HYBRID DIFFUSION MODEL FOR SEQUENTIAL DATA GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential data are ubiquitous in real-world applications, and generating realistic sequences is useful for simulation, data augmentation, and forecasting. Diffusion models have become a leading approach for this task, but standard time-domain diffusion treats the sequence as a whole and tends to capture global patterns more faithfully than local variations. Prior frequency-aware methods have added frequency information to time-domain models or moved diffusion entirely to the frequency domain; both approaches still use a single-domain generation process. We propose a Hybrid Diffusion Model that couples low-frequency generation in the time domain with high-frequency generation in the frequency domain. We decompose each sequence into low- and high-frequency components using the discrete Fourier transform and generate them with two diffusion processes that use different domains and schedules. The frequency-domain process runs on the full spectrum with a longer diffusion schedule. During joint sampling, the low-frequency component is generated by the shorter time-domain process, conditioned on the evolving high-frequency state from the frequency-domain process. On five real-world datasets, Hybrid Diffusion outperforms both single-domain diffusion baselines on performance metrics in both the time and frequency domains; on four additional datasets, it achieves the best mean performance in 10 of 12 dataset–metric comparisons against six time-series generative models.

1 INTRODUCTION

Sequential data are ubiquitous in real-world applications, including finance (John et al., 2019; Chen et al., 2016), healthcare (Tonekaboni et al., 2021; Kazijevs & Samad, 2023), transportation (Li et al., 2020; Tedjopurnomo et al., 2020), and meteorology (Yozgatligil et al., 2013). Accurate generative modeling of sequential data is therefore valuable for data augmentation, simulation, and conditional tasks such as forecasting and imputation (Yoon et al., 2019; Yuan & Qiao, 2024).

Diffusion models (Hyvärinen, 2005; Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) have recently emerged as a promising approach for sequential data generation. Most existing diffusion models for sequential data define the forward noising process on the raw temporal sequence: they gradually add Gaussian noise to the data and learn a reverse denoising process to generate samples. This paradigm has shown success in general time series generation (Yuan & Qiao, 2024), probabilistic time series imputation (Tashiro et al., 2021), video generation (Gupta et al., 2023; Liu et al., 2024), and financial data augmentation (Gao et al., 2024; Chen et al., 2025).

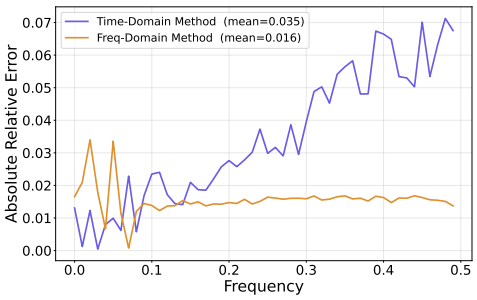


Figure 1: Absolute relative error in spectral magnitude on Gaussian processes.

Despite this progress, a naively implemented time-domain diffusion process does not align well with how sequential data are distributed in the frequency domain. In practice, sequential data commonly place most of their energy in low-frequency components. Under isotropic Gaussian noise, the imbalance of the signal's spectral distribution causes the high-frequency components to be corrupted

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

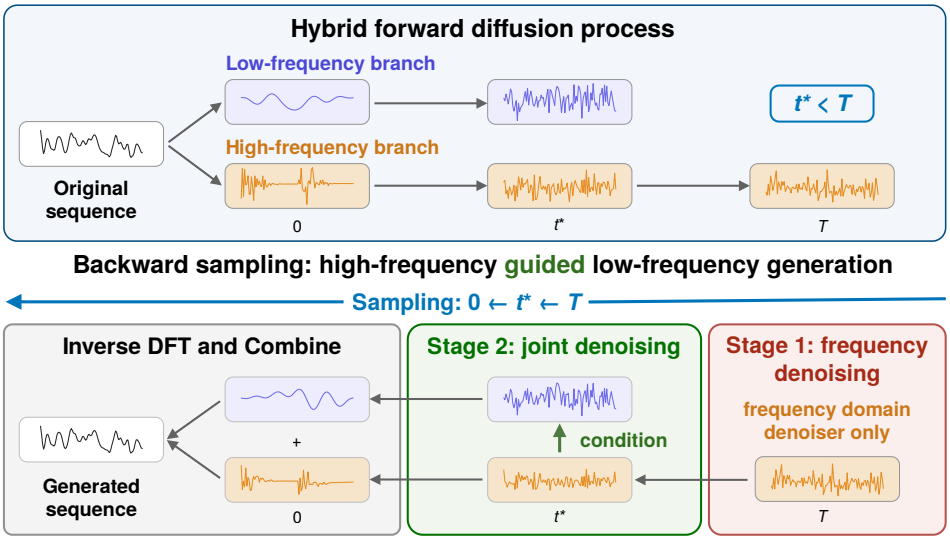


Figure 2: Overview of the Hybrid Diffusion Model with both time-domain and frequency-domain diffusion processes.

by noise far earlier than the low-frequency components; as a result, the time-domain model has difficulty recovering high-frequency components (Falck et al., 2025). We use a synthetic example to illustrate this limitation and to show that a frequency-domain diffusion model can instead learn different frequency components with more comparable accuracy; further details are provided in Appendix B. As shown in Figure 1, the time-domain model exhibits a sharp increase in spectral error at high frequencies, whereas the frequency-domain model maintains comparable accuracy across the full spectrum. This observation motivates the use of frequency-domain diffusion models to improve sequential data generation.

To mitigate the difficulty of recovering high-frequency components using time-domain diffusion models, recent frequency-aware methods inject frequency information into time-domain diffusion models (Galib et al., 2024; Yang et al., 2024). These methods keep the diffusion process in the time domain and use frequency features to guide the denoiser. More directly, Crabbé et al. (2024) perform diffusion entirely in the frequency domain through a dual SDE in Fourier space, motivated by the energy localization of real-world sequential data in the frequency domain.

However, these approaches still generate all frequency components in a single domain. In the synthetic comparison in Appendix B, the frequency-domain model reduces high-frequency spectral error when the data contain substantial high-frequency content, while the time-domain model better matches temporal covariance. These observations motivate separate processes for the two components:

How can diffusion models combine the strengths of time- and frequency-domain generation?

Hybrid Diffusion uses two diffusion processes to generate the components together (Figure 2). We decompose each sequence into low- and high-frequency components using the discrete Fourier transform. The frequency-domain process runs over the full horizon $[0, T]$ and supplies the generated high-frequency component. The time-domain process generates the low-frequency component over the shorter interval $[0, t^*]$, where $t^* < T$. During sampling, the frequency-domain reverse process first runs from T to t^* to obtain a partially recovered high-frequency state. From t^* to 0, both denoisers run, with the time-domain denoiser conditioned on the current high-frequency state. The final sequence combines the generated low-frequency component with the recovered high-frequency component through the inverse discrete Fourier transform.

Contributions We propose Hybrid Diffusion, a sequential data generation framework that decomposes each sequence into low- and high-frequency components and models them with diffusion processes that use different schedules and domains. Our contributions are summarized as follows.

- We formulate low- and high-frequency generation with domain-specific diffusion processes: a full-horizon frequency-domain process for high-frequency components and a shorter time-domain process for low-frequency components.
- We introduce a backward generation process with cross-frequency conditioning, where the high-frequency state produced by the frequency-domain process guides time-domain generation of low-frequency components to preserve consistency between global trends and local details.
- We evaluate Hybrid Diffusion on nine sequential datasets, showing lower sliced Wasserstein (SW) distances in both domains than both single-domain baselines on five datasets, and the lowest mean Discriminative Score against six generative baselines on all four additional datasets.

Notation For a real-valued vector $\mathbf{X}$, we use $\widetilde{\mathbf{X}}$ to denote its frequency-domain representation obtained by the discrete Fourier transform. We use $\mathrm{Re}[\widetilde{\mathbf{X}}]$ and $\mathrm{Im}[\widetilde{\mathbf{X}}]$ to denote the real and imaginary components of $\widetilde{\mathbf{X}}$. We denote $\mathbf{X}^{(l)}$ and $\mathbf{X}^{(h)}$ as the low- and high-frequency parts of $\mathbf{X}$, respectively. In the frequency domain, $\widetilde{\mathbf{X}}^{(l)}$ and $\widetilde{\mathbf{X}}^{(h)}$ are analogously defined.

2 Related Work

Deep generative models for sequential data Early work on sequential data generation relied on GANs and VAEs. TimeGAN (Yoon et al., 2019) introduced jointly trained embedding and adversarial objectives to capture temporal dynamics, followed by variants such as Cot-GAN (Xu et al., 2020). TimeVAE (Desai et al., 2021) extended VAE paradigms with time-series-specific decoders. While effective, these models often struggle with long-range dependencies and stable training compared to diffusion-based alternatives. Following DDPM and score-based SDEs (Ho et al., 2020; Song et al., 2020), diffusion models have been applied to sequential data across diverse settings. DiffWave (Kong et al., 2020) pioneered diffusion for waveform generation, with subsequent work extending the paradigm to probabilistic imputation and forecasting (Tashiro et al., 2021) and unconditional generation via seasonal–trend decomposition (Yuan & Qiao, 2024) or a latent-to-data diffusion cascade (Shen et al., 2026). Domain-specific extensions further include video (Gupta et al., 2023; Liu et al., 2024) and financial data augmentation (Gao et al., 2024; Chen et al., 2025).

Frequency-aware and frequency-domain diffusion A growing body of work argues that the frequency imbalance in time-domain diffusion is inherent to isotropic Gaussian noising (Falck et al., 2025; Rissanen et al., 2022; Yang et al., 2023), motivating diffusion processes that explicitly account for spectral structure. One line of work injects spectral information into time-domain diffusion via Fourier-based losses (Yuan & Qiao, 2024), frequency-feature conditioning (Galib et al., 2024; Yang et al., 2024; Jeong et al., 2025), or spectrum-reweighted training objectives (Yang et al., 2023). Other approaches change the diffusion representation or forward process. In image generation, Rissanen et al. (2022) use a blurring process that removes high-frequency details first, while Edify Image (NVIDIA et al., 2024) attenuates Laplacian frequency bands at different rates. Crabbé et al. (2024) propose a dual SDE in Fourier space, and show that frequency-domain diffusion fits the data distribution better than time-domain diffusion when the signal energy is concentrated in a few frequency components. Despite their differences, these frequency-aware methods still operate within a single domain, which can require a compromise between low-frequency structure and high-frequency detail.

3 Problem Setup and Preliminaries

We consider a multivariate time series $\mathbf{X} = [\mathbf{X}_0^\top, \ldots, \mathbf{X}_{N-1}^\top]^\top \in \mathbb{R}^{dN}$, where $\mathbf{X}_\tau \in \mathbb{R}^d$ denotes the observation at time step τ. Let P_0 denote the unknown joint distribution of $\mathbf{X}$. Our goal is to train a diffusion-based generative model from observed time series. In this section, we briefly review the Discrete Fourier Transform (DFT) as the spectral representation of time series data, and introduce the continuous-time formulation of score-based diffusion models.

3.1 Discrete Fourier Transform

Given a time series $\mathbf{X}$, its Discrete Fourier Transform (DFT) (Bracewell, 1986), denoted as $\widetilde{\mathbf{X}} = \mathcal{F}[\mathbf{X}] = [\widetilde{\mathbf{X}}_0^\top, \ldots, \widetilde{\mathbf{X}}_{N-1}^\top]^\top \in \mathbb{C}^{dN}$, is defined component-wise for each frequency index $\kappa \in$

$\{0, \ldots, N-1\}$ as

$$\widetilde{\mathbf{X}}_\kappa := \frac{1}{\sqrt{N}} \sum_{\tau=0}^{N-1} \mathbf{X}_\tau \exp\left(-2\pi i \frac{\kappa\tau}{N}\right).$$

Due to the conjugate symmetry of the DFT, $\widetilde{\mathbf{X}}$ satisfies $\widetilde{\mathbf{X}}_\kappa = \overline{\widetilde{\mathbf{X}}_{N-\kappa}}$ for the corresponding conjugate-frequency pairs. Therefore, the real and imaginary parts of $\widetilde{\mathbf{X}}$ contain duplicated entries. To extract the non-redundant real and imaginary parts from $\widetilde{\mathbf{X}}$, we define the following mapping $\phi : \mathbb{C}^{dN} \to \mathbb{R}^{dN}$,

$$\phi[\widetilde{\mathbf{X}}] = \begin{cases} \left[\left(\mathrm{Re}[\widetilde{\mathbf{X}}_\kappa]\right)_{\kappa=0}^{N/2}, \left(\mathrm{Im}[\widetilde{\mathbf{X}}_\kappa]\right)_{\kappa=1}^{N/2-1}\right], & \text{if } N \text{ is even,} \\ \left[\left(\mathrm{Re}[\widetilde{\mathbf{X}}_\kappa]\right)_{\kappa=0}^{\lfloor N/2 \rfloor}, \left(\mathrm{Im}[\widetilde{\mathbf{X}}_\kappa]\right)_{\kappa=1}^{\lfloor N/2 \rfloor}\right], & \text{if } N \text{ is odd.} \end{cases}$$

The distribution of $\widetilde{\mathbf{X}}$ can then be expressed in terms of its non-redundant components $\phi[\widetilde{\mathbf{X}}]$.

3.2 Score-based Diffusion Model

A score-based diffusion model aims to learn and sample from an unknown data distribution P_0 over $\mathbb{R}^{dN}$ by estimating the score function. It consists of coupled forward and backward processes. The forward process gradually corrupts the data distribution via Gaussian noise injection:

$$\mathrm{d}\mathbf{X}_t = -\tfrac{1}{2}\beta(t)\,\mathbf{X}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t. \qquad \text{(Forward)}$$

Here, $\mathbf{X}_t \in \mathbb{R}^{dN}$ for $t \in [0, T]$ with initial value $\mathbf{X}_0 \sim P_0$, $\beta(t) : [0, T] \to \mathbb{R}_+$ is the noise rate, $\mathbf{W}_t$ is a standard Wiener process in $\mathbb{R}^{dN}$, and T is the terminal diffusion timestep. We denote P_t as the marginal distribution of $\mathbf{X}_t$, with corresponding density function p_t. With an appropriately chosen $\beta(t)$, the terminal distribution P_T approaches a tractable Gaussian prior π.

The backward process reverses the noise corruption, enabling new sample generation via denoising. Starting from $\mathbf{X}_T \sim \pi$ and running time backward from T to 0, it follows the reverse-time SDE:

$$\mathrm{d}\mathbf{X}_t = \left[-\tfrac{1}{2}\beta(t)\,\mathbf{X}_t - \beta(t)\,\nabla \log p_t(\mathbf{X}_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t. \qquad \text{(Backward)}$$

Here, $\overline{\mathbf{W}}_t$ is a standard Wiener process when time flows backward from T to 0, and $\mathrm{d}t$ is an infinitesimal negative timestep. The term $\nabla \log p_t(\mathbf{X}_t)$ is the score function, which is unknown and must be estimated from samples of P_0. We train a neural network $\mathbf{s}_\theta(\mathbf{X}_t, t) : \mathbb{R}^{dN} \times [0, T] \to \mathbb{R}^{dN}$ to approximate the score $\nabla \log p_t$ by minimizing the denoising score matching objective:

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_{t,\,\mathbf{X}_0,\,\mathbf{X}_t}\left[\left\|\mathbf{s}_\theta(\mathbf{X}_t, t) - \nabla \log p_{t|0}(\mathbf{X}_t \mid \mathbf{X}_0)\right\|^2\right],$$

where $p_{t|0}(\mathbf{X}_t \mid \mathbf{X}_0)$ is the Gaussian transition kernel determined by $\beta(t)$. We omit the time-dependent weighting for clarity.

Score Function in the Frequency Domain The frequency process operates directly on the non-redundant frequency components $\phi[\widetilde{\mathbf{X}}] \in \mathbb{R}^{dN}$. It follows the same general score-based diffusion construction as above, with $\phi[\widetilde{\mathbf{X}}]_t$ as the diffusion state. In the main text, we use the corresponding score notation: the forward process is initialized at $\phi[\widetilde{\mathbf{X}}]_0$, and its noise-injected state at time $t \in [0, T]$ is denoted by $\phi[\widetilde{\mathbf{X}}]_t$, with marginal density p_t^ϕ. The frequency-domain score function is defined as

$$\mathbf{s}_\phi(\phi[\widetilde{\mathbf{X}}]_t, t) = \nabla_{\phi[\widetilde{\mathbf{X}}]_t} \log p_t^\phi(\phi[\widetilde{\mathbf{X}}]_t).$$

4 Hybrid Diffusion

In this section, we present the Hybrid Diffusion framework. As noted in Section 1, time-domain models alone struggle with high-frequency components. We therefore model the high-frequency part separately in the frequency domain. Specifically, we decompose the sequence into low- and high-frequency parts. We use a time-domain diffusion model to learn the low-frequency component and a frequency-domain diffusion model to learn the full spectrum, from which the high-frequency component is recovered. In what follows, we first describe the frequency decomposition and instantiate the diffusion processes on each component in Section 4.1, then present the training of the two denoisers with cross-attention conditioning in Section 4.2, and finally describe the two-stage sampling procedure in Section 4.3.

4.1 Hybrid Diffusion Processes

Frequency Decomposition We fix a cutoff frequency index f_c with $0 \le f_c \le \lfloor N/2 \rfloor$. For each DFT index $\kappa \in \{0, \dots, N-1\}$, we define $\rho(\kappa) = \min\{\kappa, N-\kappa\}$. The low- and high-frequency index sets are then defined as

$$\mathcal{K}_{\text{low}} = \{\kappa : \rho(\kappa) \le f_c - 1\}, \qquad \mathcal{K}_{\text{high}} = \{0, \dots, N-1\} \setminus \mathcal{K}_{\text{low}}.$$

We decompose the DFT vector $\widetilde{\mathbf{X}}$ into $\widetilde{\mathbf{X}}^{(l)}$ and $\widetilde{\mathbf{X}}^{(h)}$ by

$$\widetilde{\mathbf{X}}^{(l)}_\kappa = \begin{cases} \widetilde{\mathbf{X}}_\kappa, & \kappa \in \mathcal{K}_{\text{low}}, \\ \mathbf{0}, & \kappa \in \mathcal{K}_{\text{high}}, \end{cases} \qquad \widetilde{\mathbf{X}}^{(h)}_\kappa = \begin{cases} \mathbf{0}, & \kappa \in \mathcal{K}_{\text{low}}, \\ \widetilde{\mathbf{X}}_\kappa, & \kappa \in \mathcal{K}_{\text{high}}. \end{cases}$$

Thus $\widetilde{\mathbf{X}} = \widetilde{\mathbf{X}}^{(l)} + \widetilde{\mathbf{X}}^{(h)}$, as shown in the left panel of Figure 3. The corresponding time-domain components are defined by the inverse DFT as $\mathbf{X}^{(l)} = \mathcal{F}^{-1}[\widetilde{\mathbf{X}}^{(l)}]$ and $\mathbf{X}^{(h)} = \mathcal{F}^{-1}[\widetilde{\mathbf{X}}^{(h)}]$. Since $\widetilde{\mathbf{X}}^{(l)}$ and $\widetilde{\mathbf{X}}^{(h)}$ preserve conjugate symmetry, $\mathbf{X}^{(l)}, \mathbf{X}^{(h)} \in \mathbb{R}^{dN}$. We write $\phi[\widetilde{\mathbf{X}}]^{(h)}$ for the high-frequency entries of the non-redundant representation $\phi[\widetilde{\mathbf{X}}]$, and write $\phi[\widetilde{\mathbf{X}}]^{(h)}_t$ for the same entries at diffusion time t.

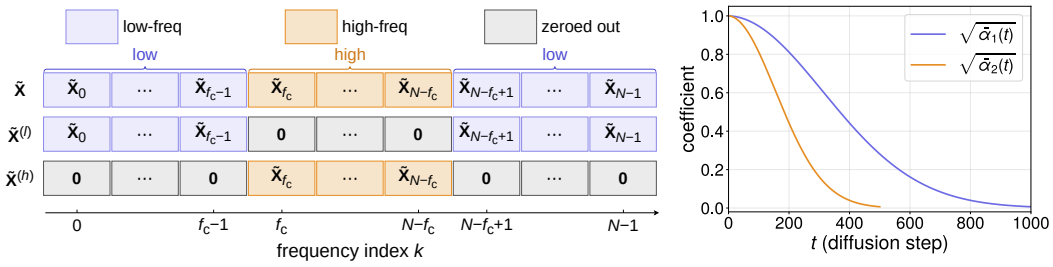


Figure 3: **(Left)** Frequency decomposition: the frequency vector $\widetilde{\mathbf{X}}$ is decomposed at cutoff frequency f_c into a low-frequency component $\widetilde{\mathbf{X}}^{(l)}$ and a high-frequency component. **(Right)** Decay of the signal coefficients $\sqrt{\bar{\alpha}_i(t)}$: the time-domain coefficient $\sqrt{\bar{\alpha}_2(t)}$ decays faster, approaching zero by t^*, while the frequency-domain coefficient $\sqrt{\bar{\alpha}_1(t)}$ decays more slowly and approaches zero by T.

After decomposition, we instantiate the time-domain process and the frequency-domain process from the general SDEs in Equations (Forward) and (Backward) with variance-preserving linear Gaussian dynamics (Song et al., 2020). The time-domain process is applied to $\mathbf{X}^{(l)}$ over $[0, t^*]$, while the frequency-domain process is applied to $\phi[\widetilde{\mathbf{X}}]$ over $[0, T]$. We train the frequency-domain process on the full spectrum $\phi[\widetilde{\mathbf{X}}]$ to provide low-frequency context for learning high-frequency components, and use its high-frequency output in the final reconstruction.

Let $\beta_1 : [0, T] \to \mathbb{R}_+$ and $\beta_2 : [0, t^*] \to \mathbb{R}_+$ denote the noise rates of the frequency-domain and time-domain diffusion processes, respectively. The resulting marginal transition is determined by the cumulative coefficient $\bar{\alpha}_i(t) = \exp(-\int_0^t \beta_i(s)\, \mathrm{d}s)$, $i = 1, 2$. Consequently, the noisy states used by the two processes are

$$\mathbf{X}^{(l)}_t = \sqrt{\bar{\alpha}_2(t)}\, \mathbf{X}^{(l)} + \sqrt{1 - \bar{\alpha}_2(t)}\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN}), \quad t \in [0, t^*], \tag{1}$$

$$\phi[\widetilde{\mathbf{X}}]_t = \sqrt{\bar{\alpha}_1(t)}\, \phi[\widetilde{\mathbf{X}}] + \sqrt{1 - \bar{\alpha}_1(t)}\, \boldsymbol{\epsilon}', \quad \boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN}), \quad t \in [0, T]. \tag{2}$$

The coefficient $\bar{\alpha}_i(t)$ measures how much clean information remains at time t, with smaller values indicating stronger corruption. We let the time-domain schedule $\bar{\alpha}_2$ decay faster than the frequency-domain coefficient $\bar{\alpha}_1$ (see the right panel of Figure 3), so that the low-frequency signal contribution vanishes by t^* ($\bar{\alpha}_2(t^*) \approx 0$) while frequency-domain information persists until T ($\bar{\alpha}_1(T) \approx 0$). We discretize time on the grid $\{0, 1, \dots, t^*, \dots, T\}$ with $t^*, T \in \mathbb{Z}_+$, and define the per-step coefficient $\alpha_i(t) := \bar{\alpha}_i(t)/\bar{\alpha}_i(t-1)$ used in the algorithms below.

4.2 Model Training

We train two denoisers $\mathbf{s}_{\theta_{\text{freq}}}$ and $\mathbf{s}_{\theta_{\text{time}}}$ to estimate the scores of the frequency-domain and time-domain diffusion processes, respectively. They share the same backbone architecture, differing

only in the cross-attention layers used by the time-domain denoiser. Each denoiser is trained by minimizing a score-matching loss:

$$\mathcal{L}_{\text{freq}}(\theta_{\text{freq}}) = \mathbb{E}_{t,\,\widetilde{\mathbf{X}},\,\phi[\widetilde{\mathbf{X}}]_t}\left[\left\|\mathbf{s}_{\theta_{\text{freq}}}\big(\phi[\widetilde{\mathbf{X}}]_t,\,t\big) - \nabla \log p^{\phi}_{t|0}\big(\phi[\widetilde{\mathbf{X}}]_t \mid \phi[\widetilde{\mathbf{X}}]\big)\right\|^2\right], \tag{3}$$

$$\mathcal{L}_{\text{time}}(\theta_{\text{time}}) = \mathbb{E}_{t,\,\mathbf{X}^{(l)},\,\mathbf{X}^{(l)}_t,\,\phi[\widetilde{\mathbf{X}}]_t}\left[\left\|\mathbf{s}_{\theta_{\text{time}}}\big(\mathbf{X}^{(l)}_t,\,t,\,\phi[\widetilde{\mathbf{X}}]^{(h)}_t\big) - \nabla \log p_{t|0}\big(\mathbf{X}^{(l)}_t \mid \mathbf{X}^{(l)}\big)\right\|^2\right], \tag{4}$$

where $p^{\phi}_{t|0}$ and $p_{t|0}$ are the Gaussian transition kernels of the frequency-domain and time-domain processes, with t sampled uniformly over $\{1,\dots,T\}$ and $\{1,\dots,t^*\}$, respectively. Detailed training procedures are deferred to Appendix A.

Conditioning via Cross-Attention A key design choice in $\mathcal{L}_{\text{time}}$ is how the high-frequency context $\phi[\widetilde{\mathbf{X}}]^{(h)}_t$ enters $\mathbf{s}_{\theta_{\text{time}}}$, since this conditioning encourages the generated low- and high-frequency components to remain consistent. Global conditioning mechanisms such as AdaLN or FiLM compress $\phi[\widetilde{\mathbf{X}}]^{(h)}_t$ into a single summary vector, potentially weakening access to fine-grained structure across individual frequencies. We instead adopt cross-attention, which preserves $\phi[\widetilde{\mathbf{X}}]^{(h)}_t$ as a sequence of per-frequency tokens, allowing each low-frequency position to query specific high-frequency components rather than only a global summary.

Algorithm 1 Two-stage sampling

Input: trained $\mathbf{s}_{\theta_{\text{freq}}}, \mathbf{s}_{\theta_{\text{time}}}$, schedules $\{\alpha_i(t), \sigma_{i,t}\}_{i=1}^{2}$
$\phi[\widetilde{\mathbf{X}}]_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN})$ ▷ **Stage 1:** frequency domain denoising
for $t = T, \dots, t^* + 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN})$
  $\phi[\widetilde{\mathbf{X}}]_{t-1} = \frac{1}{\sqrt{\alpha_1(t)}}\Big(\phi[\widetilde{\mathbf{X}}]_t + \big(1-\alpha_1(t)\big)\,\mathbf{s}_{\theta_{\text{freq}}}\Big(\phi[\widetilde{\mathbf{X}}]_t,\,t\Big)\Big) + \sigma_{1,t}\,\mathbf{z}$
end for
$\mathbf{X}^{(l)}_{t^*} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN})$ ▷ **Stage 2:** joint denoising
for $t = t^*, \dots, 1$ **do**
  $\mathbf{z}, \mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN})$ if $t > 1$, else $\mathbf{z} = \mathbf{z}' = \mathbf{0}$
  $\mathbf{X}^{(l)}_{t-1} = \frac{1}{\sqrt{\alpha_2(t)}}\Big(\mathbf{X}^{(l)}_t + \big(1-\alpha_2(t)\big)\,\mathbf{s}_{\theta_{\text{time}}}\Big(\mathbf{X}^{(l)}_t,\,t,\,\phi[\widetilde{\mathbf{X}}]^{(h)}_t\Big)\Big) + \sigma_{2,t}\,\mathbf{z}$
  $\phi[\widetilde{\mathbf{X}}]_{t-1} = \frac{1}{\sqrt{\alpha_1(t)}}\Big(\phi[\widetilde{\mathbf{X}}]_t + \big(1-\alpha_1(t)\big)\,\mathbf{s}_{\theta_{\text{freq}}}\Big(\phi[\widetilde{\mathbf{X}}]_t,\,t\Big)\Big) + \sigma_{1,t}\,\mathbf{z}'$
end for
Reconstruct $\widetilde{\mathbf{X}}^{(h)}_0$ from $\phi[\widetilde{\mathbf{X}}]^{(h)}_0$ on $\mathcal{K}_{\text{high}}$
return generated sequence $\mathbf{X}_0 = \mathbf{X}^{(l)}_0 + \mathcal{F}^{-1}[\widetilde{\mathbf{X}}^{(h)}_0]$

4.3 Sampling

Sampling mirrors the two-phase structure of the forward process and proceeds in two stages. For $t \in \{T, \dots, t^* + 1\}$, the time-domain reverse process has not yet started, so we run only the frequency-domain denoiser $\mathbf{s}_{\theta_{\text{freq}}}$ to progressively shape the high-frequency components, producing a partially denoised $\phi[\widetilde{\mathbf{X}}]^{(h)}_{t^*}$ that will serve as conditioning context in the next stage. We initialize $\mathbf{X}^{(l)}_{t^*}$ from noise, then run both denoisers in parallel for $t \in \{t^*, \dots, 1\}$: $\mathbf{s}_{\theta_{\text{freq}}}$ continues to refine $\phi[\widetilde{\mathbf{X}}]_t$, while $\mathbf{s}_{\theta_{\text{time}}}$ updates $\mathbf{X}^{(l)}_t$ conditioned on the current noisy high-frequency entries $\phi[\widetilde{\mathbf{X}}]^{(h)}_t$ via cross-attention. This coupling encourages consistency between the two domains throughout sampling. Both updates use the standard reverse-process variance

$$\sigma^2_{i,t} = \frac{1-\bar{\alpha}_i(t-1)}{1-\bar{\alpha}_i(t)}\big(1-\alpha_i(t)\big), \quad i \in \{1,2\}, \quad \bar{\alpha}_i(0) = 1.$$

The final sample is then $\mathbf{X}_0 = \mathbf{X}^{(l)}_0 + \mathcal{F}^{-1}[\widetilde{\mathbf{X}}^{(h)}_0]$; the full procedure is summarized in Algorithm 1.

5 NUMERICAL RESULTS

We empirically evaluate the proposed Hybrid Diffusion framework using two sets of experiments. First, we compare it against time-domain only and frequency-domain only diffusion models, following the setup of Crabbé et al. (2024), to demonstrate the advantage of using both domains. Second, we compare it against existing time-series generative models, following the setup of Yuan & Qiao (2024), to assess the overall quality of the generated samples. Implementation details and dataset links can be found in Appendix C.

5.1 COMPARISON WITH SINGLE-DOMAIN DIFFUSION MODELS

In this part, we follow the experimental setup of Crabbé et al. (2024) to study whether combining time and frequency information improves generation performance. Our Hybrid Diffusion Model is compared with diffusion models that use only the time or frequency domain. The models are tested on five real-world datasets described in Table 1. The quality of generated samples is evaluated via their similarity to the real data in both time and frequency domains.

Table 1: Datasets used in our experiments and their characteristics.

Dataset	Field	# Samples	# Steps N	# Features d
ECG (Kachuee et al., 2018)	Healthcare	87,553	187	1
NASDAQ-2019 (Onyshchak, 2020)	Finance	4,827	252	5
NASA-Charge (Saha & Goebel, 2007)	Engineering	2,396	251	4
NASA-Discharge (Saha & Goebel, 2007)	Engineering	1,755	134	5
US-Droughts (Minixhofer, 2021)	Climate	2,797	365	13

Models For each dataset, we parametrize the time score model $\mathbf{s}_{\theta_{\text{time}}}$ and the frequency score model $\mathbf{s}_{\theta_{\text{freq}}}$ as transformer encoders with 10 layers of multi-head self-attention and feed-forward blocks. Each layer uses 12 attention heads with model dimension $d_{\text{model}} = 72$ and a feed-forward hidden dimension of $4\,d_{\text{model}} = 288$. Cross-attention is inserted at layers 0, 3, 6, and 9 of the time-domain model. For the single-domain baselines, we follow the tuned configuration from the official implementation of Crabbé et al. (2024), using a feed-forward hidden dimension of 2048. We set $t^* = 500$ and $T = 1000$.

Metrics Following Crabbé et al. (2024), we measure the quality of generated time series via the sliced Wasserstein (SW) distance. Specifically, the Wasserstein distance measures the minimum cost of transporting one distribution onto another (Villani, 2009). The SW distance further projects both distributions onto random one-dimensional directions and averages the resulting 1D Wasserstein distances. We draw 10,000 generated sequences and, following Crabbé et al. (2024), measure their fidelity to the training data using the SW distance in both the time and frequency domains. Lower SW values indicate closer agreement between the generated samples and the training time series.

Results We compute the SW distance along 1000 random projection directions for each setting, and Table 2 reports the resulting mean $\pm$ 2 standard errors. Hybrid Diffusion achieves the best performance on all five datasets, in both the time-domain and frequency-domain SW metrics. The improvement is most pronounced on US-Droughts (about 80% reduction over time-domain diffusion, e.g., 0.584 vs. 2.849 in the frequency-domain metric) and ECG (about 65%, e.g., 0.007 vs. 0.020). On NASDAQ-2019, NASA-Charge, and NASA-Discharge, Hybrid yields consistent gains of roughly 33–40% over time-domain diffusion in both metric domains. Hybrid Diffusion also improves on the frequency-domain baseline in both metrics on every dataset. On ECG, it reduces the frequency-domain SW distance from 0.012 to 0.007 and the time-domain SW distance from 0.015 to 0.009. The computational-cost comparison further shows that the two Hybrid denoisers together use 1.42M parameters, compared with 3.21M for either single-domain baseline (Appendix C).

Table 2: SW distances in the time and frequency domains (bold indicates best performance).

Dataset	Metric Domain	Diffusion Domain		
		Hybrid	Frequency	Time
ECG	Frequency	**0.007±.000**	0.012±.000	0.020±.000
	Time	**0.009±.000**	0.015±.000	0.021±.000
NASDAQ-2019	Frequency	**41.341±1.905**	45.812±2.096	64.056±3.040
	Time	**39.606±1.853**	43.602±2.044	60.512±2.960
NASA-Charge	Frequency	**0.181±.007**	0.211±.008	0.270±.006
	Time	**0.192±.007**	0.229±.008	0.316±.008
NASA-Discharge	Frequency	**1.955±.086**	1.999±.084	2.974±.134
	Time	**1.970±.084**	2.028±.082	2.942±.134
US-Droughts	Frequency	**0.584±.016**	0.633±.018	2.849±.090
	Time	**0.680±.017**	0.738±.020	2.913±.092

5.2 Comparison with Time-Series Generative Baselines

In this part, we follow the experimental setup of Diffusion-TS (Yuan & Qiao, 2024) to compare Hybrid Diffusion with six time-series generative models: Diffusion-TS (Yuan & Qiao, 2024), TimeGAN (Yoon et al., 2019), TimeVAE (Desai et al., 2021), DiffWave (Kong et al., 2020), DiffTime (Coletta et al., 2023), and Cot-GAN (Xu et al., 2020). All methods are evaluated on four datasets covering finance, engineering, energy, and neuroscience, using sequences of length 24. The data sources and preprocessing details are provided in Appendix C.

Models For each dataset, we parametrize the time score model $\mathbf{s}_{\theta_{\text{time}}}$ and the frequency score model $\mathbf{s}_{\theta_{\text{freq}}}$ as transformer encoders with 4 attention and MLP layers, each with 4 or 8 heads and dimension $d_{\text{model}} = 64$ or 128. Cross-attention is applied at layers 0 and 2 of the time-domain model. The diffusion lengths t^*, T are set as in the previous experiment.

Metrics We adopt three complementary metrics from Yuan & Qiao (2024), all of which are lower-is-better metrics. (i) The *Context-FID Score* (Jeha et al., 2022) computes the Fréchet distance between TS2Vec embeddings of real and generated sequences. (ii) The *Correlational Score* (Ni et al., 2020) measures the absolute difference between their cross-correlation matrices. (iii) The *Discriminative Score* (Yoon et al., 2019) measures the similarity using a classification model to distinguish between the original and synthetic data as a supervised task.

Results Table 3 reports the mean and standard deviation over multiple runs of length-24 sequence generation. Hybrid Diffusion achieves the lowest mean score in 10 of the 12 dataset–metric settings, including all four Discriminative Scores. It ranks second on Energy in Context-FID and third on Stocks in Correlational Score, behind Diffusion-TS and DiffTime. On fMRI, it reduces the Correlational Score from 1.411 for Diffusion-TS to 0.880.

5.3 Ablation: Conditioning and Asymmetric Schedule

Hybrid Diffusion adds two components on top of the frequency split: the cross-attention conditioning of the time-domain denoiser on the high-frequency state (Section 4.2), and the asymmetric schedule $t^* < T$, under which the frequency-domain process runs alone from T to t^* before joint denoising begins (Sections 4.1 and 4.3). Keeping the frequency split fixed, we train four variants: **A1** uses neither, **A2** adds conditioning only, **A3** adds the asymmetric schedule only, and **A4** is the full model. Detailed settings and full results are deferred to Appendix D.

Combining conditioning with the asymmetric schedule gives the lowest SW distance in both domains on all three ablation datasets (Figure 4 and Table 8). The asymmetric schedule alone produces little change: A3 and A1 have similar SW distances. Conditioning alone improves NASA-Charge and

Table 3: Results on multiple time-series datasets (**bold** indicates the best performance, and underline indicates the second-best).

Metric	Methods	Stocks	ETTh_1	Energy	fMRI
Context-FID Score (Lower is better)	**Hybrid Diffusion**	**0.043±.004**	**0.036±.002**	0.129±.009	**0.103±.007**
	Diffusion-TS	0.147±.025	0.116±.010	**0.089±.024**	0.105±.006
	TimeGAN	0.103±.013	0.300±.013	0.767±.103	1.292±.218
	TimeVAE	0.215±.035	0.805±.186	1.631±.142	14.449±.969
	DiffWave	0.232±.032	0.873±.061	1.031±.131	0.244±.018
	DiffTime	0.236±.074	0.299±.044	0.279±.045	0.340±.015
	Cot-GAN	0.408±.086	0.980±.071	1.039±.028	7.813±.550
Correlational Score (Lower is better)	**Hybrid Diffusion**	0.009±.006	**0.039±.008**	**0.838±.151**	**0.880±.014**
	Diffusion-TS	**0.004±.001**	0.049±.008	0.856±.147	1.411±.042
	TimeGAN	0.063±.005	0.210±.006	4.010±.104	23.502±.039
	TimeVAE	0.095±.008	0.111±.020	1.688±.226	17.296±.526
	DiffWave	0.030±.020	0.175±.006	5.001±.154	3.927±.049
	DiffTime	0.006±.002	0.067±.005	1.158±.095	1.501±.048
	Cot-GAN	0.087±.004	0.249±.009	3.164±.061	26.824±.449
Discriminative Score (Lower is better)	**Hybrid Diffusion**	**0.009±.007**	**0.018±.009**	**0.089±.010**	**0.148±.013**
	Diffusion-TS	0.067±.015	0.061±.009	0.122±.003	0.167±.023
	TimeGAN	0.102±.021	0.114±.055	0.236±.012	0.484±.042
	TimeVAE	0.145±.120	0.209±.058	0.499±.000	0.476±.044
	DiffWave	0.232±.061	0.190±.008	0.493±.004	0.402±.029
	DiffTime	0.097±.016	0.100±.007	0.445±.004	0.245±.051
	Cot-GAN	0.230±.016	0.325±.099	0.498±.002	0.492±.018

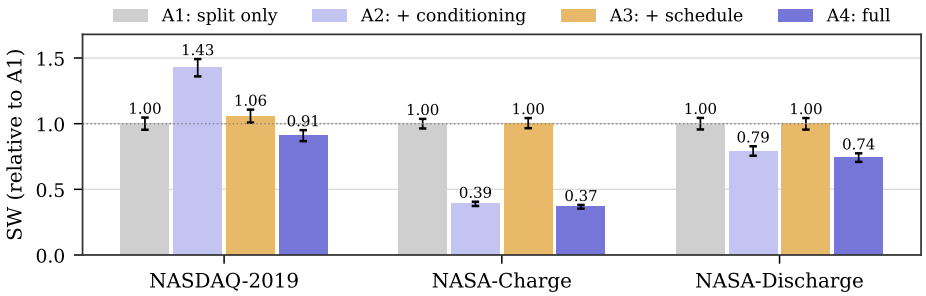


Figure 4: Ablation of cross-attention conditioning and the asymmetric schedule. Bars show the frequency-domain SW distance relative to A1 with ±2 standard errors; the time-domain metric follows the same trend.

NASA-Discharge but worsens NASDAQ-2019, where the frequency-domain SW distance rises from 45.490 to 64.870. The early frequency-only phase may provide a less noisy high-frequency condition when joint denoising begins. We further examine sensitivity to the cutoff f_c and switching time t^* in Appendix E.

6 Conclusion

Hybrid Diffusion generates low- and high-frequency components with coupled diffusion processes in the time and Fourier domains. A full-horizon frequency-domain process supplies the high-frequency state that conditions a shorter time-domain process. Hybrid Diffusion attains lower SW distances than the time-only and frequency-only baselines and the lowest mean scores in 10 of 12 dataset–metric comparisons against six generative models on four additional datasets using length-24 sequences. Combining conditioning with the asymmetric schedule achieves the lowest SW distances in both domains on all three ablation datasets.

AI USE STATEMENT

We used generative AI tools to assist with literature search, feedback on methodology and experiments, code review, and manuscript drafting and editing. The authors reviewed all AI-assisted work and independently verified the mathematical claims, citations, code, experimental outputs, and final manuscript. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The appendix describes the training and sampling algorithms, model architecture, hyperparameters, baselines, evaluation procedures, and complete time-series results needed to reproduce the reported findings. We also plan to release the code in the future.

REFERENCES

Ronald Newbold Bracewell. *The Fourier transform and its applications*. McGraw-hill New York, 1986.

Jou-Fan Chen, Wei-Lun Chen, Chun-Ping Huang, Szu-Hao Huang, and An-Pin Chen. Financial time-series data analysis using deep convolutional neural networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pp. 87–92. IEEE, 2016.

Minshuo Chen, Renyuan Xu, Yumin Xu, and Ruixun Zhang. Diffusion factor models: Generating high-dimensional returns with factor structure. *arXiv preprint arXiv:2504.06566*, 2025.

Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. *Advances in Neural Information Processing Systems*, 36, 2023.

Jonathan Crabbé, Nicolas Huynh, Jan Stanczuk, and Mihaela Van Der Schaar. Time series diffusion in the frequency domain. *arXiv preprint arXiv:2402.05933*, 2024.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A Fourier space perspective on diffusion models. *arXiv preprint arXiv:2505.11278*, 2025.

Asadullah Hill Galib, Pang-Ning Tan, and Lifeng Luo. FIDE: Frequency-inflated conditional diffusion model for extreme-aware time series generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 114434–114457. Curran Associates, Inc., 2024. doi: 10.52202/079017-3634. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cfce727868dcaf5295c0125f9d6fbc0b-Paper-Conference.pdf.

Yuan Gao, Haokun Chen, Xiang Wang, Zhicai Wang, Xue Wang, Jinyang Gao, and Bolin Ding. Diffsformer: A diffusion transformer on stock factor augmentation. *arXiv preprint arXiv:2402.06656*, 2024.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

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

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. PSA-GAN: Progressive self attention GANs for synthetic time series. In *International Conference on Learning Representations*, 2022.

Seungwoo Jeong, Junghyo Sohn, Jaehyun Jeon, and Heung-Il Suk. Frequency-conditioned diffusion models for time series generation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1114–1123, 2025. doi: 10.1145/3746252.3761050.

Chisimkwuo John, Emmanuel J Ekpenyong, and Charles C Nworu. Imputation of missing values in economic and financial time series data using five principal component analysis approaches. *CBN Journal of Applied Statistics*, 10(1):3, 2019.

Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. ECG heartbeat classification: A deep transferable representation. In *2018 IEEE international conference on healthcare informatics (ICHI)*, pp. 443–444. IEEE, 2018.

Maksims Kazijevs and Manar D Samad. Deep imputation of missing values in time series health data: A review with benchmarking. *Journal of Biomedical Informatics*, pp. 104440, 2023.

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

Huiping Li, Meng Li, Xi Lin, Fang He, and Yinhai Wang. A spatiotemporal approach for traffic data imputation with complicated missing patterns. *Transportation Research Part C: Emerging Technologies*, 119:102730, 2020.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, and Jianfeng Gao. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

Christoph Minixhofer. Predict droughts using weather and soil data, 2021.

Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional Sig-Wasserstein GANs for time series generation. *arXiv preprint arXiv:2006.05421*, 2020.

NVIDIA, Yuval Atzmon, Maciej Bala, Yogesh Balaji, Tiffany Cai, Yin Cui, Jiaojiao Fan, Yunhao Ge, Siddharth Gururani, Jacob Huffman, Ronald Isaac, Pooya Jannaty, Tero Karras, Grace Lam, J. P. Lewis, Aaron Licata, Yen-Chen Lin, Ming-Yu Liu, Qianli Ma, Arun Mallya, Ashlee Martino-Tarr, Doug Mendez, Seungjun Nah, Chris Pruett, Fitsum Reda, Jiaming Song, Ting-Chun Wang, Fangyin Wei, Xiaohui Zeng, Yu Zeng, and Qinsheng Zhang. Edify Image: High-quality image generation with pixel space laplacian diffusion models. *arXiv preprint arXiv:2411.07126*, 2024. URL https://arxiv.org/abs/2411.07126.

Oleh Onyshchak. Stock market dataset, 2020.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022.

B Saha and T. Goebel. Battery data set, NASA Ames Prognostics Data Repository, 2007.

Lifeng Shen, Kai Syun Hou, Weiyu Chen, and James Kwok. Latent-to-data cascaded diffusion models for unconditional time series generation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3b246503dce7d5f1b55d52ff43123e42-Abstract-Conference.html.

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

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

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and Alex Kai Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1544–1561, 2020.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

Cédric Villani. The Wasserstein distances. In *Optimal transport: old and new*, pp. 93–111. Springer, 2009.

Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. COT-GAN: Generating sequential data via causal optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8798–8809, 2020.

Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 22552–22562, 2023.

Xinyu Yang, Yu Sun, Xiaojie Yuan, and Xinyang Chen. Frequency-aware generative models for multivariate time series imputation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 52595–52623. Curran Associates, Inc., 2024. doi: 10.52202/079017-1667. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5e364212327fa3a59ae3595b025c469f-Paper-Conference.pdf.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ceylan Yozgatligil, Sipan Aslan, Cem Iyigun, and Inci Batmaz. Comparison of missing value imputation methods in time series: The case of turkish meteorological data. *Theoretical and Applied Climatology*, 112:143–167, 2013.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation, 2024. URL https://arxiv.org/abs/2403.01742.

A Training Algorithm Procedures for Section 4

The following algorithms give the detailed training procedures corresponding to the score-matching losses in Equations (3) and (4). Each procedure samples a diffusion time, constructs the noisy state from the appropriate forward process, and updates the corresponding denoiser by gradient descent.

Algorithm 2 Frequency-domain training

Input: training data $\phi[\widetilde{\mathbf{X}}]$, schedule $\bar{\alpha}_1$
repeat
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN})$
  $\phi[\widetilde{\mathbf{X}}]_t = \sqrt{\bar{\alpha}_1(t)}\, \phi[\widetilde{\mathbf{X}}] + \sqrt{1 - \bar{\alpha}_1(t)}\, \boldsymbol{\epsilon}$
  Take a gradient descent step on loss (3)
until converged
return trained $\mathbf{s}_{\theta_{\mathrm{freq}}}$

Algorithm 3 Time-domain training

Input: training data $\mathbf{X}^{(l)}$, $\phi[\widetilde{\mathbf{X}}]^{(h)}$, schedules $\bar{\alpha}_1, \bar{\alpha}_2$
repeat
  $t \sim \mathrm{Uniform}(\{1, \dots, t^*\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dN}),\ \boldsymbol{\epsilon}^{(h)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d|\mathcal{K}_{\mathrm{high}}|})$
  $\mathbf{X}_t^{(l)} = \sqrt{\bar{\alpha}_2(t)}\, \mathbf{X}^{(l)} + \sqrt{1 - \bar{\alpha}_2(t)}\, \boldsymbol{\epsilon}$
  $\phi[\widetilde{\mathbf{X}}]_t^{(h)} = \sqrt{\bar{\alpha}_1(t)}\, \phi[\widetilde{\mathbf{X}}]^{(h)} + \sqrt{1 - \bar{\alpha}_1(t)}\, \boldsymbol{\epsilon}^{(h)}$
  Take a gradient descent step on loss (4)
until converged
return trained $\mathbf{s}_{\theta_{\mathrm{time}}}$

B Experiments

B.1 Experiment on Gaussian Process Data

In this section, we conduct experiments on a Fourier diffusion transformer for learning synthetic Gaussian process data. Through these experiments, we compare the ability of time-domain and frequency-domain diffusion models to capture patterns of different frequencies.

Experiment Setup We consider synthetic Gaussian process data of total length $N = 100$. We set $\mu_h = 0$ for all $h \in \{1, \dots, N\}$. The covariance function is chosen as the Gaussian kernel: $\mathbf{\Gamma}(h_1, h_2) = \exp(-(h_1 - h_2)^2/(2l^2))$. We generate 50,000 sequences from each Gaussian process, using 90% for training and 10% for testing. The diffusion transformer is implemented on the basis of the Fourier diffusion framework proposed by Crabbé et al. (2024). The score function is implemented using transformer encoders, consisting of 10 stacked attention–MLP layers. Each layer employs 12 attention heads with hidden size $d_{\mathrm{model}} = 72$.

Influence of Bandwidth on Capturing Different Frequencies We study the influence of high-frequency components in the data on the performance of diffusion transformers. We vary the bandwidth of the Gaussian kernel over $\{2, 4, 8, 16\}$. A larger bandwidth l results in less high-frequency content. After training in both time and frequency domains, we generate 1000 samples to compute the relative error of the mean DFT magnitude spectrum between the generated and ground-truth data. As shown in Figure 5, the time-domain diffusion model has a potential limitation in capturing high-frequency information, whereas the frequency-domain diffusion model can compensate for this shortcoming. Time-domain diffusion achieves lower covariance errors in three of the four settings, as shown in Figure 6.

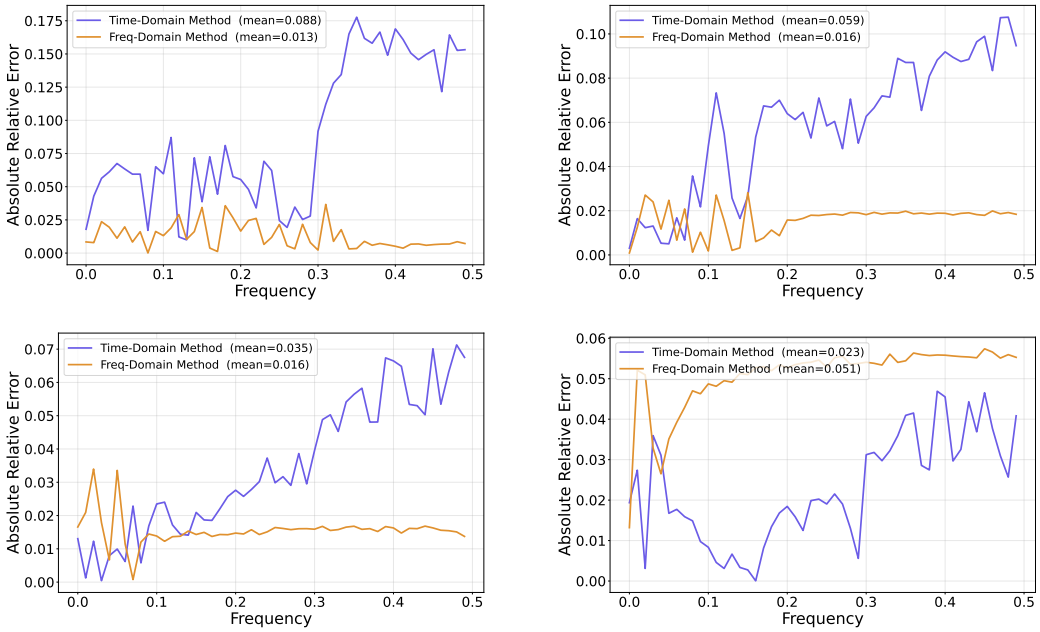


Figure 5: When $l = 2$, the time-domain diffusion model exhibits large errors in the high-frequency regions. As the bandwidth increases, the time-domain model achieves higher accuracy. Finally, when $l = 16$, the time-domain model outperforms the frequency-domain diffusion model at all frequencies.

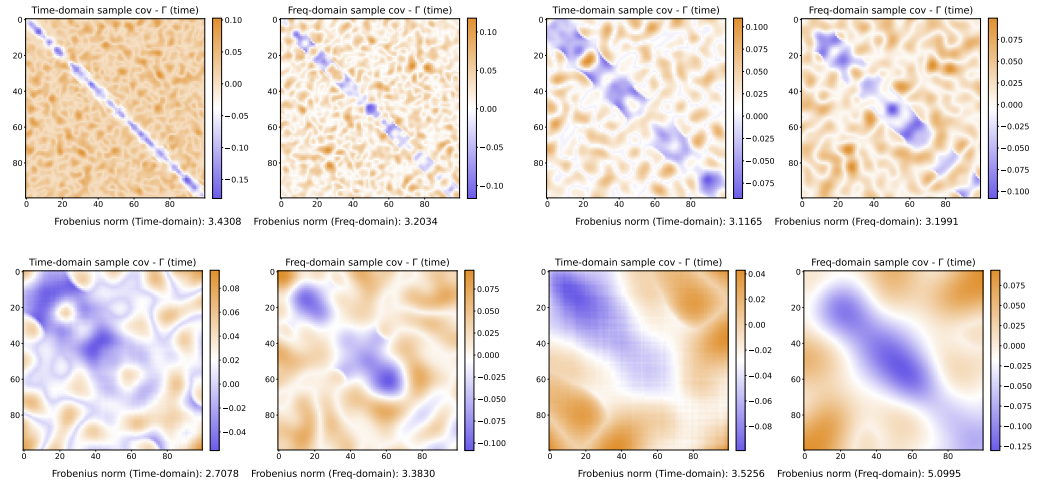


Figure 6: We demonstrate the matrix obtained by taking the element-wise difference between the covariance matrix estimated from the sampled data at each time step and the true covariance matrix, as well as its Frobenius norm.

C Experimental Details

All models are trained with the AdamW optimizer using a cosine learning-rate schedule with linear warm-up over the first 10% of training steps. The frequency-domain model adopts a linear β-schedule with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20$, while the time-domain schedule satisfies $\sqrt{\bar{\alpha}_2(t)} = \sqrt{\bar{\alpha}_1(2t)}$ for $0 \le t \le t^* = T/2$. This corresponds to $\beta_2(t) = 2\beta_1(2t)$ under the continuous-time definition in Section 4.1. All experiments are conducted on NVIDIA RTX 5090 GPUs. Table 5 shows the statistics of the datasets in Section 5.2, and all datasets are available online at the listed links. Tables 6 and 7 report the hyperparameter configurations used for the experiments in Sections 5.1 and 5.2,

respectively. Here, $lr_{\max}$ denotes the peak learning rate, L the number of transformer layers, and f_c the cutoff frequency index defined in Section 4.1. "Cross-Attn Layers" lists the layer indices at which cross-attention to $\phi[\widetilde{\mathbf{X}}]_t^{(h)}$ is applied.

Computational Cost Hybrid achieves its gains over the single-domain baselines with narrower feed-forward layers (288 versus 2048). Its frequency-domain and time-domain denoisers contain 0.66M and 0.76M parameters, respectively, totaling 1.42M—just 44% of the 3.21M parameters in either single-domain baseline. For two-GPU execution, we place one denoiser on each GPU to run them concurrently.

The training objectives in Equations (3) and (4) use separate parameter sets, and the time-domain denoiser receives a high-frequency condition constructed by noising the training data. The two denoisers can therefore be trained fully in parallel. On a single GPU, training both denoisers for 400 epochs takes 44.9 minutes, compared with 30.8 and 30.7 minutes for the time-domain and frequency-domain baselines, respectively. Concurrent execution on two GPUs reduces Hybrid's training time by 40.8% to 26.6 minutes (Table 4).

Sampling also allows concurrent execution during the second phase. In Algorithm 1, the frequency-domain chain advances independently, while the time-domain denoiser uses the current high-frequency state $\phi[\widetilde{\mathbf{X}}]_t^{(h)}$. We run the two denoisers in separate Python threads. Before each frequency-domain update in this phase, we pass the current high-frequency condition to the time-domain denoiser through host memory. Both denoisers can then advance using the same frequency-domain iterate, preserving the updates in the original sampler. Transferring the 2.5 MB condition takes less than 1 ms per step. On a single GPU, generating 1000 sequences with Hybrid takes 192.2 s, compared with 143.8 s and 144.0 s for the time-domain and frequency-domain baselines, respectively. Concurrent execution on two GPUs reduces Hybrid's sampling time by 28.5% to 137.4 s (Table 4).

Table 4: Parameter count and wall-clock cost on NVIDIA RTX 5090 GPUs. Hybrid runs sequentially on one GPU or concurrently on two GPUs, one per network; each single-domain baseline uses one GPU. Training is 400 epochs; sampling is 1000 sequences with 1000 reverse steps.

Model	GPUs	Parameters	Training (400 epochs)	Sampling (1000 sequences)
Hybrid	1	**1.42M**	44.9 min	192.2 s
Hybrid	2	**1.42M**	**26.6 min**	**137.4 s**
Time-domain only	1	3.21M	30.8 min	143.8 s
Frequency-domain only	1	3.21M	30.7 min	144.0 s

Table 5: Dataset details.

Dataset	# of Samples	dim	Link
Stocks	3773	6	https://finance.yahoo.com/quote/GOOG
ETTh_1	17420	7	https://github.com/zhouhaoyi/ETDataset
Energy	19711	28	https://archive.ics.uci.edu/ml/datasets
fMRI	10000	50	https://www.fmrib.ox.ac.uk/datasets

D Component Ablation

D.1 Purpose

Our method combines three ideas: splitting the spectrum at a cutoff frequency f_c, giving the two branches different diffusion schedules through the switching time t^*, and letting the high-frequency branch guide the low-frequency one through cross-attention. With the frequency split held fixed, this ablation measures the individual and combined effects of cross-attention conditioning and asymmetric scheduling.

Table 6: Hyperparameter choices for section 5.1.

Dataset	Normalization	$lr_{\max}$	Epochs	Cross-Attn Layers	f_c
ECG	No	0.001	200	$[0, 3, 6, 9]$	$\lfloor N/4 \rfloor + 1$
NASA-Charge	No	0.001	400	$[0, 3, 6, 9]$	$\lfloor N/4 \rfloor + 1$
NASA-Discharge	Z-score	0.0005	400	$[0, 3, 6, 9]$	$\lfloor N/4 \rfloor + 1$
NASDAQ-2019	Z-score	0.0008	400	$[0, 3, 6, 9]$	$\lfloor N/4 \rfloor + 1$
US-Droughts	Z-score	0.01	200	$[0, 3, 6, 9]$	$\lfloor N/4 \rfloor + 1$

Table 7: Hyperparameter choices for section 5.2.

Dataset	Norm.	d_{model}	n_{head}	L	$lr_{\max}$	Epochs	Cross-Attn Layers	f_c
Stocks	MinMax	64	8	4	0.002	400	$[0, 2]$	$\lfloor N/4 \rfloor + 1$
ETTh_1	MinMax	64	4	4	0.002	400	$[0, 2]$	$\lfloor N/4 \rfloor + 1$
Energy	MinMax	128	8	4	0.002	400	$[0, 2]$	$\lfloor N/4 \rfloor + 1$
fMRI	MinMax	128	8	4	0.002	400	$[0, 2]$	$\lfloor N/4 \rfloor + 1$

D.2 Design

We train four models per dataset. They form a 2×2 grid over the two parts our method adds on top of the frequency split:

- **A1** splits the spectrum, trains and samples the two branches on their own, and adds them back together with the inverse DFT only at the end. It shows what the frequency split alone gives us.
- **A2** adds cross-attention conditioning, keeping $t^* = T$.
- **A3** adds the asymmetric schedule ($t^* = 0.5\,T$) without conditioning.
- **A4** is the full model of the main text: conditioning and $t^* = 0.5\,T$.

Two switches separate the four models. The first turns the cross-attention layers of the low-frequency network on or off. The second is t^*, which sets the time-domain horizon and the transition from the frequency-only phase to joint sampling. Setting $t^* = T$ removes the frequency-only phase in A1 and A2.

Model	Cross-attention	t^*/T
A1	off	1.0
A2	on	1.0
A3	off	0.5
A4	on	0.5

All other settings follow Table 6. The high-frequency network is trained on the full spectrum, so it does not depend on either switch; one such network per dataset is shared by all four models, and the four differ only in the low-frequency network and the sampling schedule. Each model generates 10^4 samples with 1000 reverse diffusion steps.

D.3 Results and Analysis

Table 8 reports the full results summarized in Figure 4 of Section 5.3. All values quoted below refer to the frequency-domain SW distance; the time-domain metric shows the same overall pattern.

Combining the two performs best. A4 has the lowest SW distance on all three datasets. Relative to A3, adding conditioning under $t^* = 0.5\,T$ lowers the frequency-domain distance from 48.111 to 41.341 on NASDAQ-2019, from 0.495 to 0.181 on NASA-Charge, and from 2.633 to 1.955 on NASA-Discharge. Relative to A2, the asymmetric schedule also lowers the distance on each dataset.

Table 8: SW distances in the time and frequency domains for the four ablated models (bold indicates best performance).

Dataset	Metric Domain	A1	A2	A3	A4
NASDAQ-2019	Frequency	45.490±2.111	64.870±3.004	48.111±2.248	**41.341±1.905**
	Time	43.482±2.047	62.025±2.918	45.837±2.180	**39.606±1.853**
NASA-Charge	Frequency	0.493±0.018	0.192±0.008	0.495±0.019	**0.181±0.007**
	Time	0.526±0.018	0.205±0.008	0.523±0.018	**0.192±0.007**
NASA-Discharge	Frequency	2.636±0.116	2.087±0.095	2.633±0.116	**1.955±0.086**
	Time	2.654±0.114	2.063±0.092	2.648±0.114	**1.970±0.084**

The schedule alone has little effect. A1 and A3 have similar SW distances on all three datasets (45.490 vs. 48.111 on NASDAQ-2019, 0.493 vs. 0.495 on NASA-Charge, and 2.636 vs. 2.633 on NASA-Discharge). Without conditioning, the two branches do not exchange information during generation; changing the switching time alone did not improve their combined output.

Conditioning alone gives mixed results. Going from A1 to A2 improves NASA-Charge ($0.493 \rightarrow 0.192$) and NASA-Discharge ($2.636 \rightarrow 2.087$) but degrades NASDAQ-2019 ($45.490 \rightarrow 64.870$). At $t^* = T$, the high-frequency condition has not passed through a frequency-only denoising phase when joint sampling begins. This may explain the mixed outcome.

E Sensitivity to the Cutoff and the Switching Time

E.1 Purpose

The cutoff f_c and the switching time t^* decide how each sequence is split and how the two branches are timed against each other. The main text fixes both. This appendix measures how much the results move when they change, and asks whether the two need to be chosen together or can be chosen one at a time.

E.2 Design

We sweep a full grid of five cutoffs and four switching times, $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $t^*/T \in \{0.25, 0.50, 0.75, 1.00\}$, giving 20 settings per dataset. Here α is the fraction of the real-frequency axis assigned to the low-frequency branch, so $\alpha = 0.5$ is the setting $f_c = \lfloor N/4 \rfloor + 1$ used in the main text, and $t^*/T = 0.50$ is the main-text schedule. The column $t^*/T = 1.00$ is the symmetric schedule of Appendix D.

Everything else follows Table 6. The high-frequency network is trained on the full spectrum and does not depend on either value, so one such network per dataset serves all 20 settings and only the low-frequency network is retrained. Each setting generates 10^4 samples with 1000 reverse diffusion steps, and we report the sliced Wasserstein distance in both the time and the frequency domain.

E.3 Results and Analysis

Intermediate switching times perform best. Across both datasets and all five cutoffs, the best switching time is either $0.5\,T$ or $0.75\,T$. At $t^* = T$, joint denoising begins while the high-frequency condition is still noise. This setting yields the highest SW distance for every cutoff on both datasets, reaching 3.320 at $\alpha = 0.4$ on NASA-Discharge. At $t^* = 0.25\,T$, the condition has undergone more denoising, but the low-frequency branch now has only a quarter of the horizon left to denoise in, and this setting is never the best either.

The cutoff matters less. Although the cutoff has a larger effect at $t^* = T$, its effect remains relatively small at $t^*/T \in \{0.25, 0.5, 0.75\}$. At $t^* = 0.5\,T$, for example, the spread across the five cutoffs is 6% on NASDAQ-2019 and 3% on NASA-Discharge, while varying t^* alone changes the

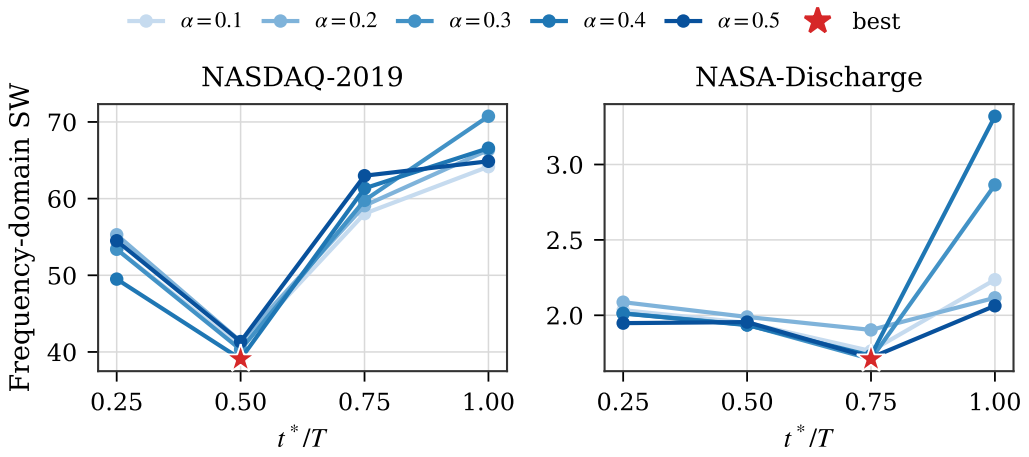


Figure 7: Sensitivity to the cutoff α and the switching time t^*. Each panel is one dataset and each line is one cutoff, with the star marking the best setting of the panel. We report the frequency-domain metric; the time-domain metric moves the same way in every setting.

score by more than 50% on NASDAQ-2019. No single cutoff is consistently best either: the curves cross each other, and the best cutoff changes both across datasets and across switching times.