HALO: ENHANCING TIME SERIES GENERATION VIA HYPERSPHERICAL LATENTS AND MASKED AUTOREGRESSIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

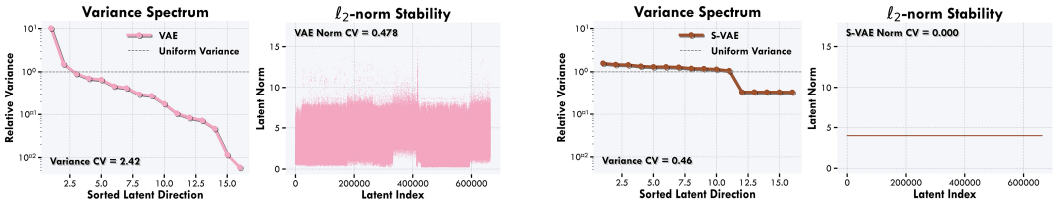
Most existing time series generators rely on a two-stage modeling paradigm: **the first stage** learns discrete latent representations of time series; **the second stage** performs autoregressive modeling on these discrete latents through next token prediction. However, this paradigm suffers from two stage-specific limitations: the first stage can lead to information loss when discretizing continuous time series, while the second stage is prone to error accumulation during autoregressive generation. To address these limitations, our core idea is to perform generative modeling in a continuous latent space with a more efficient autoregressive framework. We propose **HALO**, which enhances time series generation via **H**yperspheric**A**l **L**atents and Masked Aut**O**regressive modeling to achieve this goal by tackling two key bottlenecks: (1) variance and scale heterogeneity of continuous latent representations; (2) the difficulty of balancing generation efficiency with temporal correlation modeling. HALO first introduces a hyperspherical VAE that constrains continuous latents to a fixed-radius hyperspherical shell, effectively stabilizing the numerical fluctuations of continuous latent representations. Secondly, we develop a masked autoregressive model that balances parallel decoding and temporal correlation learning, substantially reducing the number of inference steps required for generation and improving generation stability. Our extensive experiments demonstrate that HALO achieves state-of-the-art generation performance while offering significantly improved inference efficiency over existing advanced baselines.

1 INTRODUCTION

Time series generation plays an important role in a wide range of real-world domains, including finance (Xia et al., 2024; Coletta et al., 2023; Shi et al., 2026), healthcare (Deng et al., 2025; Wang et al., 2024), and industrial fault diagnosis (Xu et al., 2025; Lee et al., 2026). The success of deep generative models in natural language processing (Zhang et al., 2023) and computer vision (Croitoru et al., 2023) has motivated their application to time series generation. By synthesizing realistic time series, these approaches can help alleviate data scarcity in settings where access to real-world data is constrained by privacy concerns or the rarity of extreme events (Zhang et al., 2026).

Existing time generation methods (Chen et al., 2024; Xu et al., 2026; Feng et al., 2026; Lyu et al., 2026) generally adopt a *two-stage modeling paradigm*. **In the first stage**, a discrete tokenizer, such as VQ-VAE (Van Den Oord et al., 2017), is trained to tokenize time series into discrete latent representations. **In the second stage**, an autoregressive model (Achiam et al., 2023) is trained, following the "*next token prediction*" to model the conditional distributions of these discrete tokens (Bengio et al., 2003). Although they have achieved promising results, they still suffer from several limitations. Firstly, discrete tokenization is inherently misaligned with the continuous nature of time series, potentially causing the loss of fine-grained temporal information. Secondly, autoregressive models rely on sequential token-by-token generation, which is prone to error accumulation and often leads to unstable and inefficient generation (Ranzato et al., 2015). To address these key limitations, we aim to perform generative modeling in a continuous latent space with a more efficient autoregressive framework, thereby mitigating information loss and improving generation stability and efficiency. Nonetheless, achieving this goal needs to overcome the two key challenges:

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



(a) VAE (b) S-VAE

Figure 1: Quantitative comparison of VAE (Kingma & Welling, 2013) and S-VAE (Ours) latents: relative variance spectra and token norms. Relative variance denotes covariance eigenvalues normalized by their mean, with dashed lines indicating uniform variance. CV denotes the coefficient of variation. S-VAE exhibits more balanced directional variances and nearly constant token norms.

Challenge 1: Continuous tokens exhibit substantial variance and scale heterogeneity. When modeling non-stationary and temporally heterogeneous time series, continuous tokens without explicit constraints on their numerical range may exhibit substantial variance and scale heterogeneity. As shown in Figure 1(a), we quantitatively analyze the continuous tokens produced by a Gaussian VAE (Kingma & Welling, 2013). Several clear observations can be drawn: ① the normalized covariance eigenspectrum decays rapidly, indicating that most of the latent variance is concentrated along only a few principal directions; ② the latent norms vary substantially across tokens, revealing considerable token-wise scale heterogeneity. These valuable insights indicate that continuous tokens exhibit both directional variance imbalance and latent-wise scale variation. Overall, these forms of continuous latents heterogeneity severely increase the learning and modeling difficulty of the second-stage autoregressive model.

Challenge 2: Efficient generation while preserving temporal correlation modeling. Standard autoregressive models explicitly model temporal dependencies by predicting each token conditioned on the previous ground-truth tokens (Bengio et al., 2003), as shown in Figure 2(a). Such an architecture provides the model with a strong capability to model temporal correlations, while this sequential, token-by-token decoding process limits generation efficiency. A straightforward and widely adopted alternative is to use MAE-style masked representation learning (He et al., 2022; Chen et al., 2025), which enables more efficient generation through parallel token prediction. Although this approach alleviates the inefficiency of token-by-token generation, it weakens the advantage of explicit temporal correlation modeling, resulting in suboptimal generation quality and fidelity (Chen et al., 2024).

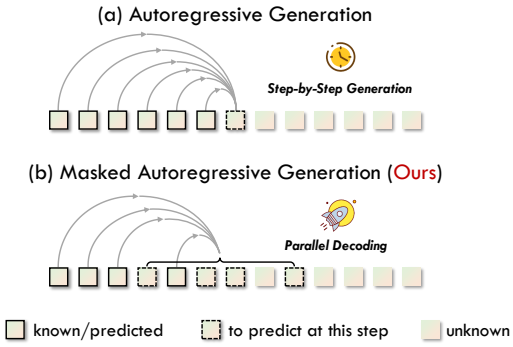


Figure 2: (a) *Autoregressive generation* produces tokens sequentially, and is prone to unstable and inefficient. (b) *Masked autoregressive generation* predicts unknown tokens conditioned on known tokens, achieving stable and efficient generation.

Motivated by the aforementioned observations and reflections, we address the above challenges by introducing **HALO**, a novel generative framework, which enhances time series generation via **H**yperspheric**A**l **L**atents and Masked Aut**O**regressive modeling. Unlike existing two-stage deep generative models for time series generation in a discrete representation space, HALO introduces a hyperspherical VAE (S-VAE) and masked autoregressive modeling to alleviate variance and scale heterogeneity of the continuous latent representations and balance efficiency and stability during autoregressive generation, respectively. More specifically, to address **the first challenge**, we introduce an S-VAE based on the von Mises–Fisher distribution. The S-VAE constrains continuous latent representations to a hypersphere with a fixed radius, making the continuous latents learned by the autoregressive model scale-invariant. As illustrated in Figure 1(b), the latents produced by our S-VAE exhibit more stable scale dynamics and greater directional homogeneity, with their norms strictly constrained to the fixed radius of the hypersphere. Therefore, the heterogeneities of continuous latents are substantially alleviated, reducing the difficulty of learning their distributions for the autoregressive model. Furthermore, as an additional benefit, this design mitigates scale drift associated with variance and scale heterogeneity during autoregressive generation (Sun et al., 2026).

On the other hand, to address **the second challenge**, inspired by recent advances in autoregressive models in visual domain (Kumbong et al., 2025; Chang et al., 2022; Li et al., 2023), we introduce masked autoregressive modeling. Following this idea, we develop a more general autoregressive model that *predicts unknown tokens conditioned on known tokens* to strike a balance between masked representation learning and unidirectional autoregressive models, as shown in Figure 2(b). As a result, it is empowered with strong capabilities for both parallel decoding and temporal correlation learning. These compact designs not only improve the model's generative capability, but also significantly reduce the number of inference steps required for generating a single sample, thereby improving generation stability and efficiency. Our contributions can be summarized as follows:

- We reflect the limitations of existing two-stage deep time series generative models, particularly the loss of fine-grained temporal information and error accumulation during autoregressive generation. Therefore, we propose HALO, a novel deep generative framework that performs generative modeling in a continuous latent space with an efficient autoregressive framework, enabling the generation of high-quality and high-fidelity synthetic time series.
- We design a novel S-VAE that constrains latents to lie on a fixed-radius hypersphere, effectively mitigating variance and scale heterogeneity in continuous latent representations. At the same time, this fundamentally alleviates the degradation in generation quality and fidelity caused by scale drift during autoregressive generation.
- We develop a masked autoregressive model that predicts unknown tokens conditioned on known tokens to strikes a balance between parallel decoding and temporal correlation learning. While preserving temporal awareness, it significantly reduces the number of inference steps required for generation, thereby improving generation efficiency and stability.
- Extensive experiments demonstrate that HALO achieves state-of-the-art generation performance. Moreover, compared with other strong baselines, HALO also demonstrates superior inference efficiency. Our code and scripts are available at here.

2 Related Works

2.1 Deep Generative Models for Time Series

Recent years have witnessed rapid advances in deep generative modeling for time series. Early studies such as TimeGAN (Yoon et al., 2019) combine supervised and adversarial objectives in a jointly learned embedding space to preserve temporal dynamics. As an alternative to adversarial generation, TimeVAE (Desai et al., 2021) employs an interpretable encoder-decoder architecture that explicitly models characteristic temporal components such as trends and seasonalities. As generative modeling techniques continue to advance, two-stage deep generative models have continued to draw considerable attention from both researchers and practitioners (Feng et al., 2024; Shibo et al., 2025; Li et al., 2026). As a representative work, SDformer (Chen et al., 2024) introduces the discrete-latent generative modeling paradigm to the time series generation community. In the first stage, it trains a VQ-VAE (Van Den Oord et al., 2017) to compress time series into a discrete latent space, while the second stage employs a GPT-style autoregressive model to learn the joint conditional probability distribution over the resulting discrete latent representations. Recently, to better model multi-scale temporal dynamics in time series, TimeMAR (Xu et al., 2026) and MSDformer (Feng et al., 2026) further advance this paradigm by extending VQ-VAE to multi-scale architectures. Despite their effectiveness, these methods represent intrinsically continuous signals using a finite codebook, which introduces quantization error and limits the fidelity of fine-grained temporal variations.

2.2 Autoregressive Modeling for Time Series Generation

Beyond natural language processing, computer vision, and audio generation, autoregressive models have also been widely adopted in the time series generation. They can be broadly categorized into two lines: GPT-style unidirectional autoregressive models and BERT-style bidirectional autoregressive models (Li et al., 2023). For unidirectional autoregressive models, representative methods such as SDformer (Chen et al., 2024) and MSDformer (Feng et al., 2026) adopt the next token prediction paradigm. For instance, TimeMAR (Xu et al., 2026) builds a scale-wise autoregressive model upon multi-scale latent representations, enabling next-scale prediction. PolarFormer (Lyu et al., 2026)

introduces a polar coordinate decomposition technique to alleviate codebook collapse and training difficulties. However, despite their strong temporal correlation learning capability, these unidirectional autoregressive models still face dilemmas in generation stability and efficiency. In another line of research, bidirectional autoregressive models, such as SDformer-m (Chen et al., 2024) and TimeVQVAE (Lee et al., 2023), are driven by the cloze-style prediction paradigm. Compared with unidirectional autoregressive models, they offer the advantage of parallel decoding, thereby substantially improving generation efficiency (He et al., 2022). Yet, due to their weakened capability for temporal correlation modeling, they often suffer from suboptimal performance. Moving beyond these two autoregressive paradigms, HALO develops a more general masked autoregressive model. This model returns to the fundamental principle of autoregressive modeling by predicting unknown tokens conditioned on known tokens, while striking a balance between parallel decoding and temporal correlation learning to enable both efficient and high-quality time series generation.

3 PRELIMINARIES

3.1 DEFINITION

Definition 3.1 (Multivariate Time Series) For the given multivariate time series $X \in \mathbb{R}^{N \times T}$, where N represents the number of variates, and T is the number of time steps. HALO adopts a channel-independent strategy (Nie et al., 2023) for robust modeling. Accordingly, for simplicity, we omit the variate dimension in the following formulation.

Definition 3.2 (VAE) The VAE (Kingma & Welling, 2013) is widely employed to compress input data into a lower-dimensional latent vector. It consists of an encoder $q_\phi(z \mid x)$, which parameterizes the approximate posterior distribution over z. Symmetrically, it also includes a decoder $q_\psi(x \mid z)$, which reconstructs x from z. The standard VAE training objective is formulated by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\psi(\mathbf{x} \mid \mathbf{z})\right] - D_{\mathrm{KL}} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\right). \tag{1}$$

Both the prior $p(z)$ and the approximate posterior $q_\phi(z \mid x)$ are parameterized as Gaussian distributions with diagonal covariance matrices by default; the prior is the isotropic standard Normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$. During training, the reparameterization trick is typically employed, i.e., $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By decoupling the stochastic sampling from the encoder parameters, this operation enables gradients to be backpropagated through z from the decoder to the encoder. Under this diagonal Gaussian posterior, the encoder outputs data-dependent, dimension-wise scale $\sigma_\phi(x)$, leading to heterogeneous variances across both latent dimensions and latents. Moreover, such heterogeneous variances can be progressively amplified during autoregressive decoding, causing detrimental scale drift (Sun et al., 2026).

3.2 PROBLEM STATEMENT

Given a training set $\mathcal{D} = \{X^{(i)}\}_{i=1}^{D}$ consisting of D time series samples, the goal is to learn a parameterized distribution $p_\theta(X)$ that approximates the underlying data distribution $p_{\mathrm{data}}(X)$. Once trained, the model can generate synthetic time series $\hat{X} \sim p_\theta(\cdot)$. The generated time series should faithfully preserve the temporal dynamics and statistical properties of the real data.

4 METHOD

4.1 STRUCTURE OVERVIEW

HALO adopts a classical two-stage generative modeling paradigm as illustrated in Figure 3. In the **first stage**, we train an S-VAE with a symmetric encoder–decoder architecture under a reconstruction objective. First, the encoder of S-VAE compresses the real time series into continuous latent representations and models them with a von Mises–Fisher distribution, constraining the latents to a fixed-radius hypersphere and thereby alleviating variance and scale heterogeneity in continuous latents. Next, given the latent representations produced by the encoder, the decoder reconstructs the time series from them.

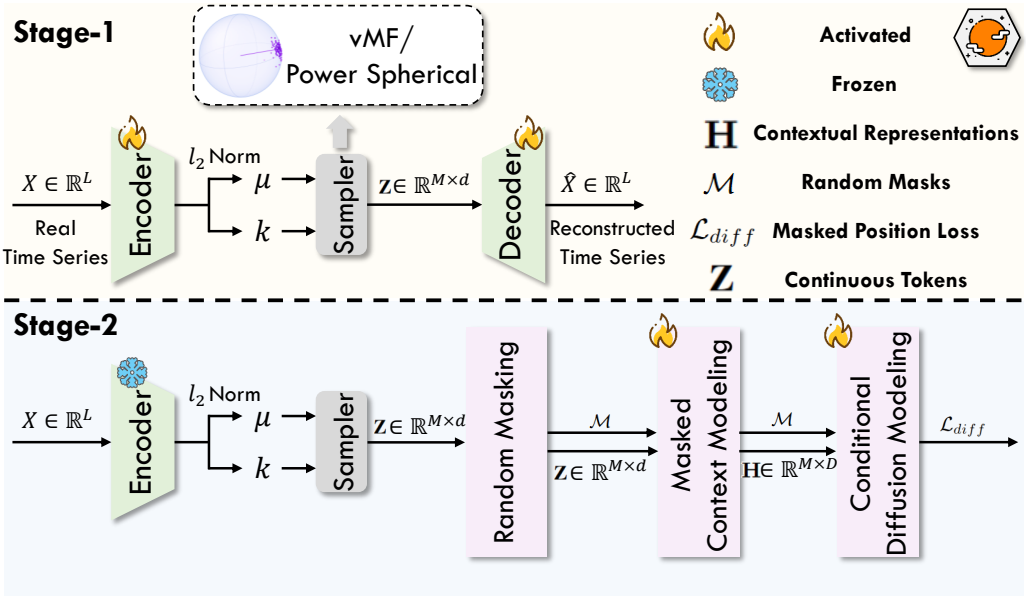


Figure 3: Overview of the HALO framework.

In the **second stage**, we train a masked autoregressive model to learn the distribution of the continuous latent representations. This model mainly includes three components: a random masking strategy, masked context modeling, and conditional diffusion modeling. The masking strategy selects prediction targets, while masked context modeling derives conditioning representations from the visible tokens. The conditional diffusion model then learns the distributions of the masked tokens conditioned on these representations. This design balances generation efficiency and temporal correlation modeling while helping mitigate error accumulation.

4.2 Stage 1: Hyperspherical VAE (S-VAE)

Encoder For the encoder, following conventional practice, we progressively compress the real time series $X \in \mathbb{R}^L$ into low-dimensional latent representations $h \in \mathbb{R}^{M \times d}$, where $M = L/s$ is the number of latents, and s represents the compression ratio. Formally, it can be expressed as follows:

$$h = \text{Encoder}(X). \tag{2}$$

For the encoder architecture, we design a hybrid structure to more effectively capture temporal characteristics. Unlike existing methods (Chen et al., 2024; Xu et al., 2026), we combine convolutional neural networks with Transformer blocks, enabling the encoder to preserve strong local feature extraction while leveraging the global modeling capability of Transformers. This complementary design enables the proposed encoder to learn more expressive latents.

von Mises–Fisher Distribution After obtaining a sequence of continuous latents h produced by the encoder, we constrain them to the unit hypersphere to mitigate heterogeneous variance in the continuous latent representations. Specifically, we first employ a linear projection layer to output a direction, and a concentration for each latent. The direction is subsequently ℓ_2-normalized to obtain the mean direction $\boldsymbol{\mu} \in \mathbb{S}^{d-1}$, while the concentration $\kappa \in \mathbb{R}_{\geq 0}$ is constrained to be non-negative. Then, based on $\boldsymbol{\mu}$ and κ, we parameterize the approximate posterior using a von Mises–Fisher distribution (Davidson et al., 2018):

$$q_\phi(\mathbf{u} \mid \mathbf{x}) = C_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}^\top \mathbf{u}\right), \quad \mathbf{u} \in \mathbb{S}^{d-1}, \tag{3}$$

where $C_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)}$ is the normalization constant, and $I_\nu(\cdot)$ denotes the modified Bessel function of the first kind of order ν. The $\boldsymbol{\mu}$ controls the preferred direction, while the concentration κ determines the degree of concentration around that direction.

After parameterizing the directional posterior with the von Mises–Fisher distribution, we sample a direction $\mathbf{u} \sim q_\phi(\mathbf{u} \mid \mathbf{x})$. We adopt a uniform prior $p(\mathbf{u}) = \mathrm{Unif}(\mathbb{S}^{d-1})$ over the hypersphere, and transform the sampled direction to the fixed-radius latent representations through $\mathbf{z} = R\mathbf{u}$. By fixing the radial component, all latent representations share the same norm, thus eliminating variance and scale heterogeneity in the continuous latents. Moreover, this geometric constraint reduces the degrees of freedom of the latent distribution, simplifying the distribution learning in the second-stage training.

Decoder Given the hyperspherical latent representations $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M]$, the decoder maps it back to the original time-series space to reconstruct the input sequence:

$$\hat{X} = \text{Decoder}(\mathbf{Z}), \quad \mathbf{z} = R\mathbf{u}. \tag{4}$$

To this end, the decoder defines the conditional likelihood $p_\psi(X \mid \mathbf{Z})$, which constitutes the ELBO of the S-VAE:

$$\mathcal{L}_{S-VAE} = \mathbb{E}_{q_\phi(\mathbf{U}|X)}[\log p_\psi(X \mid \mathbf{Z} = R\mathbf{U})] - D_{\mathrm{KL}}(q_\phi(\mathbf{U} \mid X) \| p(\mathbf{U})), \quad \mathbf{U} \in (S^{d-1})^M, \tag{5}$$

where the first term related to the reconstruction, encouraging the decoder to reconstruct the input time series from the sampled latents, and the second term regularizes the approximate posterior toward the prior distribution on the hypersphere that promoting a well-structured latent space.

Learning Objective The overall training objective includes the $\mathcal{L}_{ELBO}$. In addition, we introduce a spectral loss $\mathcal{L}_{\mathrm{FFT}}$, which computes an ℓ_1 loss in the frequency domain to encourage the S-VAE to faithfully preserve periodic and seasonal components:

$$\mathcal{L}_{Total} = -\mathcal{L}_{S-VAE} + \alpha \mathcal{L}_{FFT}, \quad \mathcal{L}_{FFT} = \|\mathcal{F}(\hat{X}) - \mathcal{F}(X)\|_1, \tag{6}$$

where α is the hyperparameter, and $\mathcal{F}(\cdot)$ represents the Fast Fourier Transform.

4.3 Stage 2: Mask Autoregressive Modeling

Random Masking Strategy To learn latent distributions, we construct masked prediction tasks over the latent sequences produced by the frozen S-VAE under varying visible contexts, following the random masking paradigm (Li et al., 2023; Chang et al., 2022). Let $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M] \in \mathbb{R}^{M \times d}$ denote a latent sequence. For each mini-batch, we sample a masking ratio r from a truncated Gaussian distribution :

$$r \sim \mathrm{TruncNormal}_{[r_{\min}, 1]}(\mu = 1, \sigma^2 = 0.25^2), \qquad m = \lceil rM \rceil, \tag{7}$$

where $r_{\min}$ controls the minimum masking ratio and m is the number of masked tokens.

Furthermore, for each sequence, we independently sample a uniformly random permutation π of the position indices $\{1, \ldots, M\}$. The masked and visible position sets are defined as

$$\mathcal{M} = \{\pi_1, \ldots, \pi_m\}, \qquad \mathcal{V} = \{1, \ldots, M\} \setminus \mathcal{M}. \tag{8}$$

This procedure uniformly selects m distinct positions for masking, while preserving the original temporal positions of the latent tokens. The visible tokens $\mathbf{Z}_\mathcal{V}$ provide contextual information, while the masked tokens $\mathbf{Z}_\mathcal{M}$ serve as prediction targets. By varying both the masking ratio and the masked positions, the model learns to predict missing latent tokens from different amounts and configurations of visible context.

Masked Context Modeling To capture temporal correlations across tokens, we employ a Transformer encoder-decoder to construct contextual representations for the masked positions. For the visible tokens $\mathbf{Z}_\mathcal{V}$, the encoder uses bidirectional self-attention to model relationships among different temporal positions, including dependencies between distant tokens. Learnable temporal positional embeddings retain the ordering information of the latent sequence, while the domain label $\mathbf{C}$ provides domain-specific prompts. Classifier-free guidance (CFG) can also be adopted to enable unconditional generation.

The decoder combines the encoded visible context with learnable mask tokens at the missing positions. Each masked position can attend to visible tokens from both earlier and later temporal

locations, allowing its contextual representation to incorporate information from the surrounding sequence as well as distant temporal context:

$$\mathbf{H} = f_\theta\left(\mathbf{Z}_\mathcal{V}, \mathcal{M}, \mathbf{C}\right), \tag{9}$$

where f_θ denotes the context modeling network, and $\mathbf{H} \in \mathbb{R}^{M\times D}$ contains the contextual representations. For each $i \in \mathcal{M}$, $\mathbf{H}_i$ summarizes the temporal context relevant to the missing latent token and serves as the condition for subsequent diffusion modeling. This enables the conditional prediction of each masked token to account for temporal correlations within the latent sequence.

Conditional Diffusion Modeling To model the distribution of each masked latent token $p_\omega(\mathbf{Z}_i \mid \mathbf{H}_i)$, we employ a lightweight conditional diffusion head, given the contextual representations $\mathbf{H}$, for $i \in \mathcal{M}$. The contextual representation $\mathbf{H}_i$ carries temporal dependencies extracted from the visible sequence, allowing the diffusion head to predict the missing latent token conditioned on its temporal context.

During training, we perturb each target latent token $\mathbf{Z}_i$ with Gaussian noise:

$$\mathbf{Z}_i^{(t)} = \sqrt{\bar{\alpha}_t}\,\mathbf{Z}_i + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

where t denotes the diffusion timestep and $\bar{\alpha}_t$ is determined by the noise schedule. A lightweight MLP $\boldsymbol{\epsilon}_\omega(\mathbf{Z}_i^{(t)}, t, \mathbf{H}_i)$ predicts the injected noise, while the timestep and contextual information are integrated into noise prediction network through AdaLN (Peebles & Xie, 2023).

We optimize the noise prediction objective over the masked positions:

$$\mathcal{L}_{diff} = \mathbb{E}\left[\frac{1}{|\mathcal{M}|d}\sum_{i\in\mathcal{M}}\left\|\boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_\omega\left(\mathbf{Z}_i^{(t)}, t, \mathbf{H}_i\right)\right\|_2^2\right], \tag{11}$$

where the expectation is taken over the training data, random masks, diffusion timesteps, and Gaussian noise. The diffusion head and the context modeling network are jointly optimized through this objective, while the S-VAE remains frozen. This formulation combines temporal context modeling with probabilistic prediction of continuous tokens. The token processing pipeline during training and autoregressive generation during inference are detailed in the Appendix A.

5 Experiments

We conduct extensive experiments to comprehensively evaluate HALO. First, we describe the experimental setup (Section 5.1). Second, we conduct in-domain and cross-domain generation experiments to evaluate the effectiveness of HALO (Section 5.2). Third, we conduct ablation studies to investigate the contribution of each component (Section 5.3). Finally, we conduct efficiency analysis experiments to demonstrate the superior generation efficiency of HALO (Section 5.4).

5.1 Experiment Settings

Datasets Our experiments are conducted on 12 datasets across four real-world domains, including: Electricity, Solar and Wind from the *energy domain*; Traffic, Taxi and Pedestrian from the *transport domain*; Air Quality, Temperature and Rain from *nature domain*; NN5, Fred-MD and Exchange from the *economic domain*. These real-world benchmarks are provided by GluonTS tool (Alexandrov et al., 2020), and Monash Time Series Forecasting Repository (Godahewa et al., 2021). For handling the datasets, we follow (Huang et al., 2025) and consider 3 different long sequence generation tasks, including: {96, 168, 336}. More details on datasets are described in the Appendix B.

Baselines We carefully select a series of high-performance deep generative models as our baselines to achieve comprehensive evaluation. These baselines include TimeGAN (Yoon et al., 2019), GT-GAN (Jeon et al., 2022), TimeVAE (Desai et al., 2021), TimeVQVAE (Lee et al., 2023), TimeVQVAE-C (class-conditional version of TimeVQVAE), TimeDP (Huang et al., 2025), and TimeMAR (Xu et al., 2026). Descriptions of the baselines are provided in the Appendix C.

Implementation Details All experiments are conducted on a single NVIDIA RTX 4090 GPU using Python 3.10.2 and PyTorch 2.9.1. We train the model (both stage) for up to 100,000 iterations with a batch size of 256 and an initial learning rate of 1×10^{-3}, using linear warm-up for the first 1,000 iterations followed by cosine decay. More technical details are provided in the Appendix D.

Evaluation Metrics To evaluate generation quality, we assess both the distributional fidelity between real and synthetic time series and the preservation of their temporal characteristics. Following this principle, we adopt three metrics: **Maximum Mean Discrepancy** (**MMD**), **Kullback–Leibler** (**KL**) divergence, and **Marginal Distribution Difference** (**MDD**). **MMD** measures the global distribution discrepancy between real and synthetic samples in a kernel-induced feature space, while **KL** divergence quantifies their distributional discrepancy at the variable level. **MMD** further evaluates marginal distribution consistency by comparing the empirical distributions of real and synthetic data at each time step. For all three metrics, lower values indicate better generation quality.

5.2 RESULTS

Evaluation of In-Domain Generation Quality We combine 12 datasets from 4 domains into a single multi-domain dataset for training, and then evaluate the proposed method against other advanced baselines. To ensure a fair comparison, we follow the experimental settings of TimeDP and conduct all experiments based on its codebase. Each run is repeated 5 times with different random seeds. Table 1 reports the mean and standard deviation of the MMD and K-L metrics for time series generation with a sequence length of 168. Results for other sequence lengths, including the corresponding MMD scores and additional evaluation metrics, are provided in the Appendix E.

Table 1: Maximum mean discrepancy (MMD) and K-L divergence (K-L) of in-domain generation for 168. Best results are highlighted in bold face and second best results are underlined.

		HALO (Ours)	TimeMAR 2026	TimeDP 2025	TimeGAN 2019	GT-GAN 2022	TimeVAE 2021	TimeVQVAE 2023	TimeVQVAE-C 2023
MMD	Electricity	**0.001** ± 0.000	**0.001** ± 0.000	0.001 ± 0.001	0.367 ± 0.255	0.254 ± 0.166	0.577 ± 0.006	0.152 ± 0.024	0.002 ± 0.001
	Solar	0.033 ± 0.000	**0.032** ± 0.000	0.041 ± 0.011	0.628 ± 0.053	0.578 ± 0.039	0.353 ± 0.014	0.437 ± 0.020	0.058 ± 0.005
	Wind	**0.007** ± 0.000	0.023 ± 0.001	0.025 ± 0.017	0.213 ± 0.017	0.170 ± 0.040	0.170 ± 0.004	0.131 ± 0.014	0.018 ± 0.007
	Traffic	0.068 ± 0.001	**0.068** ± 0.000	0.083 ± 0.034	0.567 ± 0.057	0.538 ± 0.078	0.218 ± 0.007	0.213 ± 0.016	0.089 ± 0.001
	Taxi	**0.083** ± 0.001	0.150 ± 0.002	0.095 ± 0.023	0.275 ± 0.054	0.319 ± 0.032	0.139 ± 0.007	0.128 ± 0.004	0.109 ± 0.014
	Pedestrian	0.071 ± 0.001	0.067 ± 0.001	**0.044** ± 0.020	0.090 ± 0.030	0.112 ± 0.019	0.065 ± 0.002	0.067 ± 0.007	0.058 ± 0.002
	Air	**0.010** ± 0.000	0.031 ± 0.001	0.011 ± 0.003	0.120 ± 0.045	0.211 ± 0.041	0.089 ± 0.016	0.028 ± 0.002	0.041 ± 0.008
	Temperature	0.247 ± 0.003	0.235 ± 0.005	**0.219** ± 0.022	0.926 ± 0.042	0.809 ± 0.081	1.002 ± 0.014	0.323 ± 0.008	0.259 ± 0.043
	Rain	**0.025** ± 0.000	0.074 ± 0.002	0.057 ± 0.039	0.329 ± 0.285	0.111 ± 0.109	0.292 ± 0.019	0.074 ± 0.007	0.080 ± 0.004
	NN5	0.199 ± 0.002	**0.161** ± 0.002	0.164 ± 0.010	0.874 ± 0.088	0.632 ± 0.074	0.821 ± 0.061	0.327 ± 0.012	0.243 ± 0.041
	Fred-MD	**0.001** ± 0.000	0.002 ± 0.000	0.002 ± 0.001	0.043 ± 0.021	0.133 ± 0.102	0.059 ± 0.008	0.008 ± 0.002	0.005 ± 0.002
	Exchange	0.184 ± 0.003	**0.125** ± 0.003	0.151 ± 0.024	0.530 ± 0.154	0.475 ± 0.116	0.543 ± 0.149	0.342 ± 0.050	0.233 ± 0.107
K-L	Electricity	**0.002** ± 0.000	0.012 ± 0.003	0.012 ± 0.016	0.488 ± 0.175	0.407 ± 0.079	0.734 ± 0.023	0.280 ± 0.051	0.027 ± 0.015
	Solar	0.019 ± 0.000	**0.012** ± 0.000	0.016 ± 0.005	0.612 ± 0.447	0.120 ± 0.041	0.260 ± 0.016	0.865 ± 0.108	0.234 ± 0.062
	Wind	**0.063** ± 0.002	0.107 ± 0.006	0.152 ± 0.034	1.924 ± 1.233	0.107 ± 0.016	0.484 ± 0.015	0.483 ± 0.066	0.183 ± 0.047
	Traffic	**0.004** ± 0.000	0.014 ± 0.001	0.009 ± 0.003	1.305 ± 0.320	1.409 ± 0.251	0.211 ± 0.014	0.178 ± 0.026	0.016 ± 0.003
	Taxi	**0.008** ± 0.000	0.173 ± 0.012	0.011 ± 0.004	0.650 ± 0.180	0.950 ± 0.197	0.110 ± 0.020	0.110 ± 0.026	0.038 ± 0.010
	Pedestrian	0.035 ± 0.000	0.035 ± 0.001	**0.014** ± 0.010	0.417 ± 0.181	0.411 ± 0.096	0.065 ± 0.005	0.405 ± 0.051	0.039 ± 0.008
	Air	**0.020** ± 0.001	0.070 ± 0.005	0.027 ± 0.016	0.348 ± 0.093	0.578 ± 0.049	0.164 ± 0.012	0.054 ± 0.012	0.093 ± 0.025
	Temperature	**0.158** ± 0.006	0.220 ± 0.011	0.171 ± 0.073	8.892 ± 2.681	3.174 ± 2.685	2.183 ± 0.110	0.735 ± 0.066	0.379 ± 0.110
	Rain	**0.006** ± 0.000	0.081 ± 0.002	0.013 ± 0.012	0.506 ± 0.174	0.432 ± 0.099	0.160 ± 0.022	0.047 ± 0.018	0.065 ± 0.018
	NN5	0.092 ± 0.002	0.094 ± 0.004	**0.054** ± 0.014	4.928 ± 4.112	1.386 ± 0.520	1.337 ± 0.220	1.063 ± 0.274	0.220 ± 0.151
	Fred-MD	0.228 ± 0.010	0.207 ± 0.008	**0.203** ± 0.035	0.512 ± 0.290	0.380 ± 0.070	0.346 ± 0.041	0.831 ± 0.077	1.118 ± 0.276
	Exchange	**1.742** ± 0.043	1.887 ± 0.020	1.866 ± 0.132	8.861 ± 3.397	7.201 ± 4.380	10.404 ± 1.434	5.052 ± 1.385	8.475 ± 3.056
1st Counts		**14**	6	5	0	0	0	0	0

As shown in Table 1, HALO achieves leading overall generation performance across the 12 datasets. These improvements can be attributed in part to two complementary design choices. First, continuous latent modeling avoids vector quantization, while the hyperspherical constraint stabilizes latent scales, facilitating second-stage distribution learning. Second, masked autoregressive modeling enables parallel token generation conditioned on temporal context, reducing sequential decoding steps and potentially limiting error accumulation. Together with conditional diffusion modeling, these designs enable HALO to generate time series with high distributional fidelity.

Evaluation of Cross-Domain Generation Quality Benefiting from classifier-free guidance (CFG), HALO supports unconditional generation and can be directly applied to unseen domains excluded from the multi-domain training set. Following (Huang et al., 2025), we use 3, 10, and 100 samples as both prompts and fine-tuning data for other models, with the KL divergence results for a generation length of 168 reported in Figure 4.

As shown in Figure 4, HALO achieves state-of-the-art and robust cross-domain generation performance, substantially outperforming other unconditional baselines and even fine-tuned methods. Notably, HALO also maintains a clear advantage over TimeDP, which leverages samples from unseen domains as prompts, further demonstrating its strong cross-domain generalization capability and highlighting its potential for developing large-scale foundation models for time series generation.

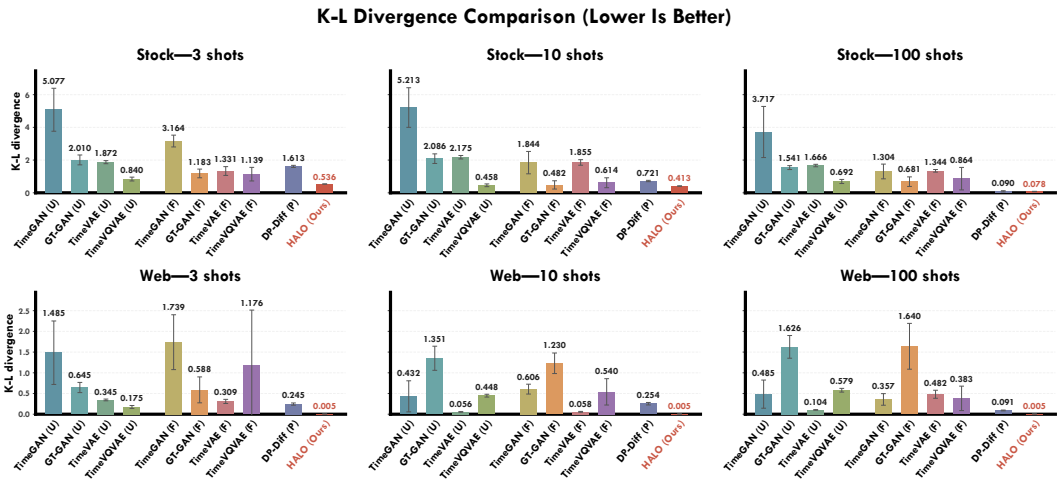


Figure 4: K-L divergence (K-L) in unseen domain settings of Stock and Web datasets for generation sequence length 168. U, F, and P denote unconditional generation, fine-tuning, and sample prompting, respectively. Results for the other metrics are provided in the Appendix E.

5.3 Ablation Studies

To assess the contribution of each component, we independently replace the S-VAE, masked autoregressive backbone, and conditional diffusion head with a standard VAE (w/ VAE), a GPT-style Transformer (w/ AR), and an MLP (w/ MLP), respectively. As shown in Table 2, all three variants underperform the full model. The degradation with w/ VAE suggests that reducing latent variance and scale heterogeneity facilitates second-stage distribution modeling. Meanwhile, w/ AR requires substantially more sequential decoding steps, which may exacerbate error accumulation and impair generation quality. Finally, the degradation with w/ MLP supports the effectiveness of conditional diffusion modeling over deterministic latent regression.

Table 2: Maximum mean discrepancy (MMD) and K-L divergence (K-L) results for ablation study on generation sequence length 168. For each evaluation metric, we report the average value across all datasets.

Metrics	HALO	w/ VAE	w/ AR	w/ MLP
MMD	0.078	0.093	0.099	0.274
K-L	0.198	0.202	0.210	1.678

5.4 Efficiency Analysis

We compare the generation efficiency of HALO, TimeMAR, and HALO (AR) for sequences of length 168. As shown in Table 3, HALO substantially outperforms TimeMAR (Xu et al., 2026), a state-of-the-art autoregressive baseline for time series generation, in inference efficiency. This advantage primarily stems from HALO's parallel decoding, which reduces the number of sequential generation steps. Another reason is that discrete quantization incurs additional computation for generation.

Table 3: Generation efficiency comparison. PGM and IS denote peak GPU memory (GB) consumption and average inference time per sample (ms), respectively.

Metrics	HALO	TimeMAR	HALO/AR
PGM	1.406	2.311	1.407
IS	2.287	8.986	5.784

6 Conclusions

In this paper, we propose HALO, a two-stage generative framework that combines continuous latent representations with efficient masked autoregressive modeling. Its hyperspherical VAE avoids discrete vector quantization and alleviates latent variance and scale heterogeneity, while its masked autoregressive model integrates temporal context and conditional diffusion to enable parallel generation with fewer sequential inference steps. Experiments across 12 datasets demonstrate strong generation quality and improved inference efficiency, with additional evaluations showing promising generalization to unseen domains. These results support continuous latent modeling as an effective approach to high-quality and efficient time series generation.

AI USE STATEMENT

Generative AI is used for grammar checking.

REPRODUCIBILITY STATEMENT

We provide a detailed description of our method in Section 4. The datasets, evaluation metrics, and experimental setup are described in Section 5. Additional implementation details, including hyperparameters, training configurations, and computational resources, are provided in Appendix B. Additional experimental analyses are provided in Appendix F. Our source code and experimental scripts are available in an anonymous repository at `https://anonymous.4open.science/r/HALO-Time-Series-Generation`.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *JMLR*, 2020.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *JMLR*, 2003.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *ICML*, 2025.

Zhicheng Chen, Shibo Feng, Zhong Zhang, Xi Xiao, Xingyu Gao, and Peilin Zhao. Sdformer: Similarity-driven discrete transformer for time series generation. *NeurIPS*, 2024.

Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. *NeurIPS*, 2023.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *TPAMI*, 2023.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Bowen Deng, Chang Xu, Hao Li, Yu-hao Huang, Min Hou, and Jiang Bian. Tardiff: Target-oriented diffusion guidance for synthetic electronic health record time series generation. In *KDD*, 2025.

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

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *AAAI*, 2024.

Shibo Feng, Zhicheng Chen, Xi Xiao, Zhong Zhang, Qing Li, Xingyu Gao, and Peilin Zhao. Msdformer: Multi-scale discrete transformer for time series generation. *TPAMI*, 2026.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *NeurIPS*, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Yu-Hao Huang, Chang Xu, Yueying Wu, Wu-Jun Li, and Jiang Bian. Timedp: Learning to generate multi-domain time series with domain prompts. In *AAAI*, 2025.

Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. In *NeurIPS*, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Hermann Kumbong, Xian Liu, Tsung-Yi Lin, Ming-Yu Liu, Xihui Liu, Ziwei Liu, Daniel Y Fu, Christopher Re, and David W Romero. Hmar: Efficient hierarchical masked auto-regressive image generation. In *CVPR*, 2025.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model. *arXiv preprint arXiv:2303.04743*, 2023.

Jaeyoung Lee, Hyuna Jeon, Uiin Kim, and Misuk Kim. Beargen: Llm-guided signal generation framework for bearing fault diagnosis. *Advanced Engineering Informatics*, 2026.

Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *CVPR*, 2023.

Wei Li, Shibo Feng, Pengcheng Wu, Xingyu Gao, Min Wu, and Peilin Zhao. Sdflow: Similarity-driven flow matching for time series generation. *arXiv preprint arXiv:2605.05736*, 2026.

Jiahong Lyu, Hongfan Gao, Wangmeng Shen, Qingsong Zhong, and Jilin Hu. Polarformer: Radial-angular latent modeling for unconditional time series generation. In *KDD*, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets. In *AAAI*, 2026.

Feng Shibo, Peilin Zhao, Liu Liu, Pengcheng Wu, and Zhiqi Shen. Hdt: Hierarchical discrete transformer for multivariate time series forecasting. In *AAAI*, 2025.

Yutao Sun, Hangbo Bao, Wenhui Wang, Zhiliang Peng, Li Dong, Shaohan Huang, Jianyong Wang, and Furu Wei. Multimodal latent language modeling with next-token diffusion. In *ICML*, 2026.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017.

Wenjie Wang, Pengfei Tang, Jian Lou, Yuanming Shao, Lance Waller, Yi-an Ko, and Li Xiong. Igamt: privacy-preserving electronic health record synthesization with heterogeneity and irregularity. In *AAAI*, 2024.

Haochong Xia, Shuo Sun, Xinrun Wang, and Bo An. Market-gan: Adding control to financial market data generation with semantic context. In *AAAI*, 2024.

Xiangyu Xu, Qingsong Zhong, and Jilin Hu. Timemar: Multi-scale autoregressive modeling for unconditional time series generation. In *WWW*, 2026.

Yi Xu, Zhigang Chen, Rui Wang, Yangfan Li, Fengxiao Tang, Ming Zhao, and Jiaqi Liu. Faultdiffusion: Few-shot fault time series generation with diffusion model. *arXiv preprint arXiv:2511.15174*, 2025.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, 2019.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 2023.

Xu Zhang, Chang Xu, Hao Li, Yuhao Huang, Qiushui Xu, Yuxuan Liang, Chenghao Liu, Ming Jin, Qingsong Wen, Peng Wang, et al. A survey on data generation for time series: Taxonomy, review and prospects. 2026.

A TOKEN PROCESSING PATH IN MASKED CONTEXT MODELING

Training Figure 5 illustrates the complete token processing pipeline in masked autoregressive modeling, during training phase. The encoder processes only the unmasked visible tokens, while the decoder jointly processes the encoded visible tokens and the learnable mask tokens. Consequently, each mask token is conditioned on information propagated from the visible tokens, including their temporal positional information. By conditioning the denoising of ground-truth masked tokens on mask tokens enriched with visible-token information, including temporal positional information, we effectively realize masked autoregressive modeling that *predicts unknown tokens conditioned on known tokens*.

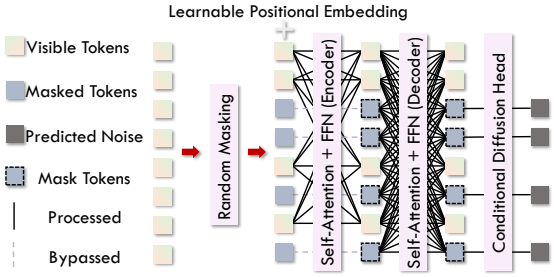


Figure 5: Illustration of the complete token processing flow in masked autoregressive modeling.

Algorithm 1 Inference Procedure of HALO

Require: Trained context network f_θ, noise predictor ϵ_ω, and S-VAE decoder $\mathrm{Decoder}$
Require: Number of latent tokens M, latent dimension d, and radius $R = \sqrt{d}$
Require: Autoregressive iterations $1 \le K \le M$, diffusion sampling steps $S = 100$, and sampling schedule $\{\tau_t\}_{t=1}^{S}$
Require: Optional domain label C ($C = \varnothing$ for the learned null condition)
Ensure: Generated time series $\widehat{X}$

Initialize $Z = [z_1, \ldots, z_M] \leftarrow \mathbf{0}_{M \times d}$
Initialize $\mathcal{M} \leftarrow \{1, \ldots, M\}$ and $\mathcal{V} \leftarrow \varnothing$
Sample a random permutation $\pi = (\pi_1, \ldots, \pi_M)$ of $\{1, \ldots, M\}$
for $k = 1, \ldots, K$ **do**
  if $k = K$ **then**
    $m_k \leftarrow 0$ ▷ Generate all remaining tokens
  else
    $m_k \leftarrow \lfloor M \cos\left(\frac{\pi k}{2K}\right) \rfloor$
    $m_k \leftarrow \max\{1, \min\{|\mathcal{M}| - 1, m_k\}\}$
  end if
  $\mathcal{M}_{\text{next}} \leftarrow \{\pi_j : 1 \le j \le m_k\}$
  $\mathcal{P}_k \leftarrow \mathcal{M} \setminus \mathcal{M}_{\text{next}}$ ▷ Positions generated in this iteration
  $H \leftarrow f_\theta(Z_{\mathcal{V}}, \mathcal{M}, C)$ ▷ Context from previously generated tokens
  for all $i \in \mathcal{P}_k$ **in parallel do**
    Sample $z_i^{(S)} \sim \mathcal{N}(0, I_d)$
    for $t = S, S-1, \ldots, 1$ **do**
      $\widehat{\epsilon}_i^{(t)} \leftarrow \epsilon_\omega(z_i^{(t)}, \tau_t, H_i)$
      $z_i^{(t-1)} \leftarrow \textsc{ReverseStep}(z_i^{(t)}, \widehat{\epsilon}_i^{(t)}, t)$
    end for
    $z_i \leftarrow R\, z_i^{(0)} / \|z_i^{(0)}\|_2$ ▷ Fixed-radius projection
  end for
  $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{P}_k$
  $\mathcal{M} \leftarrow \mathcal{M}_{\text{next}}$
end for
$\widehat{X} \leftarrow \mathrm{Decoder}(Z)$
return $\widehat{X}$

Inference Algorithm 1 summarizes the inference procedure of HALO. Starting from a fully masked latent sequence, HALO progressively generates groups of tokens according to a randomly sampled order and a cosine unmasking schedule. At each iteration, the context network conditions on all previously generated tokens, and the conditional diffusion head samples the selected tokens in parallel. Here, τ_t denotes the diffusion timestep associated with sampling step t, and $\textsc{ReverseStep}$ performs one DDPM reverse update under the selected sampling schedule, without injecting additional noise at the final step. Each completed token is projected onto the fixed-radius hypersphere before being incorporated into the visible context. Finally, the S-VAE decoder maps the completed

latent sequence back to the time series. For unconditional generation, the domain label is replaced with the learned null condition. For the comparative experiments, the number of inference steps is set to $M/6$.

B Dataset Description

We provide detailed descriptions into the datasets used in model training this paper:

- **Electricity**. This dataset contains hourly electricity consumption measurements, in kilowatts (kW), from 321 clients over the period from 2012 to 2014. The data were originally collected from the UCI Machine Learning Repository.
- **Solar**. This dataset contains 137 time series recording hourly solar power production in the state of Alabama, USA, in 2006.
- **Wind**. This dataset consists of a single long time series of wind power production, measured in megawatts (MW) at 4-second intervals starting from August 1, 2019. The data were obtained from the online platform of the Australian Energy Market Operator (AEMO).
- **Traffic**. This dataset contains 15 months of daily traffic data, comprising 440 daily records. It describes the occupancy rates, ranging from 0 to 1, of different freeway lanes in the San Francisco Bay Area over time.
- **Taxi**. This dataset contains spatio-temporal traffic time series of New York City taxi rides collected from 1,214 locations at 30-minute intervals during January 2015 and January 2016.
- **Pedestrian**. This dataset contains hourly pedestrian counts collected by 66 sensors across Melbourne, Australia, starting from May 2009. The original dataset is continuously updated as new observations become available. The version used in our experiments contains records up to April 30, 2020.
- **Air Quality**. This dataset was originally used in the KDD Cup 2018 forecasting competition. It contains hourly air-quality measurements collected from 59 monitoring stations in Beijing (35 stations) and London (24 stations) between January 1, 2017 and March 31, 2018. The measurements include PM2.5, PM10, NO_2, CO, O_3, and SO_2, resulting in 270 hourly time series categorized by city, station, and measurement type. The original data contain missing values. Leading missing values are filled with zeros, while the remaining missing values are imputed using the last observation carried forward (LOCF) strategy.
- **Temperature**. This dataset contains 32,072 daily records of temperature observations and rainfall forecasts collected by the Australian Bureau of Meteorology from 422 weather stations across Australia between May 2, 2015 and April 26, 2017. Missing values in the original dataset are replaced with zeros. In our experiments, we use the mean temperature variable.
- **Rain**. This dataset is derived from the same source as the Temperature dataset. We extract the rainfall variable to construct the corresponding time series.
- **NN5**. This dataset was originally used in the NN5 forecasting competition and contains 111 time series from the banking domain. Each series records daily cash withdrawals from an automated teller machine (ATM) in the United Kingdom. Missing values are imputed using the median value computed over the corresponding day of the week across the entire series.
- **FRED-MD**. This dataset contains 107 monthly time series of macroeconomic indicators provided by the Federal Reserve Bank and obtained from the FRED-MD database. Following common practice in the literature, the series are appropriately differenced and log-transformed.
- **Exchange**. This dataset contains daily exchange rates among eight currencies.

Moreover, we also describe the dataset used in unseen data generation experiment:

- **Stock**. This dataset contains the daily stock prices of Google (GOOG), which is listed on the NASDAQ stock exchange.
- **Web**. This dataset was originally used in the Kaggle Wikipedia Web Traffic Forecasting competition. It contains 145,063 daily time series recording the number of page views for a collection of Wikipedia pages from July 1, 2015 to September 10, 2017. Missing values in the original dataset are replaced with zeros.

C Baseline Details

We compare HALO with the following time series generation baselines, covering adversarial, variational, diffusion-based, and discrete-token generative approaches. Table 4 summarizes the code repositories of the baseline methods.

Table 4: Code repositories of the baseline methods. TimeVQVAE-C shares the implementation of TimeVQVAE.

Baseline	Code Repository
TimeMAR	https://github.com/decisionintelligence/TimeMAR
TimeDP	https://github.com/microsoft/TimeCraft/tree/main/TimeDP
TimeGAN	https://github.com/jsyoon0823/TimeGAN
GT-GAN	https://github.com/Jinsung-Jeon/GT-GAN
TimeVAE	https://github.com/abudesai/timeVAE
TimeVQVAE	https://github.com/ML4ITS/TimeVQVAE
TimeVQVAE-C	https://github.com/ML4ITS/TimeVQVAE
HALO (Ours)	https://anonymous.4open.science/r/HALO-Time-Series-Generation

TimeMAR (Xu et al., 2026). TimeMAR is a two-stage generative framework that combines multi-scale discrete representations with autoregressive modeling. It employs a dual-path VQ-VAE to encode decomposed trend and seasonal components into discrete tokens at multiple temporal resolutions, followed by a Transformer that generates these tokens in a coarse-to-fine manner. Coarse seasonal representations further guide the reconstruction of fine-grained seasonal patterns. Following the experimental protocol of (Huang et al., 2025), we adapt it to use domain labels as prompts.

TimeDP (Huang et al., 2025). TimeDP is a diffusion-based framework for multi-domain time series generation. It learns a shared set of semantic prototypes to represent temporal patterns and employs a prototype assignment module to construct domain prompts from input time series. These prompts condition the diffusion model, enabling domain-specific generation using a small set of examples from the target domain.

TimeGAN (Yoon et al., 2019). TimeGAN combines adversarial learning with a supervised temporal objective in a learned embedding space. Its embedding and recovery networks provide latent representations for generation, while the supervised objective encourages the model to preserve temporal dependencies between consecutive latent states.

GT-GAN (Jeon et al., 2022). GT-GAN is an adversarial framework designed to synthesize both regularly and irregularly sampled time series. It integrates neural ordinary and controlled differential equations with continuous-time flow processes to capture temporal dynamics under different sampling patterns.

TimeVAE (Desai et al., 2021). TimeVAE is a variational autoencoder for multivariate time series generation. It combines continuous latent representations with an interpretable decoder that explicitly models level, trend, and seasonal components, allowing generated sequences to reflect these temporal structures.

TimeVQVAE (Lee et al., 2023). TimeVQVAE adopts a two-stage framework based on vector quantization and bidirectional prior modeling. It separates time-frequency representations into low- and high-frequency components and encodes them into discrete tokens. Bidirectional Transformer priors learn the token distributions through masked modeling, enabling generation via iterative masked decoding.

TimeVQVAE-C (Lee et al., 2023; Huang et al., 2025). TimeVQVAE-C denotes the class-conditional variant of TimeVQVAE. It retains the time-frequency tokenization framework while conditioning the bidirectional token prior on class labels, enabling the generation of time series associated with a specified class.

D More Implementation Details

We provide additional implementation details for the S-VAE (Stage-1) and masked autoregressive model (Stage-2).

S-VAE Our S-VAE adopts a hybrid convolutional–Transformer encoder–decoder architecture. The encoder contains two residual downsampling blocks followed by three Transformer blocks, while the decoder contains three Transformer blocks followed by two residual upsampling blocks. Each Transformer block uses a hidden dimension of 512 and eight attention heads. The encoder predicts the parameters of a Power Spherical posterior over 16-dimensional (d) latent tokens, which are constrained to a hypersphere of radius 4 ($\sqrt{d}$). The temporal downsampling factor is set to 4 (s).

Masked Autoregressive Model Our masked autoregressive model consists of a random masking strategy, a Transformer-based context modeling network, and a conditional diffusion head. The context encoder and decoder each contain two Transformer blocks with a hidden dimension of 256 (D) and 8 attention heads. Learnable positional embeddings encode temporal positions, while masked positions are represented by learnable mask tokens in the decoder. The conditional diffusion head employs two residual MLP blocks with a hidden dimension of 512, conditioned on contextual representations and diffusion timesteps through adaptive layer normalization. The maximum number of diffusion sampling steps is set to 100 (t). The minimum masking ratio is set to 0.5 (r).

E Complete Experimental Results

Table 5 reports the MDD results corresponding to Table 1 in the main text. In addition, the MMD and MDD results for the cross-domain generation experiments are presented in Figures 6, respectively. Results for the remaining generation sequence lengths are provided in Tables 6 to 7.

Table 5: Marginal distribution distance (MDD) of in-domain generation for 168. Best results are highlighted in bold face and second best results are underlined.

		HALO (Ours)	TimeMAR 2026	TimeDP 2025	TimeGAN 2019	GT-GAN 2022	TimeVAE 2021	TimeVQVAE 2023	TimeVQVAE-C 2023
MDD	Electricity	**0.002** ± 0.000	0.003 ± 0.000	0.005 ± 0.002	0.075 ± 0.035	0.047 ± 0.008	0.098 ± 0.003	0.067 ± 0.004	0.005 ± 0.001
	Solar	63.027 ± 0.192	39.173 ± 0.226	56.414 ± 21.890	70.334 ± 11.928	83.855 ± 3.100	**16.721** ± 0.041	57.401 ± 0.041	59.043 ± 0.440
	Wind	**0.058** ± 0.002	0.072 ± 0.002	0.084 ± 0.009	0.226 ± 0.061	0.138 ± 0.015	0.201 ± 0.004	0.159 ± 0.011	0.085 ± 0.008
	Traffic	0.051 ± 0.000	**0.048** ± 0.001	0.049 ± 0.004	0.149 ± 0.018	0.153 ± 0.001	0.110 ± 0.001	0.119 ± 0.005	0.053 ± 0.002
	Taxi	**0.080** ± 0.001	0.095 ± 0.000	0.081 ± 0.008	0.104 ± 0.010	0.109 ± 0.004	0.094 ± 0.001	0.096 ± 0.004	0.084 ± 0.004
	Pedestrian	0.089 ± 0.000	0.088 ± 0.001	**0.071** ± 0.012	0.096 ± 0.024	0.097 ± 0.006	0.086 ± 0.002	0.143 ± 0.007	0.079 ± 0.002
	Air	**0.038** ± 0.001	0.046 ± 0.001	0.042 ± 0.002	0.139 ± 0.025	0.171 ± 0.011	0.085 ± 0.001	0.092 ± 0.008	0.058 ± 0.006
	Temperature	**0.142** ± 0.001	0.143 ± 0.002	**0.142** ± 0.010	0.189 ± 0.009	0.208 ± 0.012	0.259 ± 0.004	0.191 ± 0.004	0.156 ± 0.012
	Rain	0.079 ± 0.000	**0.061** ± 0.000	0.067 ± 0.015	0.228 ± 0.090	0.142 ± 0.020	0.228 ± 0.008	0.177 ± 0.016	0.104 ± 0.006
	NN5	0.150 ± 0.001	**0.134** ± 0.000	0.140 ± 0.005	0.295 ± 0.021	0.240 ± 0.014	0.304 ± 0.010	0.220 ± 0.012	0.175 ± 0.011
	Fred-MD	0.022 ± 0.000	**0.018** ± 0.000	0.021 ± 0.002	0.079 ± 0.016	0.098 ± 0.024	0.104 ± 0.009	0.126 ± 0.009	0.076 ± 0.025
	Exchange	0.358 ± 0.001	**0.326** ± 0.003	0.358 ± 0.010	0.351 ± 0.098	0.345 ± 0.018	0.442 ± 0.020	0.515 ± 0.021	0.486 ± 0.040

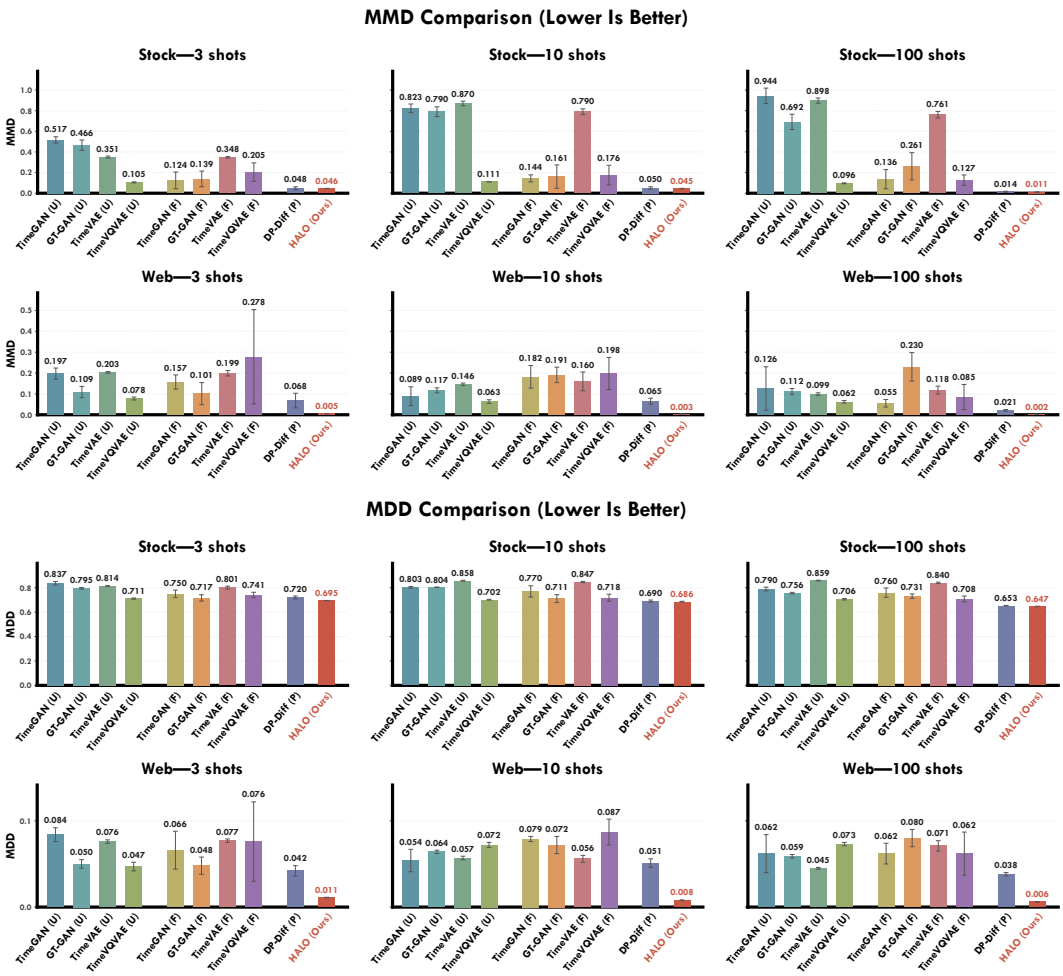


Figure 6: Maximum mean discrepancy (MMD) and Marginal distribution distance (MDD) in unseen domain settings of Stock and Web datasets for generation sequence length 168. U, F, and P denote unconditional generation, fine-tuning, and sample prompting, respectively.

Table 6: Maximum mean discrepancy (MMD), K-L divergence (K-L), and marginal distribution distance (MDD) of in-domain generation for 96. Best results are highlighted in bold face and second best results are underlined.

		HALO (Ours)	TimeMAR 2026	TimeDP 2025	TimeGAN 2019	GT-GAN 2022	TimeVAE 2021	TimeVQVAE 2023	TimeVQVAE-C 2023
MMD	Electricity	**0.001** ± 0.000	**0.001** ± 0.000	0.002 ± 0.001	0.401 ± 0.219	0.317 ± 0.311	0.648 ± 0.012	0.153 ± 0.023	0.002 ± 0.000
	Solar	0.017 ± 0.000	**0.012** ± 0.000	0.019 ± 0.001	0.584 ± 0.056	0.634 ± 0.119	0.369 ± 0.012	0.439 ± 0.021	0.035 ± 0.008
	Wind	0.041 ± 0.005	0.031 ± 0.002	0.029 ± 0.015	0.195 ± 0.030	0.217 ± 0.134	0.179 ± 0.002	0.129 ± 0.016	**0.013** ± 0.007
	Traffic	0.019 ± 0.000	0.023 ± 0.000	**0.018** ± 0.005	0.456 ± 0.100	0.513 ± 0.073	0.162 ± 0.005	0.189 ± 0.014	0.027 ± 0.002
	Taxi	**0.053** ± 0.000	0.060 ± 0.001	0.055 ± 0.007	0.197 ± 0.057	0.328 ± 0.071	0.060 ± 0.002	0.089 ± 0.005	0.061 ± 0.002
	Pedestrian	0.064 ± 0.001	0.056 ± 0.001	**0.040** ± 0.010	0.068 ± 0.025	0.159 ± 0.149	0.064 ± 0.001	0.066 ± 0.011	0.049 ± 0.005
	Air	**0.008** ± 0.000	**0.008** ± 0.000	0.010 ± 0.006	0.103 ± 0.046	0.244 ± 0.160	0.085 ± 0.011	0.022 ± 0.003	0.036 ± 0.007
	Temperature	**0.157** ± 0.003	0.177 ± 0.005	0.178 ± 0.047	0.827 ± 0.092	0.859 ± 0.109	0.958 ± 0.012	0.413 ± 0.034	0.205 ± 0.033
	Rain	**0.023** ± 0.001	0.056 ± 0.001	0.044 ± 0.027	0.285 ± 0.199	0.166 ± 0.203	0.327 ± 0.020	0.068 ± 0.008	0.072 ± 0.001
	NN5	0.213 ± 0.001	**0.212** ± 0.002	0.234 ± 0.047	0.888 ± 0.091	0.706 ± 0.116	0.875 ± 0.053	0.321 ± 0.008	0.241 ± 0.011
	Fred-MD	0.002 ± 0.000	**0.001** ± 0.000	0.005 ± 0.002	0.055 ± 0.022	0.150 ± 0.184	0.046 ± 0.013	0.004 ± 0.001	0.006 ± 0.001
	Exchange	**0.117** ± 0.004	0.141 ± 0.003	0.343 ± 0.210	0.449 ± 0.018	0.567 ± 0.114	0.592 ± 0.124	0.337 ± 0.056	0.224 ± 0.028
K-L	Electricity	**0.002** ± 0.001	0.026 ± 0.004	0.015 ± 0.017	0.552 ± 0.241	0.447 ± 0.147	0.790 ± 0.018	0.245 ± 0.039	0.031 ± 0.005
	Solar	0.021 ± 0.002	0.014 ± 0.000	**0.010** ± 0.002	0.922 ± 0.953	0.117 ± 0.047	0.288 ± 0.015	0.812 ± 0.077	0.151 ± 0.114
	Wind	0.166 ± 0.009	**0.122** ± 0.005	0.182 ± 0.029	2.288 ± 1.135	0.164 ± 0.071	0.527 ± 0.009	0.510 ± 0.112	0.284 ± 0.115
	Traffic	**0.004** ± 0.000	0.019 ± 0.000	0.010 ± 0.006	1.241 ± 0.757	1.198 ± 0.340	0.205 ± 0.008	0.192 ± 0.016	0.014 ± 0.004
	Taxi	**0.004** ± 0.001	0.045 ± 0.002	0.010 ± 0.007	0.550 ± 0.333	0.801 ± 0.270	0.121 ± 0.003	0.102 ± 0.017	0.027 ± 0.007
	Pedestrian	0.025 ± 0.000	0.040 ± 0.003	**0.012** ± 0.005	0.458 ± 0.368	0.327 ± 0.148	0.074 ± 0.004	0.373 ± 0.027	0.034 ± 0.006
	Air	**0.021** ± 0.002	0.022 ± 0.001	0.025 ± 0.011	0.317 ± 0.102	0.510 ± 0.147	0.168 ± 0.012	0.061 ± 0.012	0.096 ± 0.019
	Temperature	**0.301** ± 0.007	0.388 ± 0.013	0.392 ± 0.097	11.298 ± 1.212	5.830 ± 4.888	1.737 ± 0.068	1.135 ± 0.110	0.567 ± 0.050
	Rain	0.028 ± 0.001	0.096 ± 0.002	**0.022** ± 0.009	0.549 ± 0.257	0.481 ± 0.146	0.237 ± 0.025	0.055 ± 0.021	0.091 ± 0.010
	NN5	**0.067** ± 0.003	0.071 ± 0.002	0.069 ± 0.033	5.829 ± 6.120	2.709 ± 2.080	1.537 ± 0.224	1.038 ± 0.168	0.103 ± 0.072
	Fred-MD	**0.288** ± 0.002	0.328 ± 0.008	1.183 ± 0.280	0.628 ± 0.335	0.486 ± 0.108	0.441 ± 0.063	0.813 ± 0.093	1.240 ± 0.232
	Exchange	**1.911** ± 0.115	2.660 ± 0.116	14.336 ± 2.652	6.445 ± 2.210	10.170 ± 6.381	9.937 ± 4.322	3.740 ± 1.261	4.408 ± 0.700
MDD	Electricity	**0.002** ± 0.000	0.003 ± 0.000	0.005 ± 0.002	0.070 ± 0.020	0.046 ± 0.011	0.082 ± 0.002	0.058 ± 0.002	0.005 ± 0.001
	Solar	64.900 ± 0.081	72.076 ± 0.169	77.545 ± 18.384	60.819 ± 9.001	77.198 ± 5.522	**47.123** ± 0.114	50.619 ± 0.099	53.220 ± 0.832
	Wind	0.085 ± 0.003	0.076 ± 0.003	0.089 ± 0.012	0.207 ± 0.054	0.135 ± 0.035	0.208 ± 0.002	0.159 ± 0.009	**0.069** ± 0.014
	Traffic	0.023 ± 0.000	**0.021** ± 0.000	0.024 ± 0.003	0.109 ± 0.018	0.122 ± 0.009	0.083 ± 0.001	0.089 ± 0.002	0.028 ± 0.002
	Taxi	0.043 ± 0.000	0.054 ± 0.000	0.043 ± 0.004	0.072 ± 0.008	0.092 ± 0.010	0.064 ± 0.001	0.070 ± 0.003	**0.040** ± 0.002
	Pedestrian	0.069 ± 0.000	0.065 ± 0.001	**0.058** ± 0.007	0.074 ± 0.018	0.088 ± 0.025	0.073 ± 0.001	0.116 ± 0.003	**0.058** ± 0.003
	Air	0.028 ± 0.001	**0.024** ± 0.001	0.031 ± 0.007	0.091 ± 0.021	0.145 ± 0.022	0.070 ± 0.002	0.066 ± 0.001	0.056 ± 0.007
	Temperature	**0.117** ± 0.001	0.127 ± 0.002	0.135 ± 0.016	0.154 ± 0.004	0.185 ± 0.007	0.196 ± 0.001	0.198 ± 0.006	0.143 ± 0.010
	Rain	0.040 ± 0.000	**0.033** ± 0.000	0.039 ± 0.006	0.192 ± 0.059	0.113 ± 0.011	0.196 ± 0.003	0.134 ± 0.009	0.047 ± 0.012
	NN5	**0.148** ± 0.001	0.150 ± 0.000	0.175 ± 0.016	0.270 ± 0.014	0.225 ± 0.017	0.268 ± 0.006	0.196 ± 0.007	0.158 ± 0.004
	Fred-MD	**0.014** ± 0.000	**0.014** ± 0.000	0.023 ± 0.005	0.063 ± 0.009	0.076 ± 0.023	0.055 ± 0.012	0.083 ± 0.004	0.045 ± 0.017
	Exchange	**0.312** ± 0.003	0.340 ± 0.002	0.402 ± 0.083	0.342 ± 0.041	0.334 ± 0.024	0.425 ± 0.027	0.503 ± 0.032	0.426 ± 0.017

Table 7: Maximum mean discrepancy (MMD), K-L divergence (K-L), and marginal distribution distance (MDD) of in-domain generation for 336. Best results are highlighted in bold face and second best results are underlined. TimeMAR could not be evaluated due to out-of-memory (OOM).

		HALO (Ours)	TimeMAR 2026	TimeDP 2025	TimeGAN 2019	GT-GAN 2022	TimeVAE 2021	TimeVQVAE 2023	TimeVQVAE-C 2023
MMD	Electricity	**0.000** ± 0.000	OOM	0.001 ± 0.002	0.320 ± 0.197	0.347 ± 0.524	0.504 ± 0.012	0.135 ± 0.018	0.003 ± 0.000
	Solar	**0.066** ± 0.001	OOM	0.074 ± 0.012	0.646 ± 0.032	0.735 ± 0.045	0.400 ± 0.015	0.493 ± 0.024	0.081 ± 0.006
	Wind	0.027 ± 0.001	OOM	0.024 ± 0.003	0.208 ± 0.095	0.221 ± 0.010	0.145 ± 0.005	0.139 ± 0.015	**0.014** ± 0.007
	Traffic	0.081 ± 0.001	OOM	**0.077** ± 0.009	0.518 ± 0.028	0.704 ± 0.101	0.235 ± 0.011	0.233 ± 0.017	0.091 ± 0.005
	Taxi	0.214 ± 0.002	OOM	0.189 ± 0.034	0.329 ± 0.011	0.390 ± 0.078	**0.166** ± 0.002	0.202 ± 0.004	0.189 ± 0.007
	Pedestrian	0.055 ± 0.001	OOM	**0.035** ± 0.008	0.140 ± 0.067	0.158 ± 0.019	0.061 ± 0.003	0.074 ± 0.007	0.067 ± 0.008
	Air	**0.016** ± 0.000	OOM	0.018 ± 0.005	0.213 ± 0.037	0.261 ± 0.062	0.075 ± 0.005	0.038 ± 0.003	0.053 ± 0.007
	Temperature	0.199 ± 0.002	OOM	**0.185** ± 0.021	0.914 ± 0.031	0.934 ± 0.051	0.976 ± 0.022	0.318 ± 0.008	0.197 ± 0.014
	Rain	**0.035** ± 0.000	OOM	0.096 ± 0.086	0.151 ± 0.108	0.198 ± 0.295	0.135 ± 0.018	0.066 ± 0.002	0.081 ± 0.005
	NN5	**0.187** ± 0.001	OOM	0.191 ± 0.016	0.663 ± 0.027	0.951 ± 0.069	0.806 ± 0.086	0.333 ± 0.009	0.260 ± 0.049
	Fred-MD	**0.003** ± 0.000	OOM	0.006 ± 0.004	0.156 ± 0.097	0.063 ± 0.006	0.052 ± 0.020	0.010 ± 0.006	0.007 ± 0.001
	Exchange	**0.126** ± 0.003	OOM	0.683 ± 0.217	0.724 ± 0.130	0.553 ± 0.238	0.598 ± 0.166	0.318 ± 0.077	0.275 ± 0.198
K-L	Electricity	**0.005** ± 0.001	OOM	0.016 ± 0.016	0.396 ± 0.108	1.169 ± 2.029	0.719 ± 0.020	0.257 ± 0.023	0.042 ± 0.033
	Solar	**0.016** ± 0.001	OOM	0.017 ± 0.006	0.179 ± 0.062	2.150 ± 2.179	0.249 ± 0.012	0.870 ± 0.110	0.201 ± 0.017
	Wind	**0.098** ± 0.005	OOM	0.158 ± 0.032	0.169 ± 0.064	4.264 ± 2.790	0.373 ± 0.019	0.495 ± 0.043	0.121 ± 0.037
	Traffic	**0.004** ± 0.000	OOM	0.008 ± 0.004	1.443 ± 0.354	2.992 ± 0.592	0.241 ± 0.012	0.210 ± 0.034	0.023 ± 0.006
	Taxi	0.107 ± 0.001	OOM	**0.076** ± 0.034	0.937 ± 0.151	1.461 ± 0.367	0.160 ± 0.015	0.230 ± 0.026	0.094 ± 0.007
	Pedestrian	0.014 ± 0.000	OOM	**0.009** ± 0.003	0.452 ± 0.131	1.544 ± 0.908	0.044 ± 0.007	0.407 ± 0.069	0.053 ± 0.017
	Air	**0.009** ± 0.001	OOM	0.017 ± 0.010	1.031 ± 0.145	1.481 ± 0.553	0.180 ± 0.015	0.121 ± 0.028	0.079 ± 0.016
	Temperature	**0.024** ± 0.001	OOM	0.037 ± 0.016	2.479 ± 1.867	9.139 ± 2.247	1.726 ± 0.122	1.119 ± 0.179	0.224 ± 0.123
	Rain	**0.008** ± 0.000	OOM	0.011 ± 0.002	0.388 ± 0.112	0.271 ± 0.046	0.028 ± 0.005	0.025 ± 0.013	0.028 ± 0.005
	NN5	**0.046** ± 0.002	OOM	0.075 ± 0.034	1.418 ± 0.122	8.449 ± 5.473	1.233 ± 0.256	0.836 ± 0.250	0.180 ± 0.181
	Fred-MD	**0.203** ± 0.012	OOM	0.815 ± 0.630	0.309 ± 0.145	0.568 ± 0.321	0.212 ± 0.047	0.586 ± 0.161	1.099 ± 0.020
	Exchange	**1.971** ± 0.063	OOM	18.426 ± 4.493	14.116 ± 3.798	13.875 ± 2.965	10.670 ± 5.335	7.755 ± 2.566	7.651 ± 3.841
MDD	Electricity	**0.003** ± 0.000	OOM	0.006 ± 0.004	0.048 ± 0.009	0.061 ± 0.058	0.099 ± 0.002	0.069 ± 0.003	0.008 ± 0.003
	Solar	40.105 ± 0.113	OOM	49.170 ± 21.121	82.860 ± 4.163	74.403 ± 18.119	**13.755** ± 0.077	54.065 ± 0.183	55.878 ± 0.249
	Wind	**0.068** ± 0.003	OOM	0.091 ± 0.007	0.146 ± 0.028	0.251 ± 0.060	0.172 ± 0.004	0.168 ± 0.009	0.070 ± 0.009
	Traffic	**0.059** ± 0.000	OOM	0.066 ± 0.002	0.191 ± 0.005	0.206 ± 0.031	0.134 ± 0.001	0.158 ± 0.005	0.072 ± 0.005
	Taxi	0.684 ± 0.001	OOM	**0.680** ± 0.021	1.073 ± 0.162	0.986 ± 0.126	0.764 ± 0.003	0.750 ± 0.003	0.707 ± 0.001
	Pedestrian	0.090 ± 0.001	OOM	**0.074** ± 0.006	0.121 ± 0.027	0.153 ± 0.070	0.086 ± 0.002	0.161 ± 0.007	0.098 ± 0.006
	Air	**0.042** ± 0.000	OOM	0.049 ± 0.004	0.157 ± 0.002	0.166 ± 0.039	0.078 ± 0.002	0.099 ± 0.005	0.056 ± 0.005
	Temperature	**0.140** ± 0.001	OOM	0.153 ± 0.010	0.230 ± 0.007	0.218 ± 0.013	0.341 ± 0.005	0.204 ± 0.009	0.155 ± 0.006
	Rain	0.108 ± 0.000	OOM	**0.100** ± 0.007	0.144 ± 0.011	0.238 ± 0.184	0.210 ± 0.014	0.222 ± 0.009	0.138 ± 0.014
	NN5	**1.033** ± 0.003	OOM	1.079 ± 0.016	1.128 ± 0.017	1.199 ± 0.024	1.209 ± 0.021	1.109 ± 0.006	1.074 ± 0.012
	Fred-MD	**0.025** ± 0.000	OOM	0.031 ± 0.007	0.106 ± 0.038	0.085 ± 0.006	0.093 ± 0.035	0.128 ± 0.016	0.060 ± 0.025
	Exchange	0.386 ± 0.001	OOM	0.510 ± 0.131	**0.337** ± 0.023	0.383 ± 0.091	0.443 ± 0.027	0.537 ± 0.036	0.452 ± 0.048

F ADDITIONAL EXPERIMENTS

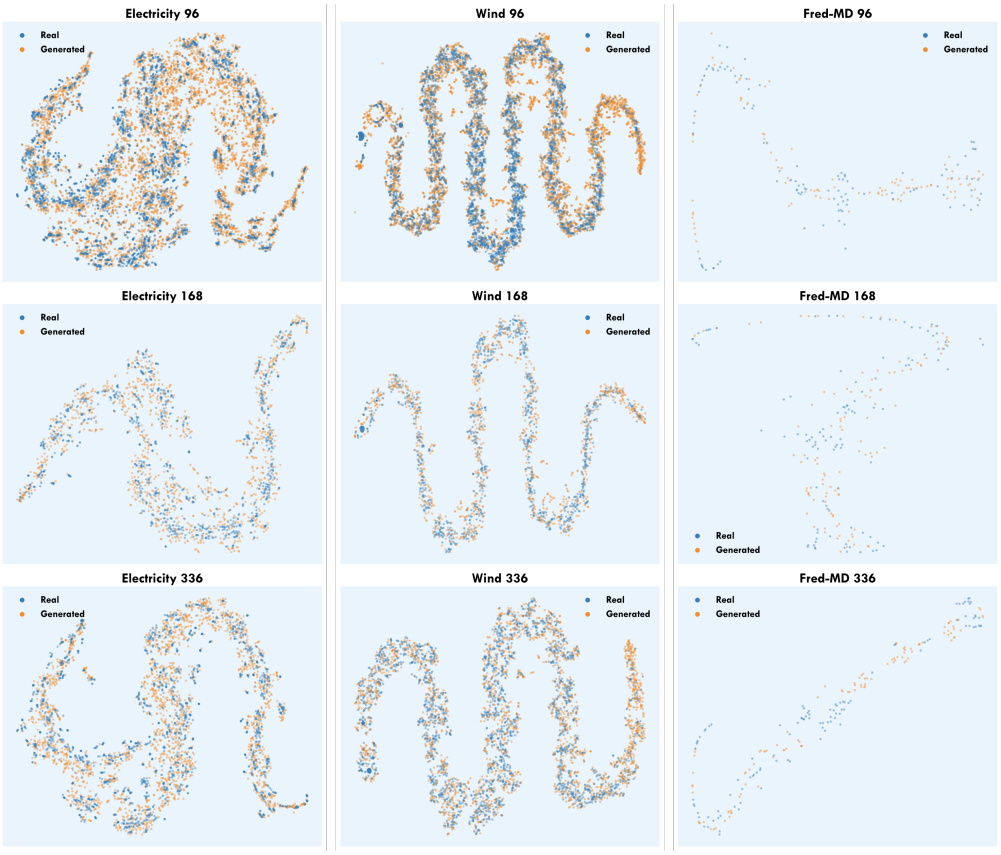


Figure 7: Visualization of synthesized time series by t-SNE across different sequence lengths.

Visualization of Synthesized Time Series We employ t-SNE to visualize the distributional similarity between real and generated samples, providing a qualitative assessment of generation fidelity, as shown in Figure 7. We observe a strong distributional alignment between the synthesized time-series samples and the real data. This demonstrates that HALO is capable of generating synthetic time series that faithfully preserve both the statistical properties and temporal structures of the original data.

Sensitivity Analysis of Autoregressive Generation Rounds To examine how the number of autoregressive generation rounds (K in Algorithm 1) affects generation quality, we conduct a sensitivity analysis. Table 8 reports the results with varying numbers of inference rounds. The results suggest that a more fine-grained autoregressive generation process may lead to higher-quality synthetic time series, potentially because it better preserves temporal dependencies during generation. This observation further supports our motivation: effectively balancing generation efficiency and temporal correlation modeling is key to achieving a desirable trade-off among generation stability, accuracy, and computational efficiency.

Table 8: Generation quality of HALO with different numbers of inference iterations K for sequence length 168. Results are reported as mean $\pm$ sample standard deviation over five random seeds. Best results are highlighted in bold face and second best results are underlined.

		$K=1$	$K=2$	$K=4$	$K=12$
MMD	Electricity	0.003 ± 0.000	0.003 ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000
	Solar	0.041 ± 0.000	**0.033** ± 0.001	$\underline{0.034 \pm 0.000}$	**0.033** ± 0.001
	Traffic	0.087 ± 0.001	0.083 ± 0.001	$\underline{0.072 \pm 0.001}$	**0.065** ± 0.001
K-L	Electricity	0.025 ± 0.001	0.059 ± 0.002	$\underline{0.020 \pm 0.002}$	**0.002** ± 0.001
	Solar	0.143 ± 0.007	0.092 ± 0.008	$\underline{0.035 \pm 0.002}$	**0.013** ± 0.000
	Traffic	0.013 ± 0.000	$\underline{0.006 \pm 0.000}$	$\underline{0.006 \pm 0.000}$	**0.002** ± 0.000
MDD	Electricity	0.008 ± 0.000	0.009 ± 0.000	$\underline{0.006 \pm 0.000}$	**0.003** ± 0.000
	Solar	82.188 ± 0.157	75.515 ± 0.152	$\underline{68.501 \pm 0.215}$	**58.702** ± 0.369
	Traffic	0.056 ± 0.000	0.056 ± 0.000	$\underline{0.055 \pm 0.000}$	**0.047** ± 0.000