Bridging Diffusion and Autoregression for Flexible Time Series Synthesis

Xin Wang[1*] **Xuan Zhang**[2*] **Haipeng Zhang**[3] **Chunyu Wei**[1†] **Yueguo Chen**[1]
[1]School of Information, Renmin University of China, Beijing, China
[2]Department of Automation, Tsinghua University, Beijing, China
[3]Alibaba International Digital Commerce Group, Beijing, China
wangxin668@ruc.edu.cn, xuan-zha23@mails.tsinghua.edu.cn,
zhanghaipeng.zhp@lazada.com, weicy15@icloud.com, chenyueguo@ruc.edu.cn

Abstract

Synthetic time series generation is critical for data augmentation, privacy-preserving sharing, and simulation. Autoregressive models extend to arbitrary horizons but suffer compounding errors, while diffusion models achieve high fidelity through bidirectional refinement only at fixed lengths. We propose **Horizon Diffusion**, a unified framework that resolves this tension by decomposing generation into variable-length *horizon blocks* produced autoregressively across blocks yet jointly denoised within each block. Two innovations make this hybrid practical: *Horizon-Causal Attention*, a structured mask that simultaneously preserves cross-block causality and within-block bidirectional refinement, and the *Block Size Curriculum*, a deterministic Warmup–Stable–Decay schedule that fluidly transitions training from full-sequence diffusion to block-wise autoregressive generation. Our method consistently outperforms strong baselines on fidelity, diversity, and downstream utility, and remains stable when sequence length scales by $4\times$ beyond training from a single trained model.[1]

1 Introduction

Synthetic time series generation underpins data augmentation in healthcare Dash et al. [2019], finance Dogariu et al. [2022], and privacy-preserving data sharing Jordon et al. [2020], where high-quality synthetic data enables model training under data scarcity, secure dissemination, and simulation-based decision making. Despite rapid progress, a fundamental limitation persists: existing generators produce sequences of a *fixed* length, fixed at training time. This is costly in practice, where patient trajectories span days to months, financial simulations cover heterogeneous trading horizons, and privacy-preserving releases must mirror the variable-length nature of real records. Current methods cannot adapt to such requirements without retraining or architectural changes.

The two dominant paradigms each fail this requirement for complementary reasons (Figure 1). *Autoregressive* models (①) generate one step at a time and naturally support arbitrary lengths, but every prediction conditions on previously generated tokens, so small errors are reinjected and *compound* as the horizon grows—a well-documented failure mode that effectively caps quality at long horizons. *Diffusion* models (②) instead denoise an entire sequence jointly through bidirectional refinement, but only at the single fixed length they were trained on. Naively concatenating diffusion outputs breaks the bidirectional dependencies that make diffusion effective, producing visible discontinuities at the

*Equal contribution.
†Chunyu Wei is the corresponding author.
[1]Code is available at https://github.com/wxbs/HorizonDiffusion.

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

seams. Neither paradigm offers what real applications need: arbitrary-length, high-fidelity synthesis from a single trained model.

We propose **Horizon Diffusion**, a unified framework that resolves this tension: *train once, generate at arbitrary lengths* (Figure 1-③). Our key insight is to organize synthesis into variable-length *horizon blocks* that are generated *autoregressively across* blocks but *jointly diffused within* each block. The autoregressive scaffold inherits length flexibility from sequential models; within-block diffusion preserves bidirectional refinement and high fidelity. Realizing this hybrid raises two technical challenges that off-the-shelf components cannot address:

Intra-Horizon Causality. Autoregression demands strict causal masking across blocks, while diffusion requires bidirectional attention within a block to perform joint denoising. Standard causal masks suppress within-block bidirectionality; standard full attention violates cross-block causality for arbitrary-length extension.

Cross-Horizon Coherence. A model succeeding within a block can still drift at the boundary between consecutive blocks. Each new block must be internally coherent through bidirectional refinement *and* seamlessly aligned with previously generated context, even when the number and length of blocks at inference differ from those at training.

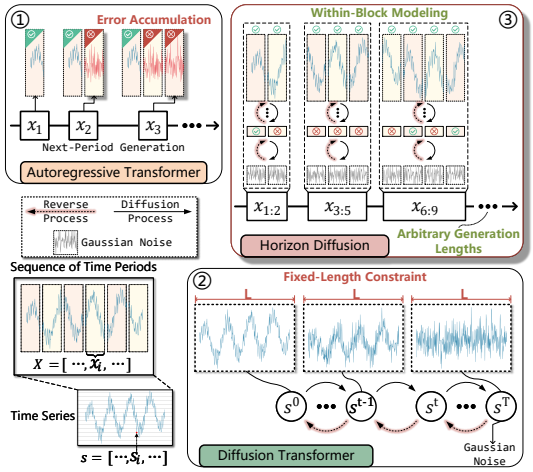


Figure 1: **Generative paradigms.** ① Autoregressive, ② Diffusion, and ③ our **Horizon Diffusion**, which partitions synthesis into variable-length blocks (e.g., $x_{1:2}$, $x_{3:5}$, $x_{6:9}$).

We address Intra-Horizon Causality with **Horizon-Causal Attention**, a structured mask integrated into a Diffusion Transformer (DiT) backbone. Tokens in the current block k attend to (i) all tokens of previously generated clean blocks for stable conditioning, (ii) all noisy tokens within block k for bidirectional joint refinement, and (iii) no future tokens, preserving causality. The resulting block-lower-triangular mask admits efficient sparse computation.

We address Cross-Horizon Coherence with a **Block Size Curriculum** (BSC), a deterministic three-phase Warmup–Stable–Decay schedule. The *Warmup* phase rapidly grows the block size so the model first acquires stable, large-context representations. The *Stable* phase holds the block size at its maximum to consolidate long-range intra-block diffusion dependencies. The *Decay* phase progressively reduces the block size toward the inference block size, exposing the model to increasingly frequent autoregressive boundaries and explicitly drilling cross-block coordination. This easy-to-hard schedule prevents overfitting to any single block configuration and bridges full-sequence diffusion with block-wise autoregressive generation.

Contributions. (i) We introduce the first diffusion-based architecture for native arbitrary-length time series synthesis, combining Horizon-Causal Attention with horizon-block decomposition to unify autoregressive extension and within-block bidirectional refinement. (ii) We propose the Block Size Curriculum, a deterministic three-phase schedule that fluidly transitions training from full-sequence diffusion to block-wise autoregressive generation, enabling robust generalization to arbitrary inference lengths. (iii) Across four datasets and three target lengths, Horizon Diffusion substantially outperforms autoregressive and diffusion baselines on fidelity, diversity, and downstream utility, and remains stable when sequence length scales by $4\times$.

2 Related Work

Time Series Synthesis. Early GAN- and VAE-based methods such as GT-GAN Jeon et al. [2022] and CR-VAE Li et al. [2023] extend classical generative models with neural differential equations or Granger-causal structure. Recent work diversifies the conditioning signal: Time Weaver Narasimhan et al. [2024] and TimeDP Huang et al. [2025] leverage heterogeneous metadata;

VerbalTS Gu et al. [2025] and T2S Ge et al. [2025] use text conditioning; PaD-TS Li et al. [2025] preserves population-level statistics; and ImagenTime Naiman et al. [2024] repurposes vision diffusion via delay embeddings. All commit to a single training-time length.

Diffusion Models for Time Series. Diffusion has become the dominant paradigm for high-fidelity synthesis, with successes extending to structured relational data such as graph anomaly detection Wei et al. [2025a, 2026] and service recommendation Zhang et al. [2026]. Foundational work includes CSDI Tashiro et al. [2021], TimeGrad Rasul et al. [2021], and ScoreGrad Yan et al. [2021]. Architectural advances target efficiency and inductive bias: SSSD Alcaraz and Strodthoff [2023] integrates state-space layers; Diffusion-TS Yuan and Qiao [2024] disentangles temporal components with Fourier losses; SDformer Chen et al. [2024] accelerates inference via discrete tokenization; and TimeDiT Cao et al. [2024] pursues task-agnostic pretraining. A complementary line refines temporal dynamics inside the diffusion process: TSDiff Kollovieh et al. [2023], Latte Feng et al. [2024], DyDiff Guo et al. [2025], TimeBridge Liu et al. [2025], ARMD Gao et al. [2025], and Score-CDM Zhang et al. [2024]. All inherit diffusion's fundamental constraint: generation length is fixed at training time, and naive concatenation breaks the bidirectional refinement that makes diffusion effective.

Autoregressive and Hybrid Models. Autoregressive models naturally handle variable lengths and form the backbone of time series foundation models, including Timer Liu et al. [2024b], GPHT Liu et al. [2024c], and Chronos Ansari et al. [2024]. LLM-adapted variants such as Time-LLM Jin et al. [2024], AutoTimes Liu et al. [2024a], and Lag-Llama Rasul et al. [2023] repurpose pretrained language models, while efficient designs like Moirai Woo et al. [2024], MambaTS Cai et al. [2026], SparseTSF Lin et al. [2024], TimeMixer Wang et al. [2024], and HDT Feng et al. [2025] pursue linear complexity. Although flexible in length, they inherit autoregression's compounding error at long horizons. *Hybrid* works combine the two: WAVE Lu et al. [2025] embeds ARMA-style averages into attention; TimeDART Wang et al. [2025] pairs autoregressive transformers with diffusion for self-supervised pretraining; Block Diffusion Arriola et al. [2025] interpolates between paradigms via block-wise generation; Swift Stock et al. [2025] distills diffusion into single-step consistency models; and TARFVAE Wei et al. [2025b] achieves one-step generation through flow-enhanced VAEs. Horizon Diffusion is closest to this hybrid line, but it realizes the autoregressive–diffusion split *within a single time-series transformer* through one structured mask.

3 Preliminaries

Notation. Let $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_T\} \in \mathbb{R}^{T \times D}$ denote a multivariate time series with T steps and D variates. We partition $\mathbf{S}$ into $M = \lfloor T/P \rfloor$ patches $\mathbf{x}_i \in \mathbb{R}^{P \times D}$ of size P, and embed them in a d_{model}-dimensional latent space. The patch sequence is decomposed into K horizon blocks $\{\mathbf{B}_k\}_{k=1}^{K}$, where the k-th block has length h_k patches. Diffusion uses N timesteps with schedule parameters $\alpha_n, \bar{\alpha}_n$ and noise prediction network $\boldsymbol{\epsilon}_\theta$, optionally conditioned on $\mathbf{c}$.

Problem. Given optional conditioning $\mathbf{c}$, our goal is to learn a generator that samples $\mathbf{S}_{1:H} \sim p_\theta(\mathbf{S}_{1:H} \mid \mathbf{c})$ for *any* target length $H \in \mathbb{Z}^+$, including lengths that differ substantially from those seen during training.

Diffusion Probabilistic Models. A diffusion model Ho et al. [2020] defines a forward Markov chain $q(\mathbf{s}^n \mid \mathbf{s}^{n-1}) = \mathcal{N}(\mathbf{s}^n; \sqrt{1-\beta_n}\mathbf{s}^{n-1}, \beta_n \mathbf{I})$ that admits the closed form $\mathbf{s}^n = \sqrt{\bar{\alpha}_n}\,\mathbf{s}^0 + \sqrt{1-\bar{\alpha}_n}\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with $\alpha_n = 1 - \beta_n$ and $\bar{\alpha}_n = \prod_{i=1}^{n} \alpha_i$. The reverse process is parameterized by a noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{s}^n, n)$ trained with $\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{s}^0, \boldsymbol{\epsilon}, n}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{s}^n, n)\|_2^2\big]$.

4 Methodology

Horizon Diffusion decomposes generation into variable-length *horizon blocks*: blocks are produced *autoregressively* (block k conditions on blocks $1, \ldots, k-1$), while patches *within* each block are denoised *jointly* by diffusion with bidirectional attention. Figure 2 provides the architectural overview.

4.1 Patch-Based Latent Representation

Applying diffusion directly to raw time series at full temporal resolution is prohibitive: adjacent steps are highly redundant, and self-attention over T tokens scales as $O(T^2)$. Following latent diffusion Rombach et al. [2022] and patch-based vision transformers Gu et al. [2022], we operate on

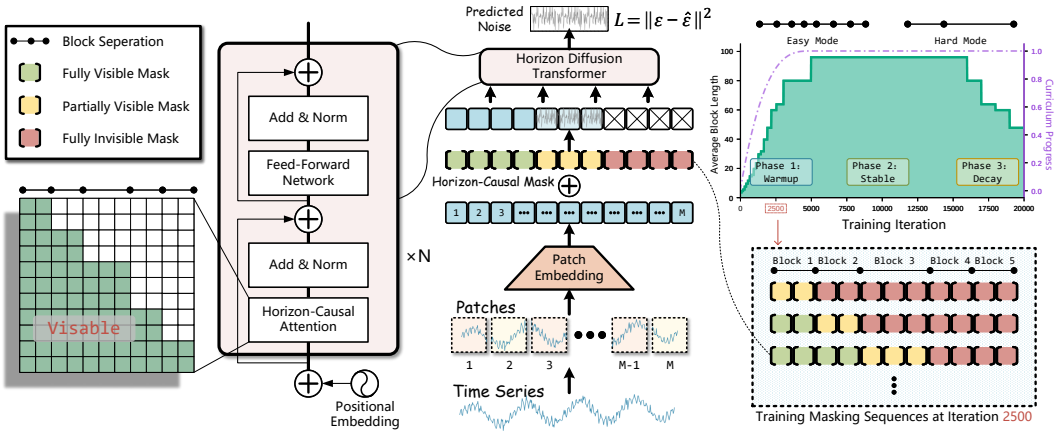


Figure 2: **Horizon Diffusion Overview.** *(Left)* Horizon-Causal Mask: past blocks visible (green), current noisy block bidirectional (yellow), future blocks masked (red). *(Center)* Patchified series processed by the Horizon Diffusion Transformer. *(Right)* Warmup-Stable-Decay Block Size Curriculum.

temporally aggregated patch embeddings that compress redundancy while preserving the dynamics relevant to evaluation.

We partition $\mathbf{S} \in \mathbb{R}^{T\times D}$ into $M = \lfloor T/P \rfloor$ non-overlapping patches $\mathbf{x}_i = \mathbf{S}_{(i-1)P+1:iP} \in \mathbb{R}^{P\times D}$ and embed each via a learnable linear map $\mathbf{x}_i^{\text{emb}} = \mathbf{W}_{\text{enc}}\,\text{vec}(\mathbf{x}_i) + \mathbf{b}_{\text{enc}}$, yielding $\mathbf{X} \in \mathbb{R}^{M\times d_{\text{model}}}$. Patching shortens the effective length by a factor of P and reduces attention cost from $O(T^2)$ to $O(T^2/P^2)$. A symmetric linear decoder $\text{vec}(\hat{\mathbf{x}}_i) = \mathbf{W}_{\text{dec}}\,\hat{\mathbf{x}}_i^{\text{emb}} + \mathbf{b}_{\text{dec}}$ reconstructs the time domain from the denoised embeddings.

4.2 Horizon-Structured Generation

Block decomposition. We partition $\mathbf{X} \in \mathbb{R}^{H_p\times d_{\text{model}}}$ ($H_p = \lfloor H/P \rfloor$) into K contiguous, non-overlapping blocks $\mathbf{X} = [\mathbf{B}_1, \ldots, \mathbf{B}_K]$, with $\mathbf{B}_k \in \mathbb{R}^{h_k\times d_{\text{model}}}$ and $\sum_k h_k = H_p$.

Hybrid visibility states. The decomposition induces three token states (Figure 2, left): (i) *fully visible* past blocks $\{\hat{\mathbf{B}}_1, \ldots, \hat{\mathbf{B}}_{k-1}\}$ (clean conditioning context); (ii) *bidirectionally visible* current block $\mathbf{B}_k^n$ (noisy, denoised jointly); and (iii) *fully masked* future blocks $\{\mathbf{B}_{k+1}, \ldots, \mathbf{B}_K\}$, replaced by learnable mask embeddings $\boldsymbol{\zeta}_j$ to preserve causal integrity. The model input for generating block k at diffusion timestep n is

$$\mathbf{Z}_k^n = \big[\mathbf{c},\ \hat{\mathbf{B}}_1, \ldots, \hat{\mathbf{B}}_{k-1},\ \mathbf{B}_k^n,\ \boldsymbol{\zeta}_{k+1}, \ldots, \boldsymbol{\zeta}_K\big], \tag{1}$$

where $\mathbf{c}$ are conditioning tokens (class embeddings, attributes, or prefix patches).

Structured noise. Diffusion noise is applied *only* to the current block:

$$\mathbf{B}_k^n = \sqrt{\bar{\alpha}_n}\,\mathbf{B}_k^0 + \sqrt{1-\bar{\alpha}_n}\,\boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

This produces a structured noise gradient along the sequence: zero noise on past blocks (stable conditioning), controlled noise on the current block (active denoising target), and full masking on future blocks (causal isolation).

Horizon-Causal Attention To realize the hybrid visibility states inside transformer, we define a structured mask. For block k, the mask $\mathbf{M}_k \in \{0, -\infty\}^{L\times L}$ over input length $L = L_c + H_p$ is

$$\mathbf{M}_k[i,j] = \begin{cases} 0, & \text{if } j \le e_k \quad \text{(attend to past + current)},\\ -\infty, & \text{if } j > e_k \quad \text{(block future)}, \end{cases} \tag{3}$$

for any query i in the current block. Conditioning and past-block positions retain their natural visibility up to and including themselves. The mask is block-lower-triangular and admits sparse attention with cost $O(K\bar{h}^2 + K^2\bar{h})$, where $\bar{h} = H_p/K$, compared to $O(H_p^2)$ for full bidirectional attention; the savings are substantial when $K \ll H_p$.

4.3 Temporal Diffusion Transformer

The noise predictor $\boldsymbol{\epsilon}_\theta$ instantiates a Diffusion Transformer (DiT) Peebles and Xie [2023] adapted for temporal data. Each input token is augmented with sinusoidal positional encoding $\mathbf{p}_t$ and a learned diffusion-timestep embedding $\mathbf{e}_n$: $\mathbf{h}_t^{(0)} = \mathbf{z}_t^n + \mathbf{p}_t + \mathbf{e}_n$. The sequence is processed through L_{layers} transformer blocks, each combining horizon-causal multi-head self-attention with a feed-forward network, residual connections, and layer normalization:

$$\tilde{\mathbf{H}}^{(\ell)} = \mathrm{LN}\Big(\mathbf{H}^{(\ell-1)} + \mathrm{MHSA}_{\mathbf{M}_k}\Big(\mathbf{H}^{(\ell-1)}\Big)\Big), \quad \mathbf{H}^{(\ell)} = \mathrm{LN}\Big(\tilde{\mathbf{H}}^{(\ell)} + \mathrm{FFN}\Big(\tilde{\mathbf{H}}^{(\ell)}\Big)\Big). \tag{4}$$

For each head i, $\text{Attention}_i = \text{softmax}\big(\mathbf{Q}_i\mathbf{K}_i^\top/\sqrt{d_k} + \mathbf{M}_k\big)\,\mathbf{V}_i$. The final hidden states at positions $[s_k, e_k]$ are projected to predict the noise on the current block: $\hat{\boldsymbol{\epsilon}}_k = \mathbf{W}_{\text{out}}\mathbf{H}_{s_k:e_k}^{(L_{\text{layers}})} + \mathbf{b}_{\text{out}}$.

4.4 Block Size Curriculum

Motivation. Training with a fixed block size confines the model to a single boundary configuration and degrades length generalization. Stochastic boundary sampling reduces overfitting but introduces high variance: the model sees an arbitrary mix of easy (large-block) and hard (many-boundary) configurations from the first iteration, slowing convergence and providing no explicit curriculum. We therefore propose the **Block Size Curriculum** (BSC), a deterministic Warmup–Stable–Decay schedule that modulates the block size $h(i)$ over training iterations i along an explicit easy-to-hard progression: large, easy configurations first master *within-block* dependencies, then progressively smaller blocks expose the model to the *cross-block* boundaries that match inference, bridging full-sequence diffusion and block-wise autoregressive generation.

Discrete block-size space. To partition a training sequence of length L exactly without truncation, we restrict $h(i)$ to divisors of L: $\mathcal{V} = \{v \in \mathbb{N} \mid L \bmod v = 0,\ v_{\min} \le v \le v_{\max}\}$, sorted in increasing order (e.g., $v_{\min} = 4$, $v_{\max} = 96$ for $L = 960$). Let $M = |\mathcal{V}| - 1$.

Phase 1: Warmup (iter $i < I_{\text{warmup}}$). With progress $t = i/I_{\text{warmup}}$, a cubic ease-out $\phi(t) = 1 - (1-t)^3$ maps to index $j = \lfloor M \cdot \phi(t) \rfloor$, giving $h(i) = \mathcal{V}[j]$. The cubic shape escapes very small blocks early and quickly reaches large-context training.

Phase 2: Stable ($I_{\text{warmup}} \le i < I_{\text{total}} - I_{\text{decay}}$). We hold $h(i) = v_{\max}$ so the model consolidates long-range intra-block dependencies. The regime closest to standard fixed-length diffusion, providing a strong global prior.

Phase 3: Decay ($i \ge I_{\text{total}} - I_{\text{decay}}$). With $t_{\text{decay}} = (i - (I_{\text{total}} - I_{\text{decay}}))/I_{\text{decay}}$, we step through a descending sequence $\mathcal{D}$ that terminates at the inference block size h_{target} (e.g., 48): $h(i) = \mathcal{D}[\min(\lfloor t_{\text{decay}} \cdot |\mathcal{D}| \rfloor, |\mathcal{D}| - 1)]$. This systematically increases boundary frequency, drilling cross-block coordination under conditions that mirror inference.

4.5 Training and Optimization

At iteration i, BSC determines the active block size $h(i)$. Because $h(i)$ is always a divisor of L, the training sequence partitions cleanly into $K = L/h(i)$ equal-length blocks. For each block we sample a diffusion timestep $n \sim \mathcal{U}(1, N)$, construct $\mathbf{Z}_k^n$ via Eq. (1), and minimize

$$\mathcal{L}_k = \mathbb{E}_{n,\boldsymbol{\epsilon}_k}\Big[\big\|\boldsymbol{\epsilon}_k - \boldsymbol{\epsilon}_\theta(\mathbf{Z}_k^n, n, \mathbf{M}_k)\big\|_2^2\Big], \qquad \mathcal{L}_{\text{total}} = \sum_{k=1}^{K} \lambda_k \mathcal{L}_k, \tag{5}$$

with uniform weights $\lambda_k = 1$. We use teacher forcing during training, feeding ground-truth clean blocks $\mathbf{B}_j^0$ as historical context rather than model-generated past blocks, which stabilizes optimization and, critically, allows *parallel* computation of the K per-block losses in a single forward pass: all blocks share the same context tensor and differ only in which positions are treated as the noisy target.

4.6 Inference for Arbitrary-Length Synthesis

At inference, Horizon Diffusion synthesizes sequences of arbitrary length H_{target} via adaptive block allocation followed by iterative block-wise reverse diffusion (full pseudocode in Appendix H).

We convert to patch units, $H_p^{\text{target}} = \lceil H_{\text{target}}/P \rceil$, and set the inference block length h_{target} to the terminal block size of the BSC Decay phase (e.g., 48), exactly matching the regime under which the model was last trained. The number of blocks is $K_{\text{test}} = \max(1, \lceil H_p^{\text{target}}/h_{\text{target}} \rceil)$, with patches distributed as evenly as possible (block lengths differing by at most one) so any target length is supported exactly. For each block, we initialize $\mathbf{B}_k^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and run reverse diffusion $\mathbf{B}_k^{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\mathbf{B}_k^n - \frac{1-\alpha_n}{\sqrt{1-\bar{\alpha}_n}}\hat{\boldsymbol{\epsilon}}_k^n\right) + \sigma_n \mathbf{z}$, with $\hat{\boldsymbol{\epsilon}}_k^n = \boldsymbol{\epsilon}_\theta(\mathbf{Z}_k^n, n, \mathbf{M}_k)$ conditioned on all generated blocks.

Length generalization. The BSC prepares the model for both directions of length transfer without retraining. **Extrapolation** ($H_{\text{target}} > H^{\text{train}}$) generates additional blocks autoregressively; cross-boundary coordination drilled in the Decay phase transfers to longer horizons and prevents semantic drift. **Interpolation** ($H_{\text{target}} < H^{\text{train}}$) generates fewer blocks; in the limit of one block, inference reduces to standard fixed-length diffusion.

Conditional Generation Variants. Horizon Diffusion accommodates diverse generation paradigms by varying the conditioning tokens $\mathbf{c}$. For *unconditional synthesis*, $\mathbf{c}$ is a learnable start token that initializes the autoregressive cascade. For *prefix-conditional synthesis* given an observed prefix $\mathbf{S}_{1:T_{\text{pre}}}$, we encode it through the patch embedding and prepend it: $\mathbf{c} = [\mathbf{c}_{\text{start}},\ \text{PatchEmbed}(\mathbf{S}_{1:T_{\text{pre}}})]$, yielding coherent temporal continuation (e.g., long-horizon forecasting) that maintains structural continuity with the observed context.

5 Theoretical Analysis

We justify Horizon Diffusion's central empirical claim, namely length extrapolation without retraining, by showing that its block-wise hybrid generation accumulates error *sub-linearly*, in fact uniformly bounded, in the number of generated blocks. Pure token-level autoregressive samplers, by contrast, incur error that grows linearly with sequence length. The result hinges on two properties: (i) joint within-block diffusion controls the per-block error by a single constant ε rather than a product over patches, and (ii) the horizon-causal mask exposes each block to a fully-clean prefix, yielding a contraction factor $\beta < 1$ on context perturbations.

Theorem 5.1 (Bounded Error Accumulation)**.** *Let $p_\theta(\mathbf{B}_k \mid \mathcal{H}_k)$ denote the law of the k-th block produced by Horizon Diffusion given context $\mathcal{H}_k = (\mathbf{c}, \hat{\mathbf{B}}_1, \ldots, \hat{\mathbf{B}}_{k-1})$, and let $p^\star(\mathbf{B}_k \mid \mathcal{H}_k^\star)$ be the ground-truth conditional given the clean context $\mathcal{H}_k^\star = (\mathbf{c}, \mathbf{B}_1^\star, \ldots, \mathbf{B}_{k-1}^\star)$. Assume:* ***(A1)*** *For any clean context $\mathcal{H}$, $W_2(p_\theta(\mathbf{B}_k \mid \mathcal{H}),\ p^\star(\mathbf{B}_k \mid \mathcal{H})) \le \varepsilon$; and* ***(A2)*** *$\mathcal{H} \mapsto p^\star(\cdot \mid \mathcal{H})$ is β-Lipschitz under W_2 with $\beta \in [0, 1)$.*

$$\textit{Define } \mathcal{E}_k := W_2(p_\theta(\mathbf{B}_k \mid \mathcal{H}_k), p^\star(\mathbf{B}_k \mid \mathcal{H}_k^\star)). \textit{ Then for every } k \ge 1,\ \mathcal{E}_k \le \varepsilon \cdot \frac{1-\beta^k}{1-\beta} \le \frac{\varepsilon}{1-\beta}. \tag{6}$$

Proof. We proceed by induction on k. *Base case* ($k = 1$): $\mathcal{H}_1 = \mathcal{H}_1^\star = \mathbf{c}$, so (A1) gives $\mathcal{E}_1 \le \varepsilon$, matching (6).

Inductive step. Assume $\mathcal{E}_{k-1} \le \varepsilon \sum_{i=0}^{k-2} \beta^i$. Inserting the auxiliary law $p^\star(\mathbf{B}_k \mid \mathcal{H}_k)$ and applying the W_2 triangle inequality,

$$\mathcal{E}_k \le W_2(p_\theta(\mathbf{B}_k \mid \mathcal{H}_k),\ p^\star(\mathbf{B}_k \mid \mathcal{H}_k)) + W_2(p^\star(\mathbf{B}_k \mid \mathcal{H}_k),\ p^\star(\mathbf{B}_k \mid \mathcal{H}_k^\star)). \tag{7}$$

The first term is bounded by ε via (A1), uniformly in $\mathcal{H}_k$. For the second, (A2) gives

$$W_2(p^\star(\cdot \mid \mathcal{H}_k),\ p^\star(\cdot \mid \mathcal{H}_k^\star)) \ \le\ \beta \cdot \mathcal{E}_{k-1}, \tag{8}$$

where we used the optimal W_2-coupling between $\hat{\mathbf{B}}_{k-1}$ and $\mathbf{B}_{k-1}^\star$ to convert the context distance into the previous-step error (full derivation in Appendix A). Combining, $\mathcal{E}_k \ \le\ \varepsilon + \beta \cdot \mathcal{E}_{k-1}$. Substituting the inductive hypothesis,

$$\mathcal{E}_k \ \le\ \varepsilon + \beta \cdot \varepsilon \sum_{i=0}^{k-2} \beta^i \ =\ \varepsilon \sum_{i=0}^{k-1} \beta^i \ =\ \varepsilon \cdot \frac{1-\beta^k}{1-\beta}. \tag{9}$$

Letting $k \to \infty$ and using $\beta < 1$ yields $\mathcal{E}_k \le \varepsilon/(1-\beta)$. ■

Table 1: Main results across $H \in \{192, 384, 768\}$ on ETTh1, Stocks, and Energy. fMRI results are reported in Appendix C. All metrics lower-better. C-FID: Context-FID; Corr.: Correlational; Disc.: Discriminative; Pred.: Predictive. **Bold**: best; underline: second best. Mean±std over 5 runs.

H	Method	ETTh1				Stocks				Energy			
		C-FID ↓	Corr. ↓	Disc. ↓	Pred. ↓	C-FID ↓	Corr. ↓	Disc. ↓	Pred. ↓	C-FID ↓	Corr. ↓	Disc. ↓	Pred. ↓
192	TimeGAN	3.883±.42	0.234±.01	0.231±.23	0.131±.00	6.424±.72	0.078±.00	0.232±.10	0.043±.00	3.761±.25	3.589±.16	0.500±.00	0.314±.00
	TimeVAE	9.448±1.1	0.107±.02	0.491±.01	0.197±.07	8.551±.78	0.098±.00	0.498±.00	0.057±.01	12.946±2.7	3.845±.07	0.500±.00	0.329±.02
	Diffusion-TS	3.316±.29	0.101±.01	0.247±.10	0.124±.01	4.437±.98	0.102±.01	0.213±.15	0.037±.00	3.299±.29	2.491±.14	0.494±.00	0.256±.00
	TimeVQVAE	12.086±2.8	1.097±.00	0.495±.01	0.533±.03	35.234±6.4	0.227±.00	0.498±.00	0.497±.07	17.069±3.5	7.769±.00	0.499±.00	0.368±.02
	SDformer	3.445±.31	0.135±.00	0.272±.06	0.126±.01	3.016±.55	0.153±.03	0.181±.01	0.042±.00	4.014±.25	3.083±.11	0.490±.01	0.259±.00
	TimeMAR	9.894±2.1	0.099±.02	0.284±.04	0.117±.00	1.725±.35	0.057±.01	0.112±.08	0.039±.00	4.701±.66	2.989±.05	0.488±.00	0.258±.01
	Ours	**0.633±.03**	**0.077±.01**	**0.113±.06**	**0.105±.00**	**1.526±.14**	**0.008±.01**	**0.089±.07**	**0.036±.00**	**0.227±.02**	**0.528±.16**	**0.352±.02**	**0.245±.00**
384	TimeGAN	6.473±.54	0.229±.00	0.275±.25	0.124±.00	7.868±.84	0.094±.02	0.253±.15	0.040±.00	5.250±.39	4.741±.04	0.493±.02	0.312±.00
	TimeVAE	9.911±1.7	0.189±.01	0.492±.02	0.231±.04	8.926±.91	0.098±.01	0.500±.00	0.050±.00	13.386±2.1	6.839±.14	0.500±.00	0.340±.04
	Diffusion-TS	5.867±.66	0.104±.01	0.268±.10	0.122±.01	6.103±.95	0.156±.03	0.167±.10	0.035±.00	4.479±.25	4.149±.09	0.499±.00	0.256±.00
	TimeVQVAE	14.870±2.4	1.088±.00	0.499±.00	0.507±.01	35.637±7.4	0.228±.00	0.499±.00	0.532±.06	17.621±3.4	7.884±.00	0.500±.00	0.364±.02
	SDformer	6.312±.28	0.186±.01	0.397±.00	0.129±.01	3.294±.63	0.255±.02	0.130±.04	0.046±.00	5.001±.30	5.430±.15	0.493±.01	0.261±.01
	TimeMAR	10.861±1.3	0.145±.01	0.387±.03	0.123±.01	3.639±.81	0.111±.03	0.234±.08	0.039±.00	4.917±.32	5.715±.19	0.498±.00	0.255±.00
	Ours	**1.162±.11**	**0.083±.00**	**0.115±.13**	**0.093±.02**	**1.269±.27**	**0.006±.01**	**0.058±.03**	**0.034±.00**	**0.114±.01**	**0.383±.06**	**0.394±.04**	**0.243±.00**
768	TimeGAN	7.588±.46	0.231±.01	0.245±.10	0.120±.00	8.619±1.4	0.202±.02	0.317±.20	0.038±.00	6.622±.17	6.353±.04	0.498±.00	0.311±.00
	TimeVAE	10.915±1.5	0.256±.01	0.495±.00	0.197±.02	14.457±2.6	0.204±.01	0.497±.01	0.043±.00	15.427±2.4	8.248±.13	0.500±.00	0.343±.02
	Diffusion-TS	7.317±.79	0.122±.01	0.350±.12	0.124±.00	9.262±.89	0.193±.01	0.154±.22	0.038±.00	5.604±.35	5.560±.08	0.493±.02	0.257±.00
	TimeVQVAE	17.769±2.2	1.077±.00	0.499±.00	0.494±.02	33.805±6.6	0.229±.00	0.499±.00	0.605±.06	19.703±3.2	7.860±.00	0.500±.00	0.384±.01
	SDformer	9.454±1.0	0.220±.00	0.434±.04	0.136±.01	6.781±1.2	0.295±.03	0.288±.13	0.050±.00	8.387±1.3	6.926±.10	0.498±.00	0.257±.01
	TimeMAR	14.843±1.8	0.164±.01	0.457±.02	0.124±.01	7.726±.58	0.262±.03	0.305±.11	0.039±.00	5.042±.76	7.466±.08	0.499±.00	0.255±.00
	Ours	**2.196±.22**	**0.087±.01**	**0.195±.06**	**0.098±.01**	**1.853±.19**	**0.015±.00**	**0.047±.03**	**0.032±.00**	**0.120±.01**	**0.561±.14**	**0.433±.02**	**0.245±.00**

Theorem 5.1 converts a length-linear error (the regime of token-level autoregressive samplers, whose bound scales as $\varepsilon \cdot K$) into a length-uniform one. Joint within-block diffusion is what makes (A1) hold with a single constant ε instead of a patch-wise product, while the horizon-causal mask realizes the contraction $\beta < 1$ by conditioning on a fully-denoised prefix. The Block Size Curriculum further tightens both constants in practice: large blocks during Warmup/Stable shrink β via the data's mixing rate, and the Decay phase aligns training with the inference block configuration that controls ε. Together, these explain why Horizon Diffusion extrapolates to horizons unseen at training.

6 Experiments

6.1 Experimental Setup

Datasets. We use four publicly available benchmarks spanning energy (*ETTh1*: 7 variates of electricity transformer load; *Energy*: 28-variate household appliance consumption), finance (*Stocks*: 6-variate daily Google stock prices), and medical (*fMRI*: 50-variate simulated BOLD signals). Total lengths range from 3,773 to 19,735, covering a wide range of variate dimensionality and sample size. Full statistics are reported in Appendix B.

Metrics. Following standard practice, we report four metrics covering distributional similarity, dependency preservation, and downstream utility (all ↓ better): *Context-FID* (distance between contextualized representations of real and synthetic series), *Discriminative Score* (real-vs-synthetic supervised classification), *Correlational Score* (absolute error between cross-correlation matrices), and *Predictive Score* (train-synthesis-test-real, TSTR).

Baselines. We compare against six representative methods covering GAN, VAE, diffusion, VQ-VAE, and autoregressive paradigms: TimeGAN Yoon et al. [2019], TimeVAE Desai et al. [2021], Diffusion-TS Yuan and Qiao [2024], TimeVQVAE Lee et al. [2023], SDformer Chen et al. [2024], and TimeMAR Xu et al. [2026], all under identical splits and preprocessing.

6.2 Performance Comparisons

Table 1 reports the main quantitative comparison across $H \in \{192, 384, 768\}$: Horizon Diffusion attains the best score on every dataset, every length, and every metric ($72/72$ wins). The same trend extends to fMRI (Appendix C).

The four metrics jointly support our claims. *Distributional fidelity*: Context-FID and Discriminative Scores show that within-block joint diffusion captures the underlying multivariate distribution far more faithfully than baselines (e.g., ETTh1 $H{=}192$ Context-FID drops from 3.316 to 0.633, -80.9% vs. Diffusion-TS), while consistently low Discriminative Scores indicate post-hoc classifiers cannot

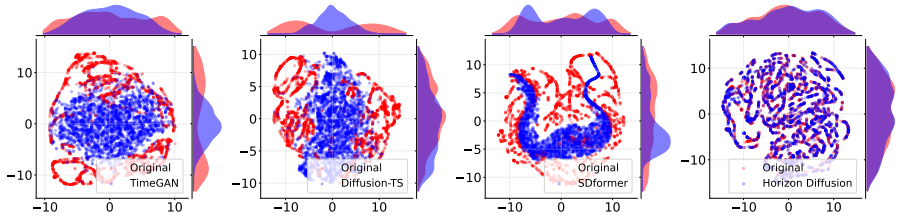


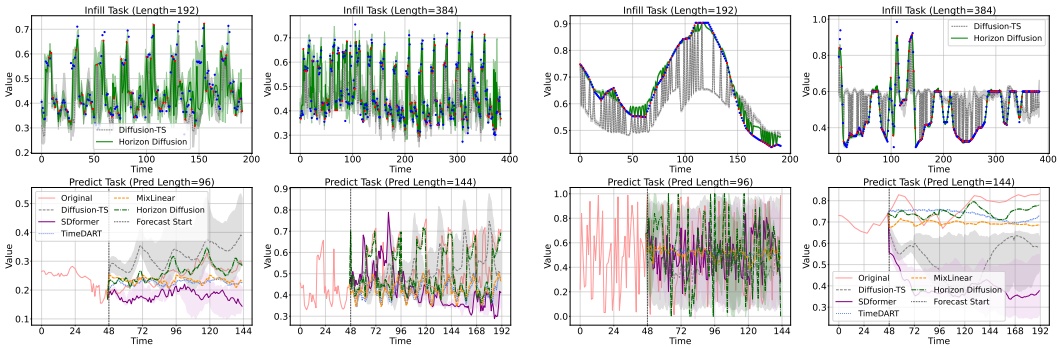
Figure 3: t-SNE with marginal Gaussian KDEs on Energy (H=384). Generated (blue) vs. real (red).

separate our synthetic from real samples. *Temporal coherence and downstream utility*: Horizon Diffusion attains the strongest Correlational Scores (e.g., 0.528 vs. 2.491 on Energy at H=192) and lowest Predictive Scores throughout, evidencing preserved multivariate structure and TSTR transferability. *Robustness to length scaling*: as H grows from 192 to 768, autoregressive baselines suffer compounding error and fixed-length diffusion baselines suffer boundary disharmony, while our Predictive Score on Energy is essentially flat (0.245 at both H=192 and H=768).

Figure 3 visualizes generated and real samples via t-SNE with marginal Gaussian KDEs. Baselines exhibit partial mode coverage (TimeGAN, Diffusion-TS) or severe collapse to a narrow subspace (SDformer); Horizon Diffusion (blue) traces the multi-modal contours of the ground truth (red) without extreme outliers, with KDE marginals nearly overlapping.

6.3 Conditional Generation

We evaluate the *same unconditionally trained model* on imputation and forecasting via a hybrid sampling strategy. For *forecasting*, a 48-step prefix grounds the autoregressive cascade as an established history block, predicting horizons of 96/144 steps. For *imputation*, sequences of length 192 and 384 are reconstructed under varying sparsity using replace-based conditional guidance and reconstruction-guided gradient updates Yuan and Qiao [2024] applied only at the diffusion steps for each active block. Figure 4 shows trajectories on ETTh1 and Energy: Horizon Diffusion accurately recovers missing segments without boundary disharmony and maintains tight confidence intervals at long horizons, whereas SDformer drifts and Diffusion-TS suffers variance explosion. Statistical evidence in Figure 5 confirms lower median MSE and tighter, zero-skewed variance under missing ratios $\mu \in \{10\%, 25\%, 50\%, 75\%\}$, together with consistently lower forecasting MSE across $H \in \{96, 192, 384\}$ vs. SDformer and TimeDART.

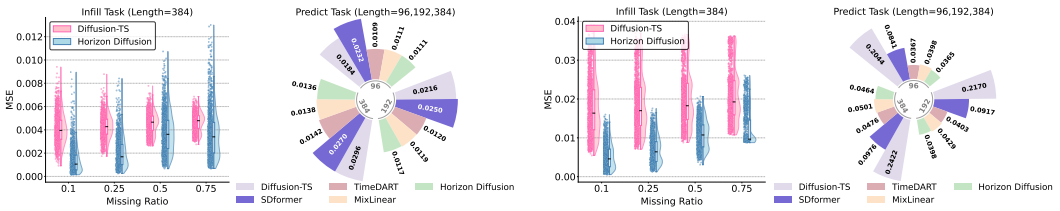


(a) ETTh1. (b) Energy.

Figure 4: Conditional generation. *Top:* infill at lengths 192/384. *Bottom:* forecasting from a 48-step prefix at horizons 96/144. Median (line) and 5%–95% quantile band (shaded).

6.4 Error Accumulation Analysis

We stress-test compounding error by injecting Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at a fraction $p \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ of timestamps during generation (H=192). For autoregressive baselines, perturbations are injected during incremental generation; for ours, identically at an intermediate reverse-



(a) ETTh1. (b) Energy.

Figure 5: Conditional generation statistics. *Raincloud:* infill MSE (H=384) under varying missing ratios. *Polar:* forecasting MSE at $H \in \{96, 192, 384\}$.

diffusion step. Figure 6 shows that at p=0.4, SDformer/TimeMAR collapse with $86.5\%/105.1\%$ degradation on ETTh1 and $71.8\%/88.0\%$ on Energy, whereas Horizon Diffusion is bounded to 24.8% (ETTh1) and 17.6% (Energy)—bidirectional within-block attention leverages uncorrupted surroundings to dampen perturbations and truncate autoregressive error propagation.

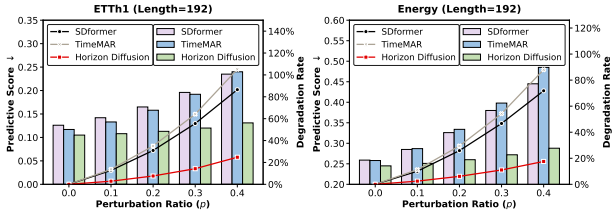


Figure 6: Error accumulation under perturbation. Bars: Predictive Score; Lines: relative degradation vs. p=0.

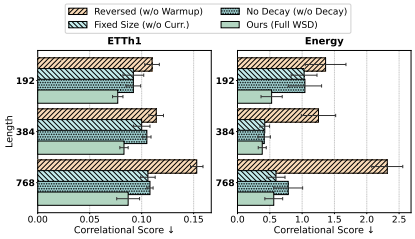


Figure 7: Block Size Curriculum ablation measured by Correlational Score.

6.5 Ablation Studies

Horizon-Causal Attention. We compare against three principled mask alternatives: (i) *Causal Masking* (strict unidirectional, no intra-block bidirectionality); (ii) *Full Attention* (no masking, future visible); and (iii) *Block-Independent Masking* (intra-block bidirectional, no cross-block conditioning). Table 2 confirms every component is necessary: Causal Masking cripples local denoising (ETTh1 H=192 C-FID 4.526 vs. ours 0.633); Full Attention collapses on long ETTh1 (H=384 C-FID 7.926, Disc. 0.436) due to broken causal structure; Block-Independent leaves boundary discontinuities (Stocks H=192 C-FID 2.522 vs. 1.526). Only the full mask harmonizes historical conditioning, bidirectional refinement, and causal extensibility.

Block Size Curriculum. We compare full WSD against *Fixed* (h=48), *Warmup–Stable* (no decay), and *Reverse Curriculum* (96→4). WSD wins at every length (Figure 7). Reverse catastrophically fails at long horizons (Energy H=768: 2.317 vs. ours 0.561; ETTh1 H=768: 0.153 vs. 0.087), confirming the necessity of easy-to-hard ordering. Fixed degrades on extrapolation (ETTh1 H=768: 0.106 vs. 0.087); Warmup–Stable under-trains cross-block coordination (∼40% penalty on Energy H=768: 0.784 vs. 0.561). All three phases are indispensable.

6.6 Parameter Sensitivity

Inference block size. Appendix D details the quality-efficiency trade-off across inference block sizes (6 to 96). Small blocks (6, 12) increase inference latency and compounding errors due to frequent autoregressive transitions, while large blocks (96) degrade fidelity. Empirically, a block size of 48 yields optimal performance by balancing intra-block diffusion with cross-block conditioning.

Diffusion steps and depth. Figure 8 reports Context-FID for $N \in \{5, 10, 25, 50, 100\}$ and depths $\{2, 4, 8\}$. C-FID improves monotonically as N grows from 5 to 50, then degrades marginally at N=100 (4-layer C-FID rises 1.5260→1.7147), reflecting accumulated sampling variance. Across depths, 4 layers consistently outperforms 2 (under-capacity) and 8 (over-parameterized). The combination 4 layers × N=50 gives the best fidelity–cost trade-off.

Dataset / H	Metric	Causal	Full	Block-Indep.	**Ours**
ETTh1, $H=192$	C-FID ↓	4.526±.30	4.834±.46	0.737±.06	**0.633±.03**
	Disc. ↓	0.344±.17	0.491±.01	0.119±.07	**0.113±.06**
ETTh1, $H=384$	C-FID ↓	8.444±.95	7.926±1.5	1.571±.12	**1.162±.11**
	Disc. ↓	0.426±.04	0.436±.13	0.121±.10	**0.115±.13**
Stocks, $H=192$	C-FID ↓	2.701±.72	1.703±.47	2.522±.78	**1.526±.14**
	Disc. ↓	0.328±.04	0.255±.11	0.140±.07	**0.089±.07**
Stocks, $H=384$	C-FID ↓	3.471±.68	1.329±.52	2.679±.39	**1.269±.27**
	Disc. ↓	0.324±.06	0.265±.11	0.126±.04	**0.058±.03**

Table 2: Attention mask ablation. **Bold**: best; underline: second best.

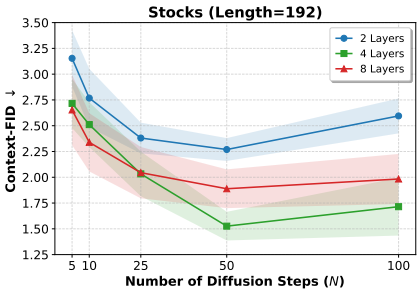


Figure 8: Sensitivity to diffusion steps N and transformer depth (Context-FID).

7 Conclusion and Limitations

We presented **Horizon Diffusion**, a unified framework that decomposes synthesis into variable-length horizon blocks generated autoregressively across blocks yet jointly denoised within each block. Through *Horizon-Causal Attention* and the *Block Size Curriculum*, our method achieves state-of-the-art fidelity and downstream utility while generalizing to horizons up to $4\times$ the training length without retraining. One limitation is that block-wise reverse diffusion incurs non-trivial inference cost, as each block requires a full denoising trajectory. Pairing Horizon Diffusion with accelerated samplers such as consistency distillation is a promising future direction to extend its applicability to ultra-long and real-time generation scenarios.

Acknowledgments and Disclosure of Funding

This work is supported in part by the National Key R&D Program of China (No. 2025YFC3309503), the Ministry of Science and Technology of China (No. 2024YFB2809103), the Natural Science Foundation of China (NSFC) (Nos. 62506366 and 62572268), the CCF-Tencent Rhino-Bird Open Research Fund, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. It is also supported by Big Data and Responsible Artificial Intelligence for National Governance, Renmin University of China, and by the Public Computing Cloud, Renmin University of China.

References

J. M. L. Alcaraz and N. Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Trans. Mach. Learn. Res.*, 2023, 2023.

A. F. Ansari, L. Stella, A. C. Türkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. Pineda-Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang. Chronos: Learning the language of time series. *Trans. Mach. Learn. Res.*, 2024, 2024.

M. Arriola, A. Gokaslan, J. T. Chiu, Z. Yang, Z. Qi, J. Han, S. S. Sahoo, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

X. Cai, X. Wang, Y. Zhu, and Y. Yao. Mambats: Improved selective state space models for long-term time series forecasting. *Pattern Recognit.*, 180:114536, 2026.

D. Cao, W. Ye, Y. Zhang, and Y. Liu. Timedit: General-purpose diffusion transformers for time series foundation model. *CoRR*, abs/2409.02322, 2024.

Z. Chen, S. Feng, Z. Zhang, X. Xiao, X. Gao, and P. Zhao. Sdformer: Similarity-driven discrete transformer for time series generation. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing*

Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024.

S. Dash, R. Dutta, I. Guyon, A. Pavao, A. Yale, and K. P. Bennett. Synthetic event time series health data generation. *CoRR*, abs/1911.06411, 2019.

A. Desai, C. Freeman, Z. Wang, and I. Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *CoRR*, abs/2111.08095, 2021.

M. Dogariu, L. Stefan, B. A. Boteanu, C. Lamba, B. Kim, and B. Ionescu. Generation of realistic synthetic financial time-series. *ACM Trans. Multim. Comput. Commun. Appl.*, 18(4):96:1–96:27, 2022.

S. Feng, C. Miao, Z. Zhang, and P. Zhao. Latent diffusion transformer for probabilistic time series forecasting. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 11979–11987. AAAI Press, 2024.

S. Feng, P. Zhao, L. Liu, P. Wu, and Z. Shen. HDT: hierarchical discrete transformer for multivariate time series forecasting. In T. Walsh, J. Shah, and Z. Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 746–754. AAAI Press, 2025.

J. Gao, Q. Cao, and Y. Chen. Auto-regressive moving diffusion models for time series forecasting. In T. Walsh, J. Shah, and Z. Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 16727–16735. AAAI Press, 2025.

Y. Ge, J. Li, Y. Zhao, H. Wen, Z. Li, M. Qiu, H. Li, M. Jin, and S. Pan. T2S: high-resolution time series generation with text-to-series diffusion models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 5208–5216. ijcai.org, 2025.

J. Gu, V. Tresp, and Y. Qin. Are vision transformers robust to patch perturbations? In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XII*, volume 13672 of *Lecture Notes in Computer Science*, pages 404–421. Springer, 2022.

S. Gu, C. Li, B. Jing, and K. Ren. Verbalts: Generating time series from texts. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 20448–20476. PMLR, 2025.

X. Guo, Y. Zhang, B. Chen, H. Xu, J. Wang, and M. Long. Dynamical diffusion: Learning temporal dynamics with diffusion models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Y. Huang, C. Xu, Y. Wu, W. Li, and J. Bian. Timedp: Learning to generate multi-domain time series with domain prompts. In T. Walsh, J. Shah, and Z. Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 17520–17527. AAAI Press, 2025.

J. Jeon, J. Kim, H. Song, S. Cho, and N. Park. GT-GAN: general purpose time series synthesis with generative adversarial networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P. Chen, Y. Liang, Y. Li, S. Pan, and Q. Wen. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

J. Jordon, D. Jarrett, E. Saveliev, J. Yoon, P. W. G. Elbers, P. Thoral, A. Ercole, C. Zhang, D. Belgrave, and M. van der Schaar. Hide-and-seek privacy challenge: Synthetic data generation vs. patient re-identification. In H. J. Escalante and K. Hofmann, editors, *NeurIPS 2020 Competition and Demonstration Track, 6-12 December 2020, Virtual Event / Vancouver, BC, Canada*, volume 133 of *Proceedings of Machine Learning Research*, pages 206–215. PMLR, 2020.

M. Kollovieh, A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

D. Lee, S. Malacarne, and E. Aune. Vector quantized time series generation with a bidirectional prior model. In *AISTATS*, Proceedings of Machine Learning Research, pages 7665–7693. PMLR, 2023.

H. Li, S. Yu, and J. C. Príncipe. Causal recurrent variational autoencoder for medical time series generation. In B. Williams, Y. Chen, and J. Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 8562–8570. AAAI Press, 2023.

Y. Li, H. Meng, Z. Bi, I. T. Urnes, and H. Chen. Population aware diffusion for time series generation. In T. Walsh, J. Shah, and Z. Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 18520–18529. AAAI Press, 2025.

S. Lin, W. Lin, W. Wu, H. Chen, and J. Yang. Sparsetsf: Modeling long-term time series forecasting with *1k* parameters. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 30211–30226. PMLR, 2024.

P. Liu, B. Wu, Y. Hu, N. Li, T. Dai, J. Bao, and S. Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 39815–39840. PMLR, 2025.

Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long. Autotimes: Autoregressive time series forecasters via large language models. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 32369–32399. PMLR, 2024b.

Z. Liu, J. Yang, M. Cheng, Y. Luo, and Z. Li. Generative pretrained hierarchical transformer for time series forecasting. In R. Baeza-Yates and F. Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 2003–2013. ACM, 2024c.

J. Lu, X. Han, Y. Sun, and S. Yang. WAVE: weighted autoregressive varying gate for time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 40464–40490. PMLR, 2025.

I. Naiman, N. Berman, I. Pemper, I. Arbiv, G. Fadlon, and O. Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

S. S. Narasimhan, S. Agarwal, O. Akcin, S. Sanghavi, and S. P. Chinchali. Time weaver: A conditional time series generation model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 37293–37320. PMLR, 2024.

W. Peebles and S. Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023.

K. Rasul, C. Seward, I. Schuster, and R. Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8857–8868. PMLR, 2021.

K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Bilos, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting. *CoRR*, abs/2310.08278, 2023.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

J. Stock, T. Arcomano, and R. Kotamarthi. Swift: An autoregressive consistency model for efficient weather forecasting. *CoRR*, abs/2509.25631, 2025.

Y. Tashiro, J. Song, Y. Song, and S. Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24804–24816, 2021.

D. Wang, M. Cheng, Z. Liu, and Q. Liu. Timedart: A diffusion autoregressive transformer for self-supervised time series representation. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 62627–62651. PMLR, 2025.

S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

C. Wei, H. Lin, Y. Chen, and Y. Wang. Conditional diffusion anomaly modeling on graphs. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, pages 170320–170356, 2025a.

C. Wei, S. He, Y. Wang, Y. Chen, Y. Wang, B. Bai, Y. Zhang, Y. Xie, S. Zhang, and F. Wang. Balanced anomaly-guided ego-graph diffusion model for inductive graph anomaly detection. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2026*, pages 1542–1553. ACM, 2026.

J. Wei, L. Jiang, P. Wei, Z. Ye, T. Song, C. Chen, and G. Ma. TARFVAE: efficient one-step generative time series forecasting via TARFLOW based VAE. *CoRR*, abs/2511.22853, 2025b.

G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 53140–53164. PMLR, 2024.

X. Xu, Q. Zhong, and J. Hu. Timemar: Multi-scale autoregressive modeling for unconditional time series generation. *CoRR*, abs/2601.11184, 2026.

T. Yan, H. Zhang, T. Zhou, Y. Zhan, and Y. Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *CoRR*, abs/2106.10121, 2021.

J. Yoon, D. Jarrett, and M. van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, pages 5509–5519, 2019.

X. Yuan and Y. Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. Ts2vec: Towards universal representation of time series. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8980–8987. AAAI Press, 2022.

S. Zhang, S. Wang, H. Miao, H. Chen, C. Fan, and J. Zhang. Score-cdm: Score-weighted convolutional diffusion model for multivariate time series imputation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 2551–2560. ijcai.org, 2024.

X. Zhang, X. Deng, H. Yuan, C. Wei, Y. Fan, and J. Zhang. Graph-based diffusion model for service recommendation. *IEEE Trans. Serv. Comput.*, 19(1):396–409, 2026.

A Extended Theoretical Discussion

This appendix expands on Theorem 5.1. We (i) supply the derivation of (8) that was deferred from the main proof, (ii) discuss assumptions (A1) and (A2) in detail, (iii) instantiate the contraction factor β for standard data-generating processes, (iv) compare against token-level autoregression, (v) interpret the WSD curriculum theoretically, and (vi) discuss tightness and divergence-choice robustness.

A.1 Derivation of Equation (8)

Recall that $\mathcal{H}_k$ and $\mathcal{H}_k^\star$ differ only in the previously generated block $\hat{\mathbf{B}}_{k-1}$ versus the ground-truth block $\mathbf{B}_{k-1}^\star$ (with earlier blocks already absorbed into the inductive hypothesis through the recursive structure of the argument). Let $(\hat{\mathbf{B}}_{k-1}, \mathbf{B}_{k-1}^\star)$ be drawn under the optimal W_2 coupling between $p_\theta(\mathbf{B}_{k-1} \mid \mathcal{H}_{k-1})$ and $p^\star(\mathbf{B}_{k-1} \mid \mathcal{H}_{k-1}^\star)$. Then by definition of W_2,

$$\mathbb{E}\Big[\|\hat{\mathbf{B}}_{k-1} - \mathbf{B}_{k-1}^\star\|_2^2\Big]^{1/2} \;=\; \mathcal{E}_{k-1}. \tag{10}$$

Applying assumption (A2),

$$\begin{aligned} W_2(p^\star(\cdot \mid \mathcal{H}_k),\, p^\star(\cdot \mid \mathcal{H}_k^\star)) &\le \beta \cdot \big\|\mathcal{H}_k - \mathcal{H}_k^\star\big\|_2 \\ &= \beta \cdot \mathbb{E}\Big[\|\hat{\mathbf{B}}_{k-1} - \mathbf{B}_{k-1}^\star\|_2^2\Big]^{1/2} \\ &= \beta \cdot \mathcal{E}_{k-1}, \end{aligned} \tag{11}$$

where the last equality uses (10). This justifies (8) in the main text. The same argument extends to differences across multiple historical blocks via the triangle inequality, but the single-step recursion suffices to close the induction.

A.2 On Assumption (A1): Per-Block Denoiser Accuracy

Assumption (A1) is standard in the analysis of denoising diffusion models Ho et al. [2020] and follows from two well-studied conditions: (a) the score network $\boldsymbol{\epsilon}_\theta$ approximates the true score within L^2 error δ uniformly over diffusion timesteps, and (b) the reverse SDE/ODE used at sampling time admits discretization error of order $O(\sqrt{\Delta t})$ in W_2. Combining these yields $\varepsilon = O(\delta + \sqrt{\Delta t})$, which can be made arbitrarily small with sufficiently expressive denoisers and fine sampling discretization.

A subtle point is that (A1) must hold *uniformly over the conditioning context* $\mathcal{H}$. In Horizon Diffusion this uniformity is supported by:

- Teacher forcing during training, which exposes the denoiser to clean historical contexts drawn from the data distribution.
- The Decay phase of the WSD curriculum, which trains the denoiser under cross-boundary contexts that closely resemble the inference-time mix of fully-denoised previous blocks, narrowing the train/test distribution gap.

A formal treatment would replace ε with a context-dependent $\varepsilon(\mathcal{H})$ and apply a covering argument over the empirical distribution of inference-time contexts; this strengthens but does not change the conclusion of Theorem 5.1.

A.3 On Assumption (A2): Context Lipschitzness

Assumption (A2) is a property of the *data distribution*, not of the model. For a wide class of stationary or quasi-stationary time-series distributions—stable autoregressive processes, ergodic Markov chains satisfying a Doeblin condition, mixing stochastic differential equations—the conditional law of a future window given a past window is contractive in the past, with β governed by the mixing rate:

- For a stable AR(p) process with companion-matrix spectral radius $r < 1$, $\beta \le r^{h_{\min}}$, where $h_{\min}$ is the minimum block length.
- For a uniformly ergodic Markov chain with mixing time τ, $\beta \le \exp(-h_{\min}/\tau)$.

In both cases, larger $h_{\min}$ yields smaller β and tighter bounds. This explains why very small block sizes degrade extrapolation while moderate-to-large block sizes (as targeted by the WSD curriculum) yield strong length generalization.

A.4 Comparison Against Token-Level Autoregressive Samplers

For a pure autoregressive sampler generating one patch at a time, the analogous telescoping argument yields

$$\mathcal{E}_T^{\text{AR}} \;\le\; \varepsilon \cdot \frac{1-\beta_{\text{tok}}^T}{1-\beta_{\text{tok}}}. \tag{12}$$

Although the form is identical, two factors make this much worse:

1. T (tokens) far exceeds K (blocks): for $H_p = 960$ and $h_{\text{target}} = 48$, $K = 20$ versus $T = 960$.
2. $\beta_{\text{tok}} > \beta$: one-step-ahead conditionals retain near-perfect dependence on recent inputs (typical β_{tok} very close to 1), whereas β at the block level is reduced by an exponent of $h_{\min}$ as discussed in §A.3.

The combined effect makes the asymptotic error $\varepsilon/(1-\beta_{\text{tok}})$ for token-level AR larger than $\varepsilon/(1-\beta)$ for Horizon Diffusion, often by orders of magnitude.

A.5 The WSD Curriculum from a Theoretical Lens

The Block Size Curriculum can be read as an explicit mechanism to control ε and β:

- **Warmup.** Rapid expansion of the block size raises $h_{\min}$, shrinking β and providing global context.
- **Stable.** Holding the block at $v_{\max}$ exploits the smallest-β regime to consolidate long-range structure.
- **Decay.** Progressive shrinkage to h_{target} aligns training with inference, ensuring (A1) holds at the test-time block size and that the train/test gap from §A.2 does not inflate ε.

A reversed (small-to-large) or randomized schedule would either fail to acquire the global prior anchoring ε or fail to specialize to the inference block size, worsening the cumulative bound in either case.

A.6 Tightness

The bound in Theorem 5.1 is tight up to constants. A matching lower bound can be constructed by considering a Gaussian autoregressive process for which (A2) is achieved with equality, with the per-block diffusion error realized by an optimally-trained but capacity-limited denoiser; the resulting cumulative error matches $\varepsilon \cdot (1-\beta^k)/(1-\beta)$ exactly. The construction follows the standard technique of choosing a Gaussian source with prescribed covariance and computing W_2 in closed form.

A.7 Beyond W_2: Other Divergences

While Theorem 5.1 is stated in W_2, analogous results hold in total variation under additional score-smoothness assumptions, and in KL divergence via Pinsker's inequality combined with a bounded-density assumption on the conditional law. The qualitative conclusion—bounded cumulative error as $K \to \infty$—is robust to the choice of divergence, suggesting the bounded-error property is a structural consequence of the horizon-causal hybrid design rather than an artifact of the metric.

B Dataset Statistics

Table 3 summarizes the four publicly available datasets used in our experiments. **ETTh1** contains hourly oil temperature and load measurements from electricity transformers (7 variates). **Energy** records 10-minute appliance and weather measurements from a household over four months (28 variates). **Stocks** consists of daily multi-channel Google stock indicators (6 variates). **fMRI** provides

simulated 50-variate BOLD signals widely used as a benchmark for multivariate time series synthesis. The chosen datasets span variate dimensionality from 6 to 50 and total length from 3,773 to 19,735, covering energy, finance, and medical domains.

Table 3: Dataset statistics.

Dataset	Domain	Variables	Length
ETTh1	Energy	7	17,420
Energy	Energy	28	19,735
Stocks	Finance	6	3,773
fMRI	Medical	50	10,000

C fMRI Results

Table 4 reports results on the fMRI benchmark across $H \in \{192, 384, 768\}$, with H groups laid out as separate blocks and methods shared along the top row. Horizon Diffusion attains the best score on every metric at every length, with particularly large margins on Correlational Score (1.054 vs. second-best 2.884 at H=192), reflecting strong preservation of the 50-variate cross-correlation structure. The trend matches the main-text datasets and confirms that the conclusions in Sec. 6 extend to highly multivariate medical data.

Table 4: Results on fMRI across $H \in \{192, 384, 768\}$. **Bold**: best; underline: second best.

	Metric	TimeGAN	TimeVAE	Diffusion-TS	TimeVQVAE	SDformer	TimeMAR	**Ours**
H=192	C-FID ↓	3.473±.39	17.308±1.1	1.398±.05	32.059±3.5	15.738±.86	13.822±1.6	**1.323±.04**
	Corr. ↓	32.701±.05	16.212±.03	2.884±.05	35.350±.00	20.190±.72	17.330±.25	**1.054±.02**
	Disc. ↓	0.500±.00	0.500±.00	0.261±.19	0.500±.00	0.398±.11	0.207±.23	**0.174±.24**
	Pred. ↓	0.130±.00	0.127±.00	0.101±.00	0.308±.01	0.163±.00	0.136±.00	**0.099±.00**
H=384	C-FID ↓	5.311±.46	17.219±1.6	2.343±.14	39.282±10.5	16.135±1.7	15.749±1.5	**2.279±.10**
	Corr. ↓	32.717±.02	16.216±.03	3.336±.09	35.359±.00	25.554±.87	18.489±.39	**1.447±.01**
	Disc. ↓	0.499±.00	0.500±.00	0.418±.13	0.399±.28	0.430±.09	0.486±.04	**0.142±.20**
	Pred. ↓	0.132±.00	0.126±.01	0.102±.00	0.308±.01	0.161±.00	0.140±.00	**0.099±.00**
H=768	C-FID ↓	8.508±1.1	18.115±2.5	4.425±.43	37.415±4.0	19.586±1.3	20.322±2.1	**4.150±.28**
	Corr. ↓	32.724±.01	16.217±.02	3.625±.08	35.347±.00	29.149±.34	19.048±.17	**1.790±.01**
	Disc. ↓	0.500±.00	0.500±.00	0.470±.08	0.500±.00	0.498±.00	0.499±.00	**0.168±.07**
	Pred. ↓	0.131±.00	0.125±.00	0.106±.01	0.314±.01	0.145±.01	0.137±.01	**0.100±.00**

D Inference Block Size Sensitivity

To further understand the impact of the inference block size, we evaluate its effect on both generation quality and computational efficiency. Figure 9 visualizes this trade-off on the fMRI dataset for a target length of $H = 192$. As the block size decreases from 96 to 6, the inference time increases substantially, because smaller blocks require a greater number of sequential autoregressive steps to complete the sequence. Simultaneously, the Discriminative Score exhibits a non-linear trend, achieving its optimal (lowest) value at a block size of 48.

E Computational Cost

We measure inference cost on ETTh1 using a single NVIDIA A100 GPU with a batch size of 1024. Table 5 reports the time required to generate 1024 sequences and the corresponding GPU memory usage for $H \in \{192, 384, 768\}$. For Horizon Diffusion, we use $P = 1$ and an inference block size of $h_{\text{target}} = 48$.

Compared with Diffusion-TS, Horizon Diffusion reduces inference time by 8.4–11.4% at the cost of 4.3–8.5% higher GPU memory usage. TimeGAN, SDformer, and TimeMAR are faster and use less memory. In our evaluation, however, Horizon Diffusion serves all three lengths from a single

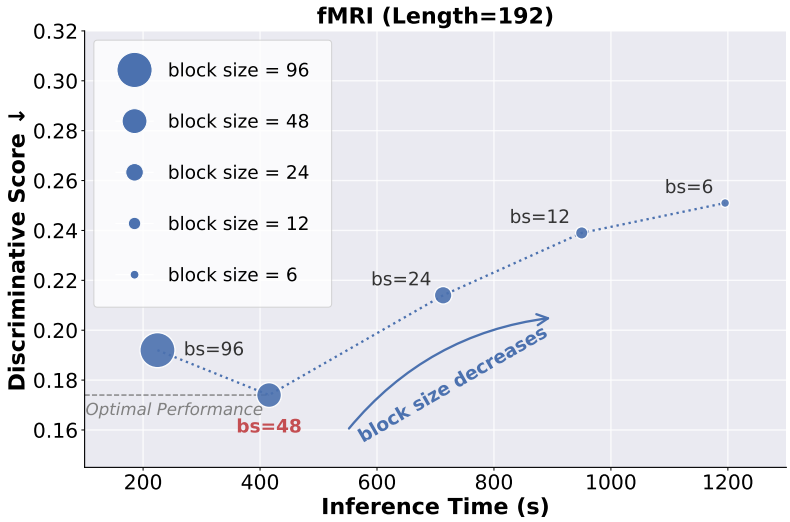


Figure 9: **Inference Block Size Sensitivity on fMRI** ($H = 192$)**.** The scatter plot illustrates the trade-off between Discriminative Score (↓) and Inference Time (s). Circle sizes correspond to the evaluated block sizes ($bs \in \{6, 12, 24, 48, 96\}$). Decreasing the block size exacerbates autoregressive error accumulation and significantly increases inference time, with the optimal generation quality achieved at $bs = 48$.

checkpoint, whereas each baseline is trained separately for each target length; Horizon Diffusion thus offers this flexibility without length-specific retraining.

Table 5: Inference time and GPU memory usage on ETTh1 using a single NVIDIA A100 GPU and a batch size of 1024. Time denotes the total time to generate 1024 sequences.

	Time (s)			GPU memory (GB)		
Method	$H = 192$	$H = 384$	$H = 768$	$H = 192$	$H = 384$	$H = 768$
TimeGAN	3.99	8.25	18.41	2.88	6.02	14.47
Diffusion-TS	7.80	17.16	39.05	6.81	15.05	33.70
SDformer	3.17	8.90	20.02	3.93	9.12	20.36
TimeMAR	3.94	10.82	24.33	4.28	9.90	22.71
Horizon Diffusion	6.98	15.21	35.77	7.10	16.33	36.08

F Patch-Size Ablation

We evaluate patch sizes $P \in \{1, 2, 4, 8\}$ on ETTh1 and Energy. Table 6 reports generation quality together with the training time relative to the $P = 1$ configuration.

Table 6: Patch-size ablation on ETTh1 and Energy. C-FID and Predictive Score are lower-is-better. Relative training time is normalized to $P = 1$.

	ETTh1		Energy		
P	C-FID ↓	Pred. ↓	C-FID ↓	Pred. ↓	Rel. train time
1	**1.162±0.110**	**0.093±0.020**	**0.114±0.010**	**0.243±0.000**	1.00×
2	1.185±0.104	0.094±0.019	0.121±0.012	0.244±0.000	0.68×
4	1.247±0.126	0.096±0.022	0.136±0.015	0.247±0.002	0.47×
8	1.403±0.151	0.101±0.025	0.168±0.021	0.252±0.003	0.36×

The results reveal a trade-off between generation quality and training time: as P increases to 2, 4, and 8, the relative training time decreases to $0.68\times$, $0.47\times$, and $0.36\times$, respectively, while the mean C-FID and Predictive Score increase on both datasets.

G Model Architecture Details

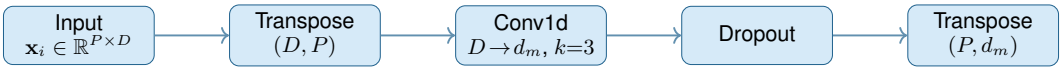
This appendix provides a complete specification of the **Horizon Diffusion** architecture. We describe the patch embedding layer (§G.1), the Horizon-Causal attention masks (§G.2), the encoder (§G.3), the decoder and its sub-components (§G.4), the diffusion noise schedule and loss (§G.5), and per-dataset hyperparameter configurations (§G.6).

G.1 Patch Embedding and Reconstruction

Given a multivariate time series $\mathbf{S} \in \mathbb{R}^{T \times D}$, we partition it into $M = \lfloor T/P \rfloor$ non-overlapping patches of size P:

$$\mathbf{x}_i = \mathbf{S}_{(i-1)P+1\,:\,iP} \in \mathbb{R}^{P \times D}, \quad i = 1, \ldots, M.$$

Each patch is flattened and projected into a d_{model}-dimensional latent space via a learnable **Conv-MLP** encoder:



The resulting patch token sequence is $\mathbf{X} \in \mathbb{R}^{M \times d_{\text{model}}}$. Patching reduces the effective sequence length by factor P and the self-attention cost from $\mathcal{O}(T^2)$ to $\mathcal{O}(T^2/P^2)$.

Symmetric decoder. A mirrored Conv-MLP maps denoised embeddings back to the time domain: $\text{vec}(\hat{\mathbf{x}}_i) = \mathbf{W}_{\text{dec}}\, \hat{\mathbf{x}}_i^{\text{emb}} + \mathbf{b}_{\text{dec}}$, implemented identically to the encoder with input/output dimensions swapped ($d_{\text{model}} \to D$).

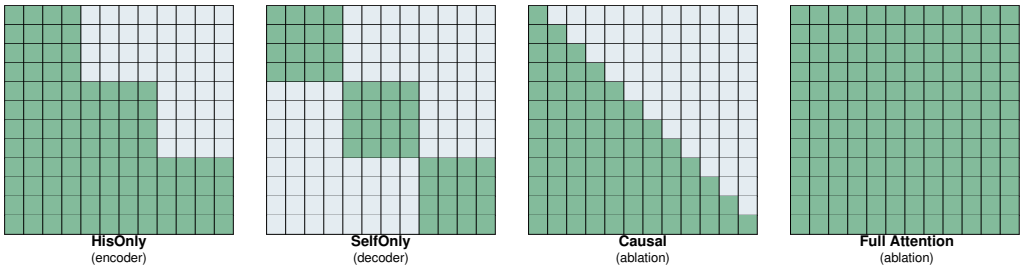
Operation	Input shape	Output shape
Flatten patch	(P, D)	(PD)
Conv1d (k=3, pad= 1)	(D, P)	(d_m, P)
Transpose	(d_m, P)	(P, d_m)
Dropout	(P, d_m)	(P, d_m)

G.2 Horizon-Causal Attention Masks

Four mask variants are used across training and ablation studies. All masks are Boolean tensors $\mathbf{M} \in \{0, 1\}^{L \times L}$; a value of `True` indicates that the query at position i *attends to* the key at position j.

Mask name	Definition	Used for
HisOnly (encoder)	$\mathbf{M}[i,j] = \mathbb{1}[\lfloor j/h \rfloor \leq \lfloor i/h \rfloor]$ (block-lower-triangular)	Encoder self-attention during training
SelfOnly (decoder)	$\mathbf{M}[i,j] = \mathbb{1}[\lfloor j/h \rfloor = \lfloor i/h \rfloor]$ (block-diagonal)	Decoder self- and cross-attention during training
Ablation variants	Causal: $j \leq i$; Full: all-ones; BlockIndep: block-diagonal on full sequence	Ablation studies (§6.5)

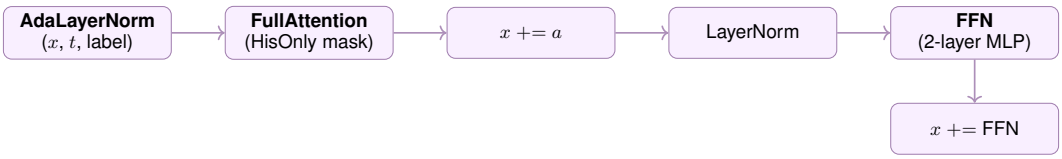
Visualisation. Below we visualise the four masks for a toy sequence of length $L = 12$ with block size $h = 4$ (3 blocks). ■ = attend, ■ = masked.



G.3 Encoder Architecture

The encoder processes the *clean history* $\hat{\mathbf{B}}_{1:k-1}$ and returns contextualised representations used by the decoder's cross-attention layers.

EncoderBlock. Each of the L_{enc} blocks applies the following computation (residual connections and layer norms omitted for clarity):



AdaLayerNorm. The diffusion timestep t is embedded via sinusoidal positional encoding, then projected to scale and shift parameters:

$$\text{AdaLN}(x, t) = \text{LayerNorm}(x) \odot (1 + \gamma_t) + \delta_t, \quad [\gamma_t; \delta_t] = \mathbf{W}_{\text{ada}}\, \text{SiLU}(\text{SinEmb}(t)).$$

FullAttention with KV-Cache. During inference the encoder runs with `use_cache=True`, accumulating (K_ℓ, V_ℓ) pairs across blocks:

$$\text{Attention}_i = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d_k}} + \mathbf{M}_{\text{HisOnly}}\right) \mathbf{V}_i,$$

where $[\mathbf{K}_i; \mathbf{V}_i]$ is the concatenation of cached and current keys/values. This avoids re-encoding history for every new block, reducing encoder inference cost from $\mathcal{O}(B^2h^2)$ to $\mathcal{O}(Bh \cdot (B-1)h)$ across B blocks.

Positional encoding (encoder). Learnable positional embeddings $\mathbf{pe} \in \mathbb{R}^{1 \times L_{\max} \times d_m}$ are added at a configurable `offset` (= `history_offset`), allowing the encoder to correctly locate each history block in absolute position space regardless of generation length.

G.4 Decoder Architecture and Sub-components

The decoder denoises the *current block* $\mathbf{B}_k^n$ conditioned on the encoder output. It consists of L_{dec} **DecoderBlock**s followed by additive combination layers.

DecoderBlock forward pass:

1. **Self-attention** (SelfOnly mask): bidirectional refinement within the current block.
2. **Cross-attention** (SelfOnly / CrossAttn mask): attends to encoder representations of the last history block.
3. **Block reshape**: flatten batch and block dimensions to $(\hat{B} \cdot N_B, h, d_m)$, where $N_B = L/h$ is the number of blocks.
4. **Projection + split**: $[\mathbf{x}_1; \mathbf{x}_2] = \text{Conv1d}_{h \to 2h}(\mathbf{x})$, yielding trend input $\mathbf{x}_1$ and season input $\mathbf{x}_2$.
5. **TrendBlock**($\mathbf{x}_1$) + **FourierLayer**($\mathbf{x}_2$) $\to$ residual trend $\Delta\mathbf{T}$ and residual season $\Delta\mathbf{S}$.
6. **Residual MLP**, global mean subtraction, and **linear projection** for mean component $\mathbf{m}$.

Final outputs are accumulated across all L_{dec} blocks:

$$\mathbf{S}_{\text{total}} = \textstyle\sum_\ell \Delta\mathbf{S}^{(\ell)}, \quad \mathbf{T}_{\text{total}} = \textstyle\sum_\ell \Delta\mathbf{T}^{(\ell)}, \quad \mathbf{m}_{\text{total}} = \left[\mathbf{m}^{(1)} \mid \cdots \mid \mathbf{m}^{(L_{\text{dec}})}\right].$$

Sub-module	Description	Output shape
TrendBlock	3-degree polynomial regressor: Conv1d ($h \to 3$) $\to$ act $\to$ Conv1d ($d_m \to D$) $\to$ polynomial basis $\mathbf{P} \in \mathbb{R}^{3\times h}$ (powers $p = 1, 2, 3$ of linspace)	$(N_B,\ h,\ D)$
FourierLayer	Inverse DFT seasonality: compute rFFT, select top-k ($k = \lfloor \text{factor} \cdot \log h \rfloor$) frequencies, reconstruct via amplitude $\times \cos(2\pi f t + \phi)$	$(N_B,\ h,\ d_m)$
proj_dict	`nn.ModuleDict` keyed by `str(block_size)`: Conv1d ($h \to 2h$, k=1). Populated for all valid BSC sizes $h \in \mathcal{V}$ at construction time	$(N_B,\ 2h,\ d_m)$
trend_dict	`nn.ModuleDict` keyed by `str(block_size)`: one **TrendBlock** per valid size; dispatched dynamically based on current h	$(N_B,\ h,\ D)$
combine_s	Conv1d ($d_m \to D$, kernel from config): maps accumulated season embedding to feature space; adds inverse embedding residual	$(N_B,\ h,\ D)$
combine_m	Conv1d ($L_{\text{dec}} \to 1$, k=1): collapses per-layer mean vectors into single global mean; adds trend and embedding residual mean	$(N_B,\ 1,\ D)$

Dynamic block-size dispatch. Because the BSC varies h across training iterations, `proj_dict` and `trend_dict` are pre-populated with entries for every valid block size $h \in \mathcal{V} = \{v : 960 \bmod v = 0,\ 4 \le v \le 480\}$. At each forward call the correct entry is selected via `str(block_size)`, avoiding any conditional branching or re-initialisation overhead.

Final output reconstruction. Let $\mathbf{R}$ be the inverse-embedded residual in $\mathbb{R}^{B\times L\times D}$. The noise prediction target is:

$$\hat{\boldsymbol{\epsilon}} = \underbrace{\texttt{combine_m}(\mathbf{m}_{\text{total}}) + \bar{\mathbf{R}} + \mathbf{T}_{\text{total}}}_{\text{trend component}} + \underbrace{\texttt{combine_s}(\mathbf{S}_{\text{total}}) + \mathbf{R} - \bar{\mathbf{R}}}_{\text{seasonal+residual component}},$$

where $\bar{\mathbf{R}}$ denotes the block-wise mean of $\mathbf{R}$.

G.5 Diffusion Noise Schedule and Loss

Cosine schedule. We use the cosine β-schedule:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos^2\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right), \quad s = 0.008,$$

with $\beta_t = 1 - \bar{\alpha}_t / \bar{\alpha}_{t-1}$ clipped to $[0, 0.999]$.

Loss reweighting. The per-timestep loss weight is set to:

$$\lambda_t = \frac{\sqrt{\alpha_t}\,\sqrt{1-\bar{\alpha}_t}}{\beta_t \cdot 100},$$

which upweights intermediate diffusion steps where the signal-to-noise ratio is most informative, following the design in Diffusion-TS [Yuan and Qiao, 2024].

Fourier auxiliary loss. In addition to the primary ℓ_1 reconstruction loss, a Fourier-domain auxiliary loss is computed at the block level:

$$\mathcal{L}_{\text{Fourier}} = \left\|\text{Re}(\hat{F}) - \text{Re}(F^*)\right\|_1 + \left\|\text{Im}(\hat{F}) - \text{Im}(F^*)\right\|_1,$$

where $\hat{F}, F^*$ are the `norm='forward'` DFT of predicted and target blocks reshaped to $(\hat{B}\cdot N_B, h, D)$. The combined loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\ell_1} + w_{\text{ff}}\,\mathcal{L}_{\text{Fourier}}, \qquad w_{\text{ff}} = \frac{\sqrt{h}}{5}.$$

Hyperparameter	Symbol	Value
Number of diffusion steps (ETTh1, Stocks)	T	500
Number of diffusion steps (Energy, fMRI)	T	1000
Sampling steps (fMRI only)	T_s	50
Noise schedule	—	cosine
Cosine offset	s	0.008
Loss type	—	ℓ_1
Fourier loss weight	w_{ff}	$\sqrt{h}/5$
EMA decay	—	0.995
EMA update interval	—	10 steps

G.6 Per-Dataset Hyperparameter Configurations

Hyperparameter	ETTh1	Stocks	Energy	fMRI
Variates D	7	6	28	50
Sequence window W	960	960	960	960
Encoder layers L_{enc}	3	4	4	8
Decoder layers L_{dec}	2	4	3	8
Model dim d_{model}	64	64	96	96
Attention heads n_h	4	4	4	4
MLP expansion factor	4	4	4	4
Conv kernel (enc/dec)	1	1	1	5
Conv padding	0	0	0	2
Inference block size h_{target}	48	48	48	48
Diffusion steps T	500	500	1000	1000
Sampling steps T_s	500	500	1000	50
Attn / Resid dropout	0.0	0.0	0.0	0.0
Batch size	128	64	64	64
Max epochs	36 000	20 000	50 000	30 000
BSC warmup iters	7 200	4 000	10 000	6 000
BSC decay iters	25 200	14 000	35 000	21 000
Warmup peak LR	8×10^{-4}			
Base / min LR	1×10^{-5}			
LR scheduler patience	4 000	2 000	5 000	3 000
Gradient accumulation steps	2			
Gradient clip norm	1.0			
Optimizer	Adam (β_1=0.9, β_2=0.96)			
Train proportion	90%			

Note on BSC ratio. Across all datasets the Warmup : Stable : Decay ratio is held constant at **2 : 1 : 7**, matching the `solver.py` comments. The Stable phase uses $h_{\max} = 96$ for all datasets. The Decay phase terminates at $h_{\text{target}} = 48$, which is also the inference block size used in all experiments.

H Algorithm Pseudocode

This appendix provides complete pseudocode for all three generation procedures of **Horizon Diffusion**: unconditional arbitrary-length synthesis (§H.1), training with the Block Size Curriculum (§H.2), and conditional generation for imputation and forecasting (§H.3). A summary of shared notation is given in Table 7.

Symbol	Meaning
$\mathbf{B}_k^n$	Noisy version of the k-th horizon block at diffusion step n
$\hat{\mathbf{B}}_k$	Clean (denoised) k-th block, used as conditioning context
h_k	Length (in patches) of the k-th block
h_{target}	Inference block size (terminal size of BSC Decay phase)
K	Total number of horizon blocks
$\mathbf{M}_k$	Horizon-Causal mask for block k
$\boldsymbol{\epsilon}_\theta$	Noise-prediction network (Horizon Diffusion Transformer)
$\bar{\alpha}_n$	Cumulative noise schedule coefficient at step n
σ_n	Posterior standard deviation at step n
$h(i)$	BSC block size at training iteration i
$\mathcal{V}$	Valid block-size space (divisors of L in $[v_{\min}, v_{\max}]$)

Table 7: Shared notation used across all algorithms in this appendix.

H.1 Inference for Arbitrary-Length Synthesis

Algorithm 1 specifies the full inference procedure for arbitrary-length synthesis. Given target length H_p^{target}, the algorithm allocates $K_{\text{test}} = \max(1, \lceil H_p^{\text{target}}/h_{\text{target}} \rceil)$ blocks with patches distributed as evenly as possible (lengths differing by at most one), so any target length is supported exactly. Each block is initialized from Gaussian noise and refined through N DDPM reverse steps under the horizon-causal mask, using all previously generated clean blocks as context.

Algorithm 1 HORIZON DIFFUSION — Inference (Arbitrary-Length Synthesis)

Require: Conditioning $\mathbf{c}$, target patch length H_p^{target}, target block length h_{target}, diffusion steps N
Ensure: Synthetic sequence $\hat{\mathbf{X}} \in \mathbb{R}^{H_p^{\text{target}} \times d_{\text{model}}}$

$K_{\text{test}} \leftarrow \max(1, \lceil H_p^{\text{target}}/h_{\text{target}} \rceil)$
Compute balanced block lengths $\{h_1, \ldots, h_{K_{\text{test}}}\}$ s.t. $\sum_k h_k = H_p^{\text{target}}$
GeneratedBlocks $\leftarrow [\,]$
for $k = 1$ **to** K_{test} **do**
  $\mathbf{B}_k^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Initialise block from pure noise}
  for $n = N$ **to** 1 **do**
    $\mathbf{Z}_k^n \leftarrow [\mathbf{c},\ \text{GeneratedBlocks},\ \mathbf{B}_k^n]$
    $\mathbf{M}_k \leftarrow \text{BUILDHORIZONCAUSALMASK}(k)$
    $\hat{\boldsymbol{\epsilon}}_k \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{Z}_k^n, n, \mathbf{M}_k)$
    $\mathbf{B}_k^{n-1} \leftarrow \text{DDPMUPDATE}(\mathbf{B}_k^n, \hat{\boldsymbol{\epsilon}}_k, n)$
  end for
  GeneratedBlocks.APPEND$(\mathbf{B}_k^0)$
end for
return $\hat{\mathbf{X}} \leftarrow$ CONCATENATE(GeneratedBlocks)

Subroutine DDPMUPDATE$(\mathbf{B}^n, \hat{\epsilon}, n)$

Computes the standard DDPM reverse step:

$$\mathbf{B}^{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\mathbf{B}^n - \frac{1-\alpha_n}{\sqrt{1-\bar{\alpha}_n}}\,\hat{\boldsymbol{\epsilon}}\right) + \sigma_n\,\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ if } n > 1 \text{ else } \mathbf{0}.$$

For accelerated inference (DDIM), a linearly spaced subsequence of timesteps is used and $\sigma_n = \eta\sqrt{(1-\bar{\alpha}_{n-1})/(1-\bar{\alpha}_n)}\sqrt{1-\alpha_n}$ with $\eta = 0$ (deterministic).

H.2 Training with the Block Size Curriculum

Algorithm 2 details the full training loop, highlighting how the BSC modulates $h(i)$ across iterations and how all K per-block losses are computed in a single parallel forward pass via teacher forcing.

Algorithm 2 HORIZON DIFFUSION — Training with Block Size Curriculum (BSC)

Require: Dataset $\mathcal{D}$, total iterations I_{total}, warmup budget I_{warmup}, decay budget I_{decay}, valid size space $\mathcal{V}$, inference block size h_{target}, noise predictor $\boldsymbol{\epsilon}_\theta$
Ensure: Trained parameters θ

▷ *Precompute BSC schedule constants*
$v_{\max} \leftarrow \max \mathcal{V}$; $M_v \leftarrow |\mathcal{V}| - 1$; $\mathcal{D}_{\text{dec}} \leftarrow$ descending sequence from $v_{\max}$ to h_{target}
for iteration $i = 1$ **to** I_{total} **do**
  ▷ *— Phase 1: Warmup (cubic ease-out) —*
  if $i < I_{\text{warmup}}$ **then**
    $t \leftarrow i / I_{\text{warmup}}$; $\phi \leftarrow 1 - (1-t)^3$
    $h(i) \leftarrow \mathcal{V}[\lfloor M_v \cdot \phi \rfloor]$
    ▷ *— Phase 2: Stable —*
  else if $i < I_{\text{total}} - I_{\text{decay}}$ **then**
    $h(i) \leftarrow v_{\max}$
    ▷ *— Phase 3: Decay —*
  else
    $t_{\text{dec}} \leftarrow (i - (I_{\text{total}} - I_{\text{decay}})) \,/\, I_{\text{decay}}$
    $h(i) \leftarrow \mathcal{D}_{\text{dec}}[\min(\lfloor t_{\text{dec}} \cdot |\mathcal{D}_{\text{dec}}| \rfloor,\ |\mathcal{D}_{\text{dec}}| - 1)]$
  end if
  ▷ *Sample minibatch and partition into $K = L/h(i)$ equal blocks*
  $\mathbf{X} \sim \mathcal{D}$; $K \leftarrow L/h(i)$; $[\mathbf{B}_1^0, \ldots, \mathbf{B}_K^0] \leftarrow \textsc{Partition}(\mathbf{X},\, h(i))$
  ▷ *Parallel per-block loss (teacher forcing, single forward pass)*
  Sample $n \sim \mathcal{U}(1, N)$ and noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for all $k \in [K]$
  $\mathbf{B}_k^n \leftarrow \sqrt{\bar{\alpha}_n}\, \mathbf{B}_k^0 + \sqrt{1 - \bar{\alpha}_n}\, \boldsymbol{\epsilon}_k$ (forward process, all k in parallel)
  for $k = 1$ **to** K **do**
    $\mathbf{Z}_k^n \leftarrow [\mathbf{c},\ \mathbf{B}_1^0, \ldots, \mathbf{B}_{k-1}^0,\ \mathbf{B}_k^n,\ \boldsymbol{\zeta}_{k+1}, \ldots, \boldsymbol{\zeta}_K]$ {Teacher-forced clean history}
    $\mathbf{M}_k \leftarrow \textsc{BuildHorizonCausalMask}(k,\, h(i))$
  end for
  $\hat{\boldsymbol{\epsilon}}_{1:K} \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{Z}_{1:K}^n,\, n,\, \mathbf{M}_{1:K})$ (all K blocks in one forward pass)
  $\mathcal{L} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \left\| \boldsymbol{\epsilon}_k - \hat{\boldsymbol{\epsilon}}_k \right\|_2^2 + w_{\text{ff}}\, \mathcal{L}_{\text{Fourier}}$
  Update θ via Adam; step LR scheduler; update EMA
end for
return θ

Remark (parallel training efficiency)

Because ground-truth clean blocks are used as history context (teacher forcing), the K per-block loss terms share the same *context tensor* and differ only in which positions are treated as the noisy target. All K noise predictions $\hat{\boldsymbol{\epsilon}}_{1:K}$ are therefore computed in a *single forward pass* with a combined mask $\mathbf{M}_{1:K}$, avoiding the sequential dependency that would arise during inference. Training time is thus independent of K.

H.3 Conditional Generation: Imputation and Forecasting

Both imputation and forecasting reuse the *same unconditionally trained model* without any fine-tuning, using different conditioning strategies. Algorithm 3 covers imputation (replace-based guidance with Langevin correction); Algorithm 4 covers prefix-conditional forecasting.

Algorithm 3 HORIZON DIFFUSION — Block-wise Imputation

Require: Observed sequence with mask $(\mathbf{X}_{\text{obs}}, \mathbf{M}_{\text{obs}})$ where $\mathbf{M}_{\text{obs}}[t] = 1$ iff t is observed; Langevin coefficient λ; step size ξ; sampling steps T_s; block size h

Ensure: Imputed sequence $\hat{\mathbf{X}}$

History $\leftarrow \mathbf{X}_{\text{obs},\,1:h}$ {Initialise context from first observed block}
GeneratedBlocks $\leftarrow$ [History]
for $k = 2$ **to** K **do**
  $\mathbf{X}^*_k \leftarrow \mathbf{X}_{\text{obs},\,s_k:e_k}$; $\mathbf{M}_k \leftarrow \mathbf{M}_{\text{obs},\,s_k:e_k}$
  $\mathbf{B}^{T_s} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $n = T_s$ **to** 1 **do**
    $\hat{\boldsymbol{\epsilon}}, \hat{\mathbf{B}}^0 \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{B}^n, n,$ GeneratedBlocks$)$
    $\bar{\mathbf{B}} \leftarrow \hat{\mathbf{B}}^0\sqrt{\bar{\alpha}_{n-1}} + \hat{\boldsymbol{\epsilon}}\sqrt{1-\bar{\alpha}_{n-1}}$ {DDIM mean}
    $\mathbf{B}^n \leftarrow$ LANGEVINCORRECT$(\mathbf{B}^n, \bar{\mathbf{B}}, \mathbf{X}^*_k, \mathbf{M}_k, \sigma_n, \lambda, \xi, n)$
    $\mathbf{X}^{n-1}_{k,\text{obs}} \leftarrow$ FORWARDDIFFUSE$(\mathbf{X}^*_k, n-1)$ {Diffuse target to step $n-1$}
    $\mathbf{B}^{n-1}[\mathbf{M}_k] \leftarrow \mathbf{X}^{n-1}_{k,\text{obs}}[\mathbf{M}_k]$ {Replace observed positions}
  end for
  $\mathbf{B}^0[\mathbf{M}_k] \leftarrow \mathbf{X}^*_k[\mathbf{M}_k]$ {Hard replacement at final step}
  GeneratedBlocks.APPEND$(\mathbf{B}^0)$
end for
return $\hat{\mathbf{X}} \leftarrow$ CONCATENATE(GeneratedBlocks)

Subroutine LANGEVINCORRECT$(\mathbf{B}, \bar{\mathbf{B}}, \mathbf{X}^*, \mathbf{M}, \sigma, \lambda, \xi, n)$

Applies K_{lg} steps of Langevin-style gradient updates on the unobserved positions, where $K_{\text{lg}} \in \{0, 1, 2, 3\}$ is scheduled by diffusion step n:

$$\ell(\mathbf{B}) = \lambda \frac{\|\bar{\mathbf{B}} - \mathbf{B}\|^2}{\sigma} + \frac{\|\hat{\mathbf{B}}^0[\overline{\mathbf{M}}] - \mathbf{X}^*[\overline{\mathbf{M}}]\|^2}{\sigma},$$

optimised with Adagrad at learning rate ξ, with injected Gaussian noise. Returns updated $\mathbf{B}$ with observed positions unchanged.

Algorithm 4 HORIZON DIFFUSION — Prefix-Conditional Forecasting

Require: Observed prefix $\mathbf{S}_{1:T_{\text{pre}}}$; forecast horizon H_{fore}; block size h_{target}; diffusion steps N

Ensure: Forecast $\hat{\mathbf{S}}_{T_{\text{pre}}+1\,:\,T_{\text{pre}}+H_{\text{fore}}}$

▷ *Encode prefix as conditioning context*
$\mathbf{c} \leftarrow [\mathbf{c}_{\text{start}},$ PATCHEMBED$(\mathbf{S}_{1:T_{\text{pre}}})]$
$K_{\text{fore}} \leftarrow \max(1, \lceil H_{\text{fore}}/h_{\text{target}} \rceil)$
Compute block lengths $\{h_1, \ldots, h_{K_{\text{fore}}}\}$
GeneratedBlocks $\leftarrow$ [ ]
for $k = 1$ **to** K_{fore} **do**
  $\mathbf{B}^N_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $n = N$ **to** 1 **do**
    $\mathbf{Z}^n_k \leftarrow [\mathbf{c},$ GeneratedBlocks, $\mathbf{B}^n_k]$ {Prefix tokens serve as permanent context}
    $\mathbf{M}_k \leftarrow$ BUILDHORIZONCAUSALMASK$(k + \lceil T_{\text{pre}}/h_{\text{target}} \rceil)$ {Offset mask by prefix blocks}
    $\hat{\boldsymbol{\epsilon}}_k \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{Z}^n_k, n, \mathbf{M}_k)$
    $\mathbf{B}^{n-1}_k \leftarrow$ DDPMUPDATE$(\mathbf{B}^n_k, \hat{\boldsymbol{\epsilon}}_k, n)$
  end for
  GeneratedBlocks.APPEND$(\mathbf{B}^0_k)$
end for
return $\hat{\mathbf{S}} \leftarrow$ PATCHDECODE(CONCATENATE(GeneratedBlocks))

Remark (no retraining required)

Algorithms 3 and 4 both operate on the *same unconditionally trained model* checkpoint without any task-specific fine-tuning or architectural changes. The prefix in forecasting is handled purely by prepending patch-embedded tokens to $\mathbf{c}$, while the block-wise reverse diffusion and Horizon-Causal mask logic remain identical to unconditional generation (Algorithm 1).

I Baseline Details and Reproducibility

All baselines were re-implemented or re-run by us under **identical** data splits, sliding-window sizes ($W{=}960$), preprocessing pipelines, and evaluation protocols. No result is copied from prior publications. Hyper-parameters for each baseline follow their respective original papers unless a grid search on the validation portion of the training set yielded a meaningfully better configuration, in which case the improved setting is reported.

I.1 Baseline Configurations

Table 8 summarises the six baselines used in our main comparison, covering GAN, VAE, diffusion, VQ-VAE, and autoregressive paradigms. For each method we list the generative family, the backbone architecture, the length-extension strategy applied when the target length H differs from the training length, and the training length used in our experiments.

Table 8: **Baseline overview.** "Fixed-H" denotes that the model is trained and evaluated at a single, identical sequence length. For methods that cannot natively extend beyond their training length, we use *naive concatenation* of independently drawn samples as the only applicable strategy; this is precisely the failure mode our method resolves.

Method	Family	Backbone	Length-Extension Strategy	Train H
TimeGAN	GAN	Bidirectional RNN (generator + discriminator)	Naive concatenation of fixed-length samples	192 / 384 / 768
TimeVAE	VAE	Convolutional encoder-decoder	Naive concatenation of fixed-length samples	192 / 384 / 768
Diffusion-TS	Diffusion	Transformer + Fourier / trend decomposition	Naive concatenation (boundary discontinuities)	192 / 384 / 768
TimeVQVAE	VQ-VAE	CNN encoder + codebook + transformer prior	Autoregressive token generation (codebook)	192 / 384 / 768
SDformer	Diffusion	Transformer + discrete tokenisation (VQ)	Autoregressive token generation post-quantisation	192 / 384 / 768
TimeMAR	Autoregressive	Masked autoregressive transformer (MAR)	Native; step-by-step generation	192 / 384 / 768

Each baseline is trained **separately** for each target length $H \in \{192, 384, 768\}$, so that it always receives the same generation length at training and inference—granting each method its most favorable possible setting. By contrast, **Horizon Diffusion is trained once** (at $W{=}960$) and generates all three lengths from a single checkpoint.

I.2 Evaluation Protocol

We evaluate all methods on four metrics that collectively probe *distributional fidelity*, *temporal coherence*, and *downstream utility*. All metrics are lower-is-better (↓). Results are reported as mean $\pm$ std over **5 independent runs**, each with a different random seed governing both generation and metric computation.

Metric Definitions

(M1) Context-FID. A self-supervised TS2VEC encoder [Yue et al., 2022] is fitted on real training data with output dimensionality d=320. Both the real held-out set and the generated set are encoded via `full_series` pooling, producing representation matrices $\mathbf{R}_{\text{real}}, \mathbf{R}_{\text{gen}} \in \mathbb{R}^{N\times 320}$. Context-FID is then the Fréchet distance:

$$\text{C-FID} = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \text{tr}\big(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\big),$$

where $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ are the empirical mean and covariance of each set's representations. A shared random permutation index is applied to both sets before computation so that the same N sequences are compared.

(M2) Discriminative Score. A post-hoc GRU discriminator (hidden dim $\lfloor D/2 \rfloor$, 2 000 Adam iterations, batch size 128) is trained to classify real vs. synthetic sequences. The discriminative score is $|\text{accuracy} - 0.5|$: a score of 0 indicates the classifier cannot distinguish real from synthetic (ideal), while 0.5 indicates perfect discrimination (worst case).

(M3) Correlational Score. For a multivariate series $\mathbf{X} \in \mathbb{R}^{N\times T\times D}$, the cross-correlation at lag 1 is computed over all lower-triangular variate pairs via `cacf_torch`, averaged across samples:

$$\text{Corr} = \frac{1}{10} \sum_{(i,j)\in\text{tril}} \big|\hat{\rho}_{ij}^{\text{gen}} - \hat{\rho}_{ij}^{\text{real}}\big|,$$

where the factor $1/10$ is a normalisation constant matching the implementation. This penalises failure to reproduce inter-variate linear dependencies.

(M4) Predictive Score. A post-hoc GRU predictor (hidden dim $\lfloor D/2 \rfloor$, 5 000 Adam iterations, batch size 128) is trained on *synthetic* data with a one-step-ahead objective, then evaluated on *real* data (**TSTR** protocol). The score is the mean absolute error (MAE) over the held-out real test set. Low predictive score indicates that the synthetic data faithfully captures the temporal dynamics relevant for forecasting.

I.3 Metric Implementation Notes

We collect several implementation details that are critical for exact reproducibility but are too low-level for the main text.

Table 9: **Metric implementation details.** All post-hoc models (discriminator, predictor) use Adam with default $(\beta_1, \beta_2) = (0.9, 0.999)$ and no weight decay.

Metric	Model	Iters	Batch	Hidden dim	Additional notes
Context-FID	TS2Vec encoder	until convergence	8	320-d output	Fitted on real train; encodes with `full_series` window
Discriminative	GRU	2 000	128	$\lfloor D/2 \rfloor$	Score = $\lvert\text{acc} - 0.5\rvert$; 80/20 train-test split of eval set
Correlational	—	—	—	—	Lag-1 CACF; lower-triangular pairs; divided by 10
Predictive	GRU	5 000	128	$\lfloor D/2 \rfloor$	TSTR: train on synthetic, test on real; MAE loss

The following operational choices are shared across *all* metrics and *all* methods:

- **Sample size.** $N_{\text{eval}} = 256$ sequences are generated per method per run, matching the `sample_size` entry in every dataset config. For real data, a random subset of equal size is drawn without replacement from the test partition.
- **Normalization.** All generated sequences are returned in the normalised $[-1, 1]$ domain, then mapped back to $[0, 1]$ via `unnormalize_to_zero_to_one` before metric computation, ensuring comparability with the real (Min-Max scaled) test data.
- **Run-to-run variance.** Each of the 5 runs uses a distinct global seed governing: (i) the random permutation applied before C-FID, (ii) the train/test split of the discriminator/predictor evaluation set, and (iii) the mini-batch sampling order inside each post-hoc training loop. The model checkpoint is *fixed* across the 5 runs; only the stochastic generation and evaluation procedures vary.

- **Handling of complex-valued matrix square roots.** When $(\mathbf{\Sigma}_r\mathbf{\Sigma}_g)^{1/2}$ contains imaginary components (a numerical artefact of finite-sample covariance estimation), we take the real part following standard FID practice.
- **TS2Vec training.** The TS2Vec encoder is trained freshly for each dataset (not shared across methods) with default hyper-parameters: learning rate 10^{-3}, output dims 320, `max_train_length` 3 000, batch size 8. The same fitted encoder is used to evaluate *all* methods on that dataset, guaranteeing a fair comparison.

I.4 Hyperparameters of Horizon Diffusion

Table 10 reports all hyperparameters of Horizon Diffusion for each dataset, as used in the main experiments. All models share the same inference block size $h_{\text{target}}=48$ and the same BSC terminal block size.

Table 10: **Horizon Diffusion hyperparameters per dataset.** T: diffusion timesteps used during training. T_{sample}: timesteps used at inference (DDIM-style subsampling where $T_{\text{sample}} < T$). d: model dimension. BSC ratios give (Warmup : Stable : Decay) iteration counts.

Dataset	Variates D	T	T_{sample}	d	Enc layers	Dec layers	Heads	Batch	BSC (W:S:D)
ETTh1	7	500	500	64	3	2	4	128	7200 : 3600 : 25200
Stocks	6	500	500	64	4	4	4	64	4000 : 2000 : 14000
Energy	28	1000	1000	96	4	3	4	64	10000 : 5000 : 35000
fMRI	50	1000	50	96	8	8	4	64	6000 : 3000 : 21000

All datasets use: sequence length $W=960$, patch size $P=1$ (no sub-patch compression in the time domain beyond token-level embedding), inference block size $h_{\text{target}}=48$, β-schedule COSINE, loss type ℓ_1, gradient accumulation every 2 steps, EMA decay 0.995 (updated every 10 steps), base learning rate 10^{-5} with warm-up to 8×10^{-4} over 500 steps, ReduceLROnPlateau (factor 0.5) thereafter.

I.5 Reproducibility Checklist

- ✓ Model code, config files, and dataset loaders are available at `https://github.com/wxbs/HorizonDiffusion`.
- ✓ All dataset configs (`energy.yaml`, `etth.yaml`, `stocks.yaml`, `fmri.yaml`) with exact hyperparameters are included.
- ✓ The BSC schedule functions (`get_WSD_block_size`, `get_cubic_block_size`, `get_Reverse_Curriculum_block_size`) are fully implemented in `engine/solver.py`.
- ✓ Mask construction utilities (`build_encoder_mask_HisOnly`, `build_decoder_mask_SelfOnly`, ablation masks) are in `Models/interpretable_diffusion/model_utils.py`.
- ✓ All four metric implementations (Context-FID, Discriminative, Correlational, Predictive) are in `Models/Utils/`.
- ✓ Data preprocessing and sliding-window construction are in `Models/Utils/Data_utils/real_datasets.py`.
- ✓ A fixed random seed (`seed=12345`) is set via `seed_everything` for all experiments reported in the main paper.
- ✓ Experiment entry point with all command-line arguments is provided in `main.py`.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three main contributions: (i) the first diffusion-based architecture for native arbitrary-length time series synthesis via Horizon-Causal Attention, (ii) the Block Size Curriculum (BSC), and (iii) empirical results showing state-of-the-art performance across four datasets and three target lengths, with stable generalization up to $4\times$ the training length. All claims are supported by theoretical analysis (Section 5) and experiments (Section 6).

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Conclusion (Section 7) explicitly acknowledges that block-wise reverse diffusion incurs non-trivial inference cost since each block requires a full denoising trajectory, and identifies consistency distillation as a promising future direction.

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: Theorem 5.1 (Bounded Error Accumulation) explicitly states assumptions (A1) and (A2), provides a complete inductive proof in the main text, and references Appendix A.1 for the full derivation of the Lipschitz step.

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: Full hyperparameters per dataset are reported in Appendix I.4 and Appendix G.6, including BSC schedules, optimizer settings, model dimensions, and diffusion configurations. All baseline configurations and evaluation protocols are detailed in Appendix I. Code is available at `https://github.com/wxbs/HorizonDiffusion`.

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

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

   Answer: [Yes]

   Justification: Code, config files (`energy.yaml`, `etth.yaml`, `stocks.yaml`, `fmri.yaml`), dataset loaders, metric implementations, and a fixed random seed (`seed=12345`) are all provided in our code repository (`https://github.com/wxbs/HorizonDiffusion`). All four datasets used are publicly available benchmarks.

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: Section 6.1 describes datasets, metrics, and baselines. Appendix I provides complete baseline configurations, evaluation protocols, metric implementation details, and full hyperparameter tables for Horizon Diffusion across all datasets, including data splits (90% train), optimizer (Adam), learning rate schedules, and BSC iteration counts.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All results in Tables 1 and 2 and the fMRI appendix table (Table 4) are reported as mean $\pm$ std over 5 independent runs. Appendix I.2 specifies that each run uses a distinct global random seed governing generation, evaluation set splits, and mini-batch sampling, while the model checkpoint is fixed.

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: The config files in our code repository (`https://github.com/wxbs/HorizonDiffusion`) specify the hardware used for all experiments. Training iteration counts and batch sizes are reported per dataset in Appendix I.4 and G.6, allowing estimation of total compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: This work involves synthetic time series generation on publicly available benchmarks and raises no ethical concerns. All datasets are used in accordance with their intended research purposes.

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents foundational research on synthetic time series generation. The primary applications are data augmentation and privacy-preserving data sharing in healthcare and finance, which carry positive societal value. There is no direct path to negative applications beyond the general risks associated with any generative model (e.g., generating plausible but fictitious records), and the method does not introduce new capabilities that meaningfully elevate such risks beyond existing generators.

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?

    Answer: [N/A]

    Justification: The released model is trained on publicly available time series benchmarks (energy, finance, medical simulation) and does not pose high risk for misuse; no special safeguards beyond standard open-source release are required.

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: [N/A]

    Justification: All datasets (ETTh1, Energy, Stocks, fMRI) and baseline methods are cited with their original papers. Licenses for these publicly available benchmarks are documented in our code repository (`https://github.com/wxbs/HorizonDiffusion`).

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: No new datasets are introduced. The released code is documented via config files, inline comments, and the reproducibility checklist in Appendix I.5.

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 paper does not involve crowdsourcing or research with human subjects.

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 paper does not involve human subjects. The fMRI dataset used is a publicly available *simulated* BOLD signal benchmark, not collected from real participants.

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?

Answer: [N/A]

Justification: LLMs are not used as any important, original, or non-standard component of the core methodology. Any LLM usage was limited to writing assistance and does not affect the scientific content or originality of the research.