HyFAD: Hybrid Time-Frequency Diffusion with Frequency-Aware Embedding for Time Series Imputation

Hongfan Gao[1], Wangmeng Shen[1], Bin Yang[1], Jilin Hu[1,2]*
[1]School of Data Science and Engineering, East China Normal University
[2]Engineering Research Center of Blockchain Data Management, Ministry of Education.
{hf.gao,wmshen}@stu.ecnu.edu.cn, {byang,jlhu}@dase.ecnu.edu.cn

Abstract

Diffusion models have demonstrated strong performance in time series modeling due to their ability to progressively capture complex data distributions through iterative denoising. However, existing approaches struggle with frequency-sensitive denoising, high-frequency reconstruction and balancing global trends with local dynamics. To address these limitations, we propose **HyFAD**, a **Hy**brid time-frequency **D**iffusion model with **F**requency-**A**ware embedding for time series imputation. Built upon the DDPM paradigm, HyFAD adopts a coupled time-frequency diffusion framework, in which the reverse denoising proceeds sequentially from the time domain to the frequency domain, enabling coarse-to-fine generation. Specifically, the time-domain diffusion process captures low-frequency global trends, while the frequency-domain diffusion process refines high-frequency spectral components. We further introduce a frequency-aware step embedding that exploits the relationship between diffusion steps and spectral components, providing step-dependent spectral guidance and facilitates more accurate band-wise reconstruction. Extensive experiments on multiple benchmark datasets demonstrate that HyFAD achieves state-of-the-art performance. Our source code is available at `https://github.com/decisionintelligence/HyFAD`.

1 Introduction

Diffusion models [1–3] have emerged as a powerful class of generative frameworks for time series modeling, owing to their strong ability to approximate complex data distributions. By progressively denoising latent representations initialized from Gaussian noise, these models learn a reverse process that captures rich temporal dependencies. As a result, diffusion-based approaches have achieved state-of-the-art performance across a range of time series tasks, including imputation [4–8], forecasting [9–14], and generation [15–18].

Despite their success, existing diffusion models for time series still suffer from several notable limitations: (i) **Insufficient reconstruction of high-frequency details** [19–21]. In standard time-domain diffusion, each diffusion step injects isotropic Gaussian noise whose energy is uniformly distributed across frequencies. However, real-world time series usually exhibit highly imbalanced spectral energy distributions, with most energy concentrated in low-frequency components. This mismatch makes high-frequency reconstruction more challenging and limits the model's ability to recover fine-grained temporal details. (ii) **Imbalanced modeling of global trends and local dynamics** [22–25]. Global trends and local dynamics often compete during reconstruction. Prioritizing long-term structures may lead to over-smoothed outputs, while emphasizing short-term fluctuations can disrupt global

*Corresponding author.

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

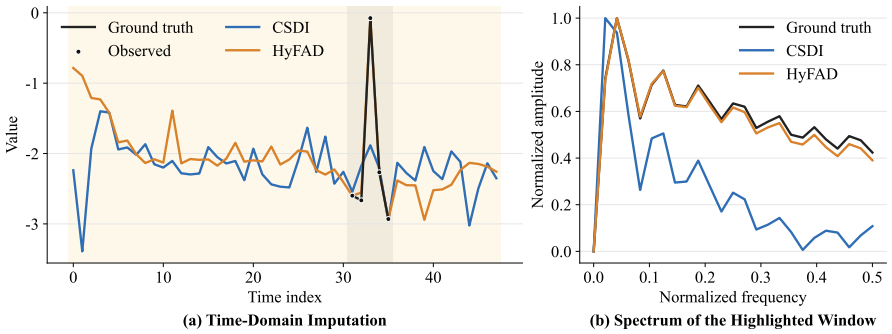


Figure 1: An imputation example from Channel 4 on PhysioNet. While CSDI captures coarse trends, it attenuates sharp local variations in the highlighted region. In contrast, HyFAD better preserves these local structures and matches the ground-truth spectrum, correcting CSDI's underestimation of mid/high-frequency amplitudes.

consistency. (iii) **Frequency-insensitive modeling** [26–28]. Existing diffusion models typically use frequency-agnostic step embeddings and inject noise uniformly across frequency bands. As a consequence, they have limited ability to adapt their denoising behavior to the distinct recovery characteristics of low- and high-frequency components, which can lead to over-smoothed reconstructions when local variations are weak or sparsely observed.

Fig. 1 provides a representative imputation example on PhysioNet. The time-domain diffusion baseline, CSDI, captures the coarse level of the sequence but attenuates the sharp local variation in the highlighted region, resulting in an over-smoothed reconstruction. The spectrum computed over the same window further shows that the reconstructed signal underestimates middle- and high-frequency amplitudes compared with the ground truth. This example indicates that ignoring frequency dynamics can lead to observable reconstruction errors, motivating diffusion models that explicitly account for frequency-domain denoising.

In summary, these limitations can be attributed to two fundamental causes. (i) **Time-domain-only noise modeling.** In most existing diffusion models, both noise injection and denoising are performed exclusively in the time domain. This design overlooks the imbalanced spectral energy distribution of real-world time series, where high-frequency components often have weaker energy and are therefore harder to recover. Incorporating frequency-domain modeling enables explicit characterization of band-wise spectral statistics, allowing high-frequency transients and local dynamics to be more effectively captured during denoising. (ii) **Lack of frequency-specific attention in step embeddings.** Existing diffusion step embeddings do not incorporate frequency-dependent information and therefore fail to capture how different frequency bands should be emphasized or suppressed at different denoising stages. This prevents the model from adaptively allocating capacity between low- and high-frequency components, exacerbating the loss of fine-grained details and weakening multi-scale representation.

To address these challenges, we propose **HyFAD**, a **Hy**brid time-frequency **D**iffusion model with **F**requency-**A**ware embedding for time series imputation. Unlike conventional time-domain diffusion approaches, HyFAD incorporates frequency-domain dynamics into the diffusion process, enabling explicit differentiation across spectral bands and improving high-frequency reconstruction. Specifically, HyFAD formulates a coupled time-frequency diffusion process, where frequency-domain and time-domain noise are injected sequentially in the forward process and removed in a mirrored order during reverse denoising. Furthermore, we analyze the relationship between diffusion steps and spectral components and introduce a frequency-aware step embedding that provides step-dependent spectral guidance, allowing the model to focus on appropriate frequency bands throughout denoising.

Our contributions are summarized as follows:

1. We propose HyFAD, a coupled time-frequency diffusion framework for time series imputation. The proposed forward and reverse processes explicitly model both time-domain and frequency-domain noise within a unified DDPM framework, enabling coarse-to-fine reconstruction.
2. We introduce a frequency-aware diffusion step embedding that provides step-dependent spectral guidance by modeling the correspondence between diffusion steps and frequency bands, allowing adaptive focus on appropriate spectral components during denoising.

3. We conduct extensive experiments on multiple real-world benchmark datasets, where HyFAD consistently achieves state-of-the-art performance, demonstrating the effectiveness of the proposed method.

2 Preliminaries

2.1 Discrete Fourier Transform

Given a real-valued discrete time series $\mathbf{x} = [x_0, \ldots, x_{L-1}] \in \mathbb{R}^L$ with length L, its unitary Discrete Fourier Transform (DFT), denoted by $\tilde{\mathbf{x}} \in \mathbb{C}^L$, is defined as:

$$\tilde{\mathbf{x}}_k = \frac{1}{\sqrt{L}} \sum_{\tau=0}^{L-1} \mathbf{x}_\tau \exp\left(-\frac{k2\pi i}{L}\tau\right), \tag{1}$$

where k is the frequency index, τ is the time index, and i denotes the imaginary unit. As $\mathbf{x}$ is real-valued, $\tilde{\mathbf{x}}$ exhibits conjugate symmetry *i.e.,* $\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}^*_{L-k}$, making almost half of the coefficients redundant. To obtain a compact representation that preserves isometry (*i.e.,* satisfying Parseval's identity) in the real domain, we define an *orthonormal real DFT* (rDFT) transform.

Specifically, we exploit the spectral symmetry to discard the redundant negative frequencies. To offset the energy reduction caused by this truncation, we apply a scaling factor of $\sqrt{2}$ to the real and imaginary components of the intermediate frequencies. The elements of the transformed vector $\mathbf{y}$ are arranged as follows:

$$\mathbf{y}_m = \begin{cases} \Re(\tilde{\mathbf{x}}_0) & m = 0 \\ \Re(\tilde{\mathbf{x}}_m) & m = \frac{L}{2} \text{ (L is even)} \\ \sqrt{2}\Re(\tilde{\mathbf{x}}_m) & 0 < m < \lceil \frac{L}{2} \rceil \\ \sqrt{2}\Im(\tilde{\mathbf{x}}_{k(m)}) & m \geq N_r, \end{cases} \tag{2}$$

where $\Re(\tilde{\mathbf{x}})$ and $\Im(\tilde{\mathbf{x}})$ denote the real and imaginary parts of $\mathbf{x}$ and $\mathbf{y} = [y_0, \ldots, y_{L-1}]^\top \in \mathbb{R}^L$ is the real-valued frequency representation constructed from $\Re(\tilde{\mathbf{x}})$ and $\Im(\tilde{\mathbf{x}})$. The index $m = 0, \ldots, L-1$ denotes the entry index of $\mathbf{y}$, $k(m) = m - N_r + 1$ maps entries in the imaginary-part block of $\mathbf{y}$ to the corresponding positive-frequency index and $N_r = \lfloor \frac{L}{2} \rfloor$ is the number of frequency bins in the complex spectrum.

It is worth noticing that rDFT is a linear transformation, *i.e.,* $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{W} \in \mathbb{R}^{L \times L}$ is the transformation matrix and orthogonal (*i.e.,* $\mathbf{W}\mathbf{W}^T = \mathbf{I}$). This implies that the rDFT operation is invertible, allowing for reconstructing the time-domain representation $\mathbf{x}$ from the frequency-domain representation $\mathbf{y}$ with $\mathbf{x} = \mathbf{W}^T\mathbf{y}$.

2.2 Time Series Diffusion in the Frequency Domain

While most existing diffusion-based approaches [29, 30] for time series focus exclusively on modeling in the time domain, [22] explores the diffusion process in the frequency domain by introducing a noise scaling matrix Λ to the frequency-domain diffusion process under DFT. Λ plays two essential roles: (i) Rescaling spectral coefficients to make frequency-domain noise isotropic across all frequency components, (ii) Ensuring energy consistency between the time and frequency domains under the frequency domain transform. For an input sequence $\mathbf{x} \in \mathbb{R}^{K \times L}$, the diagonal noise scaling matrix $\Lambda \in \mathbb{R}^{L \times L}$ is defined as :

$$\Lambda = \begin{cases} 1 & n = 1 \text{ or } n = \frac{L}{2} \ (L \text{ is even}) \\ \frac{1}{\sqrt{2}} & \text{otherwise} \end{cases} \tag{3}$$

Different from [22], we set $\Lambda = \mathbf{I}$ since rDFT in Sec.2.1 is employed in our implementations instead of DFT for isotropic noise and energy consistency.

2.3 Problem Definition and Notations

Definition 2.1 (**Time Series with Missing Value**)**.** A time series with missing values is defined as $\tilde{\mathbf{X}} = (\mathbf{X}, \mathbf{M}, \mathbf{T})$, where $\mathbf{X} \in \mathbb{R}^{K \times L}$ is the observation matrix with K observations at a time, which

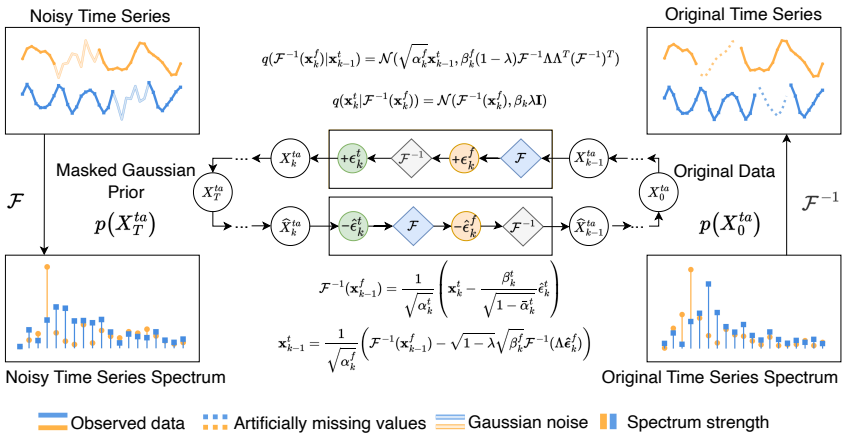


Figure 2: Noise injection and denoising process in HyFAD. At each forward step, frequency-domain noise is injected first and then time-domain noise. In the reverse process, time-domain noise is removed first, followed by frequency-domain noise.

are ordered along L time intervals chronologically; $\mathbf{M} \in \mathbb{R}^{K \times L}$ is an indicator matrix that indicates whether the observation at (i, j) in $\mathbf{X}$ is missing or not. We use $M^{\text{obs}} \in \{0, 1\}^{K \times L}$ to denote the observation mask, where $M^{\text{obs}}_{i,j} = 1$ indicates that $X_{i,j}$ is observed. We use M^{tar} to denote the target mask for evaluation or training, where $M^{\text{tar}}_{i,j} = 1$ indicates an imputation target.

Problem Statement (Time Series Imputation). Given a time series with missing value $\tilde{\mathbf{X}} = (\mathbf{X}, \mathbf{M}, \mathbf{T})$, the goal of time series imputation is to learn an imputation function $\mathcal{M}_\theta$, such that

$$\bar{\mathbf{X}} = \mathcal{M}_\theta(\tilde{\mathbf{X}}), \tag{4}$$

where $\bar{\mathbf{X}} \in \mathbb{R}^{K \times L}$ is the imputed time series, $\bar{\mathbf{X}}_{i,j}$ denotes the imputation output if $\mathbf{M}_{i,j} = 1$, otherwise $\bar{\mathbf{X}}_{i,j} = \tilde{\mathbf{X}}_{i,j}$.

Notations. We adopt the following notations. We use superscripts to denote the domain of the data, and subscripts to denote denoising steps, *e.g.,* $\mathbf{x}^f_k$ refers to the data at diffusion step k in the frequency domain and $\boldsymbol{\epsilon}^t_k$ refers to the noise at diffusion step k in the time domain. β is a predefined scheduler over T denoising steps and $\alpha_k = 1 - \beta_k, k \in \{1, 2, \cdots, T\}$. We denote $\bar{\alpha}_{i:j} = \Pi^j_{k=i} \alpha_k$ and if $i = 1$, we omit the lower index i and use $\bar{\alpha}_j$ for short. Specially, we define $\bar{\alpha}_{0:j} = 1$ for arbitrary j. Besides, for time series data, we follow the same notations as CSDI, *i.e.,* $\mathbf{X}^{\text{ta}}, \mathbf{X}^{\text{obs}}$ denotes the imputation targets and ground truth values and $\mathbf{M}^{\text{cond}}$ denotes the condition matrix.

3 Methodology

3.1 Hybrid Time-Frequency Diffusion Model

3.1.1 Forward Process of HyFAD

The forward process of HyFAD consists of two successive parts in a single step, *i.e.,* noise is first injected in the frequency domain and subsequently in the time domain. By iteratively performing these two successive noise injection procedures, the original data distribution $p(x_0)$ is transformed into a target Gaussian distribution. Given frequency transform $\mathcal{F}$, we first map the data into corresponding domain via $\mathcal{F}$, and then inject random gaussian noise $\boldsymbol{\epsilon}^f_k \sim \mathcal{N}(0, \mathbf{I})$ into this frequency domain representation following a DDPM-style forward diffusion process, which is formulated as:

$$\mathbf{x}^f_k = \sqrt{\alpha^f_k}\mathcal{F}(\mathbf{x}^t_{k-1}) + \sqrt{\beta^f_k}\sqrt{1-\lambda}(\Lambda\boldsymbol{\epsilon}^f_k), \tag{5}$$

where λ is the coefficient for balancing variance between frequency and time domain noise and $\boldsymbol{\epsilon}^f_k \sim \mathcal{N}(0, \mathbf{I})$. Then the frequency-domain noisy sample $\mathbf{x}^f_k$ in Eq.5 is transformed back to time domain:

$$\mathcal{F}^{-1}(\mathbf{x}^f_k) = \sqrt{\alpha^f_k}\mathbf{x}^t_{k-1} + \sqrt{\beta^f_k}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}^f_k), \tag{6}$$

where $\boldsymbol{\epsilon}_k^f \sim \mathcal{N}(0, \mathbf{I})$ and we can obtain the posterior $q(\mathcal{F}^{-1}(\mathbf{x}_k^f)|\mathbf{x}_{k-1}^t) \coloneqq \mathcal{N}(\sqrt{\alpha_k^f}\mathbf{x}_{k-1}^t, \beta_k^f(1-\lambda)\mathcal{F}^{-1}\Lambda\Lambda^T(\mathcal{F}^{-1})^T)$.

Then another **independent gaussian noise** $\boldsymbol{\epsilon}_k^t \sim \mathcal{N}(0, \mathbf{I})$ is injected to $\mathcal{F}^{-1}(\mathbf{x}_k^f)$, *i.e.,* noise is injected in the time domain:

$$\mathbf{x}_k^t = \sqrt{\alpha_k^t}\mathcal{F}^{-1}(\mathbf{x}_k^f) + \sqrt{\beta_k^t}\sqrt{\lambda}\boldsymbol{\epsilon}_k^t. \tag{7}$$

Thus, the posterior can be written as $q(\mathbf{x}_k^t|\mathcal{F}^{-1}(\mathbf{x}_k^f)) \coloneqq \mathcal{N}(\sqrt{\alpha_k^t}\mathcal{F}^{-1}(\mathbf{x}_k^f), \beta_k^t\lambda\mathbf{I})$. With this, the noise injection loop k is completed. We can formulate the relationship between $\mathbf{x}_k^t$ and $\mathbf{x}_{k-1}^t$ as:

$$\mathbf{x}_k^t = \sqrt{\alpha_k^t\alpha_k^f}\mathbf{x}_{k-1}^t + \sqrt{\alpha_k^t(1-\alpha_k^f)}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_k^f) + \sqrt{1-\alpha_k^t}\sqrt{\lambda}\boldsymbol{\epsilon}_k^t. \tag{8}$$

By iteratively performing the noise injection process according to Eq.6, Eq.7 and Eq.8, the closed-form relationship between $\mathbf{x}_k^t$ and $\mathbf{x}_0^t$ can be derived as:

$$\mathbf{x}_k^t = \sqrt{\bar{\alpha}_k^t\bar{\alpha}_k^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_k^t\bar{\alpha}_k^f}{\bar{\alpha}_{s-1}^t\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f) + \sqrt{\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_k^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t. \tag{9}$$

According to Eq.9, for any diffusion step k, $\mathbf{x}_k^t$ can be expressed as a linear transformation of the original signal $\mathbf{x}_0^t$ with two groups of independent gaussian noise $\boldsymbol{\epsilon}_s^t$ and $\boldsymbol{\epsilon}_s^f$, $s \in \{1, 2, \cdots, k\}$. Therefore, the posterior $q(\mathbf{x}_k^t|\mathbf{x}_0^t)$ remains gaussian, *i.e.,*

$$q(\mathbf{x}_k^t|\mathbf{x}_0^t) \coloneqq \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{10}$$

where $\boldsymbol{\mu}_k = \sqrt{\bar{\alpha}_k^t\bar{\alpha}_k^f}\mathbf{x}_0^t$ and $\boldsymbol{\Sigma}_k = (1-\lambda)\sum_{s=1}^{k}\beta_s^f\frac{\bar{\alpha}_k^t}{\bar{\alpha}_{s-1}^t}\frac{\bar{\alpha}_k^f}{\bar{\alpha}_s^f}\left(\mathcal{F}^{-1}\Lambda\Lambda^T(\mathcal{F}^{-1})^T\right) + \lambda\sum_{s=1}^{k}\beta_s^t\frac{\bar{\alpha}_k^t}{\bar{\alpha}_s^t}\mathbf{I}$.

3.1.2 Reverse Process of HyFAD

To progressively reconstruct the original data from the diffused samples, we construct a reverse Markov chain that is consistent with the forward noise injection process. Following the DDPM formulation, we parameterize the reverse distribution $p_\theta(\mathbf{x}_{k-1}^t|\mathbf{x}_k^t)$ using a Gaussian approximation, aiming to model the true posterior $q(\mathbf{x}_{k-1}^t|\mathbf{x}_k^t, \mathbf{x}_0^t)$. This enables closed-form stepwise sampling throughout the denoising trajectory. Since our method introduces a combination of time-domain noise and frequency-domain noise (Eq.8) at each diffusion step, the reverse process must explicitly account for the structure of this composite noise to ensure faithful reconstruction of original data.

To improve sampling stability, we adopt a DDIM-style sampling strategy [31], which follows a deterministic trajectory in the reverse process. We initialize from a Gaussian prior that is consistent with the terminal distribution of the forward diffusion. As in Eq.10, the sampling process starts from

$$\mathbf{x}_T^t = \sqrt{\lambda}\boldsymbol{\epsilon}^t + \sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}^f), \boldsymbol{\epsilon}^t, \boldsymbol{\epsilon}^f \sim \mathcal{N}(0, \mathbf{I}). \tag{11}$$

At each step k, the model predicts the time domain noise $\hat{\boldsymbol{\epsilon}}_k^t$ and frequency domain noise $\hat{\boldsymbol{\epsilon}}_k^f$. Reverse denoising is then performed in a step-wise, coupled manner within the same diffusion loop. First, the time-domain noise in $\mathbf{x}_k^t$ is removed according to:

$$\mathcal{F}^{-1}(\mathbf{x}_{k-1}^f) = \frac{1}{\sqrt{\alpha_k^t}}\left(\mathbf{x}_k^t - \frac{\beta_k^t}{\sqrt{1-\bar{\alpha}_k^t}}\hat{\boldsymbol{\epsilon}}_k^t\right). \tag{12}$$

Then the frequency-domain noise in $\mathcal{F}^{-1}(\mathbf{x}_{k-1}^f)$ is removed for $\mathbf{x}_{k-1}^t$:

$$\mathbf{x}_{k-1}^t = \frac{1}{\sqrt{\alpha_k^f}}\left(\mathcal{F}^{-1}(\mathbf{x}_{k-1}^f) - \sqrt{1-\lambda}\sqrt{\beta_k^f}\mathcal{F}^{-1}(\Lambda\hat{\boldsymbol{\epsilon}}_k^f)\right). \tag{13}$$

Eq.12 and Eq.13 define distinct update rules for time and frequency domain denoising process. To summarize, the step-wise denoising strategy *i.e.,* time-to-frequency denoising, serves as the mirror counterpart of the forward diffusion process (frequency-to-time noise injection). Empirically, the time-domain branch is effective in stabilizing global structures and low-frequency trends, while the frequency-domain branch focuses on high-frequency component, together forming a coarse-to-fine denoising mechanism. The forward and reverse process is illustrated in Fig.2.

3.1.3 Loss Function

As stated in Sec.3.1.1 and Sec.3.1.2, the overall noise injection process consists of two successive sub-processes, *i.e.,* time and frequency domain noise injection. We adopt the commonly used noise estimation approach in diffusion models. The loss function consists of three components: (i) time domain noise estimation loss, *i.e.,* the l_2 distance between the estimated time domain noise and forward time domain noise, (ii) frequency domain noise estimation loss, *i.e.,* the l_2 distance between the estimated frequency domain noise and forward frequency domain noise. (iii) consistency loss, *i.e.,* the l_2 distance between total noise from step k to step $k-1$. This consistency term mitigates potential mismatches arising from estimating the two branches separately and discourages arbitrary residual reallocation across branches. Accordingly, the overall loss is defined as:

$$\mathcal{L}^d = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0),k}(\|\boldsymbol{\epsilon}_k^t - \hat{\boldsymbol{\epsilon}}_k^t\|_2^2 + \|\boldsymbol{\epsilon}_k^f - \mathcal{F}^{-1}(\Lambda \hat{\boldsymbol{\epsilon}}_k^f)\|_2^2 + \omega\|(\boldsymbol{\epsilon}_k^t + \boldsymbol{\epsilon}_k^f) - (\hat{\boldsymbol{\epsilon}}_k^t + \mathcal{F}^{-1}(\Lambda \hat{\boldsymbol{\epsilon}}_k^f))\|_2^2), \quad (14)$$

where ω is the weight for consistency loss. In addition, since our model is specifically designed for time series imputation task, we focus solely on the reconstruction error over masked regions during training, *i.e.,* $\mathcal{L} = \mathbf{M}^t \odot \mathcal{L}^d$. The detailed training and sampling algorithm is presented in Sec.A.7.

3.2 Frequency-aware Diffusion Embedding

Considering the relationship between denoising steps and the frequency components of the data, we present the following proposition [32–34]:

Proposition 3.1 (Frequency Components in the Diffusion Process)**.** *Consider the forward process where the noise is isotropic and has cumulative spectral energy N_t at diffusion step t. Partition the frequency axis into ordered bands $\{B_b\}_{b=1}^B$ from low to high frequency, and let E_b denote the band-averaged clean-signal energy in band B_b. If the signal satisfies the relative low-pass condition $E_{b_1} \geq E_{b_2}$ for any $b_1 < b_2$, then for any SNR threshold $\gamma > 0$, the threshold time $\tau_b(\gamma) = \inf\left\{t \in [0,T] : \frac{E_b}{N_t} \leq \gamma\right\}$ satisfies $\tau_{b_2}(\gamma) \leq \tau_{b_1}(\gamma), \forall b_1 < b_2$. That is, higher-frequency bands reach the same degradation threshold no later than lower-frequency bands during the forward process. Equivalently, the reverse process tends to recover low-frequency structures before progressively restoring high-frequency details.*

Proof. The proof is in Appendix.A.4. □

Motivated by Prop. 3.1, we propose a frequency-aware diffusion embedding to capture the stage-dependent recoverability of different frequency components during denoising. As the noise level changes across diffusion steps, different frequency bands should be emphasized differently. To this end, the proposed embedding integrates three components: **noise-aware frequency gate**, **stage-aware frequency schedule**, and **missing-aware band reliability**.

Noise-aware Frequency Gate. Given the accumulated diffusion noise, whose expected spectral energy satisfies $\mathbb{E}[|\hat{\epsilon}_t(\omega)|^2] = \int_0^t |g(s)|^2 \mathrm{d}s$, the diffusion step t corresponds to a frequency-agnostic noise scale $N_t = \int_0^t |g(s)|^2 \mathrm{d}s$, which decreases monotonically in the reverse process. Since different bands have different signal-to-noise ratios, treating all frequency components uniformly is suboptimal. We therefore introduce a noise-aware gate, with calibration parameters γ and s controlling the gating threshold and its scale:

$$\hat{g}_b(t) = \sigma\left(\frac{\log P_b^{\mathrm{sig}} - \log(\gamma N_t s)}{\tau}\right), \quad (15)$$

where P_b^{sig} is the spectral energy of frequency band b computed from the signal proxy detailed in Sec. 3.3.1, and τ controls the sharpness of the gate. To avoid numerical issues, we set a minimum value $g_{\min}$:

$$g_b(t) = g_{\min} + (1 - g_{\min})\hat{g}_b(t). \quad (16)$$

Stage-aware Frequency Schedule. Following Prop. 3.1, we further encode the progressive low-to-high frequency recovery process into the diffusion embedding. For each band b, the stage-dependent scheduling weight $S_b(t)$ measures its relative importance at diffusion step t:

$$S_b(t) = \exp\left(-\left(\frac{\omega_b}{c(t)}\right)^p\right), \quad (17)$$

where ω_b is the normalized frequency coordinate, $c(t)$ gradually relaxes the effective bandwidth as denoising progresses, and $p > 0$ is a constant. Unlike the noise-aware gate, $S_b(t)$ depends only on t and ω_b, providing a stage-dependent frequency prior independent of actual spectral energy or noise magnitude.

Band Reliability. For time series imputation, missing values can introduce spectral artifacts and weaken the reliability of frequency-domain modeling. We therefore define a missing-aware reliability weight for each frequency band:

$$R_b = \frac{P_b^{\text{sig}}}{P_b^{\text{sig}} + \kappa P_b^{\text{mask}}}, \tag{18}$$

where P_b^{mask} denotes the spectral energy induced by the missing pattern in band b, and κ is a constant.

Finally, we combine the noise-aware gate, stage-aware schedule, and band reliability through element-wise multiplication, and project the result into the embedding space:

$$\mathbf{g}(t) = \text{Proj}\left(g(t) \odot S(t) \odot R,\ \frac{d}{2}\right), \tag{19}$$

where $\text{Proj}(\cdot, \frac{d}{2})$ resamples $\lfloor \frac{L}{2} \rfloor + 1$ frequency bins to $\frac{d}{2}$ embedding dimensions. The resulting gating vector modulates sinusoidal basis functions to form the frequency-aware diffusion embedding:

$$\mathbf{e}(t) = \left[\sin(t\,\mathbf{f}_d) \odot \mathbf{g}(t),\ \cos(t\,\mathbf{f}_d) \odot \mathbf{g}(t)\right], \tag{20}$$

where $\mathbf{f}_d$ denotes a linearly spaced frequency grid and t is the diffusion step. Details are provided in Appendix A.5.

3.3 Model Details

3.3.1 Missing-aware Frequency Modeling

Direct frequency transformation of the masked series $\tilde{\mathbf{X}} = \mathbf{M} \odot \mathbf{X}$ often induces spectral leakage, distorting the true spectral morphology. To mitigate this, we introduce a Signal Proxy mechanism that serves as a diffusion-synchronized spectral reference. Designed to ensure observational consistency and reliable estimation, the proxy evolves with the diffusion process to reflect the changing signal-to-noise ratio. Formally, we define the signal proxy $\tilde{\mathbf{x}}_{\text{proxy}}$ at step t as:

$$\tilde{\mathbf{x}}_{\text{proxy}} = \mathbf{M} \odot \mathbf{x}_{\text{obs}} + (1 - \mathbf{M}) \odot \mathbf{x}_t \tag{21}$$

where $\mathbf{x}_{\text{obs}}$ denotes the observed values and $\mathbf{x}_t$ is the temporal latent variable. This formulation preserves the authentic signal structure at observed locations while populating missing intervals with state estimates consistent with the current diffusion phase. This diffusion-aware mechanism effectively minimizes spectral interference, enabling robust frequency representation learning.

3.3.2 Model Architecture

We present the details of the denoising network. We adopt a cascaded dual-branch network as the core denoising architecture, where both branches share the same network structure. Considering the intra-channel and inter-channel dependencies in multivariate time series, the temporal-feature transformer architecture from CSDI [4] is adopted as the diffusion backbone.

Specifically, our model takes the noisy data, conditional information and step embedding as input. The input is first processed by a temporal attention module and a feature attention module for feature extraction, followed by a linear projection to estimate time-domain noise $\hat{\epsilon}_t$. To ensure that the frequency-domain branch focuses exclusively on estimating frequency-domain noise, we transform the output of the time-domain branch into the frequency domain via a given transformation $\mathcal{F}$ and use it as the input to the frequency-domain branch. The frequency-domain branch is symmetric to the time-domain branch, except that the step embedding is replaced with the frequency-aware diffusion embedding introduced in Sec. 3.2. Finally, the frequency-domain output is mapped back to the time domain via $\mathcal{F}^{-1}$. Overall, the model produces two outputs: the estimated time-domain noise $\hat{\boldsymbol{\epsilon}}_t$ and frequency-domain noise $\hat{\boldsymbol{\epsilon}}_f$. The details of our model architecture is in the Appendix A.6.

Table 1: Imputation performance on PhysioNet and Air Quality datasets. Best results are in **bold**; second-best are underlined

METHOD	PHYSIONET 10%		PHYSIONET 50%		PHYSIONET 90%		AIR QUALITY	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MEAN	0.714	1.035	0.711	1.091	0.710	1.097	50.685	66.558
LERP	0.372	0.708	0.417	0.840	0.565	0.993	15.363	27.658
BRITS	0.278	0.693	0.385	0.833	0.560	0.975	16.519	26.775
GPVAE	0.469	0.783	0.521	0.907	0.642	1.038	23.941	40.586
SSGAN	0.323	0.662	0.449	0.852	0.670	1.060	32.999	48.951
TIMESNET	0.375	0.690	0.453	0.840	0.642	1.031	22.685	39.336
CSDI	0.215	0.491	0.307	0.673	0.492	0.851	9.347	18.713
SAITS	0.232	0.583	0.315	0.735	0.565	0.971	15.424	30.558
MODERNTCN	0.351	0.697	0.440	0.803	0.647	1.026	24.089	40.052
LSCD	0.212	0.457	0.305	0.658	0.492	0.845	9.340	18.270
FSDI	**0.202**	0.465	0.298	0.658	0.468	0.817	9.349	17.869
HYFAD (OURS)	0.206	**0.423**	**0.288**	**0.622**	**0.448**	**0.782**	**9.296**	**17.730**

4 Experiments

Experimental Settings. All experiments are implemented in Python 3.12 with Pytorch [35] on a single Nvidia RTX 3090 GPU. We use Adam [36] with an initial learning rate of 1×10^{-3} and a multi-step scheduler that decays the learning rate by 10 at 75% and 90% of the total epochs. Quadratic noise schedules are adopted for both time- and frequency-domain diffusion, and the frequency transformation $\mathcal{F}$ is set to rDFT as described in Sec. 2.1.

Datasets and Evaluation Metrics. We evaluate HyFAD on two widely used imputation benchmarks: PhysioNet [37] and Air Quality [38]. We report MAE and RMSE between the imputed values and ground truth at missing positions. More details are provided in Appendix A.8.

Baselines. We compare HyFAD with classical methods (Mean and Lerp), deep learning approaches (BRITS [39], SAITS [40], TimesNet [41]), deep generative models (GP-VAE [42], SSGAN [43], CSDI [4], LSCD [44], FSDI [45]), and a time-series foundation model (ModernTCN [46]). All baselines are re-run in our environment with identical data splits, missing masks and evaluation metrics.

4.1 Time Series Imputation Results

We evaluate HyFAD under different missing-data scenarios on two real-world datasets, *i.e.,* PhysioNet and Air Quality. Table 1 reports the quantitative results, with visualizations provided in Appendix A.13.

As shown in Table 1, HyFAD achieves the best results on 7 of the 8 metrics. The only exception is MAE under 10% missingness on PhysioNet, where the frequency-shaped noise schedule of FSDI is slightly better (0.202 vs. 0.206). The advantage of HyFAD grows with the missing rate: under 90% missingness it reduces both MAE and RMSE by 4.3% relative to the second-best method. On Air Quality, HyFAD also obtains the best MAE and RMSE.

4.2 Ablation Studies and Parameter Analysis

To validate the effectiveness of the proposed frequency-aware diffusion embedding, we replace it with the diffusion embedding in [4]. We further analyze several key hyperparameters, including the terminal values of the noise schedules in time- and frequency-domain diffusion, *i.e.,* β^t_{end} and β^f_{end}, the balancing parameter λ that controls the relative strengths of the two diffusion processes, and ω, the weight of the consistency loss.

As shown in Table 2, the proposed Frequency-Aware Diffusion Embedding consistently outperforms the standard diffusion embedding across all missing rates on PhysioNet. In addition, since CSDI can be viewed as the time-domain-only counterpart of HyFAD, the consistent improvements over CSDI in Table 1 further validate the necessity of hybrid time-frequency modeling. These results support the core design of HyFAD: the time branch preserves global structure, while the frequency branch complements local dynamics and high-frequency details.

For noise allocation, the best performance is achieved when the temporal process is dominant ($\lambda = 0.75$). The performance drops at $\lambda = 0.25$ and 0.5 indicate that frequency-domain diffusion is more suitable for refining local dynamics that are not fully captured in the time domain, rather than

Table 2: Results of ablation studies and parameter analysis. "FA" stands for the proposed frequency-aware diffusion embedding and "DE" stands for the diffusion embedding in [4].

Embedding	λ	β^t_{end}	β^f_{end}	ω	PhysioNet 10%		PhysioNet 50%		PhysioNet 90%	
					RMSE	MAE	RMSE	MAE	RMSE	MAE
FA	0.75	0.5	0.05	0.4	0.423	0.206	0.622	0.288	0.782	0.448
DE	0.75	0.5	0.05	0.4	0.439	0.208	0.638	0.291	0.798	0.451
FA	**0.25**	0.5	0.05	0.4	0.471	0.235	0.661	0.320	0.840	0.489
FA	**0.5**	0.5	0.05	0.4	0.503	0.216	0.660	0.299	0.815	0.462
FA	0.75	**0.25**	0.05	0.4	0.451	0.210	0.664	0.294	0.806	0.455
FA	0.75	**0.75**	0.05	0.4	0.475	0.209	0.667	0.312	0.810	0.473
FA	0.75	0.5	**0.01**	0.4	0.490	0.213	0.650	0.294	0.806	0.456
FA	0.75	0.5	**0.1**	0.4	0.433	0.210	0.639	0.290	0.797	0.451
FA	0.75	0.5	**0.25**	0.4	0.464	0.225	0.661	0.306	0.812	0.467
FA	0.75	0.5	**0.5**	0.4	0.581	0.261	0.700	0.341	0.819	0.475
FA	0.75	0.5	0.05	**0.0**	0.432	0.209	0.629	0.290	0.798	0.451
FA	0.75	0.5	0.05	**0.2**	0.441	0.208	0.649	0.290	0.800	0.45
FA	0.75	0.5	0.05	**0.6**	0.441	0.206	0.641	0.289	0.797	0.449
FA	0.75	0.5	0.05	**0.8**	0.430	0.205	0.652	0.289	0.799	0.451
FA	0.75	0.5	0.05	**1.0**	0.441	0.205	0.651	0.288	0.794	0.447

serving as the primary noise carrier. For diffusion schedules, the optimal temporal noise level appears at $\beta^t_{\text{end}} = 0.5$, while either excessive (0.75) or insufficient (0.25) noise weakens the model's ability to learn the underlying data distribution. Regarding β^f_{end}, small values provide insufficient spectral information for refinement, whereas overly large values over-regularize the temporal branch and lead to degraded performance.

Finally, $\omega = 0.4$ achieves the best overall trade-off, yielding the lowest RMSE across all missing rates while maintaining competitive MAE. Smaller weights lead to insufficient coupling, whereas larger weights may slightly improve some MAE values but generally degrade RMSE, suggesting over-constrained cross-domain fitting.

5 Related Works

Time series imputation aims to recover complete sequences from partially observed data. Early methods such as mean and linear interpolation (Lerp) rely on simple statistical assumptions and often fail to capture complex temporal dynamics. Deep learning-based approaches, including BRITS [39], SAITS [40], and TimesNet [41], improve imputation by modeling complex temporal dependencies.

Diffusion models have recently been introduced to time series imputation [45, 47], motivated by their success in other domains [48, 9, 49]. CSDI [4] formulates imputation as conditional score matching with observed values as conditions, while SSSD [5] incorporates state space models for temporal modeling. Recent works also explore frequency-domain information: FGTI [20] introduces frequency-space conditions for multi-domain representation learning, and FourierDiffusion [22] transforms signals into the spectral domain and conducts diffusion entirely in frequency space. FSDI [45] reformulates the forward process with a Parseval-calibrated, per-frequency variance schedule derived from the dataset's spectral energy, without architectural changes.

Frequency-domain modeling [13, 50–52] has also been widely studied in time-series analysis, as spectral characteristics provide complementary information beyond the time domain. Fedformer [53] combines decomposition with a frequency-domain Transformer and frequency mode selection for efficient long-term forecasting. FiLM [54] uses Legendre memory with Fourier- and low-rank-based frequency-enhanced layers to preserve dominant low-frequency patterns and suppress noise. TFAD [55] combines time- and frequency-domain analysis with sequence decomposition, context comparison, and data augmentation for anomaly detection, where the frequency branch mainly provides a discriminative representation space.

HyFAD differs from prior works by explicitly coupling time- and frequency-domain diffusion for time-series imputation. Rather than using frequency information only for representation learning, HyFAD performs forward and reverse diffusion directly in both domains. It further introduces frequency-aware diffusion embedding and step-aware spectral guidance to support progressive denoising and reconstruction, improving recovery of both global trends and fine-grained temporal variations. In contrast to FSDI, which reshapes the noise allocation across frequencies, HyFAD models the reconstruction trajectory itself, coupling time- and frequency-domain denoising with stage-dependent spectral guidance.

6 Conclusion

In this paper, we propose HyFAD, a hybrid time-frequency diffusion model with frequency-aware diffusion embedding. By integrating time- and frequency-domain processes within a unified diffusion framework, HyFAD enables coarse-to-fine reconstruction. The proposed frequency-aware diffusion-step embedding adaptively emphasizes appropriate spectral components during denoising, enhancing reconstruction of high-frequency components. Experiments demonstrate consistently strong performance across multiple datasets and varying missing rates.

Acknowledgments and Disclosure of Funding

Jilin Hu is the corresponding author of the work. We gratefully acknowledge the support of the ECNU Multifunctional Platform for Innovation (001) for providing computational resources.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020.

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

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

[4] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, pages 24804–24816, 2021.

[5] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Trans. Mach. Learn. Res.*, 2023.

[6] Xingjian Wu, Junkai Lu, Zhengyu Li, Xiangfei Qiu, Jilin Hu, Chenjuan Guo, Christian S Jensen, and Bin Yang. TimeART: Towards agentic time series reasoning via tool-augmentation. *arXiv preprint arXiv:2601.13653*, 2026.

[7] Hanyin Cheng, Jingrong Zhou, Yang Shu, and Chenjuan Guo. KITE: Knowledge-guided probabilistic modeling for time series forecasting with exogenous variables. In *ICML*, 2026.

[8] Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *NeurIPS*, 2025.

[9] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *ICML*, pages 8857–8868, 2021.

[10] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *NeurIPS*, pages 28341–28364, 2023.

[11] Hanyin Cheng, Ruitong Zhang, Yuning Lu, Peng Chen, Meng Wang, Yang Shu, Bin Yang, and Chenjuan Guo. STAR: Boosting time series foundation models for anomaly detection through state-aware adapter. In *NeurIPS*, 2026.

[12] Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. CoRA: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *ICLR*, 2026.

[13] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. In *Proc. VLDB Endow.*, pages 2363–2377, 2024.

[14] Xiangfei Qiu, Yuhan Zhu, Zhengyu Li, Xingjian Wu, Bin Yang, and Jilin Hu. DAG: A dual correlation network for time series forecasting with exogenous variables. In *ICML*, 2026.

[15] Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. In *NeurIPS*, pages 61048–61059, 2023.

[16] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *ICLR*, 2024.

[17] Hanyin Cheng, Xingjian Wu, Xiangfei Qiu, Yang Shu, Bin Yang, and Chenjuan Guo. CCD: Capturing cross-correlations with deformable convolutional networks for multivariate time series forecasting. In *KDD*, 2026.

[18] Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. In *NeurIPS*, 2025.

[19] Asadullah Hill Galib, Pang-Ning Tan, and Lifeng Luo. FIDE: frequency-inflated conditional diffusion model for extreme-aware time series generation. In *NeurIPS*, pages 114434–114457, 2024.

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

[21] Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard E. Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models. *CoRR*, abs/2505.11278, 2025.

[22] Jonathan Crabbé, Nicolas Huynh, Jan Stanczuk, and Mihaela van der Schaar. Time series diffusion in the frequency domain. In *ICML*, pages 9407–9438, 2024.

[23] Lifeng Shen, Weiyu Chen, and James T. Kwok. Multi-resolution diffusion models for time series forecasting. In *ICLR*, 2024.

[24] Ilan Naiman, Nimrod Berman, Itai Pemper, Idan Arbiv, Gal Fadlon, and Omri Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series. In *NeurIPS*, pages 121699–121730, 2024.

[25] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *ICLR*, 2025.

[26] Chenxi Wang, Linxiao Yang, Zhixian Wang, Liang Sun, and Yi Wang. A non-isotropic time series diffusion model with moving average transitions. In *ICML*, pages 65144–65166, 2025.

[27] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *CVPR*, pages 22552–22562, 2023.

[28] Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. TAB: Unified benchmarking of time series anomaly detection methods. In *Proc. VLDB Endow.*, pages 2775–2789, 2025.

[29] Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. In *NeurIPS*, pages 10758–10774, 2023.

[30] Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *CoRR*, abs/2106.10121, 2021.

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[32] Mang Ning, Mingxiao Li, Jianlin Su, Haozhe Jia, Lanmiao Liu, Martin Benes, Wenshuo Chen, Albert Ali Salah, and Itir Önal Ertugrul. Dctdiff: Intriguing properties of image generative modeling in the DCT space. In *ICML*, pages 46498–46524, 2025.

[33] Wenshuo Chen, Haozhe Jia, Songning Lai, Keming Wu, Hongru Xiao, Lijie Hu, and Yutao Yue. Free-t2m: Frequency enhanced text-to-motion diffusion model with consistency loss. *CoRR*, abs/2501.18232, 2025.

[34] Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *CVPR*, pages 8911–8920, 2024.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[37] 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.

[38] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. ST-MVL: filling missing values in geo-sensory time series data. In *IJCAI*, pages 2704–2710, 2016.

[39] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: bidirectional recurrent imputation for time series. In *NeurIPS*, pages 6776–6786, 2018.

[40] Wenjie Du, David Côté, and Yan Liu. SAITS: self-attention-based imputation for time series. *Expert Syst. Appl.*, 219:119619, 2023.

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

[42] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: deep probabilistic time series imputation. In *AISTATS*, pages 1651–1661, 2020.

[43] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *AAAI*, pages 8983–8991, 2021.

[44] Elizabeth Fons, Alejandro Sztrajman, Yousef El-Laham, Luciana Ferrer, Svitlana Vyetrenko, and Manuela Veloso. LSCD: lomb-scargle conditioned diffusion for time series imputation. In *ICML*, pages 17411–17436, 2025.

[45] Wangmeng Shen, Hongfan Gao, Qingsong Zhong, Dingli Xu, and Jilin Hu. FSDI: frequency-shaped diffusion for time-series imputation. In *WWW 2026*, pages 5310–5321, 2026.

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

[47] Hongfan Gao, Wangmeng Shen, Xiangfei Qiu, Ronghui Xu, Bin Yang, and Jilin Hu. SSD-TS: exploring the potential of linear state space models for diffusion models in time series imputation. In *KDD*, 2025.

[48] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182, 2023.

[49] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, pages 36479–36494, 2022.

[50] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *SIGKDD*, pages 1185–1196, 2025.

[51] Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *ICML*, 2026.

[52] Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A koopman-kalman enhanced variational autoencoder for probabilistic time series forecasting. In *ICML*, 2025.

[53] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, pages 27268–27286, 2022.

[54] Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. Film: Frequency improved legendre memory model for long-term time series forecasting. In *NeurIPS*, pages 12677–12690, 2022.

[55] Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis. In *CIKM*, pages 2497–2507, 2022.

[56] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205), 2017.

A Appendix

A.1 Choice of $\mathcal{F}$ and Λ

Notably, our framework imposes no strict restrictions on the transform $\mathcal{F}$, as any invertible linear frequency-domain transformation satisfies our requirements. On the other hand, the Λ matrix is intrinsically coupled with the chosen transform $\mathcal{F}$. Any modification to the transform $\mathcal{F}$ necessitates a corresponding adjustment to the Λ matrix to ensure energy conservation between the time and frequency domains. Tab.3 presents the MAE and RMSE results of different transforms.

DFT performs best because the frequency-aware mechanisms operate on frequency bins, which DFT represents directly. DWT coefficients jointly encode scale and temporal location, which does not align with bin-wise modulation, and DCT does not preserve the phase information available in the DFT.

Table 3: Choice of the frequency transform (MAE/RMSE).

Transform	PhysioNet 10%	PhysioNet 50%	PhysioNet 90%	Air Quality
DCT	0.208/0.446	0.296/0.655	0.456/0.808	10.586/21.026
DWT	0.209/0.451	0.297/0.649	0.460/0.815	10.069/18.835
DFT	**0.206/0.423**	**0.288/0.622**	**0.448/0.782**	**9.296/17.730**

A.2 Details in Forward Process

Posterior of Eq.6: Eq.6 presents

$$\mathcal{F}^{-1}(\mathbf{x}_k^f) = \sqrt{\alpha_k^f}\mathbf{x}_{k-1}^t + \sqrt{\beta_k^f}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_k^f), \qquad \boldsymbol{\epsilon}_k^f \sim \mathcal{N}(0, \mathbf{I}). \tag{22}$$

$\mathcal{F}^{-1}(\mathbf{x}_{k-1}^t)$ is a linear transformation of $\mathbf{x}_k^t$ along with a Gaussian term, Therefore, the posterior $q(\mathcal{F}^{-1}(\mathbf{x}_k^f)|\mathbf{x}_{k-1}^t)$ remains gaussian. The posterior mean is $\sqrt{\alpha_k^f}\mathbf{x}_{k-1}^t$ and the standard deviation is

$$(\sqrt{\beta_k^f}\sqrt{1-\lambda}\mathcal{F}^{-1}\Lambda)(\sqrt{\beta_k^f}\sqrt{1-\lambda}\mathcal{F}^{-1}\Lambda)^T = \beta_k^f(1-\lambda)\mathcal{F}^{-1}\Lambda\Lambda^T(\mathcal{F}^{-1})^T \tag{23}$$

Here $\mathcal{F}^{-1}$ and $(\mathcal{F}^{-1})^T$ denotes a matrix $\mathbf{U}$ due to the linearity of transformation $\mathcal{F}$, *i.e.,* $\mathcal{F}(\mathbf{x}) = \mathbf{U}\mathbf{x}$.

Posterior of Eq.7: Eq.7 presents $\mathbf{x}_k^t = \sqrt{\alpha_k^t}\mathcal{F}^{-1}(\mathbf{x}_k^f) + \sqrt{\beta_k^t}\sqrt{\lambda}\boldsymbol{\epsilon}_k^t$, $\boldsymbol{\epsilon}_k^t \sim \mathcal{N}(0, \mathbf{I})$.$\mathbf{x}_k^t$ is a linear transformation of $\mathcal{F}^{-1}(\mathbf{x}_k^f)$ with a Gaussian term $\sqrt{\beta_k^t}\sqrt{\lambda}\boldsymbol{\epsilon}_k^t$. Therefore, the posterior is still Gaussian with mean $\sqrt{\alpha_k^t}\mathcal{F}^{-1}(\mathbf{x}_k^f)$ and standard deviation $\beta_k^t\lambda\mathbf{I}$.

Proof of Eq.9 (the relationship between $\mathbf{x}_k^t$ and $\mathbf{x}_k^0$ in the forward process):

From Eq.8:

$$\mathbf{x}_k^t = \sqrt{\alpha_k^t\alpha_k^f}\mathbf{x}_{k-1}^t + \sqrt{\alpha_k^t(1-\alpha_k^f)}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_k^f) + \sqrt{1-\alpha_k^t}\sqrt{\lambda}\boldsymbol{\epsilon}_k^t \tag{24}$$

we can get:

$$\mathbf{x}_k^t = \sqrt{\bar{\alpha}_k^t\bar{\alpha}_k^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_k^t}{\bar{\alpha}_{s-1}^t}}\sqrt{\frac{\bar{\alpha}_k^f}{\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f) + \sqrt{\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_k^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t \tag{25}$$

Proof. We proceed by mathematical induction.

For $k = 1$:

$$\begin{aligned}
\mathbf{x}_1^t &= \sqrt{\alpha_1^t\alpha_1^f}\mathbf{x}_0^t + \sqrt{\alpha_1^t(1-\alpha_1^f)}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_1^f) + \sqrt{1-\alpha_1^t}\sqrt{\lambda}\boldsymbol{\epsilon}_1^t \\
&= \sqrt{\bar{\alpha}_1^t\bar{\alpha}_1^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sqrt{\beta_1^f}\sqrt{\alpha_1^t}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_1^f) + \sqrt{\lambda}\sqrt{\beta_1^t}\boldsymbol{\epsilon}_1^t \\
&= \sqrt{\bar{\alpha}_1^t\bar{\alpha}_1^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{1}\sqrt{\beta_1^f}\sqrt{\frac{\bar{\alpha}_1^t}{\bar{\alpha}_0^t}}\sqrt{\frac{\bar{\alpha}_1^f}{\bar{\alpha}_1^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_1^t) + \sqrt{\lambda}\sum_{s=1}^{1}\sqrt{\beta_1^t}\sqrt{\frac{\bar{\alpha}_1^t}{\bar{\alpha}_1^t}}\boldsymbol{\epsilon}_1^t
\end{aligned} \tag{26}$$

Therefore, Eq.25 holds when $k = 1$. Suppose Eq.25 holds when $k = m$, *i.e.,*

$$\mathbf{x}_m^t = \sqrt{\bar{\alpha}_m^t\bar{\alpha}_m^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{m}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_m^t}{\bar{\alpha}_{s-1}^t}}\sqrt{\frac{\bar{\alpha}_m^f}{\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f) + \sqrt{\lambda}\sum_{s=1}^{m}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_m^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t \tag{27}$$

For $k = m + 1$:

$$\begin{aligned}
\mathbf{x}_{m+1}^t &= \sqrt{\alpha_{m+1}^t \alpha_{m+1}^f}\mathbf{x}_m^t + \sqrt{\alpha_{m+1}^t(1-\alpha_{m+1}^f)}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_{m+1}^f) + \sqrt{1-\alpha_{m+1}^t}\sqrt{\lambda}\boldsymbol{\epsilon}_{m+1}^t \\
&= \sqrt{\alpha_{m+1}^t \alpha_{m+1}^f}(\sqrt{\bar{\alpha}_m^t \bar{\alpha}_m^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{m}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_m^t}{\bar{\alpha}_{s-1}^t}}\sqrt{\frac{\bar{\alpha}_m^f}{\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f)\sqrt{\lambda}\sum_{s=1}^{m}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_m^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t) \\
&+ \sqrt{\alpha_{m+1}^t(1-\alpha_{m+1}^f)}\sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_{m+1}^f) + \sqrt{1-\alpha_{m+1}^t}\sqrt{\lambda}\boldsymbol{\epsilon}_{m+1}^t \\
&= \sqrt{\bar{\alpha}_{m+1}^t \bar{\alpha}_{m+1}^f}\mathbf{x}_0^t + \sqrt{1-\lambda}\sum_{s=1}^{m+1}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_{m+1}^t}{\bar{\alpha}_{s-1}^t}}\sqrt{\frac{\bar{\alpha}_{m+1}^f}{\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f) + \sqrt{\lambda}\sum_{s=1}^{m+1}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_{m+1}^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t
\end{aligned} \tag{28}$$

Therefore, Eq.25 holds for arbitrary k. □

Posterior of Eq.9: Eq.9 indicates $\mathbf{x}_k^t$ is linear transformation of $\mathbf{x}_0^t$ with two groups of independent gaussian noise. Therefore, $q(\mathbf{x}_k^t|\mathbf{x}_0^t)$ is still gaussian with the standard deviation of the sum of two the two groups of gaussian noise.

A.3 Details in the reverse process

Noise prior in the reverse process. At the end of the forward process, $\bar{\alpha}_k^t, \bar{\alpha}_k^f \to 0$, therefore, the mean of $\mathbf{x}_k^t$ is 0. For the standard deviation term, it is the linear combination of two independent gaussian noises, so the reverse process starts from $\mathbf{x}_T = \sqrt{\lambda}\boldsymbol{\epsilon}^t + \sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}^f), \boldsymbol{\epsilon}^t, \boldsymbol{\epsilon}^f \sim \mathcal{N}(0, \mathbf{I})$.

A.4 Proof of Proposition 3.1

Proposition A.1 (Frequency Components in the Diffusion Process)**.** *Consider the forward process where the noise is isotropic and has cumulative spectral energy N_t at diffusion step t. Partition the frequency axis into ordered bands $\{B_b\}_{b=1}^B$ from low to high frequency, and let E_b denote the band-averaged clean-signal energy in band B_b. If the signal satisfies the relative low-pass condition $E_{b_1} \geq E_{b_2}$ for any $b_1 < b_2$, then for any SNR threshold $\gamma > 0$, the threshold time $\tau_b(\gamma) = \inf\left\{t \in [0,T] : \frac{E_b}{N_t} \leq \gamma\right\}$ satisfies $\tau_{b_2}(\gamma) \leq \tau_{b_1}(\gamma), \forall b_1 < b_2$. That is, higher-frequency bands reach the same degradation threshold no later than lower-frequency bands during the forward process. Equivalently, the reverse process tends to recover low-frequency structures before progressively restoring high-frequency details.*

Proof. Considering a diffusion process

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{t})\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w_t} \tag{29}$$

where $\boldsymbol{w_t}$ is the standard Wiener process in $\mathbb{R}^{\mathrm{d}_{\boldsymbol{x}}}$, $\boldsymbol{f} : \mathbb{R}^{\mathrm{d}_{\boldsymbol{x_t}}} \times [0,T] \to \mathbb{R}^{\mathrm{d}_{\boldsymbol{x_t}}}$ is the drift and $g(t) : [0,T] \to \mathbb{R}^{N\times N}$ is the diffusion coefficient. The solution to Eq.29 is formulated as:

$$\mathbf{x}_t = \mathbf{x}_0 + \int_0^t \boldsymbol{f}(\mathbf{x}_s, s)\mathrm{d}s + \int_0^t g(s)\mathrm{d}\boldsymbol{w_s} \tag{30}$$

By applying DFT to Eq.30, we have:

$$\hat{\mathbf{x}}_t(\omega) = \hat{\mathbf{x}}_0(\boldsymbol{\omega}) + \hat{f}(\boldsymbol{\omega}) + \hat{\boldsymbol{\epsilon}}_t(\boldsymbol{\omega}), \tag{31}$$

where $\hat{\mathbf{x}}_t(\omega)$ is the DFT of the noisy item and satisfies: $\mathbb{E}[\hat{\boldsymbol{\epsilon}}_t(\omega)] = 0$ and $\mathbb{E}[|\hat{\boldsymbol{\epsilon}}_t(\omega)|^2] = \int_0^t |g(s)|^2\mathrm{d}s$. The signal-noise-ratio (SNR) of frequency ω is defined as:

$$\mathrm{SNR}(\omega) = \frac{|\hat{\mathbf{x}}_t(\omega)|^2}{\mathbb{E}[|\hat{\boldsymbol{\epsilon}}_t(\omega)|^2]} = \frac{|\hat{\mathbf{x}}_t(\omega)|^2}{\int_0^t |g(s)|^2\mathrm{d}s} \tag{32}$$

We partition the frequency axis into B disjoint bands $\{\mathcal{B}_b\}_{b=1}^B$ from low to high frequency. Let $N_t := \int_0^t |g(s)|^2\,\mathrm{d}s$ denote the cumulative noise intensity up to diffusion step t. We define the band-averaged signal energy as $E_b := \mathbb{E}_{\omega\in\mathcal{B}_b}\left[\mathbb{E}\left[|\hat{\mathbf{x}}_0(\omega)|^2\right]\right]$. Under the assumption that the second-order

statistics of the transformed noise do not explicitly depend on frequency, the band-level SNR for any band b can be written as

$$\mathrm{SNR}_b(t) = \frac{E_b}{N_t}.$$

For any $\gamma > 0$, we have

$$\mathrm{SNR}_b(t) \le \gamma \iff \frac{E_b}{N_t} \le \gamma \iff N_t \ge \frac{E_b}{\gamma}.$$

For any $b_1 < b_2$, the relative low-pass assumption gives

$$E_{b_1} \ge E_{b_2} \quad \Longrightarrow \quad \frac{E_{b_1}}{\gamma} \ge \frac{E_{b_2}}{\gamma}.$$

Therefore, if at some time t,

$$N_t \ge \frac{E_{b_1}}{\gamma},$$

then it must also hold that

$$N_t \ge \frac{E_{b_2}}{\gamma}.$$

Equivalently, if the lower-frequency band b_1 already satisfies

$$\mathrm{SNR}_{b_1}(t) \le \gamma,$$

then the higher-frequency band b_2 must also satisfy

$$\mathrm{SNR}_{b_2}(t) \le \gamma.$$

This shows that, in the forward diffusion process, higher-frequency bands reach the same SNR threshold no later than lower-frequency bands. Furthermore, since N_t is non-decreasing with respect to t, the first threshold-crossing times satisfy

$$\tau_{b_2}(\gamma) \le \tau_{b_1}(\gamma).$$

□

A.5 Details of Frequency-aware Diffusion Embedding

Factor s in Noise-aware Frequency Gate. Due to substantial variations in spectral energy scales across different datasets and variables, we introduce a global calibration factor s to align the magnitude of band-wise spectral energy, which ensures the thresholding operation in the frequency-aware diffusion embedding is performed on a consistent relative scale.

During training, the factor s is updated per batch using a global exponential moving average (EMA). Specifically, given the band-wise spectral energy P_b^s of the current batch, s_0 is initialized as the median of P_b^s:

$$s_0 = \mathrm{median}(P_b^s), \tag{33}$$

s is then updated as:

$$s_k \leftarrow \alpha s_{k-1} + (1-\alpha)\hat{s}_k, \tag{34}$$

where $\alpha \in (0,1)$ denotes the EMA decay rate and we set $\alpha = 0.99$ in our implementation, s_{k-1} is the factor after batch $k-1$ and $\hat{s}_k$ is the observations from batch k, *i.e.,* $\hat{s}_k = \mathrm{median}(P_k^s)$. During inference, the factor s is fixed to the value accumulated during training and is no longer updated. This design enforces a strict separation between training and testing statistics, thereby preventing potential data leakage and ensuring consistent gating behavior across test samples.

ω_b and $c(t)$ in Stage-aware Frequency Schedule. Let $b \in \{0, 1, \cdots, B-1\}$ index the $B = \lfloor \frac{L}{2} \rfloor + 1$ frequency bands. The relative position of the b-th band is:

$$\tilde{\omega}_b = \frac{b}{B-1}, \tag{35}$$

To align the relative frequency coordinate with the numerical scale of the stage-dependent cutoff, we further multiply a max frequency $\omega_{\max}$:

$$\omega_b = \tilde{\omega}_b \cdot \omega_{\max} \tag{36}$$

The function $c(t)$ is a stage-dependent frequency scaling function that controls the effective frequency range emphasized by the model at diffusion step t, which is defined as:

$$c(t) = c_{\min} + (c_{\max} - c_{\min}) \left(1 - \frac{t}{T}\right)^q, \tag{37}$$

where t is the current diffusion step and T is the total number of diffusion steps. $c_{\min}$ and $c_{\max} = \omega_{\max}$ are the minimum and maximum cutoff frequency, $q > 0$ is a transition exponent. $c(t)$ remains close to $c_{\min}$ during the early stages of the reverse diffusion process, thereby emphasizing low-frequency components. As the denoising process progresses, $c(t)$ gradually approaches $c_{\max}$, allowing the model to progressively incorporate higher-frequency details.

Embedding Details. The projection operator $\mathrm{Proj}(\cdot, \frac{d}{2})$ in Eq.19 is implemented via one-dimensional linear interpolation along the frequency axis using `torch.nn.functional.interpolate` (`mode='linear'`), which resamples the frequency-wise gating vector from $\lfloor L/2 \rfloor + 1$ frequency bins to $\frac{d}{2}$ embedding dimensions.

The frequency grid $\mathbf{f}_d = [f_1, f_2, \cdots, f_{\frac{d}{2}}] \in \mathbb{R}^{\frac{d}{2}}$ in Eq.20 is constructed by uniform sampling from $[0, f_{\max}]$:

$$f_i = \frac{i-1}{\frac{d}{2}-1} f_{\max} \tag{38}$$

A.6 Architecture Details

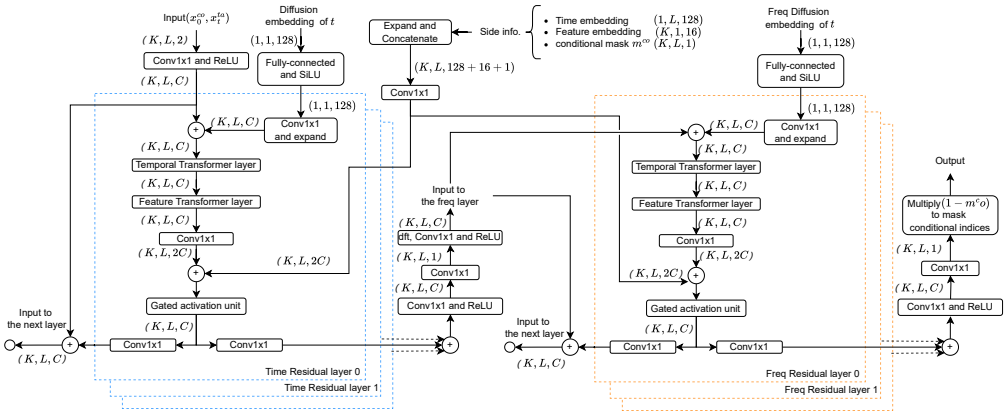


Figure 3: Architecture details of our denoising model ϵ_θ.

Fig.3 presents the diffusion backbone of our denoising block. The block adopts a time-frequency coupled dual-branch residual The input sequence is first projected using a 1×1 convolution followed by a ReLU activation. Time and Frequency diffusion step embeddings are generated via fully connected layers with SiLU activations and then concatenated with external side information.

During residual modeling, the network first processes the representation in the time-domain branch. After an initial 1×1 convolution and ReLU projection, the features are passed through a temporal Transformer layer and a feature Transformer layer to capture inter-channel and intra-channel dependencies. The outputs are fused using a gated activation unit (GAU) and further refined by stacked 1×1 convolutions. Residual connections feed the results back into the input stream to form the input for the next layer.

Then the time-domain representation is then fed into the frequency-domain branch with a similar structure to get the frequency domain output,. By processing in a time-to-frequency order, the architecture establishes a complementary relationship between global trend modeling and fine-grained frequency refinement.

A.7 Training and Sampling Details

The detailed training and sampling algorithm is presented in Alg.1 and 2.

Algorithm 1 Training Procedure of HyFAD

Input: Observed sequence $\mathbf{x}_0$, condition mask $\mathbf{M}^{\text{cond}}$, observation mask $\mathbf{M}^{\text{obs}}$, side information $\mathbf{s}$, number of steps T,number of iterations N, time- and frequency-domain scheduler β^t, β^f, total denoising step T, transformation $\mathcal{F}$, noise balance coefficient λ, consistency weight ω.
Output: Denoising function $\boldsymbol{\epsilon}_\theta^t, \boldsymbol{\epsilon}_\theta^f$
for $i = 1$ **to** T **do**
  $k \sim \text{Uniform}(\{1, 2, \cdots, T\})$
  Calculate time-domain noise:$\boldsymbol{\epsilon}_k^t = \sqrt{\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^t}\sqrt{\frac{\bar{\alpha}_k^t}{\bar{\alpha}_s^t}}\boldsymbol{\epsilon}_s^t$, $\boldsymbol{\epsilon}_s^t \sim \mathcal{N}(0, \mathbf{I})$.
  Calculate frequency-domain noise:$\boldsymbol{\epsilon}_k^f = \sqrt{1-\lambda}\sum_{s=1}^{k}\sqrt{\beta_s^f}\sqrt{\frac{\bar{\alpha}_k^t}{\bar{\alpha}_{s-1}^t}}\sqrt{\frac{\bar{\alpha}_k^f}{\bar{\alpha}_s^f}}\mathcal{F}^{-1}(\Lambda\boldsymbol{\epsilon}_s^f), \boldsymbol{\epsilon}_s^f \sim \mathcal{N}(0, \mathbf{I})$.
  Calculate noisy sample at step k:$\mathbf{x}_k^t = \sqrt{\alpha_k^t\alpha_k^f}\mathbf{x}_0^t + \boldsymbol{\epsilon}_k^t + \boldsymbol{\epsilon}_k^f$
  Construct time domain input: $\mathbf{h}^t \leftarrow \texttt{set_input}(\mathbf{x}_k^t, \mathbf{X}^{\text{obs}}, \mathbf{M}^{\text{cond}})$
  Estimate time domain noise and calculate time noise loss:$\hat{\boldsymbol{\epsilon}}_k^t = \boldsymbol{\epsilon}_\theta^t(\mathbf{h}_t, \mathbf{s}, k)$, $\mathcal{L}^t = \|\hat{\boldsymbol{\epsilon}}_k^t - \boldsymbol{\epsilon}_k^t\|_2^2$
  Update and construct frequency domain input:$\mathcal{F}^{-1}(\mathbf{x}_k^f) = \frac{1}{\sqrt{\alpha_k^t}}\left(\mathbf{x}_k^t - \frac{\beta_k^t}{\sqrt{1-\bar{\alpha}_k^t}}\hat{\boldsymbol{\epsilon}}_k^t\right)$
  Construct frequency domain input:$\mathbf{h}^f = \texttt{set_input}(\mathcal{F}^{-1}(\mathbf{x}_k^t), \mathbf{s}, k)$
  Estimate frequency domain noise and calculate frequency noise loss: $\hat{\boldsymbol{\epsilon}}_k^f = \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{h}^f, \mathbf{s}, k)$, $\mathcal{L}^f = \|\mathcal{F}^{-1}(\Lambda\hat{\boldsymbol{\epsilon}}_k^f) - \boldsymbol{\epsilon}_k^f\|_2^2$
  Calculate consistency loss:$\mathcal{L}_c = \|(\boldsymbol{\epsilon}_k^t + \boldsymbol{\epsilon}_k^f) - (\hat{\boldsymbol{\epsilon}}_k^t + \mathcal{F}^{-1}(\Lambda\hat{\boldsymbol{\epsilon}}_k^f))\|_2^2$
  Take gradient descent step on $\nabla_\theta\left[(1-\mathbf{M}^{\text{cond}})\cdot(\mathcal{L}^t + \mathcal{L}^f + \omega\mathcal{L}_c)\right]$
end for
return $\boldsymbol{\epsilon}_\theta^t, \boldsymbol{\epsilon}_\theta^f$

Algorithm 2 Sampling Procedure of HyFAD

Input: Trained denoising function $\boldsymbol{\epsilon}_\theta^t, \boldsymbol{\epsilon}_\theta^f$, sampling step T, observed data $\mathbf{X}^{\text{obs}}$, condition mask $\mathbf{M}^{\text{cond}}$, side information $\mathbf{s}$, time domain scheduler β^t, frequency domain scheduler β^f
Output: Generated Sequence $\mathbf{x}_0$
Initialize by sampling from noise prior: $\mathbf{x}_T = \sqrt{\lambda}\mathbf{z}^t + \sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\mathbf{z}^f)$. $\mathbf{z}^t, \mathbf{z}^f \sim \mathcal{N}(0, \mathbf{I})$
for $k = T$ **to** 1 **do**
  Construct time-domain input:$\mathbf{h}_k^t \leftarrow \texttt{set_input}(\mathbf{x}_k^t, \mathbf{X}^{\text{obs}}, \mathbf{M}^{\text{cond}})$
  Estimate time-domain noise:$\hat{\boldsymbol{\epsilon}}_k^t \leftarrow \boldsymbol{\epsilon}_\theta^t(\mathbf{h}_k^t, \mathbf{s}, k)$
  Update input according to time-domain noise:$\mathcal{F}^{-1}(\mathbf{x}_k^f) = \frac{1}{\sqrt{\alpha_k^t}}\left(\mathbf{x}_k - \frac{\beta_k^t}{\sqrt{1-\bar{\alpha}_k^t}}\hat{\boldsymbol{\epsilon}}_k^t\right)$
  Construct frequency-domain input:$\mathbf{h}_k^f \leftarrow \texttt{set_input}(\mathcal{F}^{-1}(\mathbf{x}_k^f), \mathbf{X}^{\text{obs}}, \mathbf{M}^{\text{cond}})$
  Estimate frequency-domain noise:$\hat{\boldsymbol{\epsilon}}_k^f \leftarrow \boldsymbol{\epsilon}_\theta^f(\mathbf{h}_k^f, \mathbf{s}, k)$
  Update input according to frequency domain noise: $\mathbf{x}_{k-1}^t \leftarrow \frac{1}{\sqrt{\alpha_k^f}}\left(\mathcal{F}^{-1}(\mathbf{x}_k^f) - \sqrt{1-\lambda}\mathcal{F}^{-1}(\Lambda\hat{\boldsymbol{\epsilon}}_k^f)\right)$
end for
return $\mathbf{x}_0$

A.8 Dataset Details

The PhysioNet dataset [37] consists of 4000 irregularly-sampled medical time series data including 35 variables (*e.g.,* Albumin and heart rate) collected from ICU with a total length of 48 hours and has been made available by their authors under the terms of the Creative Commons Attribution License 3.0 (CCAL). Consistent with previous studies [4], the dataset is processed hourly to get 48 timesteps. The processed dataset contains nearly 80% originally missing values without ground truth. In our experiments, we random select 10/50/90% of the observed values as the imputation targets (*i.e.,* ground truth of test dataset).

The air quality dataset [56] contains PM2.5 data collected from 36 monitor stations in Beijing and is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. All the air quality data are collected hourly for 12 months. The original dataset contains 13.3% missing values with artificial ground-truth with a non-random missing pattern.

The details of the datasets are presented in Tab.4.

Table 4: Details of PhysioNet and Air Quality datasets.

Dataset	# Samples	# Features	Time Steps	Missing Type	Missing Ratio
PhysioNet	4000	35	48	Originally Missing & Random	80% (Original)
Air Quality	5633	36	36	Non-Random & Artificial	13% (Original)

A.9 Evaluation Metrics

In this section, we present the details of evaluation metrics in our experiments. $y, \hat{y} \in \mathbb{R}^{K \times L}$ denote the ground truth and output of our model and M is the indicator matrix.

Mean Absolute Error (MAE) calculates the average L_1 error between the imputed samples and the ground truth of the time series:

$$\mathbf{MAE}(y, \hat{y}) = \frac{\sum_{i=1}^{K} \sum_{j=1}^{L} M_{ij} |y_{ij} - \hat{y}_{ij}|_1}{\sum_{i=1}^{K} \sum_{j=1}^{L} M_{ij}} \tag{39}$$

MAE reflects the overall deviation across all points, emphasizes the overall accuracy of the model outputs. Compared with RMSE, MAE is more robust to outliers in the data.

Rooted Mean Square Error (RMSE) calculates the average L_2 error between the imputed samples and the ground truth of time series:

$$\mathbf{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=1}^{K} \sum_{j=1}^{L} M_{ij} \|y_{ij} - \hat{y}_{ij}\|_2^2}{\sum_{i=1}^{K} \sum_{j=1}^{L} M_{ij}}} \tag{40}$$

RMSE highlights potential large errors, so a few big deviations may dominate the score. Compared with MAE, RMSE is less robust to outliers but better captures worst-case performance.

A.10 Limitations

Here we provide an in-depth discussion regarding the limitations of our current work. First, our exploration of frequency-domain representations is primarily confined to the Discrete Fourier Transform (DFT). Future research could extend this framework to alternative transforms, such as the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), to better capture different kinds of features from different types of time series data. Second, this study focuses on deterministic time series imputation using DDIM for efficient sampling. A promising future direction involves adapting DDPM-based sampling schemes to our hybrid time-frequency architecture, as well as evaluating the model's performance on probabilistic imputation tasks to better quantify uncertainty. Another limitation is that the frequency-domain sampling process can be less numerically stable than sampling in the time domain, particularly when reconstructing signals from sparse spectral representations. We intend to investigate specialized stabilization methods for frequency-domain diffusion in future works.

Table 5: Training time comparison of CSDI and HyFAD.

	CSDI	HyFAD
TRAINING/EPOCH	6.70s	18.06s
INFERENCE/SAMPLE	36.51s	46.42s

A.11 Implementation Details

Global factor s_k is updated as $s_k = \alpha s_{k-1} + (1-\alpha)\hat{s}_k$, where $\hat{s}_k$ is computed as the median spectral energy of the current training batch. It depends only on the observed values of the training samples and their diffusion states without any validation or test information. Moreover, this EMA is updated only during training and kept fixed at inference time, so it does not introduce data leakage.

The noise scheduler does not rely on validation data. The scheduler is predefined, and the noise scale N_t in the frequency branch (Eq. 18) is computed analytically from the diffusion coefficients rather than estimated from validation or test statistics. Therefore, neither the construction nor the use of the scheduler involves validation information.

All baselines are evaluated under the same setting. We strictly follow the CSDI setting for data preprocessing, splitting, and evaluation on both PhysioNet and Air Quality, ensuring a fair comparison across methods. For validation, we follow the same setting as CSDI: the validation set is only used to select the best checkpoint, and we do not use early stopping in our training.

A.12 Computational Cost

Compared with CSDI, the extra cost of HyFAD mainly comes from frequency-domain branch, time–frequency transforms and the joint optimization objective. Since the backbone is Transformer-based, the overall complexity is $O(L^2)$ with transform cost $O(L \log L)$. We also provide training and inference speed on PhysioNet (batch size = 16) in Table.5.

A.13 Experiment details

A.13.1 Hyperparameters

The hyperparameters in our experiments is detailed in Table.6 and the hyperparameters in frequency-aware diffusion embedding in presented in Table.7

A.13.2 Ablation on explicit frequency modeling

To empirically assess the contribution of explicit frequency modeling, we compare HyFAD with its pure time-domain counterpart. Because the time-domain and frequency-domain diffusion processes in HyFAD are tightly coupled throughout both the forward and reverse diffusion procedures, simply removing the frequency branch from the denoising network would lead to an incomplete and potentially unfair ablation. Instead, we remove the frequency-domain process from both diffusion stages, which reduces the framework to a purely time-domain diffusion model. This counterpart is equivalent to CSDI, an established generative baseline for time-series imputation. Therefore, the comparison between HyFAD and CSDI serves both as a controlled study of the effect of frequency modeling and as a direct comparison with a representative time-domain generative imputation method.

To further examine whether the imputed series preserves periodic structures across different spectral ranges, we adopt Spectral-MAE and partition the spectrum into three sub-intervals and the results are in Tab.8. The Spectral-MAE is defined as the normalized PSD estimates of the ground-truth series $P_{\mathrm{GT}}(\omega)$ against those of the reconstruction $P_{\mathrm{Pred}}(\omega)$:

$$\text{S-MAE} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left| \frac{P_{\mathrm{GT}}(\omega)}{\sum_{\omega'} P_{\mathrm{GT}}(\omega')} - \frac{P_{\mathrm{pred}}(\omega)}{\sum_{\omega'} P_{\mathrm{pred}}(\omega')} \right| \tag{41}$$

where Ω is the set of evaluated frequencies. The results show that CSDI exhibits systematic weaknesses in capturing both low- and high-frequency patterns, highlighting the importance of explicit frequency-domain modeling. Since Spectral-MAE can be dominated by low-frequency components, the performance improvement is not significant. We additionally compute the error in the logarithmic

Table 6: Hyperparameter details in our experiments

Dataset	PhysioNet	Air Quality
Epochs	400	400
Batch Size	16	16
Learning Rate	0.001	0.001
#Time Layers	4	4
#Freq Layers	4	4
Channels	128	128
#Time Heads	8	8
#Freq Heads	8	8
#Time Diffusion Embedding Dim	128	128
#Freq Diffusion Embedding Dim	128	128
β^t_{start}	0.0001	0.0001
β^t_{end}	0.5	0.75
β^f_{start}	0.0001	0.0001
β^f_{end}	0.05	0.05
ω	0.4	1.0
λ	0.75	0.75
#Steps	50	50
Noise Scheduler	quad	quad
#Time Embedding Dim	128	128
#Feature Embedding Dim	16	16

Table 7: Hyperparameters in frequency-aware diffusion embedding.

Dataset	γ	τ	$g_{\min}$	κ	$c_{\min}$	$c_{\max}$	q
PhysioNet	1.0	0.7	0.3	0.5	1.0	100.0	1.0
Air Quality	1.0	0.7	0.3	0.5	1.0	100.0	1.0

spectral domain and report band-wise Log-SMAE. Moreover, we report the Leading Frequency Error (LFE), which evaluates whether the dominant frequency component of the original signal is correctly recovered by the imputed sequence. LFE complements Log-SMAE by focusing on the recovery of the dominant periodic structure: a large LFE indicates that the dominant frequency is not well recovered, whereas Log-SMAE measures the fidelity of the full spectral distribution. The log-SMAE and LFE are formulated as:

$$\log \text{S-MAE} = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} |\log \frac{P_{\text{GT}}(\omega)}{\sum_{\omega'} P_{\text{GT}}(\omega')} - \log \frac{P_{\text{pred}}(\omega)}{\sum_{\omega'} P_{\text{pred}}(\omega')}| \tag{42}$$

$$\text{LFE} = \frac{1}{N} \sum_{i=1}^{N} |f^{\star}_{\text{GT},i} - f^{\star}_{\text{pred,i}}|, \qquad f^{\star}_i = \frac{1}{2\pi} \arg\max_{\omega \in \Omega} P_i(\omega) \tag{43}$$

The additional spectral evaluation results provide further empirical evidence for the effect of explicit frequency modeling. As shown in Table.9, HyFAD achieves lower Log-SMAE than the pure time-domain counterpart across all three frequency bands and all missing ratios. In the low-frequency band, the relative reductions are 24.4%, 19.7%, and 10.8% under 10%, 50%, and 90% missingness, respectively. In the middle-frequency band, the reductions are 22.8%, 17.9%, and 11.2%, while in the high-frequency band, the reductions are 21.3%, 18.1%, and 6.8%.

Table 8: Bandwise S-MAE results of CSDI and HyFAD.

MISSING RATIO	0.1		0.5		0.9	
FREQUENCY BAND	CSDI	HYFAD	CSDI	HYFAD	CSDI	HYFAD
LOW	0.0144	0.0143	0.0624	0.0595	0.121	0.117
MIDDLE	0.0131	0.0128	0.0543	0.0522	0.0952	0.0923
HIGH	0.0123	0.0120	0.0491	0.0473	0.081	0.079

Table 9: Bandwise log-SMAE and LFE results on CSDI and HyFAD, left results: CSDI, right results: HyFAD.

Missing ratio	10%		50%		90%	
Frequency band	Log-SMAE	LFE	Log-SMAE	LFE	Log-SMAE	LFE
Low-Frequency	0.1046/0.0791	4.1e-5/2.3e-5	0.3876/0.3113	0.0002/0.0002	0.6246/0.5569	0.0003/0.0003
Middle-Frequency	0.1271/0.0981	0.0012/0.0009	0.4531/0.3720	0.0048/0.0040	0.6827/0.6060	0.0074/0.0065
High-Frequency	0.1504/0.1184	0.0024/0.0019	0.4986/0.4084	0.0092/0.0074	0.7190/0.6701	0.0133/0.0120

Table 10: Detailed comparison of HyFAD, LSCD and CSDI.

		PhysioNet 10%		PhysioNet 50%		PhysioNet 90%		Air Quality	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
HyFAD	mean(std)	0.423(0.007)	0.206(0.001)	0.622(0.025)	0.288(0.002)	0.782(0.028)	0.448(0.003)	17.730(0.021)	9.296(0.007)
	95% CI	[0.414, 0.432]	[0.206, 0.207]	[0.591, 0.652]	[0.286, 0.290]	[0.746, 0.817]	[0.444, 0.452]	[17.705, 17.756]	[9.286, 9.305]
LSCD	mean(std)	0.457(0.004)	0.212(0.0004)	0.6582(0.0004)	0.3046(0.0005)	0.845(0.000)	0.492(0.000)	18.270(0.145)	9.340(0.034)
	95% CI	[0.452, 0.462]	[0.212,0.213]	[0.658, 0.659]	0.304, 0.305	0.845,0.845	0.492,0.492	18.125,18.416	9.306,9.374
	p-value(HyFAD-LSCD)	2.21e-4	8.42e-6	2.92e-2	1.65e-5	7.68e-3	9.14e-6	8.25e-4	2.99e-2
CSDI	mean(std)	0.491(0.004)	0.215(0.0005)	0.673(0.001)	0.307(0.001)	0.851(0.000)	0.492(0.0005)	18.713(0.080)	9.347(0.029)
	95% CI	[0.486,0.497]	[0.215, 0.216]	[0.672, 0.675]	[0.306, 0.307]	[0.851, 0.851]	[0.491,0.492]	[18.614, 18.812]	[9.311, 9.382]
	p-value(HyFAD-CSDI)	4.5e-5	1.94e-5	9.95e-3	3.65e-5	5.56e-3	9.62e-6	6.10e-6	8.17e-3

HyFAD also consistently reduces LFE in the middle- and high-frequency bands across all missing ratios. Specifically, the relative reductions in the middle-frequency band are 25.0%, 16.7%, and 12.2%, and those in the high-frequency band are 20.8%, 19.6%, and 9.8%. In contrast, the low-frequency LFE values of both methods are already very small, indicating that the dominant low-frequency periodicity can be relatively well recovered even by the pure time-domain baseline. This suggests that the main advantage of HyFAD lies in recovering the more challenging middle- and high-frequency periodic structures.

Overall, these results show that the benefit of explicit frequency modeling is not limited to aggregate improvements in standard time-domain metrics. Instead, HyFAD brings consistent improvements in spectral reconstruction, particularly in the middle- and high-frequency regions where the pure time-domain baseline exhibits larger errors. This further validates the effectiveness of incorporating frequency-domain diffusion for modeling heterogeneous spectral components in time-series imputation.

A.13.3 Comparison with LSCD and CSDI

To better demonstrate the superiority of HyFAD over the two next-best baselines, CSDI and LSCD, We present the results of HyFAD, LSCD and CSDI on 5 independent runs and report the corresponding 95% CI and p-values in Table.10.

The detailed comparison in Table 10 further confirms the superiority and stability of HyFAD over the two strongest baselines, LSCD and CSDI. HyFAD achieves the lowest mean RMSE and MAE across all datasets and missingness rates. On PhysioNet, HyFAD consistently outperforms LSCD and CSDI under 10%, 50%, and 90% missingness, with especially clear gains in RMSE. For example, compared with LSCD, HyFAD reduces RMSE by 7.4%, 5.5%, and 7.5% under the three missing ratios, respectively. Similar improvements are observed over CSDI, showing that HyFAD remains robust as the missing ratio increases.

The confidence intervals also indicate that HyFAD maintains stable performance across repeated runs, and its intervals are generally lower than those of LSCD and CSDI. Moreover, all reported p-values are below 0.05, demonstrating that the improvements of HyFAD over both LSCD and CSDI are statistically significant. On the Air Quality dataset, HyFAD also achieves the best RMSE and MAE, although the MAE improvement is relatively smaller. Overall, these results show that HyFAD not only improves average imputation accuracy but also provides statistically reliable gains over competitive diffusion-based baselines.

A.13.4 Visualization Results

The imputation results on Air Quality dataset and PhysioNet dataset with missing ratio 10%, 50% and 90% are presented in Fig.4,5,6,7.

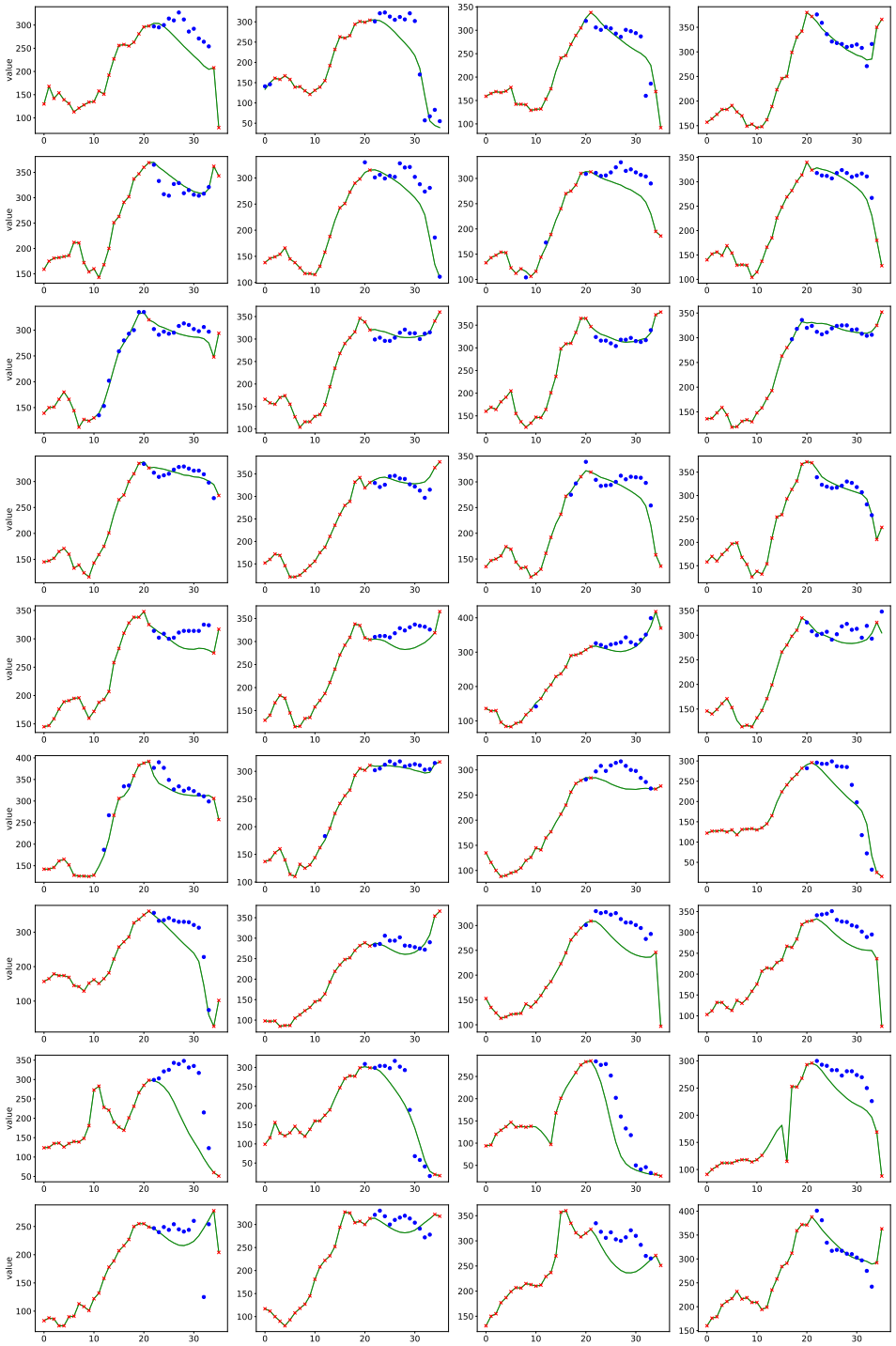


Figure 4: Visualization of imputation results on AQI dataset from Channel 1 to Channel 36. The solid line represents the imputation results, the blue dots represent the ground truth of the missing points, and the red crosses represent the observed values.

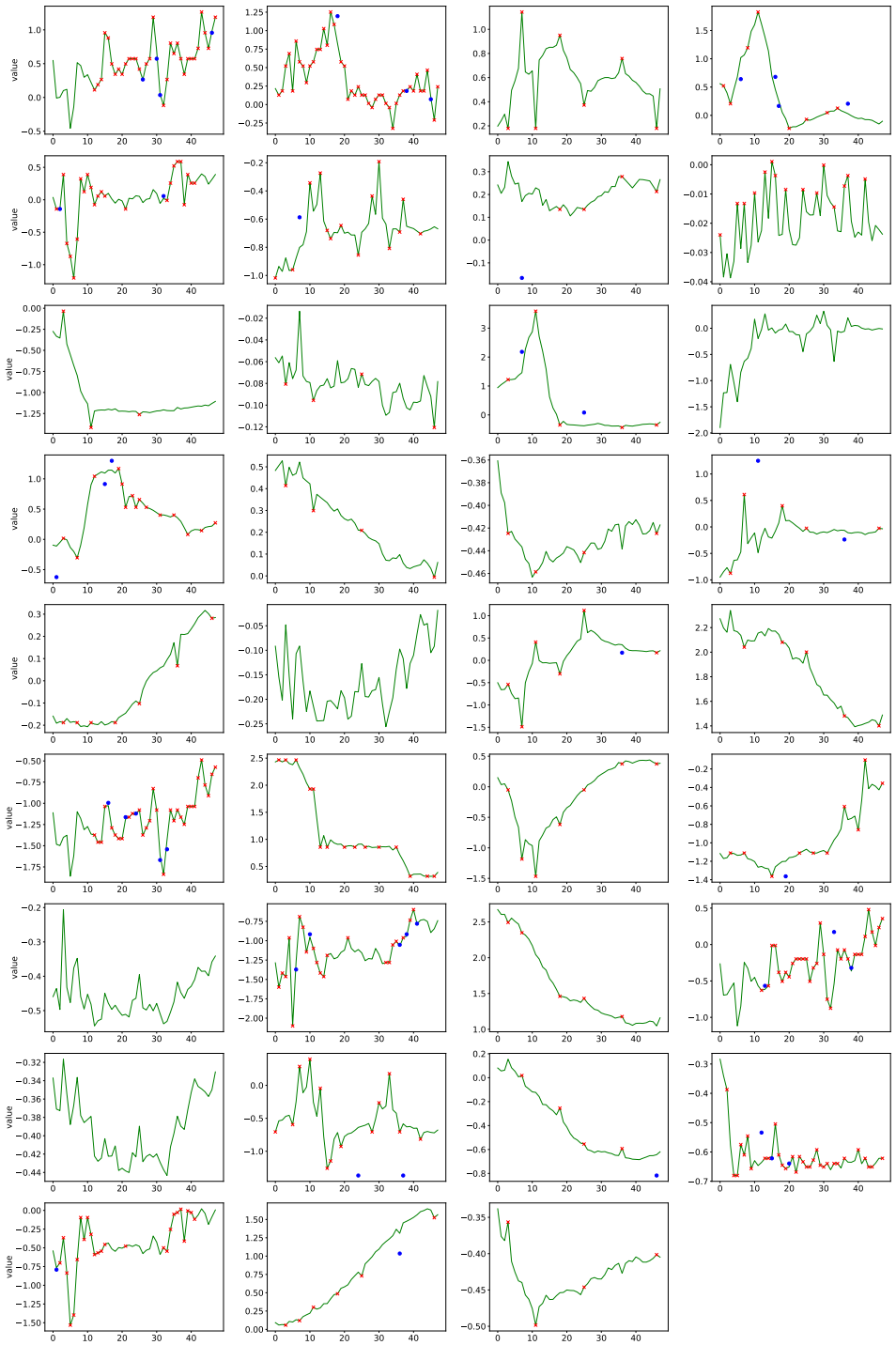


Figure 5: Visualization of imputation results on PhysioNet dataset from Channel 1 to Channel 35 with 10% missing. The solid line represents the imputation results, the blue dots represent the ground truth of the missing points, and the red crosses represent the observed values.

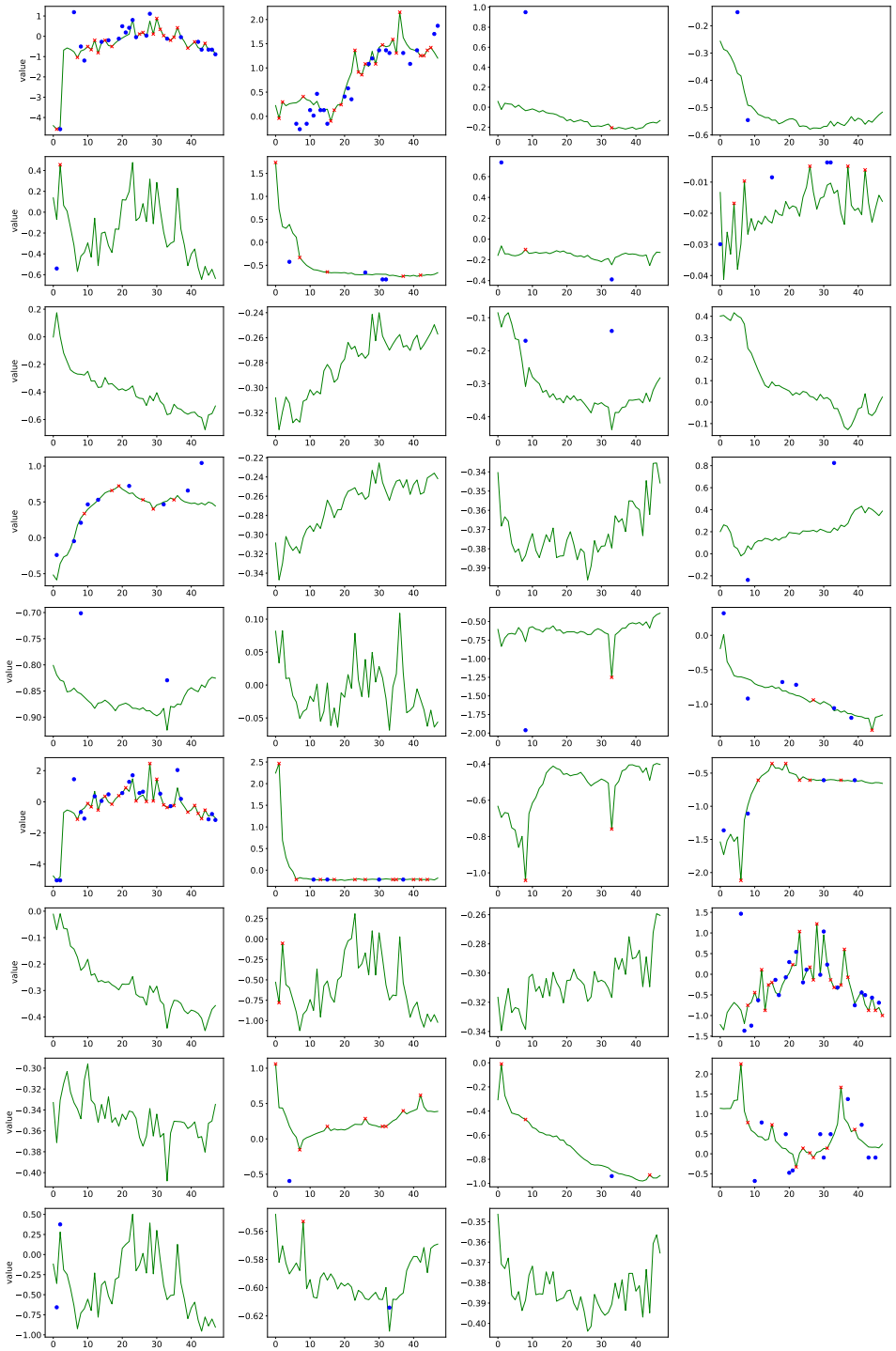


Figure 6: Visualization of imputation results on PhysioNet dataset from Channel 1 to Channel 35 with 50% missing. The solid line represents the imputation results, the blue dots represent the ground truth of the missing points, and the red crosses represent the observed values.

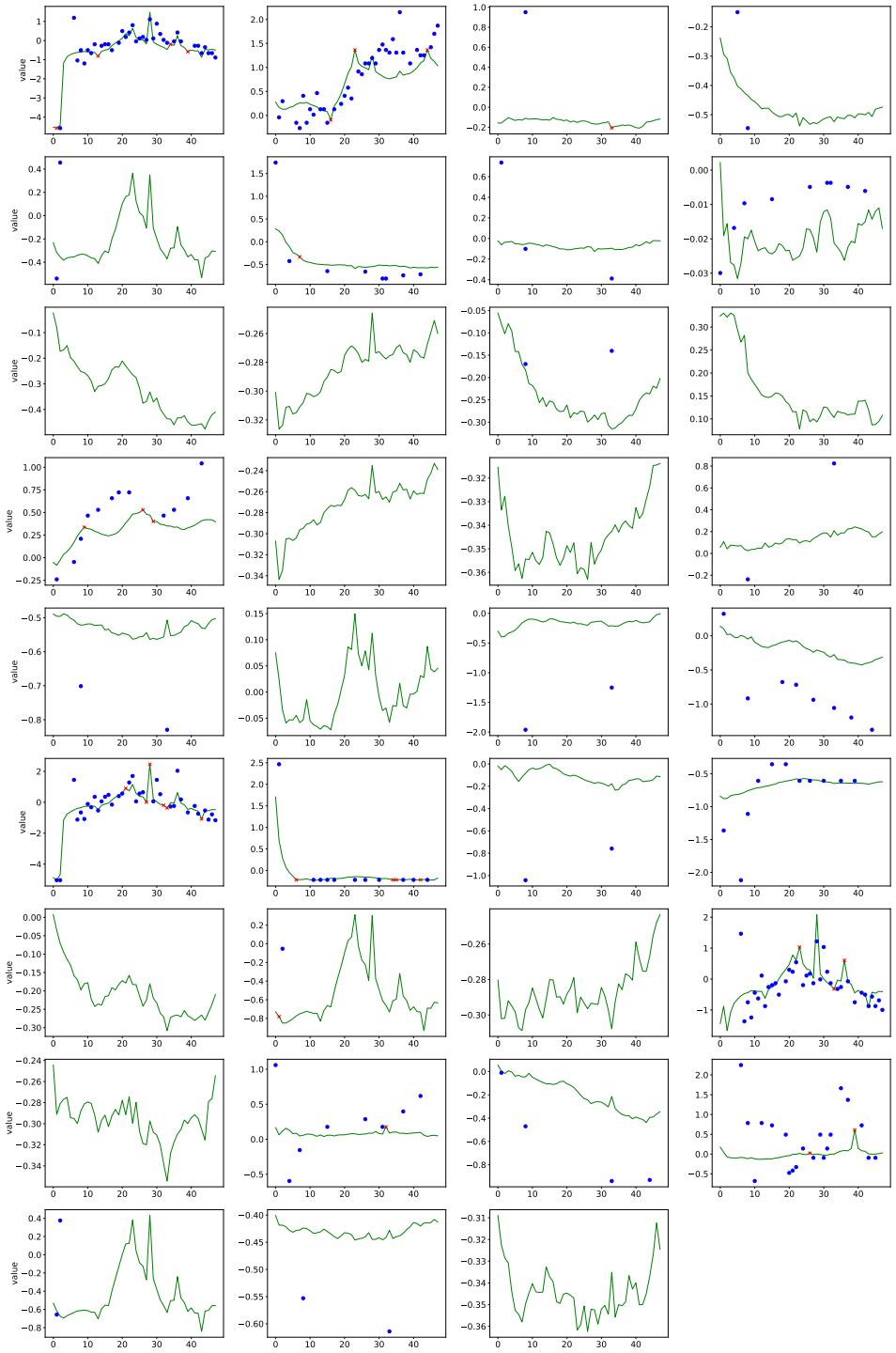


Figure 7: Visualization of imputation results on PhysioNet dataset from Channel 1 to Channel 35 with 90% missing. The solid line represents the imputation results, the blue dots represent the ground truth of the missing points, and the red crosses represent the observed values.

A.14 Social impacts

This work may have positive societal impacts by improving time series imputation in domains such as healthcare monitoring, air-quality assessment, industrial sensing, and scientific data analysis, where missing observations are common and better reconstruction can support more reliable downstream analysis. However, the method may also pose risks if imputed values are treated as ground-truth measurements, especially in high-stakes settings such as clinical decision-making, public safety, or critical infrastructure. Incorrect or biased imputations could lead to misleading conclusions or harmful decisions, particularly for underrepresented populations, regions, or sensor conditions. Therefore, imputed values should be clearly distinguished from observed data, and deployment in sensitive domains should include domain-specific validation, uncertainty assessment, privacy protection, and expert review.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Our core contributions and claims stated in the introduction are directly supported by the theoretical analysis and empirical results detailed in the main text.

   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 explicitly discuss the limitations of our proposed method in Appendix Sec.A.10.

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: [Yes]

   Justification: We clearly state all assumptions alongside the theoretical statements in Section.3.2, and provide the complete rigorous proofs in Appendix.A.4.

   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 ensure reproducibility by providing detailed experimental parameters in Section.4, complete algorithm pseudocode in Appendix.A.7, and a link to the source code 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: We provide an anonymized link to our source code repository directly in the abstract, which includes detailed instructions to reproduce the main experimental results.

   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: We explicitly specify all necessary training and test details, including hyperparameters and configurations, in Section.4 of the main text.

   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: We report the mean and standard deviation across multiple independent runs with different random seeds.

   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: We provide comprehensive details regarding the hardware specifications (e.g., GPU type) in Sec.4 and the estimated training and inference time cost in Appendix.A.12.

   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 strictly adheres to the NeurIPS Code of Ethics. We utilize only standard, publicly available datasets, properly credit all previous work, and our methodology does not introduce any foreseeable negative societal impacts or dual-use risks.

   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: [Yes]

    Justification: We explicitly discuss both the potential positive applications and the potential negative societal impacts of our work in Appendix.A.14.

    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: This paper does not release any pre-trained generative models, large-scale scraped datasets, or other assets that pose a high risk for misuse or dual-use. Therefore, specific safeguards for responsible release are not applicable.

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 third-party assets (e.g., datasets and baselines) are properly cited in the main text, and their corresponding licenses are explicitly listed in Appendix.A.8.

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: [N/A]

Justification: This paper proposes a new algorithmic method and evaluates it exclusively on existing, publicly available datasets (e.g., Physionet, Air Quality). We do not introduce any new datasets, models, or standalone software assets.

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 any new crowdsourcing experiments, data annotation by human workers, 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 research does not involve any human subjects, user studies, or new crowdsourcing efforts. All experiments were conducted on standard, publicly available datasets.

    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: This research does not utilize LLMs as a component of its core methods or scientific pipeline. As noted in the instructions, any use of LLMs was strictly limited to standard text editing and formatting, which does not impact the scientific rigor or originality.

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.