). The best results are highlighted in **bold**.

Dataset		Energy				ETTh				Weather				Sine			
Missing Type		Uniform		Geometric		Uniform		Geometric		Uniform		Geometric		Uniform		Geometric	
	Model	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%
RNN	BRITS	0.246	0.379	0.329	0.366	0.194	0.299	0.227	0.292	0.050	0.073	0.058	0.071	0.010	0.021	0.020	0.019
	MRNN	1.086	1.143	1.087	1.143	0.743	0.782	0.751	0.781	0.651	0.661	0.654	0.659	0.169	0.170	0.169	0.170
	GRUD	0.364	0.457	0.426	0.445	0.310	0.394	0.348	0.383	0.104	0.369	0.164	0.350	0.008	0.014	0.013	0.012
Transformer	Transformer	0.323	0.401	0.354	0.393	0.168	0.252	0.183	0.247	0.076	0.070	0.083	0.068	0.039	0.046	0.042	0.045
	Informer	0.344	0.379	0.371	0.373	0.207	0.302	0.226	0.296	0.047	0.059	0.052	0.058	0.027	0.039	0.036	0.037
	PatchTST	0.586	0.529	0.528	0.517	0.202	0.272	0.231	0.262	0.074	0.082	0.081	0.077	0.015	0.017	0.019	0.015
	SAITS	0.177	0.212	0.204	0.208	0.140	**0.210**	0.152	0.206	0.045	0.049	0.050	0.048	0.026	0.026	0.030	0.025
	ImputeFormer	0.066	0.219	0.113	0.194	0.146	0.245	0.165	0.235	0.049	0.171	0.090	0.149	0.006	0.041	0.012	0.037
CNN	TimesNet	0.617	0.818	0.689	0.808	0.593	0.719	0.628	0.717	0.211	0.351	0.225	0.348	0.167	0.220	0.189	0.217
	SCINet	0.557	0.571	0.546	0.565	0.238	0.323	0.261	0.317	0.074	0.089	0.086	0.087	0.019	0.031	0.026	0.029
Linear	DLinear	0.795	0.527	0.684	0.517	0.379	0.445	0.373	0.435	0.372	0.223	0.345	0.220	0.070	0.053	0.062	0.051
	FiLM	0.877	0.520	0.791	0.512	0.696	0.627	0.707	0.623	0.380	0.209	0.360	0.206	0.127	0.109	0.132	0.108
	FreTS	0.157	0.219	0.225	0.206	0.222	0.299	0.262	0.286	0.071	0.080	0.080	0.074	0.068	0.068	0.075	0.065
Generative	GPVAE	0.476	0.730	0.501	0.727	0.333	0.449	0.369	0.440	0.158	0.255	0.172	0.252	0.152	0.162	0.153	0.162
	USGAN	0.264	0.412	0.330	0.401	0.208	0.302	0.240	0.295	0.087	0.122	0.101	0.118	0.014	0.029	0.026	0.026
	CSDI	0.094	0.139	0.110	0.136	0.160	0.244	0.177	0.240	0.039	0.049	0.044	0.048	0.003	0.004	0.004	0.004
	FGTI	0.050	0.100	0.065	0.097	0.218	0.349	0.293	0.328	0.038	0.051	0.046	0.049	**0.001**	**0.003**	**0.002**	**0.003**
	MDTIM (Ours)	**0.044**	**0.085**	**0.053**	**0.082**	**0.127**	0.211	**0.146**	**0.205**	**0.032**	**0.044**	**0.036**	**0.043**	0.002	0.004	0.003	**0.003**

to irregular and non-stationary signals. Compared to the diffusion-based CSDI [33], MDTIM consistently yields lower MAE across all datasets, owing to our *Expectation-based Reconstruction* that recovers continuous values deterministically from the predicted token distribution and mitigates the sampling variance inherent in continuous diffusion. MDTIM also exhibits notable robustness under geometric masking, where most baselines suffer significant degradation; for instance, on Energy at 30%, MDTIM increases only from 0.044 to 0.053, indicating that the masked diffusion objective encourages inference of global temporal structure rather than reliance on local interpolation.

Table 2: Robustness analysis on extended sequence lengths ($L \in \{96, 192\}$). We report the MAE averaged over 3 random mask seeds.

Length	96						192					
Missing Type	Uniform			Geometric			Uniform			Geometric		
Model	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
BRITS	0.191	0.229	0.287	0.220	0.250	0.280	0.192	0.230	0.287	0.222	0.250	0.280
SAITS	0.139	0.162	0.204	0.150	0.172	0.200	0.145	0.162	0.198	0.156	0.171	0.195
CSDI	0.149	0.177	0.222	0.162	0.187	0.218	0.170	0.199	0.245	0.183	0.209	0.241
MDTIM	**0.124**	**0.152**	**0.199**	**0.138**	**0.163**	**0.194**	**0.123**	**0.148**	**0.190**	**0.135**	**0.157**	**0.186**

Scalability to Long-Term Dependencies. To verify that MDTIM captures long-range temporal dependencies, we extend the evaluation to longer sequence lengths $L \in \{96, 192\}$. As shown in Table 2, the performance advantage of MDTIM grows with the sequence length: while MDTIM is comparable to SAITS at $L = 48$, the gap widens substantially at longer horizons. At $L = 192$ with 30% Uniform missing, MDTIM achieves an MAE of 0.123, considerably lower than BRITS (0.192) and SAITS (0.145), indicating that MDTIM captures global context without suffering from the attention dilution often observed in baselines.

Probabilistic Imputation Performance Beyond point estimation accuracy, we evaluate the quality of the predictive distributions generated by the models using the Continuous Ranked Probability Score (CRPS). CRPS estimates the calibration (reliability) and sharpness (precision) of the probabilistic inference.

As shown in Table 3, MDTIM consistently outperforms CSDI, the diffusion-based baseline, across all datasets and missing scenarios. On the highly periodic Sine dataset, MDTIM achieves a near-constant CRPS regardless of the missing rate, indicating that it captures deterministic periodicity with high confidence. While continuous diffusion models often exhibit excessive variance due to

Table 3: Quantitative comparison of probabilistic imputation performance (CRPS, $L = 48$). We compare MDTIM with the diffusion-based baseline CSDI. Lower is better.

		Energy		ETTh		Sine	
Missing Type	**Rate**	CSDI	MDTIM	CSDI	MDTIM	CSDI	MDTIM
Uniform	30%	0.0629	**0.0429**	0.1120	**0.0863**	0.0023	**0.0020**
	50%	0.0747	**0.0541**	0.1356	**0.1078**	0.0025	**0.0020**
	70%	0.0927	**0.0771**	0.1748	**0.1520**	0.0030	**0.0020**
Geometric	30%	0.0732	**0.0547**	0.1262	**0.1017**	0.0027	**0.0020**
	50%	0.0818	**0.0626**	0.1460	**0.1197**	0.0028	**0.0020**
	70%	0.0905	**0.0733**	0.1708	**0.1462**	0.0029	**0.0020**

sampling noise, MDTIM mitigates such ambiguity through its expectation-based decoding over discrete distributions. Furthermore, on the complex real-world ETTh dataset, MDTIM achieves significantly lower CRPS (e.g., 0.0863 vs. 0.1120 at 30% uniform missing), demonstrating that our *Ordinal-Aware Soft Labeling* guides the model to produce well-calibrated uncertainty estimates.

Imputation under Naturally Missing Real-World Data. To assess MDTIM under realistic missingness patterns, we further evaluate on PhysioNet 2012 [29], a multivariate clinical time series benchmark with $\sim$80% natural missingness due to irregular ICU sampling, and apply additional uniform and geometric masking on top of the existing missingness. Since about 80% of entries are naturally missing, the observed values are too sparse to form a reliable spectrum of the target, so we disable the spectral consistency loss for this dataset; all other settings remain identical.

As shown in Table 4, MDTIM achieves the lowest MAE across all settings, outperforming SAITS by a clear margin. Notably, FGTI exhibits the largest degradation here, even underperforming BRITS. This indicates that frequency-domain priors become unreliable for highly irregular signals where dominant spectral components are weak or unstable, whereas our ordinal-aware discrete formulation generalizes effectively to real-world sparse and irregular observations.

Table 4: Imputation performance on PhysioNet 2012.

PhysioNet2012	**Uniform**			**Geometric**		
Model	30%	50%	70%	30%	50%	70%
BRITS	0.333	0.364	0.408	0.356	0.379	0.405
SAITS	0.283	0.314	0.360	0.298	0.324	0.358
CSDI	0.303	0.332	0.376	0.320	0.342	0.374
ImputeFormer	0.289	0.329	0.389	0.311	0.343	0.388
FGTI	0.338	0.376	0.425	0.370	0.397	0.423
MDTIM (Ours)	**0.236**	**0.279**	**0.339**	**0.257**	**0.291**	**0.337**

5.3 Computational Efficiency and Model Scalability

Diffusion-based methods often trade efficiency for accuracy. To examine this trade-off, we compare MDTIM against representative baselines at two model scales on the Energy dataset with uniform missing (Table 5).

Table 5: Imputation performance and efficiency on Energy (uniform missing). Inference time is measured on the full test set (MDTIM: M=10 noise samples; CSDI: T=50 diffusion steps).

Scale	**Small**					**Large**				
Model	**30%**	**50%**	**70%**	**Params (M)**	**Time (s)**	**30%**	**50%**	**70%**	**Params (M)**	**Time (s)**
BRITS	0.246	0.296	0.379	2.18	4.88	0.245	0.293	0.371	8.55	4.96
SAITS	0.212	0.216	0.240	25.25	0.41	0.177	0.185	0.212	88.24	0.79
CSDI	0.094	0.112	0.139	1.19	401.69	0.074	0.093	0.128	4.49	894.76
MDTIM (Ours)	**0.046**	**0.065**	**0.092**	0.65	4.05	**0.044**	**0.061**	**0.085**	8.33	12.63

MDTIM addresses the latency bottleneck of continuous diffusion: while CSDI requires up to 894s at the Large scale, the Small MDTIM completes inference in 4.05s while halving the MAE (0.074→0.046). It is also notably parameter-efficient—the 0.65M Small variant surpasses the 88.24M Large SAITS, suggesting that the masked diffusion objective captures temporal dynamics more densely than scaling deterministic Transformers.

5.4 Downstream Forecasting Task

To assess whether the imputation accuracy of MDTIM translates to downstream tasks, we evaluate forecasting performance when the input window contains missing values. Each imputation model reconstructs the partially observed input, after which a fixed PatchTST [24] forecaster, pre-trained on clean sequences, predicts the subsequent H steps. Since the forecaster is identical across all settings, differences in forecasting MAE directly reflect the quality of imputation.

Table 6: Forecasting MAE on Energy (L=48) using a fixed PatchTST predictor with imputed inputs from each model.

Energy	**Forecast Horizon**			
Imputer	H=4	H=6	H=8	H=12
SAITS	0.187	0.207	0.224	0.254
CSDI	0.171	0.190	0.208	0.237
MDTIM (Ours)	**0.138**	**0.153**	**0.170**	**0.200**

As shown in Table 6, MDTIM yields substantially lower forecasting error than both baselines across all horizons on Energy, reducing MAE by 19–26% over SAITS and 16–19% over CSDI. These results indicate that the imputation accuracy of MDTIM propagates to downstream forecasting, and that high-fidelity imputation is essential for tasks that rely on partially observed inputs.

5.5 Ablation Studies

To rigorously evaluate the proposed framework, we conduct ablation studies on the mechanism by which missingness enters the model, the contribution of our expectation-based unmasking and soft labeling strategies, the spectral consistency term, and the sensitivity to the vocabulary size.

Missingness Mechanisms. As argued in the introduction, what matters is how missingness enters the network, and we test this with two controls that keep the backbone and training identical. The first replaces the `[MASK]` token by an attention mask: missing positions still read the observed ones, but they are removed from the keys of every attention block, so no observed position can see where the gaps are. Table 7 shows that at 70% missing this variant is worse on every dataset under both patterns, with the largest drop on ETTh, where the shape of a gap carries the most information about the values inside it. The second control, MDTIM (Cont), keeps the masked diffusion training but operates on continuous values with a placeholder indicator, so the missing entries share the embedding space of the observations; Figure 4 in Appendix D.2 shows it trailing the discrete model at every missing rate on Energy. Both mechanisms tell the model which entries are missing. The difference is that only the `[MASK]` token lets the rest of the sequence condition on that fact.

Table 7: MAE at 70% missing when the `[MASK]` token is replaced by an attention mask that hides missing positions from observed ones.

	Uniform		**Geometric**	
Dataset	Attn. mask	`[MASK]`	Attn. mask	`[MASK]`
ETTh	0.234	**0.211**	0.225	**0.205**
Energy	0.087	**0.085**	0.084	**0.082**
Weather	0.047	**0.044**	0.046	**0.043**
Sine	0.0038	**0.0035**	0.0036	**0.0034**

Table 8: Ablation study on unmasking and labeling strategies. All results are reported as MAE averaged over 3 random mask seeds under the uniform missing scenario.

Components		**ETTh**			**Energy**			**Sine**		
Exp. Unmasking	Soft-Label	30%	50%	70%	30%	50%	70%	30%	50%	70%
✗	✗	0.134	0.166	0.229	0.053	0.073	0.101	0.0071	0.0072	0.0075
✗	✓	0.131	0.162	0.222	0.056	0.079	0.115	0.0069	0.0070	0.0074
✓	✗	0.129	0.159	0.212	0.044	**0.060**	0.085	0.0026	0.0030	0.0036
✓	✓	**0.127**	**0.157**	**0.211**	**0.044**	0.061	**0.085**	**0.0022**	**0.0026**	**0.0035**

Impact of Unmasking and Labeling Strategies. Table 8 summarizes the effect of our two proposed components: *Expectation-based Unmasking* and *Ordinal-Aware Soft Labeling*. Expectation-based Unmasking consistently reduces error across all datasets, confirming that computing the expected value over discrete bins mitigates quantization error inherent in Argmax selection. The contribution of Soft Labeling, in contrast, scales with the structural regularity of the data. On the highly periodic Sine dataset, Soft Labeling yields a substantial additional gain on top of Expectation-based Unmasking, as ordinal-aware guidance aligns naturally with the smooth, predictable transitions of periodic signals. A similar but milder gain is observed on ETTh, whose complex temporal dynamics still benefit from ordinal regularization under higher uncertainty. On Energy, where the dynamics are less structured, Soft Labeling offers no further improvement beyond Expectation-based Unmasking. These results

indicate that Soft Labeling is most effective when the underlying signal exhibits clear ordinal or periodic structure that ordinal-aware guidance can exploit.

Impact of Spectral Consistency. Table 9 removes the spectral term (λ=0) with all other components in place, on ETTh, the most broadband of our benchmarks. At L=48 the two variants are tied at 30% and 50% and λ=0 is marginally ahead at 70%, so on a short window the term has little to work with. This is a matter of frequency resolution rather than a harmful objective: the resolution of the FFT is set by the window length, and as L grows to 192 the spectral term pulls ahead at every missing rate. The same reasoning governs PhysioNet, where the observed entries are too sparse to form a reliable spectrum and we set λ=0. Even without the spectral term, MDTIM remains ahead of every baseline in Table 1, so the reported gains rest on the discrete formulation rather than on the auxiliary loss.

Table 9: MAE on ETTh with and without the spectral term (λ) across window lengths L, uniform missing.

ETTh		**Length (L)**		
Rate	λ	48	96	192
30%	0	**0.127**	0.125	0.126
	1	**0.127**	**0.124**	**0.123**
50%	0	**0.157**	**0.152**	0.151
	1	**0.157**	**0.152**	**0.148**
70%	0	**0.208**	**0.196**	0.191
	1	0.211	0.199	**0.190**

Sensitivity to Vocabulary Size (K). We further investigate the impact of the vocabulary size K on imputation accuracy (Table 10), which exhibits a trade-off between discretization resolution and classification complexity. A small vocabulary (K=20) suffers from high quantization error, failing to capture fine-grained fluctuations. Conversely, K=60 slightly degrades performance, as a larger vocabulary increases the difficulty of the discrete classification task. K=40 achieves the best balance, providing sufficient resolution to approximate the continuous signal while remaining tractable to optimize.

Table 10: MAE on ETTh with varying vocabulary size (K).

ETTh	**Uniform**			**Geometric**		
Bins	30%	50%	70%	30%	50%	70%
20	0.138	0.166	0.217	0.157	0.181	0.211
40	**0.127**	**0.157**	**0.211**	**0.146**	**0.173**	**0.205**
60	0.129	0.160	0.214	0.148	0.175	0.208

6 Conclusion and Limitations

We proposed MDTIM, a framework that adapts the masked diffusion training paradigm to continuous time series imputation. By introducing Stochastic Discretization and Ordinal-Aware Soft Labeling, MDTIM bridges the gap between continuous dynamics and categorical tokenization, while the Expectation-based Unmasking strategy enables precise continuous reconstruction. Extensive experiments demonstrate that MDTIM consistently outperforms both deterministic and generative baselines in reconstruction accuracy and robustness across diverse missing scenarios, while requiring substantially less inference time than continuous diffusion baselines.

Limitations and Future Work. While MDTIM demonstrates strong performance across diverse missing scenarios, several aspects warrant further investigation. First, Stochastic Discretization is unbiased in expectation but, like any quantization, lossy for an individual sample, and a data-adaptive discretization scheme that automatically selects the resolution would further improve generalization. Second, our spectral consistency regularization assumes the presence of stable frequency components and is therefore disabled on benchmarks dominated by irregular sampling such as PhysioNet. Designing an adaptive frequency objective that modulates its influence based on local signal regularity is a promising direction. Third, the per-channel embedding tables cost $O(CKD)$ parameters, which grows linearly with the number of variates; a channel-shared alternative is examined in Appendix D.3. Finally, our evaluation focuses on standard imputation benchmarks of moderate length; extending MDTIM to extremely long sequences and to multi-scale or non-stationary domains such as financial tick data remains for future work.

Acknowledgments

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2025-25420430). This work was also partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2024-00338859), by the Technology Innovation Program (No. RS-2026-25544725) funded by the Ministry of Trade, Industry and Resources (MOTIR, Korea), and

by the Ministry of Trade, Industry and Resources (MOTIR), Republic of Korea, in 2026, under the project "Development of Core Foundation Model and AI Agent Technology for Thermal-Process Manufacturing (TP-MFM)" (No. TPAI2616). All four grants contributed equally to this work.

References

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

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

[4] 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.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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

[7] Wenjie Du, Jun Wang, Linglong Qian, Yiyuan Yang, Zina Ibrahim, Fanxing Liu, Zepu Wang, Haoxin Liu, Zhiyuan Zhao, Yingjie Zhou, et al. Tsi-bench: Benchmarking time series imputation. *arXiv preprint arXiv:2406.12747*, 2024.

[8] Gal Fadlon, Idan Arbiv, Nimrod Berman, and Omri Azencot. A diffusion model for regular time series generation from irregular data with completion and masking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=M9JmlA6Cgf`.

[9] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020.

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35: 8633–8646, 2022.

[13] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forr, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022.

[15] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025.

[16] SeungHyun Kim, Hyunsu Kim, Eunggu Yun, Hwangrae Lee, Jaehun Lee, and Juho Lee. Probabilistic imputation for time-series classification with missing data. In *International Conference on Machine Learning*, pages 16654–16667. PMLR, 2023.

[17] Hyukhun Koh, Minha Jhang, Dohyung Kim, Sangmook Lee, and Kyomin Jung. Conditional [mask] discrete diffusion language model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8910–8934, 2025.

[18] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[19] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

[20] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured atate apace models. *Transactions on machine learning research*, pages 1–36, 2023.

[21] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[22] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8983–8991, 2021.

[23] Tong Nie, Guoyang Qin, Wei Ma, Yuewen Mei, and Jian Sun. Imputeformer: Low rankness-induced transformers for generalizable spatiotemporal imputation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2260–2271, 2024.

[24] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Jbdc0vTOcol`.

[25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[27] Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored diffusion language model. *Advances in Neural Information Processing Systems*, 2025.

[28] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[29] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pages 245–248. IEEE, 2012.

[30] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[32] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[33] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Jun Wang, Wenjie Du, Yiyuan Yang, Linglong Qian, Wei Cao, Keli Zhang, Wenjia Wang, Yuxuan Liang, and Qingsong Wen. Deep learning for multivariate time series imputation: a survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/1187. URL https://doi.org/10.24963/ijcai.2025/1187.

[36] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[37] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

[38] Xinyu Yang, Yu Sun, Xiaojie Yuan, and Xinyang Chen. Frequency-aware generative models for multivariate time series imputation. *Advances in Neural Information Processing Systems*, 37: 52595–52623, 2024.

[39] 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. *Advances in Neural Information Processing Systems*, 36:76656–76679, 2023.

[40] Jinsung Yoon, William R Zame, and Mihaela Van Der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018.

[41] Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

[42] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[43] 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 *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[44] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022.

A Preliminaries on Diffusion Models

A.1 Diffusion Models

Diffusion models [30, 11, 31] are a class of latent-variable generative models that learn to invert a fixed noise-injection process. Since their introduction, they have become the dominant paradigm in many continuous-data generative tasks, achieving state-of-the-art results in image [5, 26, 14], audio [18], and video [12] synthesis, and have also been adopted as strong probabilistic models for time series forecasting and imputation [33, 1].

Notation. Let $x_0 \in \mathbb{R}^d$ denote a clean data sample drawn from the data distribution $q(x_0)$, and let $x_t \in \mathbb{R}^d$ for $t \in \{1, \dots, T\}$ denote its noisy latents along the forward chain. A noise schedule $\{\beta_t\}_{t=1}^T \subset (0,1)$ controls the amount of corruption at each step, with the per-step retention factor $\alpha_t = 1 - \beta_t$ and the cumulative retention factor $\bar{\alpha}_t = \prod_{s \le t} \alpha_s$; by construction $\bar{\alpha}_0 = 1$ and $\bar{\alpha}_T \approx 0$.

A.1.1 Diffusion Framework

Forward (Gaussian corruption). The forward process gradually corrupts a data sample into pure Gaussian noise over T steps:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I). \tag{9}$$

A useful property of this Markov chain is that its t-step marginal admits a closed form:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I),\ x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,\ \epsilon \sim \mathcal{N}(0, I). \tag{10}$$

That is, x_t at any timestep t can be sampled *directly* from x_0 without simulating the full chain, which makes per-step training tractable. As $t \to T$, the schedule is chosen so that $\bar{\alpha}_T \approx 0$ and x_T becomes approximately isotropic Gaussian, independent of x_0.

Reverse (learned denoising). The reverse process incrementally denoises noise back into the data manifold. While the marginal $q(x_{t-1}|x_t)$ is intractable, the *conditional* posterior $q(x_{t-1}|x_t, x_0)$ is Gaussian with closed-form mean and variance depending only on β_t and $\bar{\alpha}_t$ [11]. A neural network parameterizes the reverse kernel

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{11}$$

where the mean is typically expressed through a noise predictor $\epsilon_\theta(x_t, t)$:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\Big). \tag{12}$$

A.1.2 Conditional Diffusion

The diffusion framework introduced in Appendix A.1 defines an unconditional generative model. Many practical applications, including the time series imputation setting of this paper, instead require sampling from a conditional distribution $p(x_0 \mid y)$.

Let $\mathcal{O} \subseteq \{1, \dots, d\}$ index observed coordinates and $\mathcal{M} = \bar{\mathcal{O}}$ the missing ones. The goal is to sample

$$x_0^{\mathcal{M}} \sim p(x_0^{\mathcal{M}} \mid x_0^{\mathcal{O}}). \tag{13}$$

While external-conditioning paradigms such as classifier guidance [5] and classifier-free guidance [10] are effective when y is an external attribute such as a class label or text prompt, they do not apply directly to time series imputation, where the conditioning signal is a subset of the data itself. Diffusion-based imputation methods therefore commonly adopt the following setting [33, 1].

The forward process applies Gaussian corruption only to missing coordinates while preserving observed ones throughout all steps,

$$q(x_t^{\mathcal{M}}|x_0^{\mathcal{M}}) = \mathcal{N}(x_t^{\mathcal{M}}; \sqrt{\bar{\alpha}_t}\, x_0^{\mathcal{M}}, (1-\bar{\alpha}_t) I), \quad x_t^{\mathcal{O}} = x_0^{\mathcal{O}}\ \forall t. \tag{14}$$

The noise predictor receives the partially-noisy tensor $\tilde{x}_t$ (noisy on $\mathcal{M}$, clean on $\mathcal{O}$) together with the missingness mask $\mathcal{M}$ and the timestep t, $\epsilon_\theta(\tilde{x}_t, \mathcal{M}, t)$, giving the network direct access to which coordinates are observed and what their values are.

A.1.3 Training objective

Training maximizes a variational lower bound (ELBO) on $\log p_\theta(x_0)$. The bound decomposes into a sum of KL divergences between the tractable forward posterior $q(x_{t-1} \mid x_t, x_0)$ and the learned reverse kernel $p_\theta(x_{t-1} \mid x_t)$ at each step [11]. Reparameterizing the network as a noise predictor ϵ_θ and substituting the closed-form expression for x_t in Equation (10), the per-step KL reduces (up to a t-dependent reweighting) to the simple denoising loss of Ho et al. [11]:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}\{1,\dots,T\},x_0,\epsilon\sim\mathcal{N}(0,I)}[\|\epsilon - \epsilon_\theta(\sqrt{\bar\alpha_t}x_0 + \sqrt{1-\bar\alpha_t}\epsilon, t)\|_2^2]. \tag{15}$$

For the conditional setting in Appendix A.1.2, the noise predictor $\epsilon_\theta(\tilde x_t, \mathcal{M}, t)$ takes the partially noisy tensor and the missingness mask as input, and the loss is restricted to the missing coordinates [33, 1]:

$$\mathcal{L}_{\text{imp}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}\{1,\dots,T\},x_0,\epsilon\sim\mathcal{N}(0,I),\mathcal{M}}\left[\,\|\epsilon^{\mathcal{M}} - \epsilon_\theta^{\mathcal{M}}(\tilde x_t, \mathcal{M}, t)\|_2^2\,\right], \tag{16}$$

where the missingness mask $\mathcal{M}$ is sampled from a distribution of patterns during training, encouraging the model to generalize across missingness configurations.

A.2 Masked Diffusion Language Models

Masked Diffusion Language Models (MDLM) [28] adapt the diffusion framework of Appendix A.1 to discrete sequences by replacing Gaussian corruption with an absorbing-state Markov chain [2]. The role of pure noise $\mathcal{N}(0, I)$ in continuous diffusion is played here by a single distinguished symbol `[MASK]`: the forward process gradually replaces tokens with `[MASK]`, the prior at $t = 1$ is the all-`[MASK]` sequence, and the reverse process iteratively *unmasks* positions back to data tokens.

Notation. Let $t(i) = \frac{i}{T}$ and $s(i) = \frac{i-1}{T}$ for $i \in \{1, \dots, T\}$. Tokens take values in $\{0, 1, \dots, K\}$ where 0 denotes `[MASK]`. We represent tokens as one-hot vectors in $\mathcal{V} \coloneqq \{e_0, \dots, e_K\} \subset \{0, 1\}^{K+1}$ and denote the `[MASK]` one-hot by $m \coloneqq e_0$. For a length-L sequence $x = (x^l)_{l=1}^L$, let $z_t = (z_t^l)_{l=1}^L$ be the latent at time t. We write $\mathrm{Cat}(\cdot; \pi)$ for a categorical distribution and $\langle \cdot, \cdot \rangle$ for the dot product.

A.2.1 MDLM Framework

Forward (absorbing mask corruption). The forward process factorizes across token positions and replaces each token by `[MASK]` with probability $1 - \alpha_t$:

$$q(z_t \mid x) = \prod_{l=1}^{L} \mathrm{Cat}(z_t^l; \alpha_t x^l + (1 - \alpha_t)m), \tag{17}$$

with $\alpha_t \to 1$ as $t \to 0$ and $\alpha_t \to 0$ as $t \to 1$. We parameterize the noise schedule via $\alpha_t = e^{-\sigma(t)}$. Our experiments use the *log-linear* schedule defined by $\sigma(t) = -\log(1 - t)$, which yields a linear decay $\alpha_t = 1 - t$.

The defining property of this process is that `[MASK]` is an *absorbing* state: once $z_t^l = m$, it remains masked for all later times. Consequently, $q(z_t|x)$ depends only on which positions have been masked and the reverse process is naturally structured around iterative *unmasking*.

Reverse (learned denoising kernel). MDLM parameterizes the reverse kernel $p_\theta(z_s \mid z_t) = \prod_{l=1}^L p_\theta(z_s^l \mid z_t)$ via a predicted mixture token $x_\theta^l(z_t)$ over the K non-mask vocabulary entries. Two structural constraints mirror the absorbing forward process: (i) *zero-masking*, $\langle x_\theta^l(z_t), m\rangle = 0$, so the predictor never re-introduces `[MASK]` and (ii) *carry-over*, $z_t^l \neq m \Rightarrow z_s^l = z_t^l$, so already-unmasked tokens are preserved. As a result, the reverse transition is

$$p_\theta(z_s^l \mid z_t) = \begin{cases} \mathrm{Cat}(z_s^l; z_t^l), & z_t^l \neq m, \\ \mathrm{Cat}(z_s^l; \frac{\alpha_s - \alpha_t}{1-\alpha_t} x_\theta^l(z_t) + \frac{1-\alpha_s}{1-\alpha_t} m), & z_t^l = m. \end{cases} \tag{18}$$

For masked positions, the update is a mixture: with probability $\frac{\alpha_s - \alpha_t}{1-\alpha_t}$ the token is unmasked according to x_θ^l, and otherwise it remains `[MASK]`.

A.2.2 Conditional Masking

For text infilling, observed tokens can be treated as anchors and excluded from corruption. Let $\mathcal{A} \subseteq \{1, \dots, L\}$ be the anchored index set and $\bar{\mathcal{A}}$ its complement. Anchored positions are preserved exactly throughout the forward process while non-anchored positions undergo standard absorbing mask corruption:

$$q(z_t \mid x) = \prod_{l \in \mathcal{A}} \mathbb{I}[z_t^l = x^l] \prod_{l \in \bar{\mathcal{A}}} \mathrm{Cat}\big(z_t^l;\ \alpha_t x^l + (1 - \alpha_t) m\big)\,, \tag{19}$$

so that $z_t^l = x^l$ for all t and all $l \in \mathcal{A}$ (hard preservation).

The reverse kernel mirrors this structure. Anchored positions are copied from z_t unchanged, and only non-anchored masked positions are denoised via the standard reverse kernel in Equation (18). This guarantees that observed tokens remain unchanged throughout generation while the model fills in the unobserved ones.

This anchored formulation directly matches the imputation setting in Appendix A.1.2. The anchored set $\mathcal{A}$ corresponds to the observed coordinates $\mathcal{O}$, and the complement $\bar{\mathcal{A}}$ corresponds to the missing coordinates $\mathcal{M}$.

A.2.3 Training Objective

MDLM trains the denoising kernel by minimizing a negative ELBO (NELBO). Owing to the absorbing structure of the forward process, each per-step KL between the forward posterior and the learned reverse kernel reduces in closed form to a weighted cross-entropy over currently masked positions, $w_t \sum_{l \in \mathcal{M}_t} \mathrm{CE}(x_\theta^l(z_t), x^l)$ where $w_t = \frac{\alpha_t - \alpha_s}{1 - \alpha_t}$ and $\mathcal{M}_t = \{l \mid z_t^l = m\}$ is the set of currently masked positions. The denoiser $x_\theta^l(z_t)$ predicts the clean token at position l directly, paralleling the prediction of clean signal x_0 in continuous diffusion. Taking $T \to \infty$, the discrete-time sum converges to a continuous-time objective that samples $t \sim \mathcal{U}(0, 1)$ and uses the weight $w(t) = \frac{\alpha_t'}{1 - \alpha_t}$:

$$\mathcal{L}_{\mathrm{CT}}(x; \theta) = \mathbb{E}_t \left[\frac{\alpha_t'}{1 - \alpha_t} \sum_{l \in \mathcal{M}_t} \mathrm{CE}\big(x_\theta^l(z_t), x^l\big) \right], \tag{20}$$

which integrates the per-step loss over a continuous time grid and is the form actually used in our experiments.

With anchors (Section A.2.2), observed positions are excluded from the forward process and therefore never appear masked at any time t. The training loss is consequently computed only on non-anchor masked positions $\mathcal{M}_t^{\mathrm{anch}} = \{l \in \bar{\mathcal{A}} \mid z_t^l = m\}$, i.e., $\mathcal{M}_t$ is replaced by $\mathcal{M}_t^{\mathrm{anch}}$ in Equation (20). The denoiser is trained to reconstruct only the missing coordinates conditional on the observed ones, which is exactly the imputation objective $p_\theta(x_0^{\mathcal{M}} \mid x_0^{\mathcal{O}})$ targeted in Appendix A.1.2.

B Derivation of the Soft-Label Loss Decomposition

We provide the derivation of the decomposition stated in Eq. 6. For brevity, we suppress the position index l and the timestep weight $w(t)$, and consider a single per-token loss

$$\mathcal{L} = -\big\langle \mathbf{s}, \log \mathbf{p}_\theta \big\rangle = -\sum_{i=1}^{K} s_i \log p_\theta(i), \tag{21}$$

where $\mathbf{s}$ is the ordinal-aware soft label centered at the ground-truth index y (Eq. 2) and $\mathbf{p}_\theta \in \Delta^{K-1}$ is the predicted distribution.

Step 1: Decomposition of the soft label. Let $\epsilon := 1 - s_y \in [0, 1)$ denote the total mass that $\mathbf{s}$ assigns outside the ground-truth index y. We define

$$\mathbf{u} := \frac{\mathbf{s} - s_y \mathbf{e}_y}{1 - s_y}, \tag{22}$$

where $\mathbf{e}_y$ denotes the one-hot vector at position y. By construction, $u_y = 0$ and $\mathbf{u}$ is a valid probability distribution supported on the ordinal neighborhood $\{i : 0 < |i - y| \le w\}$. The soft label can then be written as a convex combination

$$\mathbf{s} = (1 - \epsilon)\, \mathbf{e}_y + \epsilon\, \mathbf{u}. \tag{23}$$

Step 2: Splitting the cross-entropy. Substituting Eq. 23 into $\mathcal{L}$ and applying the linearity of the inner product gives

$$\mathcal{L} = -\big\langle (1-\epsilon)\,\mathbf{e}_y + \epsilon\,\mathbf{u},\ \log \mathbf{p}_\theta \big\rangle \tag{24}$$

$$= -(1-\epsilon)\,\langle \mathbf{e}_y, \log \mathbf{p}_\theta \rangle - \epsilon\,\langle \mathbf{u}, \log \mathbf{p}_\theta \rangle. \tag{25}$$

The first term is the standard MDLM cross-entropy against the one-hot target,

$$-\langle \mathbf{e}_y, \log \mathbf{p}_\theta \rangle = -\log p_\theta(y) = \mathcal{L}_{\mathrm{NELBO}}, \tag{26}$$

recovering the per-token NELBO contribution of vanilla MDLM [28].

Step 3: KL form of the second term. The second term is the cross-entropy $H(\mathbf{u}, \mathbf{p}_\theta)$, which by the standard identity decomposes as

$$H(\mathbf{u}, \mathbf{p}_\theta) = -\langle \mathbf{u}, \log \mathbf{p}_\theta \rangle = D_{\mathrm{KL}}(\mathbf{u} \,\|\, \mathbf{p}_\theta) + H(\mathbf{u}), \tag{27}$$

where $H(\mathbf{u}) = -\sum_i u_i \log u_i$ is the entropy of $\mathbf{u}$ and depends only on $\mathbf{s}$ (i.e., independent of $\mathbf{p}_\theta$).

Step 4: Combining. Substituting back into Eq. 25 yields

$$\mathcal{L} = (1-\epsilon)\,\mathcal{L}_{\mathrm{NELBO}} + \epsilon\, D_{\mathrm{KL}}(\mathbf{u} \,\|\, \mathbf{p}_\theta) + \underbrace{\epsilon\, H(\mathbf{u})}_{\text{const}}, \tag{28}$$

which is exactly the decomposition in Eq. 6. Two limiting cases are worth noting:

- As the soft label sharpens ($\mathbf{s} \to \mathbf{e}_y$, equivalently $\epsilon \to 0$), the KL term vanishes and $\mathcal{L} \to \mathcal{L}_{\mathrm{NELBO}}$, recovering the vanilla MDLM objective.
- For $\epsilon > 0$, the additional non-negative term $\epsilon\, D_{\mathrm{KL}}(\mathbf{u} \,\|\, \mathbf{p}_\theta)$ penalizes probability mass placed outside the ordinal neighborhood of y, since $\mathbf{u}$ is supported only on $\{i : 0 < |i-y| \le w\}$.

The same decomposition holds inside the expectation $\mathbb{E}_t[w(t)\cdot]$ in Eq. 5 by linearity, giving the full-objective form

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_t\left[w(t) \sum_{l \in \mathcal{M}_t} \Big((1-\epsilon^{(l)})\,\mathcal{L}^{(l)}_{\mathrm{NELBO}} + \epsilon^{(l)}\, D_{\mathrm{KL}}\big(\mathbf{u}^{(l)} \,\|\, \mathbf{p}_\theta(\mathbf{z}_t^{(l)})\big) \Big) \right] + \text{const}. \tag{29}$$

C Experimental Settings

C.1 Datasets

Table 11: Statistics of benchmark datasets.

Datasets	Features	Frequency	Samples	Domain
Energy	28	10 min.	19,735	Weather
ETTh	7	60 min.	17,420	Temperature
Weather	21	10 min.	52,696	Weather
Sine	5	-	10000	Simulation

We evaluate the proposed MDTIM on four time-series datasets: three real-world benchmarks (Energy, ETTh, Weather) to assess practical validity, and one synthetic Sine dataset for controlled evaluation of periodic dynamics. These datasets cover a broad range of dynamics observed in long-horizon multivariate time-series. The Energy[1] dataset contains electricity-related measurements with strong seasonality and occasional irregular variations. The ETTh[2] dataset is a standard benchmark derived from Electricity Transformer Dataset [43] and is commonly used to evaluate long-range dependency modeling. The Weather[3] [36] dataset includes multivariate meteorological observations characterized

[1] https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction

[2] https://github.com/zhouhaoyi/ETDataset

[3] https://www.bgc-jena.mpg.de/wetter/

by nonlinear interactions, seasonal trends, and stochastic fluctuations. In addition, the synthetic Sine dataset offers a controlled environment with known periodic structure, which helps validate fundamental forecasting behavior under minimal uncontrolled factors. Overall, these datasets were selected to ensure domain diversity and to evaluate performance under both structured periodic signals and noisy, non-stationary real-world patterns. Dataset specifications are summarized in Table 11. For the implementation and training of the baseline models, we adopted the codebase curated by Du et al. [7][4].

C.2 Training Configurations

We provide the detailed hyperparameter settings used to train MDTIM. To ensure reproducibility, we list the common architectural parameters and training schemes applied across all datasets. All experiments are conducted with dual Intel Xeon Gold 6444Y CPUs and a single NVIDIA H100 PCIe GPU (80GB).

C.2.1 Hyperparameters

The architecture of MDTIM is based on the Factorized Temporal-Variate Backbone described in Section 4. Table 12 summarizes detailed hyperparameters. Every architectural and discretization hyperparameter, including K, w, and σ, is shared across all datasets and missing rates. K was selected once from the ETTh sweep in Table 10, and w and σ were fixed a priori without tuning, so none of the reported gains come from per-dataset configuration.

Table 12: Hyperparameters for MDTIM architecture and training.

Category	Parameter	Value	Description
Model	Hidden Size (D)	256	Dimension of hidden states
	Attention Heads	16	Number of heads in MSA
	DiT Blocks (L)	5	Number of factorized layers
	Dropout	0.2	Dropout probability
	Bins (K)	40	Vocabulary size for discretization
	Soft-label window (w)	2	Truncation window of Eq. 2
	Soft-label smoothness (σ)	1.0	Gaussian width in token bins
	Inference passes (M)	10	Noise samples averaged in Eq. 8
	Output Range	1.5	Value range $[-1.5, 1.5]$
	Cond. Dim	16	Time-step embedding dimension
Training	Batch Size	256	-
	Learning Rate	3×10^{-4}	-
	Scheduler	Const. w/ Warmup	-
	Warmup Steps	2,500	-
	Max Steps	10,000	-
	Gradient Clip	1.0	Norm clipping value
	EMA Decay	0.995	Exponential moving average
	FFT Weight (λ)	1.0	Spectral loss weight

C.2.2 Noise Schedule and Time-Dependent Weighting

We employ a Log-Linear Noise Schedule to compute the continuous-time diffusion process. For a time step $t \in [0, 1]$ and a small constant $\epsilon = 10^{-3}$, the total noise $\sigma(t)$ and its rate of change are defined as:

$$\sigma(t) = -\log\left(1 - (1-\epsilon)t\right), \quad \frac{d\sigma}{dt} = \frac{1-\epsilon}{1-(1-\epsilon)t} \tag{30}$$

[4] https://github.com/WenjieDu/Awesome_Imputation

To ensure balanced training across varying noise levels, we apply importance sampling via a time-dependent loss weight $w(t)$. This weight effectively normalizes the contribution of each time step to the objective function:

$$w(t) = \frac{d\sigma(t)/dt}{e^{\sigma(t)} - 1} \tag{31}$$

This weighting scheme assigns higher importance to low-noise regions ($t \approx 0$), prioritizing the learning of fine-grained details from clean data, while down-weighting highly corrupted states ($t \approx 1$) where reconstruction is ambiguous.

C.3 Evaluation Metrics

We evaluate both point imputation accuracy and the quality of predictive distributions. Let $y_{i,t,c}$ and $\hat{y}_{i,t,c}$ denote the ground truth and the imputed value for sample i, time step t, and channel c. All metrics are computed exclusively on the masked positions, denoted by the set of indices $\mathcal{M}$.

MAE and MSE. We report Mean Absolute Error (MAE) and Mean Squared Error (MSE), averaged over the masked entries $\mathcal{M}$:

$$\text{MAE} = \frac{1}{|\mathcal{M}|} \sum_{(i,t,c)\in\mathcal{M}} |\hat{y}_{i,t,c} - y_{i,t,c}|\,, \qquad \text{MSE} = \frac{1}{|\mathcal{M}|} \sum_{(i,t,c)\in\mathcal{M}} (\hat{y}_{i,t,c} - y_{i,t,c})^2\,. \tag{32}$$

In probabilistic settings, we set $\hat{y}_{i,t,c}$ to the sample mean of M generated trajectories, $\hat{y}_{i,t,c} = \frac{1}{M}\sum_{m=1}^{M} \tilde{y}_{i,t,c}^{(m)}$.

CRPS. To assess the probabilistic imputation performance, we use the Continuous Ranked Probability Score (CRPS) [21]. For a predictive CDF $F_{i,t,c}$ at a masked position, CRPS is defined as

$$\text{CRPS}(F_{i,t,c}, y_{i,t,c}) = \int_{-\infty}^{\infty} \Big(F_{i,t,c}(z) - \mathbb{I}[z \geq y_{i,t,c}]\Big)^2\, dz. \tag{33}$$

Lower CRPS indicates better probabilistic imputation, rewarding both calibration and sharpness.

Sample-based CRPS Estimation. Since the model produces M stochastic samples $\{\tilde{y}_{i,t,c}^{(m)}\}_{m=1}^{M} \sim F_{i,t,c}$, we approximate CRPS by

$$\overline{\text{CRPS}}_{i,t,c} = \frac{1}{M}\sum_{m=1}^{M} \left|\tilde{y}_{i,t,c}^{(m)} - y_{i,t,c}\right| - \frac{1}{2M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} \left|\tilde{y}_{i,t,c}^{(m)} - \tilde{y}_{i,t,c}^{(n)}\right|. \tag{34}$$

We report the average of $\overline{\text{CRPS}}_{i,t,c}$ over all masked entries in $\mathcal{M}$.

D Additional Experimental Results

In this section, we provide supplementary experimental results that complement the main paper. Section D.1 reports the full quantitative comparison including the 50% missing ratio omitted from Table 1 for space. Section D.2 presents the two controls behind the missingness-mechanism ablation of Section 5.5, Section D.3 examines a channel-shared embedding, and Section D.4 reports variability across training seeds.

D.1 Full Imputation Performance Across All Missing Ratios

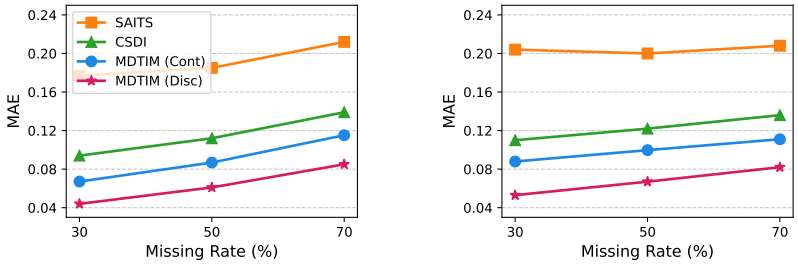
Table 13 reports the complete results of our main imputation experiment, including the 30%, 50%, and 70% missing ratios under both Uniform and Geometric scenarios. The 50% column extends the trend reported in the main paper: MDTIM consistently maintains the lowest MAE across the majority of settings, and the relative improvement over baselines remains stable across missing ratios. This confirms that the performance advantage of MDTIM is not specific to any particular corruption level, but holds robustly throughout the regime of partial observation.

Table 13: Quantitative comparison of multivariate time-series imputation performance ($L = 48$). We report the MAE of MDTIM and baselines averaged over 3 random mask seeds. The best results are highlighted in **bold**.

Dataset		Energy						ETTh						Weather						Sine					
Missing Type		Uniform			Geometric			Uniform			Geometric			Uniform			Geometric			Uniform			Geometric		
	Model	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
RNN	BRITS	0.246	0.296	0.379	0.329	0.344	0.366	0.194	0.235	0.299	0.227	0.258	0.292	0.050	0.057	0.073	0.058	0.063	0.071	0.010	0.013	0.021	0.020	0.019	0.019
	MRNN	1.086	1.107	1.143	1.087	1.108	1.143	0.743	0.762	0.782	0.751	0.765	0.781	0.651	0.653	0.661	0.654	0.656	0.659	0.169	0.169	0.170	0.169	0.169	0.170
	GRUD	0.364	0.387	0.457	0.426	0.425	0.445	0.310	0.336	0.394	0.348	0.362	0.383	0.104	0.176	0.369	0.164	0.231	0.350	0.008	0.008	0.014	0.013	0.012	0.012
Transformer	Transformer	0.323	0.335	0.401	0.354	0.356	0.393	0.168	0.201	0.252	0.183	0.212	0.247	0.076	0.067	0.070	0.083	0.072	0.068	0.039	0.041	0.046	0.042	0.044	0.045
	Informer	0.344	0.338	0.379	0.371	0.357	0.373	0.207	0.240	0.302	0.226	0.256	0.296	0.047	0.046	0.059	0.052	0.050	0.058	0.027	0.025	0.039	0.036	0.032	0.037
	PatchTST	0.586	0.336	0.529	0.528	0.378	0.517	0.202	0.218	0.272	0.231	0.243	0.262	0.074	0.061	0.082	0.081	0.069	0.077	0.015	0.011	0.017	0.019	0.014	0.015
	SAITS	0.177	0.185	0.212	0.204	0.200	0.208	0.140	0.166	**0.210**	0.152	0.176	0.206	0.045	0.044	0.049	0.050	0.048	0.048	0.026	0.023	0.026	0.030	0.026	0.025
	Imputeformer	0.066	0.100	0.219	0.113	0.143	0.194	0.146	0.178	0.245	0.165	0.196	0.235	0.049	0.068	0.171	0.090	0.111	0.149	0.006	0.012	0.041	0.012	0.019	0.037
CNN	TimesNet	0.617	0.683	0.818	0.689	0.726	0.808	0.593	0.653	0.719	0.628	0.666	0.717	0.211	0.202	0.351	0.225	0.223	0.348	0.167	0.188	0.220	0.189	0.200	0.217
	SCINet	0.557	0.418	0.571	0.546	0.442	0.565	0.238	0.252	0.323	0.261	0.273	0.317	0.074	0.068	0.089	0.086	0.076	0.087	0.019	0.018	0.031	0.026	0.024	0.029
Linear	DLinear	0.795	0.351	0.527	0.684	0.388	0.517	0.379	0.319	0.445	0.373	0.357	0.435	0.372	0.149	0.223	0.345	0.161	0.220	0.070	0.032	0.053	0.062	0.041	0.051
	FiLM	0.877	0.380	0.520	0.791	0.413	0.512	0.696	0.608	0.627	0.707	0.627	0.623	0.380	0.156	0.209	0.360	0.165	0.206	0.127	0.102	0.109	0.132	0.110	0.108
	FreTS	0.157	0.147	0.219	0.225	0.184	0.206	0.222	0.238	0.299	0.262	0.266	0.286	0.071	0.059	0.080	0.080	0.067	0.074	0.068	0.037	0.068	0.075	0.049	0.065
Generative	GPVAE	0.476	0.509	0.730	0.501	0.532	0.727	0.333	0.349	0.449	0.369	0.385	0.440	0.158	0.153	0.255	0.172	0.170	0.252	0.152	0.150	0.162	0.153	0.151	0.162
	USGAN	0.264	0.323	0.412	0.330	0.362	0.401	0.208	0.243	0.302	0.240	0.266	0.295	0.087	0.094	0.122	0.101	0.104	0.118	0.014	0.018	0.029	0.026	0.026	0.026
	CSDI	0.094	0.112	0.139	0.110	0.122	0.136	0.160	0.192	0.244	0.177	0.206	0.240	0.039	0.043	0.049	0.044	0.046	0.048	0.003	0.004	0.004	0.004	0.004	0.004
	FGTI	0.050	0.071	0.100	0.065	0.080	0.097	0.218	0.255	0.349	0.293	0.311	0.328	0.038	0.043	0.051	0.046	0.047	0.049	**0.001**	**0.002**	**0.003**	**0.002**	**0.003**	**0.003**
	MDTIM (Ours)	**0.044**	**0.061**	**0.085**	**0.053**	**0.067**	**0.082**	**0.127**	**0.157**	0.211	**0.146**	**0.173**	**0.205**	**0.032**	**0.036**	**0.044**	**0.036**	**0.039**	**0.043**	0.002	0.003	0.004	0.003	**0.003**	**0.003**

D.2 Comparison with Continuous Modeling

To isolate the contribution of our discrete formulation, we compare MDTIM (Disc), the proposed model, against a continuous-modeling variant MDTIM (Cont). MDTIM (Cont) shares the same Factorized Temporal-Variate backbone and masked diffusion training paradigm, but operates directly on continuous input values and is optimized with a Mean Squared Error (MSE) loss. In contrast, MDTIM (Disc) employs our *Stochastic Discretization* to map continuous signals into discrete tokens and optimizes the ordinal-aware discrete diffusion objective described in Section 4.



(a) Uniform Missing (b) Geometric Missing

Figure 4: Comparison between MDTIM (Disc) and MDTIM (Cont) on the Energy dataset, alongside representative baselines (SAITS, CSDI). Results are reported under both uniform and geometric missing scenarios across varying missing rates.

As shown in Figure 4, MDTIM (Cont) already performs competitively with strong baselines, indicating that the masked diffusion paradigm of learning to reconstruct from varying corruption levels with timestep conditioning is itself effective even in continuous spaces. Nevertheless, MDTIM (Disc) consistently achieves lower MAE across all missing rates and missing types. This performance gap confirms that the explicit structural separation between valid observations and missing placeholders, achieved through our orthogonal tokenization, provides a benefit beyond the masked diffusion training scheme alone, and is crucial for time series imputation.

Attention-mask variant. The second control in Section 5.5 replaces the `[MASK]` token by an attention mask. The input still carries an embedding at every position, but positions that are missing are removed from the key set of both the temporal and the variate attention in every block, so each query attends only to observed positions and to itself. Missing positions therefore read the observed ones, while no observed position can see where the gaps are. Everything else, including the discretization, the soft-label objective, the spectral term, and the training schedule, is identical to MDTIM, and the variant is evaluated under the same three mask seeds.

D.3 Channel-Shared Embedding

The per-channel embedding tables E_c of Section 4 hold CKD parameters, a cost that grows linearly with the number of variates. To see how much of the accuracy depends on them, we retrain MDTIM on Energy with a single embedding table shared across all channels, which costs KD parameters, a 28-fold reduction, with everything else identical.

Table 14: MAE on Energy with per-channel and channel-shared embedding tables, averaged over 3 random mask seeds.

	Uniform			Geometric		
Embedding	30%	50%	70%	30%	50%	70%
Per-channel (CKD)	0.044	0.061	0.085	0.053	0.067	0.082
Shared (KD)	0.044	0.061	0.085	0.053	0.067	0.082

The two variants perform identically to three decimals in every setting (Table 14). The RoPE attention that captures temporal correlation also picks up the cross-channel structure, which makes the shared table a practical option for data with many channels. We would be careful about extrapolating this at much higher C, since Energy has only 28 channels, still far fewer than the several hundred of a truly high-dimensional dataset.

D.4 Variance Across Training Seeds

The mask seed determines which entries are missing, so changing it changes the imputation problem itself, and a standard deviation taken across mask seeds would blend problem difficulty with model variability. We therefore report variability across training seeds under a fixed mask. For each of the three mask seeds used in Section 5, we train three models with seeds 2024, 2025, and 2026 and report the mean and standard deviation of their MAE (Table 15). The training-seed standard deviation stays at or below 0.001 in every setting, so the comparisons in the main text are not an artifact of seed noise.

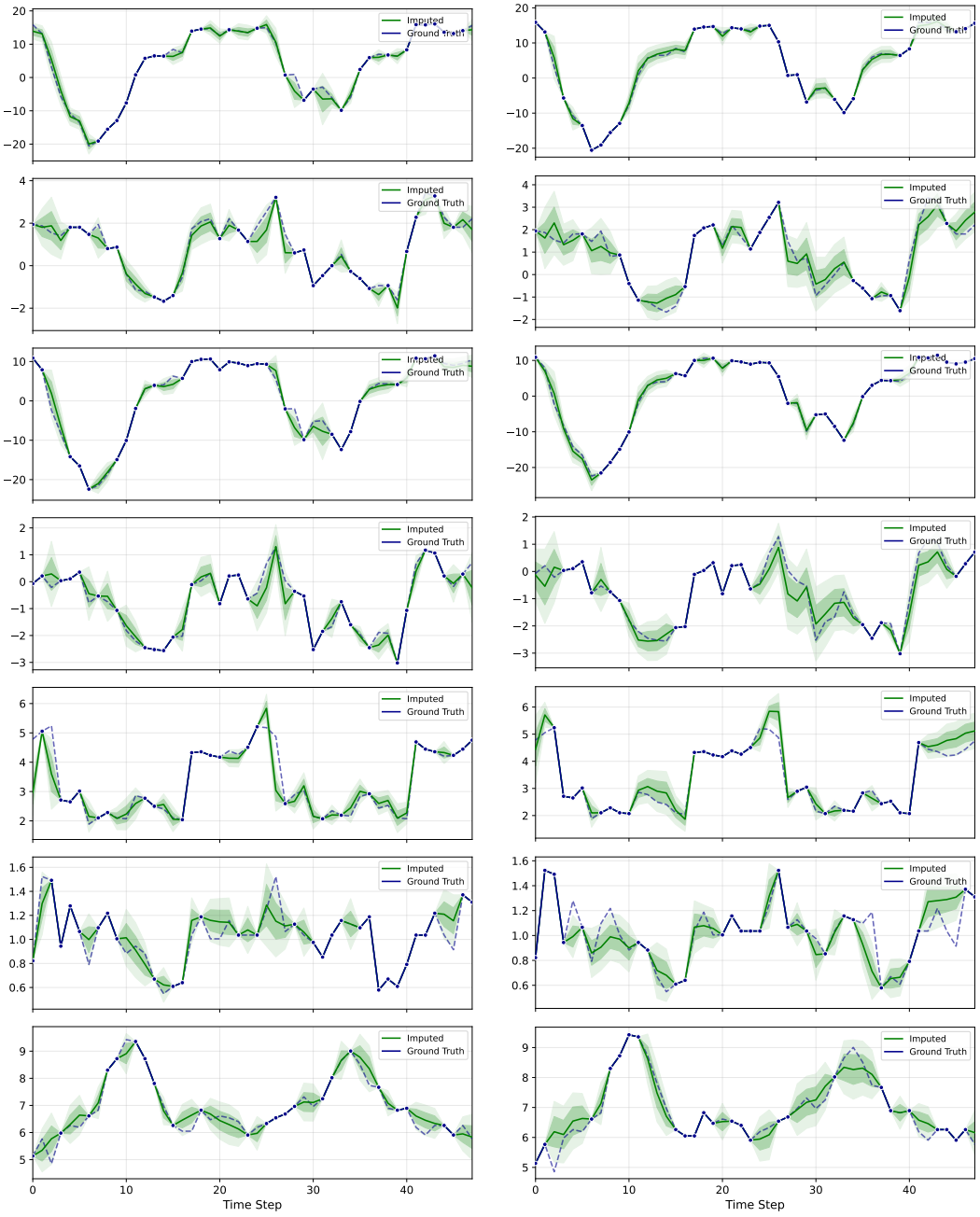
Table 15: MAE mean $\pm$ standard deviation over three training seeds (2024, 2025, 2026), reported separately for each mask seed.

		Uniform			Geometric		
Dataset	Rate	Mask 2024	Mask 2025	Mask 2026	Mask 2024	Mask 2025	Mask 2026
ETTh	30%	0.1270±0.0005	0.1274±0.0006	0.1269±0.0007	0.1461±0.0004	0.1449±0.0005	0.1456±0.0005
	50%	0.1569±0.0001	0.1578±0.0002	0.1569±0.0003	0.1716±0.0001	0.1733±0.0003	0.1721±0.0003
	70%	0.2118±0.0009	0.2113±0.0009	0.2110±0.0010	0.2045±0.0009	0.2048±0.0008	0.2056±0.0006
Energy	30%	0.0439±0.0001	0.0445±0.0001	0.0441±0.0001	0.0527±0.0002	0.0546±0.0001	0.0532±0.0002
	50%	0.0606±0.0001	0.0610±0.0002	0.0608±0.0001	0.0669±0.0001	0.0672±0.0001	0.0668±0.0001
	70%	0.0845±0.0002	0.0850±0.0002	0.0845±0.0001	0.0819±0.0001	0.0823±0.0002	0.0820±0.0002
Weather	30%	0.0316±0.0001	0.0316±0.0001	0.0317±0.0001	0.0360±0.0001	0.0363±0.0000	0.0363±0.0000
	50%	0.0360±0.0000	0.0360±0.0001	0.0361±0.0000	0.0387±0.0001	0.0390±0.0000	0.0390±0.0000
	70%	0.0440±0.0001	0.0441±0.0001	0.0440±0.0001	0.0427±0.0001	0.0429±0.0001	0.0428±0.0001
Sine	30%	0.0021±0.0002	0.0021±0.0002	0.0021±0.0002	0.0026±0.0003	0.0027±0.0003	0.0027±0.0003
	50%	0.0024±0.0002	0.0025±0.0002	0.0024±0.0002	0.0027±0.0002	0.0028±0.0002	0.0028±0.0002
	70%	0.0033±0.0002	0.0032±0.0002	0.0032±0.0002	0.0031±0.0002	0.0032±0.0002	0.0031±0.0002

E Visualization of Imputation Results

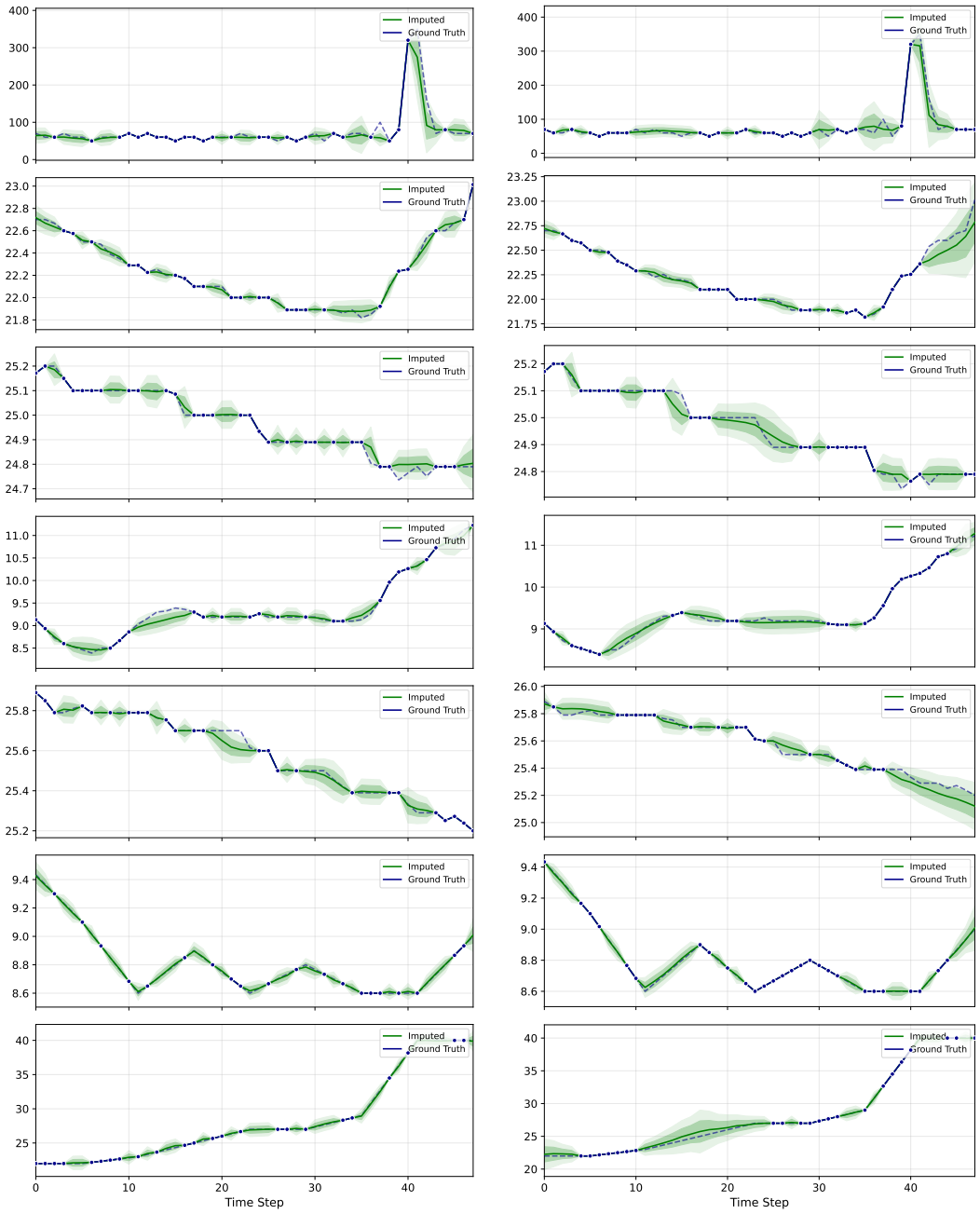
We provide comprehensive visualizations of the imputation results across all benchmark datasets: ETTh, Energy, Weather, and Sine with MDTIM. Figures 5, 6, 7, and 8 illustrate the reconstructed time series for all channels (ETTh, Sine) or subset of channels (Energy, Weather). In each figure, the left column displays the results under 50% uniform masking, while the right column shows under 50% geometric masking.

The blue lines represent the ground truth values, while the green lines denote the imputed values ($\hat{x}$) reconstructed by MDTIM. The green shaded areas indicate the estimated uncertainty intervals. As observed, MDTIM effectively captures the complex temporal dynamics and periodicity of the multivariate time series.



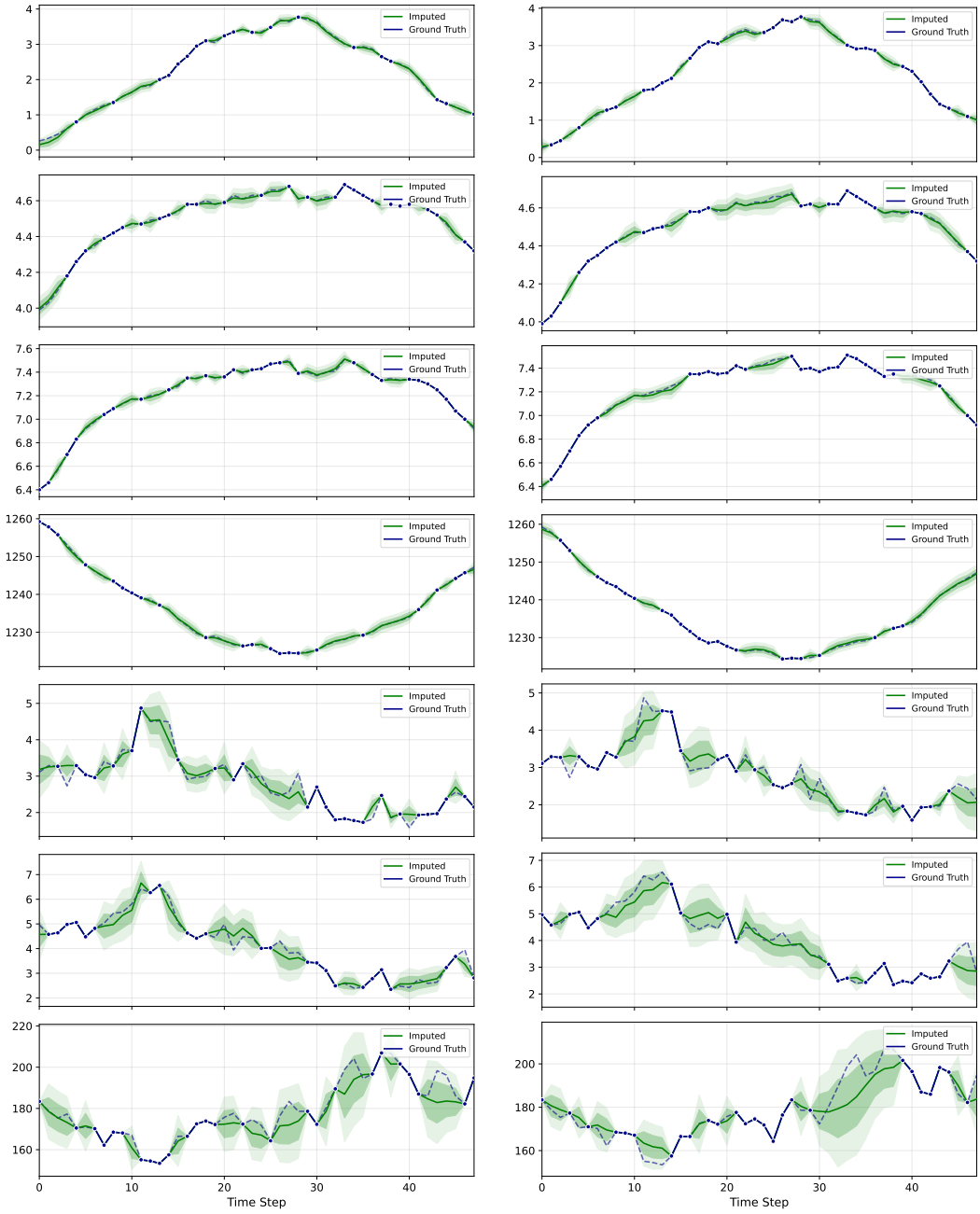
(a) Uniform Missing (b) Geometric Missing

Figure 5: Visualization of imputation results on the ETTh dataset.



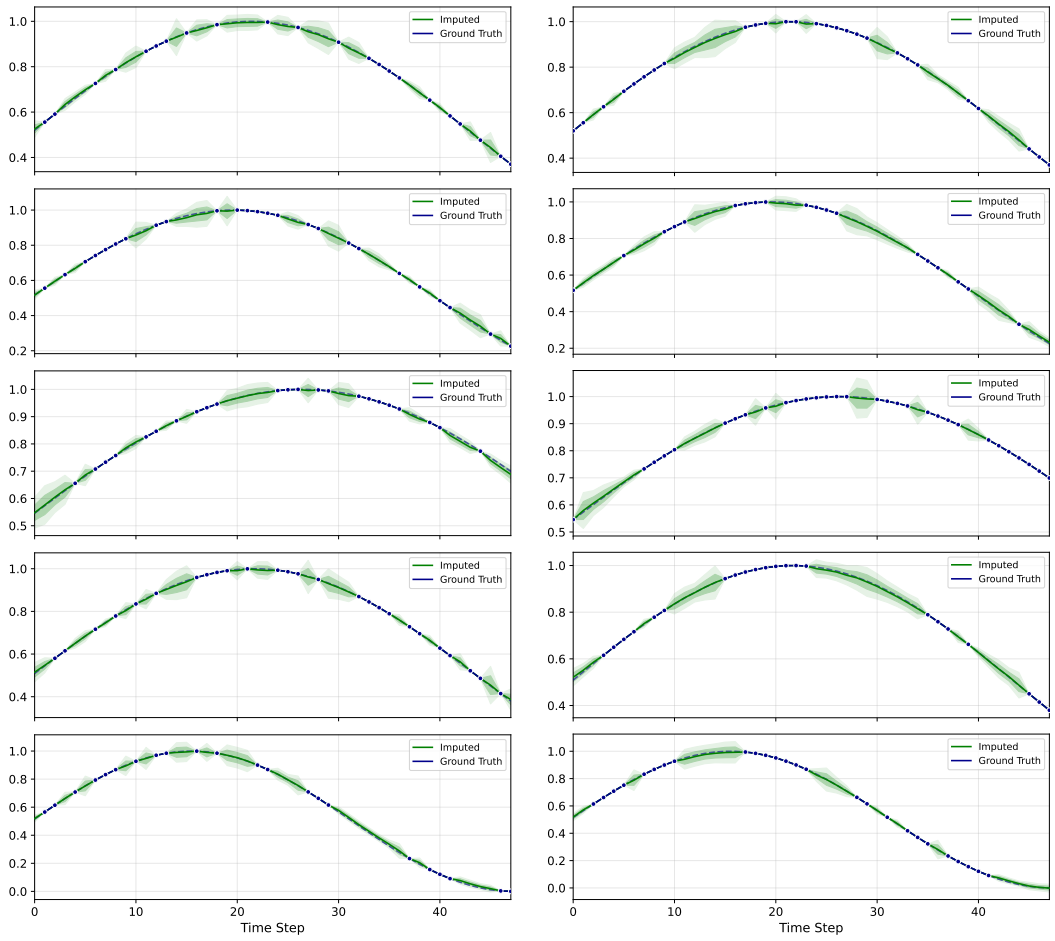
(a) Uniform Missing (b) Geometric Missing

Figure 6: Visualization of imputation results on the Energy dataset.



(a) Uniform Missing (b) Geometric Missing

Figure 7: Visualization of imputation results on the Weather dataset.



(a) Uniform Missing (b) Geometric Missing

Figure 8: Visualization of imputation results on the Sine dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly state our three main contributions: (i) the masked diffusion framework for time series imputation, (ii) the Stochastic Discretization and Ordinal-Aware Soft Labeling pipeline, and (iii) the Expectation-based Unmasking strategy with Spectral Consistency Regularization. Each claim is supported by experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6: Stochastic Discretization is unbiased in expectation but lossy for an individual sample, the spectral consistency term is disabled on sparsely observed data such as PhysioNet, the per-channel embedding tables cost $O(CKD)$ parameters, and the evaluation is limited to standard imputation benchmarks of moderate length.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain formal theoretical results; the contribution is methodological and empirical.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the model architecture in Section 4, the training objective in Section 4.2 and Appendix A.2.3, full hyperparameter settings in Appendix C.2.1 (Table 12), the noise schedule and time-dependent weighting in Appendix C.2.2, and the evaluation protocol in Section 5.1 and Appendix C.3. Code is publicly available at the project page linked in the abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and instructions for reproducing the main results are publicly available at https://dongbeank.github.io/MDTIM/. All datasets used (Energy, ETTh, Weather, Sine, PhysioNet 2012) are publicly available; URLs are provided in Appendix C.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and evaluation details are provided in Section 5.1 and Appendix C.2. Appendix C.2.1 (Table 12) lists the full hyperparameter configuration, including hidden size, number of layers, dropout, vocabulary size, batch size, learning rate, scheduler, warmup steps, gradient clipping, EMA decay, and FFT loss weight.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All main results are averaged over three random mask seeds (Section 5). Appendix D.4 additionally reports the mean and standard deviation over three independently trained models (training seeds 2024, 2025, 2026) under each mask seed, for every dataset, missing pattern, and missing rate. The standard deviation is at or below 0.001 in every setting, well below the margins to the baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Compute resources are described in Appendix C.2: all experiments are conducted with dual Intel Xeon Gold 6444Y CPUs and a single NVIDIA H100 PCIe GPU (80GB). Inference time is reported in Table 5.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our research conforms to the NeurIPS Code of Ethics. The work uses publicly available benchmark datasets and does not involve human subjects, sensitive data, or applications with foreseeable harm.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on time series imputation methodology. While accurate imputation can benefit downstream applications such as healthcare and energy monitoring, the contribution is general-purpose and does not have a direct path to specific negative societal applications beyond those of generic time series modeling.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed model is a time series imputation method trained on public benchmark datasets and does not pose risks of misuse such as those associated with generative language or image models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baseline models are cited with their original publications. Dataset URLs are provided in Appendix C.1 footnotes. Baseline implementations follow the codebase of Du et al. [7], also referenced in Appendix C.1.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code for MDTIM is publicly released at https://dongbeank.github.io/MDTIM/ with documentation describing how to reproduce the experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve research with human subjects. The PhysioNet 2012 dataset is a publicly released benchmark with the original IRB approval handled by its providers.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.