DiM-TS: Bridge the Gap Between Selective State Space Models and Time Series for Generative Modeling

Zihao Yao[1], **Jiankai Zuo**[1], **Yaying Zhang**[1*]

[1]The Key Laboratory of Embedded System and Service Computing, Ministry of Education,
Tongji University, Shanghai 200092, China
{yaozihao, tj_zjk, yaying.zhang}@tongji.edu.cn

Abstract

Time series data plays a pivotal role in a wide variety of fields but faces challenges related to privacy concerns. Recently, synthesizing data via diffusion models is viewed as a promising solution. However, existing methods still struggle to capture long-range temporal dependencies and complex channel interrelations. In this research, we aim to utilize the sequence modeling capability of a State Space Model called Mamba to extend its applicability to time series data generation. We firstly analyze the core limitations in State Space Model, namely the lack of consideration for correlated temporal lag and channel permutation. Building upon the insight, we propose Lag Fusion Mamba and Permutation Scanning Mamba, which enhance the model's ability to discern significant patterns during the denoising process. Theoretical analysis reveals that both variants exhibit a unified matrix multiplication framework with the original Mamba, offering a deeper understanding of our method. Finally, we integrate two variants and introduce Diffusion Mamba for Time Series (DiM-TS), a high-quality time series generation model that better preserves the temporal periodicity and inter-channel correlations. Comprehensive experiments on public datasets demonstrate the superiority of DiM-TS in generating realistic time series while preserving diverse properties of data.

Code — https://github.com/yzh8221/DiMTS

Introduction

Time series data has been extensively applied in diverse domains for effective data analysis and prediction tasks, including finance, energy and climate (Godahewa et al. 2021). However, privacy concerns frequently hinder the data collection process, limiting the accessibility of real-world data (Alaa and Chan 2021). Additionally, in data-scarce domains like energy, the requirement for rich and high-quality datasets is challenging (Li et al. 2025). To address above issues, synthesizing realistic time series that closely resemble but do not replicate the original dataset has emerged as a promising solution, attracting increasing attention in recent years. Due to the superior training stability compared to GANs and higher-quality samples than VAEs, denoising diffusion probabilistic models (DDPMs) (Ho, Jain, and Abbeel 2020) have become the prevailing paradigm in generative modeling (Yuan and Qiao 2024).

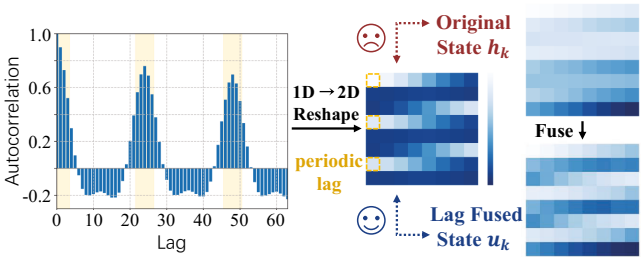


Figure 1: Comparison of ACF values, the latent state in original SSMs, and the lag fused state. We reshape them into a 2D format for clearer presentation. While the original latent state fails to capture periodic dependencies observed in ACF values, the lag fused state performs better in this regard.

Despite advancements, the Transformer-based architectures in most existing methods remain susceptible to noise (Huang et al. 2023), which may generate unrealistic distribution during the denoising process. Furthermore, the self-attention mechanism is inherently permutation-invariant (Zeng et al. 2023), leading the model to perform numerical approximation rather than capturing temporal dependencies. As a result, the synthetic samples suffer from low quality, as essential temporal properties are not explicitly preserved.

Meanwhile, State Space Models (SSMs) demonstrate great potential for long sequence modeling (Dao and Gu 2024). Among them, Mamba (Gu and Dao 2023) has recently gained popularity due to its selection mechanism that parameterize input tokens to filter out irrelevant information. Despite the inherent suitability of SSMs for modeling time series, their integration into generation task remains largely underexplored. It is mainly hindered by two key challenges.

(1) The lack of inductive bias for modeling correlated lags in temporal dimension. As shown in Figure 1, the autocorrelation function (ACF) typically exhibits high values at fixed periodic intervals, reflecting the similarity and dependency between the current time step and specific lag. While SSMs outperform Transformers in capturing temporal dynamics, the inherent unidirectional scanning paradigm inevitably ignore such correlations. The latent state of SSMs in Figure 1 reveals unidirectional attenuation along the tempo-

*Corresponding author.

ral scanning, contradicting the variation pattern of the ACF. The inconsistency disrupts the temporal semantics associated with periodicity, hindering the model's ability to preserve temporal dependencies during the denoising process.

(2) The difficulty of capturing complex variable interactions in channel dimension. Channel correlation is crucial for time series, as the modeling of a particular channel can be enhanced by leveraging information from related channels. However, global attention is susceptible to interference from irrelevant channels, hindering the recovery of lost inter-channel dependencies from noise. While Mamba with selection mechanism offering a potential solution, it tends to shift focus toward recent input (Xiao et al. 2024). As a result, highly correlated channels are insufficiently modeled if they are distant in scanning order. This highlights the need for effective time series channel permutation strategy.

In this study, we tackle the aforementioned challenges by presenting DiM-TS, a novel diffusion model that pioneers bridging the gap between selective SSMs and time series for generative modeling. It adopts an encoder-based dual-channel architecture to better capture time series properties across multiple dimensions. As the core technique, we propose Lag Fusion Mamba for temporal denoising and Permutation Scanning Mamba for channel denoising. The former fuses latent state of SSMs with correlated lags, introducing inductive bias to explicitly model periodicity while preserving the temporal dynamics as in Figure 1. The latter introduces a correlation-aware permutation strategy that leverages the attention shift of Mamba to enhance modeling between highly correlated channels. We further show that both modules and original Mamba can be represented within a unified structured matrix framework, offering a clearer conceptual understanding of our method. Additionally, we design a multi-feature loss to reconstruct samples rather than noises in each diffusion step, which encourage samples to approach realistic distribution from multiple perspectives.

Our contributions are summarized as follows.

- We present DiM-TS that better leverages the advantages of SSMs in time series generation. To the best of our knowledge, we are the first to bridge the gap between selective SSMs and time series for generative modeling.
- Motivated by the limitations of SSMs in modeling temporal dependencies and channel correlation, we propose two effective variants: Lag Fusion Mamba and Permutation Scanning Mamba. We further prove their unification with Mamba under the structured matrix framework.
- Experiments under challenge settings demonstrate that DiM-TS achieves superior performance in generating time series that preserve multiple key properties.

Related Work

Generative Models in Time Series

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Mogren 2016), which jointly optimize generator and discriminator, have been widely applied to time series generation (Jeha et al. 2022) but suffer from training instability. VAE-based models (Desai et al. 2021; Kingma and Welling 2022) enable fast and diverse sampling, yet often produce low-quality samples and struggle with KL divergence optimization (Jeong et al. 2025). Denoising diffusion probabilistic models (DDPMs) (Ho, Jain, and Abbeel 2020) emerge as a new class of generative framework and have demonstrated effectiveness in domains like images (Hu et al. 2024) and trajectory (Zhu et al. 2023). Recently, diffusion models have also been developed for time series. Diffusion-TS (Yuan and Qiao 2024) improves generalization and interpretability by disentangling temporal components such as trend and seasonality. PaD-TS (Li et al. 2025) explicitly considers time series population-level property preservation overlooked by previous approaches. Despite the advancements, Transformer architecture adopted by most methods are inherently time-invariant (Zeng et al. 2023). This hinders the preservation of essential temporal properties, thereby degrading the fidelity and quality of generated samples.

State Space Models

SSMs are mathematical framework depicting the system dynamic behavior over time (Rangapuram et al. 2018). LSSL (Gu et al. 2021) connects SSMs with Recurrent models and introduces the HiPPO (Gu et al. 2020) framework to handle long term dependencies. To mitigate resource scarcity issue, S4 (Gu, Goel, and Ré 2021) leverages structured SSMs to improve the efficiency and scalability. However, the linear time invariance formulation limits the context-awareness. To this end, Mamba (Gu and Dao 2023) introduces a hardware-efficient selection mechanism that filters noise and propagates relevant information by parameterizing the input. Researchers further adapt Mamba to domain-specific requirements. Spatial-Mamba (Xiao et al. 2024) utilizes dilated convolutions to capture image spatial structural dependencies. ZigMa (Hu et al. 2024) integrates a continuous scanning scheme with DDPMs for visual data generation.

Despite the effectiveness of Mamba that have demonstrated across various domains, its application for generative time series modeling remains unexplored. In this work, we aim to address the limitations of Mamba for time series data to fully exploit its potential and bridge this gap.

Preliminaries

We begin by presenting the definition of time series generation, then, we briefly review the formulations of DDPMs and SSMs. Please refer to Appendix B and D for details.

Problem Statement

Given observations of a multivariate time series dataset $\mathcal{D} = \{x_i\}_{i=1}^{M}$ with M samples. Each sample $x_i \in \mathbb{R}^{L \times C}$ is a multivariate time series, where L is the sequence length and C denotes the number of channels. Our unconditional generation goal is utilizing diffusion-based models to map Gaussian noise to a synthetic dataset $\mathcal{D}_{\text{syn}} = \{\bar{x}_i\}_{i=1}^{M}$ that approximates the distribution of original dataset $\mathcal{D}$.

Denoising Diffusion Probabilistic Models

Diffusion models are a type of generative model that contain forward process and reverse process. The forward process is

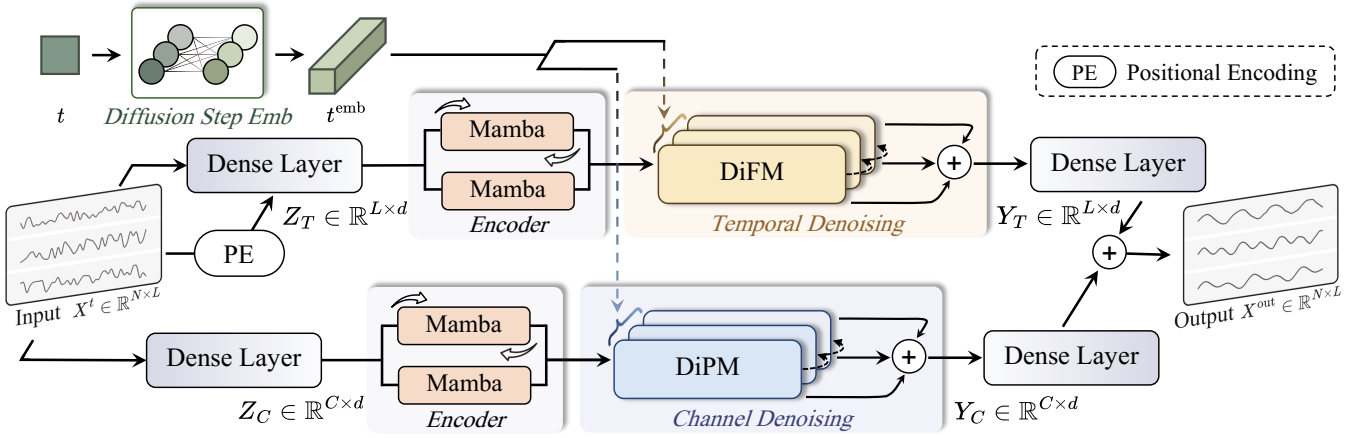


Figure 2: Proposed DiM-TS framework. Diffusion State Fusion Mamba (DiFM) and Diffusion Scanning Permutation Mamba (DiPM) are tailored for temporal denoising and channel denoising during generation process, respectively.

a Markov process where a sample $x^0 \sim q(x)$ is gradually noised into standard Gaussian noise $x^T \sim \mathcal{N}(0, \mathbf{I})$ by incrementally adding noise at each diffusion step t:

$$q\left(\mathbf{x}^t \mid \mathbf{x}^{t-1}\right) = \mathcal{N}\left(\mathbf{x}^t; \sqrt{1-\beta^t}\mathbf{x}^{t-1}, \beta^t\mathbf{I}\right), \quad (1)$$

where $t \in [1, T]$, $\beta^t \in (0, 1)$. The reverse process gradually denoise samples via reverse transitions:

$$p_\theta\left(\mathbf{x}^{t-1} \mid \mathbf{x}^t\right) = \mathcal{N}\left(\mathbf{x}^{t-1}; \mu_\theta\left(\mathbf{x}^t, t\right), \Sigma_\theta\left(\mathbf{x}^t, t\right)\right), \quad (2)$$

where $\mu_\theta(\cdot)$ is a learnable parameter, $\sum_\theta(\cdot)$ is fixed as $\sigma_t^2\mathbf{I}$.

The reverse process can be reduced to learning a surrogate approximator to parameterize $\mu_\theta(x^t, t)$ for all t. Hence, the denoising model parameters θ are optimized by minimizing:

$$\mathcal{L}_0(\theta) = \sum_{t=1}^{T} \mathbb{E}_{q(x^t|x^0)} ||\mu(x^t, x^0) - \mu_\theta(x^t, t)||, \quad (3)$$

where $\mu(x^t, x^0)$ is the mean of posterior $q(x^{t-1}|x^0, x^t)$.

State Space Models

SSMs are typically linear time-invariant system mapping input $x(k) \in \mathbb{R}^H$ to $y(k) \in \mathbb{R}^H$ via latent state $h(k) \in \mathbb{R}^{N\times H}$. This dynamic system can be described by the linear state transition and observation equations as:

$$h'(k) = \mathbf{A}h(k) + \mathbf{B}x(k), \quad y(k) = \mathbf{C}h(k) + \mathbf{D}x(k). \quad (4)$$

$\mathbf{A} \in \mathbb{R}^{N\times N}$ is state transition matrix. $\mathbf{B} \in \mathbb{R}^{N\times 1}, \mathbf{C} \in \mathbb{R}^{1\times N}$ are projection parameters. $\mathbf{D} \in \mathbb{R}$ is typically omitted (assume $\mathbf{D} = 0$), as it can be viewed as a skip connection.

Due to the hardness of analytical solutions for solving Eq. (4), ZOH (Comanescu 2012) is applied to approximate the continuous-time SSMs into a discrete-time form. Given a parameter Δ, the discretized SSMs can be represented as:

$$h_k = \bar{\mathbf{A}}_k h_{k-1} + \bar{\mathbf{B}}_k x_k, \quad y_k = \mathbf{C}_k h_k, \quad (5)$$

where $\bar{\mathbf{A}} = \exp(\Delta\mathbf{A})$, $\bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}$. Since time-independent parameters lack content-aware representation, Mamba introduces the selective mechanism that makes $\mathbf{B}$, $\mathbf{C}$ and Δ depend on input x_t.

Methodology

Model Framework

The architecture of DiM-TS is depicted in Figure 2. It comprises two parts: temporal dependencies modeling and channel interactions modeling. Each part utilize an encoder-decoder module to capture time series patterns. The representations are subsequently fused to obtain the final output.

Embedding Given a diffusion step t and its corresponding noised time series $x^t \in \mathbb{R}^{L\times C}$, we obtain temporal first input $x_T \in \mathbb{R}^{L\times C}$ and channel first input $x_C \in \mathbb{R}^{C\times L}$ by permuting L and C separately. Then, x_T and x_C pass through linear dense layer to learn context representations:

$$z_T = (W_T^1 x_T + b_T^1) + \text{PE}, \quad z_C = W_C^1 x_C + b_C^1 \quad (6)$$

where $W_T^1, W_C^1, b_T^1, b_C^1$ are learnable parameters. PE denotes an additional positional encoding, where $\text{PE}_{pos,2i} = \sin(\frac{pos}{10000^{2i/d}})$, $\text{PE}_{pos,2i+1} = \cos(\frac{pos}{10000^{2i/d}})$.

Encoder Selection mechanism has demonstrated the effectiveness in filtering out irrelevant information. However, the unidirectional scanning paradigm can only incorporate preceding input. Here, we utilize two vanilla Mamba to form a bidirectional Mamba encoder layer Bi-Mamba$(\cdot)$ to extract relative features at each diffusion step:

$$Z_T = \text{Bi-Mamba}(z_T), \quad Z_C = \text{Bi-Mamba}(z_C). \quad (7)$$

Decoder Since DiT (Peebles and Xie 2023) has been validated as an effective diffusion framework in high throughput and condition incorporating, we mirror the backbone of DiT to devise Diffusion State Fusion Mamba (DiFM) and Diffusion Scanning Permutation Mamba (DiPM) as the final layers in DiM-TS (see Appendix C for details). The diffusion timestep t, serving as conditional information, is transformed to t^{emb} via dense layers and then incorporated into each denoising layer. Given the encoded Z_T from previous section, the generation process can be formally described as:

$$\begin{aligned} Y_T^i =& \mathbb{1}_{i=0}\text{DiFM}(Z_T, t^{\text{emb}}) + \mathbb{1}_{i=1}\text{DiFM}(Y_T^0 + Z_T, t^{\text{emb}}) \\ &+ \mathbb{1}_{i>0}\text{DiFM}(Y_T^{i-1} + Y_T^{i-2}, t^{\text{emb}}), \end{aligned} \quad (8)$$

where Y_T^i is the output of i^{th} DiFM block. $\mathbb{1}_c$ represents the indicator function, which evaluates to 1 if the condition c holds, and 0 otherwise. Subsequently, the temporal representation Y_T can be learned by adding the output of all DiFM blocks. For the channel dimension, we simply replace Z_T with Z_C and apply the same procedure to obtain Y_C.

Eventually, we convert the temporal and channel representation to their original shape with dense layers, and obtain the final output through summation:

$$x^{\text{out}}(x^t, t, \theta) = (W_T^2 Y_T + b_T^2) + (W_C^2 Y_C + b_C^2), \quad (9)$$

where $W_T^2, W_C^2, b_T^2, b_C^2$ are learnable parameters.

Training Objective

In the reverse denoising process, the model is trained to generate time series via the following objective function:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t,x_0}[\|x^0 - x^{\text{out}}(x^t, t, \theta)\|^2]. \quad (10)$$

However, the training loss solely focuses on the authenticity of data at the individual level, neglecting higher-level statistical properties (Li et al. 2025). For instance, traffic flow typically exhibits periodic peak patterns, and weather data often contains correlated fluctuations between pressure and humidity. To address this, we introduce additional multi-feature loss to guide the diffusion process.

Since most temporal information is localized on the low frequencies, imposing constraint in the frequency domain can enhance sample fidelity by preserving underlying temporal property (Crabbé et al. 2024). Fourier transform that converts time domain signal to frequency domain representation has proven to be an effective operation (Yuan and Qiao 2024). The Fourier-based auxiliary loss can be defined as:

$$\mathcal{L}_T = \|\mathcal{FFT}(x^0) - \mathcal{FFT}(x^{\text{out}}(x^t, t, \theta))\|^2, \quad (11)$$

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

Meanwhile, to capture the correlation distribution shift in channel dimension, we adopt Maximum Mean Discrepancy (MMD) inspired by (Li et al. 2025). By calculating all $P = \frac{C(C-1)}{2}$ pairwise channel correlation distribution shift, the regularization loss can be defined as:

$$\mathcal{L}_C = \frac{1}{P} \sum_{i=1}^{P} \text{MMD}(D^i(x^0), D^i(x^{\text{out}})), \quad (12)$$

D^i denotes the correlation distribution of i^{th} pair channels. We further employ the Same Diffusion Step Sampling strategy (Li et al. 2025) for reasonable distribution comparison.

Hence, the training objective can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{DDPM}} + \lambda_1 \mathcal{L}_T + \lambda_2 \mathcal{L}_C, \quad (13)$$

where λ_1 and λ_2 are hyperparameter to balance loss terms.

Enhanced Mamba Module

In this section, we present the proposed Lag Fusion Mamba and Permutation Scanning Mamba. They are incorporated as core components into DiFM and DiPM, respectively.

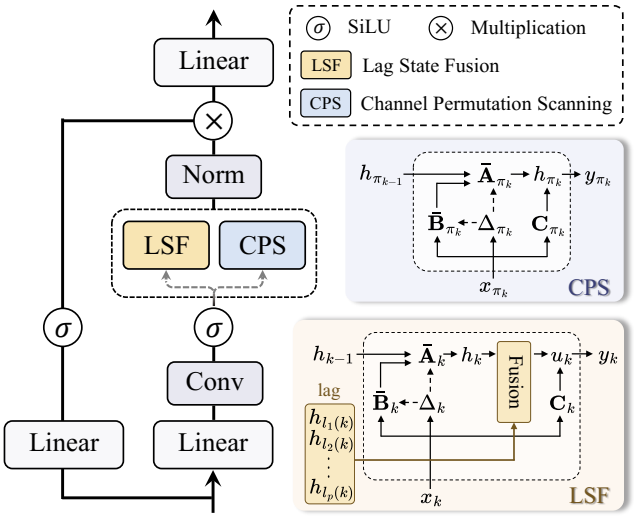


Figure 3: Architecture of proposed Mamba variants. Lag Fusion Mamba employs the LSF equation, while Permutation Scanning Mamba adopts the CPS equation.

Lag Fusion Mamba It is designed to capture the temporal dependencies of lag inspired by the characteristic of autocorrelation. Here, we introduce Lag State Fusion (LSF) equation into SSMs formula as shown in Figure 3. It fuses lag features in latent state space without disrupting the inherent sequential nature of time series and facilitates cross-temporal modeling. The process is formulated as:

$$\begin{aligned} h_k &= \bar{\mathbf{A}}_k h_{k-1} + \bar{\mathbf{B}}_k x_k, \\ u_k &= \sum_{p \in \Omega} \eta_p h_{l_p(k)}, \quad y_k = \mathbf{C}_k u_k, \end{aligned} \quad (14)$$

where h_k is the original latent state, u_k is the lag fusion state, Ω is the lag set, η_p is a learnable weight, and $l_p(k)$ is the index of the p^{th} lag of position k. Compared with the original Mamba where the state h_k is solely influenced by previous state h_{k-1}, the state u_k is combined with additional lag state through linear weighting fusion, resulting in a richer representation of both local and long-term temporal semantics. Moreover, an inductive bias of correlated lags is introduced to latent state prior to projecting, which enables DiFM capable of capturing periodic dependencies from noised data. Specifically, when the lag set Ω contains only the current state h_k, the formulation reduces to the original Mamba.

In practice, considering the regular time intervals and fixed lag, we implement linear weighted state fusion via multi-scale dilated convolutions. The 1D state sequence is first reshaped into 2D and processed by depth-wise convolutions with varying dilation factors defined by Ω. Finally, the fused 2D state is flattened to generate the output.

Permutation Scanning Mamba It aims to scan time series channels using a coherent permutation. To this end, we augment SSMs by introducing Channel Permutation Scanning (CPS) equation as depicted in Figure 3. It rearranges tokens before feeding into Eq. (5). For a given permutation

$\pi = \{\pi_1, \pi_2, \ldots, \pi_C\}$, the process can be expressed as:

$$h_{\pi_k} = \bar{\mathbf{A}}_{\pi_k} h_{\pi_{k-1}} + \bar{\mathbf{B}}_{\pi_k} x_{\pi_k}, \quad y_{\pi_k} = \mathbf{C}_{\pi_k} h_{\pi_k}, \tag{15}$$

where π_k denotes the index of the k^{th} scanning token.

Since inter-channel correlations reflect consistent variation patterns, preserving the proximity of highly correlated channels during scanning facilitates more accurate noise estimation and synthetic data with realistic inter-channel dependencies. Based on the observation, we propose a permutation strategy that keep related channels adjacent while separating unrelated channels apart during scanning. We take channel similarity matrix $\mathbf{G}$ derived from arbitrary metric (e.g., Pearson) as input. Each channel can be represented by $v \in \mathbb{R}$ for sequence order, $g_{ij} \in \mathbf{G}$ represents the closeness between channel i and j. To approximate the difference between v_i and v_j with g_{ij}, we optimize following function:

$$\min \sum_{i=1}^{C} \sum_{j=1}^{C} \|v_i - v_j\|^2 g_{ij}. \tag{16}$$

After obtaining vector $V = (v_1, v_2, \ldots, v_C)$, the ordered vector $V_\pi = (v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_C})$ can be generated by sorting element values. Since the value of correlated channels are numerically close, their permutation in V_π is also adjacent, thus yielding the desired permutation $\pi = \{\pi_1, \ldots, \pi_C\}$.

Channel order rearrangement can be implemented by transformation matrix H defined on π, where $H_{ij} = \mathbb{1}_{i=\pi_j}$. Specially, when $\pi = \{1, 2, \ldots, C\}$, H reduces to the identity, and CPS coincides with the original SSMs.

In practice, the input x is first transformed into Hx. Following processing with Eq. (15), the output is standardized to the original permutation by inverse transformation H^{-1}.

Connection with Original Mamba

In this section, we conduct an in-depth analysis of the interrelations among original Mamba and proposed variants, aiming to elucidate the underlying mechanisms of our approach (see Appendix D for more detailed derivations).

Mamba The formulation is defined in Eq. (5). We can derive the latent state h_t by induction:

$$h_k = \bar{\mathbf{A}}_k \ldots \bar{\mathbf{A}}_1 h_0 + \ldots + \bar{\mathbf{A}}_k \bar{\mathbf{B}}_{k-1} x_{k-1} + \bar{\mathbf{B}}_k x_k. \tag{17}$$

By multiplying with $\mathbf{C}_k$ to produce y_k, vectorizing the equation over k, and setting the initial latent state $h_0 = \bar{\mathbf{B}}_0 x_0$, we establish the matrix transformation form of SSMs:

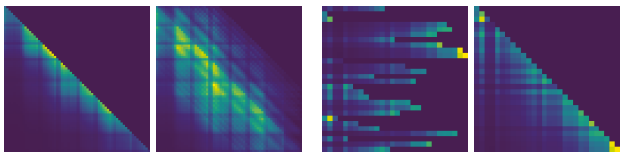
$$y_k = \sum_{i=0}^{k} \mathbf{C}_k^\top \bar{\mathbf{A}}_{i:k}^{\times} \bar{\mathbf{B}}_i x_i, \quad y = \text{SSM}(x) = \mathbf{M}x, \tag{18}$$

where $\mathbf{M}$ is a lower triangular matrix, $\mathbf{M}_{ki} := \mathbf{C}_k^\top \bar{\mathbf{A}}_{i:k}^{\times} \bar{\mathbf{B}}_i$, $\bar{\mathbf{A}}_{i:k}^{\times} := \Pi_{j=i+1}^{k} \bar{\mathbf{A}}_j$ denotes the product of the state transition matrices indexed from $i+1$ to k for $i < k$, and is defined as identity matrix when $i = k$.

Lag Fusion Mamba Under the same setting as Eq. (14) and Eq. (18), the fused latent state and corresponding output in LSF equation can be reformulated as follows:

$$u_k = \sum_{p \in \Omega} \sum_{i \le l_p(k)} \eta_p \bar{\mathbf{A}}_{i:l_p(k)}^{\times} \bar{\mathbf{B}}_i x_i, \; y = \text{LSF}(x) = \mathbf{M}^F x, \tag{19}$$

$\mathbf{M}^F$ is adjacency matrix, $\mathbf{M}_{ki}^F := \sum_{p \in \Omega} \eta_p \mathbf{C}_k^\top \bar{\mathbf{A}}_{i:l_p(k)}^{\times} \bar{\mathbf{B}}_i$.



(a) $\mathbf{M}$ (left) and $\mathbf{M}^F$ (right). (b) $\mathbf{M}^C$ (left) and $\mathbf{M}^C H$ (right).

Figure 4: Matrix Visualizations. $\mathbf{M}$, $\mathbf{M}^F$, $\mathbf{M}^C$ denotes the matrices in Mamba, Lag Fusion Mamba and Permutation Scanning Mamba, respectively. H is transformation matrix.

Permutation Scanning Mamba Given the specific H, CPS can be rewritten based on Eq. (15) and Eq. (18):

$$y = \text{CPS}(x) = H^{-1}\text{SSM}(Hx) = \mathbf{M}^C x, \tag{20}$$

where $\mathbf{M}^C$ can be transformed into a lower triangular matrix through row interchange, $\mathbf{M}^C := H^{-1}\mathbf{M}H$.

Analysis Based on the above derivation, we can conclude that all the paradigms — Mamba, Lag Fusion Mamba, and Permutation Scanning Mamba — can be modeled within a unified matrix multiplication framework, *i.e.*, $y = \mathbf{M}x$. Their distinctions arise from the structural differences of the matrix $\mathbf{M}$. As illustrated in Figure 4, the matrix $\mathbf{M}$ in Mamba exhibits a decaying pattern over scanning. This effect arises from the cumulative multiplication of state transition matrix $\bar{\mathbf{A}}_k$, which leads to exponential decay in attention weights as the intervals between tokens increases. For Lag Fusion Mamba that fuses additional lag state set Ω via weighted summation, the resulting matrix $\mathbf{M}^F$ not only extents unidirectional time series modeling into a global scope, but also effectively capture high-correlated lag interactions, even when they are far apart. Permutation Scanning Mamba, on the other hand, leverages the attention transition of Mamba while employing the channel permutation strategy. This makes model shift focus toward previously high-correlated channels and suppress attention to irrelevant ones.

Experiments

In this section, we describe the experiment settings and evaluate DiM-TS across various domains and sequence lengths. We also provide visualization results to enhance the understanding of our model behavior. Finally, we conduct an ablation study to assess the effectiveness of components.

Experiment Settings

We briefly discuss the datasets, baselines, and evaluation metrics. All experiments are conducted on a machine with NVIDIA V100 GPU and 32GB memory. Implementation details and code are provided in Supplementary Material.

Datasets We utilize four major public datasets spanning diverse domains, including finance, electricity, energy and environment. (1) Stocks: Daily stock data from Google (2004-2019) with six features such as Open, Volume, etc. (2) ETTh: Electricity transformer data collected hourly, including oil temperature and six power-related metrics. (3)

Metric	Methods	Stocks	ETTh	Energy	KDD-Cup
Context-FID score (Lower the Better)	DiM-TS	**0.0440±0.0074**	**0.0259±0.0021**	**0.0320±0.0003**	**0.0220±0.0029**
	PaD-TS	0.0715±0.0255	1.2674±0.1881	0.0657±0.0078	0.1372±0.0208
	Diffusion-TS	0.4055±0.0557	0.2570±0.0112	0.0708±0.0135	1.0141±0.1186
	FourierDiff	0.1294±0.0314	0.1198±0.0076	0.4477±0.0467	0.8294±0.1501
	TimeVAE	0.3892±0.1174	0.8995±0.1147	3.3228±0.2680	1.8987±0.2582
	TimeGAN	0.4182±0.1147	1.9650±0.3051	1.5532±0.1681	1.1560±0.3504
Correlational score (Lower the Better)	DiM-TS	**0.0048±0.0029**	**0.0219±0.0046**	**0.4108±0.1369**	**3.6615±1.2297**
	PaD-TS	0.0085±0.0080	0.1237±0.0017	0.5724±0.0827	6.7944±1.1817
	Diffusion-TS	0.0244±0.0053	0.0595±0.0053	0.6360±0.0877	9.4173±0.7498
	FourierDiff	0.0139±0.0079	0.0473±0.0090	1.1992±0.2587	15.6568±2.0939
	TimeVAE	0.0859±0.0048	0.0593±0.0192	2.1681±0.1034	30.7528±1.5355
	TimeGAN	0.0059±0.0033	0.2175±0.0084	3.5817±0.1221	16.6840±1.5923
Discriminative Score (Lower the Better)	DiM-TS	**0.0291±0.0151**	**0.0053±0.0019**	0.2410±0.0201	**0.0844±0.0233**
	PaD-TS	0.0485±0.0792	0.1576±0.0137	**0.0919±0.0193**	0.3769±0.0460
	Diffusion-TS	0.0910±0.0237	0.0832±0.0067	0.1072±0.0162	0.2957±0.0168
	FourierDiff	0.0553±0.0587	0.0446±0.0074	0.2062±0.0339	0.4833±0.0040
	TimeVAE	0.1794±0.0801	0.1739±0.0935	0.4999±0.0001	0.4639±0.0100
	TimeGAN	0.2013±0.0712	0.3228±0.0738	0.4995±0.0004	0.4988±0.0008
Predictive score (Lower the Better)	DiM-TS	**0.0367±0.0001**	**0.1086±0.0101**	**0.2474±0.0004**	**0.0241±0.0003**
	PaD-TS	0.0368±0.0001	0.1180±0.0018	0.2514±0.0002	0.0282±0.0001
	Diffusion-TS	0.0368±0.0001	0.1173±0.0057	0.2490±0.0005	0.0324±0.0017
	FourierDiff	**0.0367±0.0001**	0.1171±0.0070	0.2508±0.0001	0.0285±0.0005
	TimeVAE	0.0385±0.0003	0.1200±0.0044	0.2888±0.0008	0.0290±0.0003
	TimeGAN	0.0505±0.0007	0.1450±0.0046	0.3129±0.0021	0.0368±0.0004

Table 1: Generation results with length 64 on multiple datasets. The best scores are in bold and the second best are underlined.

Energy: A UCI appliances energy prediction dataset with 28 features related to household energy consumption. (4) KDD-Cup: Hourly air quality from 2017 to 2018 estimated by 24 stations in London. More details and additional Traffic dataset results are available in Appendix F.

Baselines We carefully select five competitive models that cover generative frameworks: (1) Diffusion-based models: PaD-TS (Li et al. 2025), Diffusion-TS (Yuan and Qiao 2024), FourierDiff (Crabbé et al. 2024). (2) VAE-based model: TimeVAE (Desai et al. 2021). (3) GAN-based model: TimeGAN (Yoon, Jarrett, and Van der Schaar 2019).

Metrics The quantitative evaluation of the synthesized data is conducted from three key aspects: 1) the distribution diversity of time series. 2) the fidelity of temporal and channel dependencies. 3) the usefulness in downstream application. We employ the following evaluation metrics (Yuan and Qiao 2024): (1) Context-Fréchet Inception Distance score (Context-FID score): Computes the difference between representations of real and generated data fitting into local context. (2) Correlational score: Assess temporal dependency by absolute error between cross correlation matrices by real and generated time series. (3) Discriminative score: Evaluates the similarity between original and generated data based on distinguishability assessed via a supervised classification model. (4) Predictive score: Measures the usefulness of generated data by capturing Mean Absolute Error of a time series forecasting model trained on generated data. We additionally include feature-based metrics summarized in (Ang et al. 2023): Marginal Distribution Difference (MDD), AutoCorrelation Difference (ACD), Skewness Difference (SD), Kurtosis Difference (KD). We also adopt population-level metrics from (Li et al. 2025): Value distribution shift (VDS) and Functional dependency distribution shift (FDDS). For calculation formulas, please refer to Appendix E.

Baselines Comparison

Main Results We list the results of 64-length time series generation in Table 1. Among the baselines concerned, DiM-TS achieves the best performance on most datasets across various metrics. It demonstrates the superiority of our method in generating high-quality synthetic time series. Notably, DiM-TS improves the context-FID score over 60% and the correlation score over 35% compared to previous state-of-the-art models. Moreover, the predictive score indicates that the synthetic data generated by DiM-TS is more applicable to real-world task. As shown in Figure 5, DiM-TS achieves overall superior performance under feature-based metrics and population-level property preservation settings. The observations substantiate the capability of DiM-TS in synthesizing high-fidelity time series.

Visualization To provide an intuitive understanding of model behavior, we employ the t-SNE (Van der Maaten and Hinton 2008) and kernel density estimation (Węglarczyk 2018) to visualize the fidelity of generated data. As shown in

Metrics	Length	DiM-TS	PaD-TS	Diffusion-TS	FourierDiff	TimeVAE
Context-FID score	128	**0.0451±0.0025**	1.4856±0.2231	0.7100±0.0624	0.4753±0.0262	0.7571±0.0895
	256	**0.0516±0.0028**	1.8520±0.2971	1.7604±0.0848	1.0262±0.0838	1.3814±0.1354
Correlational score	128	**0.0215±0.0058**	0.1171±0.0117	0.0890±0.0046	0.0855±0.0126	0.0556±0.0098
	256	**0.0225±0.0070**	0.1357±0.0038	0.1144±0.0129	0.0916±0.0050	0.0444±0.0104
Discriminative score	128	**0.0033±0.0017**	0.1707±0.0375	0.1436±0.0099	0.1619±0.0126	0.1835±0.0982
	256	**0.0044±0.0047**	0.1979±0.0429	0.2103±0.0131	0.1939±0.1274	0.1984±0.0909
Predictive score	128	**0.1115±0.0088**	0.1270±0.0054	0.1122±0.0027	0.1175±0.0054	0.1151±0.0121
	256	**0.1050±0.0099**	0.1106±0.0088	0.1171±0.0054	0.1150±0.0037	0.1163±0.0059

Table 2: Results of long-term time series generation on ETTh dataset. The best scores are in bold.

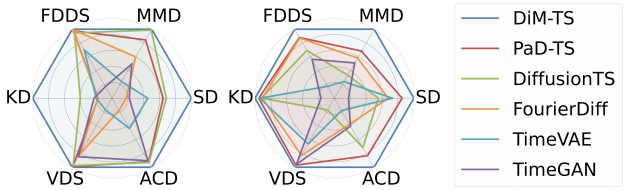


Figure 5: Feature-based and population-level measures comparison on Energy (left) and KDD-Cup (right).

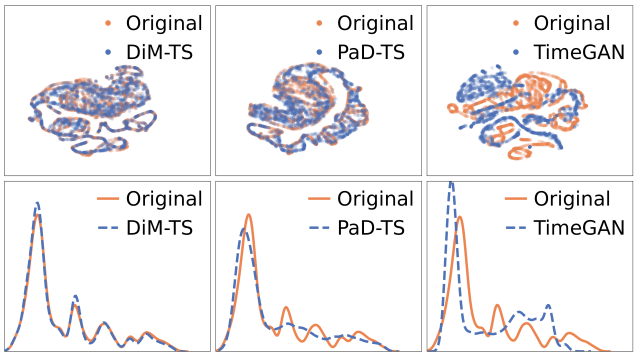


Figure 6: Visualizations of t-SNE plots and data distributions on original (orange) and synthetic (blue) time series.

the 1^{st} row in Figure 6, the 2D projection of DiM-TS using t-SNE exhibits diversity and closer alignment with the original data, whereas other methods either fail to achieve comprehensive coverage or produce unrealistic samples. The 2^{nd} row in Figure 6 shows that the synthetic time series value distribution of DiM-TS is the most similar to the original data. The results indicate that DiM-TS effectively learns the underlying statistical properties.

Long sequence generation We further verify model stability in longer sequence generation task with lengths of 128 and 256. As shown in Table 2, DiM-TS achieves the best performance under the challenging setting, implying the efficacy of improved components tailored for time series. Notably, the performance of DiM-TS changes steadily as the sequence length increases, demonstrating superior robustness which is meaningful for real-world applications.

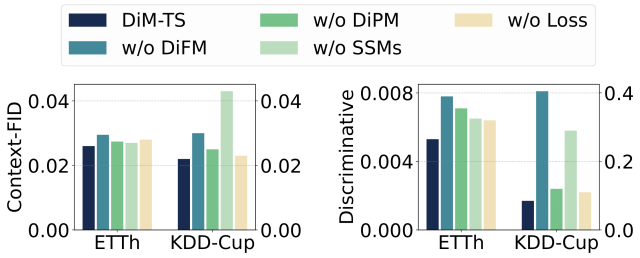


Figure 7: The results of ablation study.

Ablation Study

To validate the effectiveness of components, we set up variant models for ablation experiments: (1) w/o DiFM: The lag state fusion is removed in DiFM, i.e., it is replaced by original Mamba. (2) w/o DiPM: The channel permutation strategy is removed in DiPM. (3) w/o SSMs: This variant uses DiT to replace DiFM and DiPM. (4) w/o Loss: The multi-feature loss terms $\mathcal{L}_T$ and $\mathcal{L}_C$ are omitted during training.

As shown in Figure 7, the results highlight that Lag Fusion Mamba is the most critical part in our method, demonstrating the effectiveness of incorporating the inductive bias of correlated lags in enhancing the temporal dependency awareness of SSMs. While the Diffusion Permutation Mamba and auxiliary loss terms contribute less significantly, they still play crucial roles. Overall, integrating all components, the full DiM-TS achieves the best performance.

Conclusion

In this paper, we present DiM-TS, a novel framework that empower selective state space models for time series generation. As key contributions, we propose a Lag Fusion Mamba designed for modeling temporal dependencies, and a Permutation Scanning Mamba tailored to capturing channel correlation during the denoising process. We further provide an in-depth analysis between the proposed variants and original Mamba, demonstrating their unification under the matrix multiplication framework and offering deeper insights into our approach. Extensive experiments show that DiM-TS excels at synthesizing high-quality time series while preserving multiple properties across various settings.

Acknowledgments

This work was partly supported by the National Key Research and Development Program of China under Grant 2022YFB4501704, the National Natural Science Foundation of China under Grant 72342026, and Fundamental Research Funds for the Central Universities under Grant 2024-6-ZD-02.

References

Alaa, A.; and Chan, A. J. 2021. Generative time-series modeling with fourier flows. In *International Conference on Learning Representations*.

Ang, Y.; Huang, Q.; Bao, Y.; Tung, A. K.; and Huang, Z. 2023. Tsgbench: Time series generation benchmark. *arXiv preprint arXiv:2309.03755*.

Comanescu, M. 2012. Integration of observer equations used in AC motor drives by zero and First Order Hold discretization. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, 3694–3698. IEEE.

Crabbé, J.; Huynh, N.; Stanczuk, J.; and Van Der Schaar, M. 2024. Time series diffusion in the frequency domain. *arXiv preprint arXiv:2402.05933*.

Dao, T.; and Gu, A. 2024. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, 10041–10071.

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

Godahewa, R.; Bergmeir, C.; Webb, G. I.; Hyndman, R. J.; and Montero-Manso, P. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; and Ré, C. 2020. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hu, V. T.; Baumann, S. A.; Gui, M.; Grebenkova, O.; Ma, P.; Fischer, J.; and Ommer, B. 2024. Zigma: A dit-style zigzag mamba diffusion model. In *European Conference on Computer Vision*, 148–166. Springer.

Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; and Wang, Y. 2023. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36: 46885–46902.

Jeha, P.; Bohlke-Schneider, M.; Mercado, P.; Kapoor, S.; Nirwan, R. S.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2022. PSA-GAN: Progressive self attention GANs for synthetic time series. In *The Tenth International Conference on Learning Representations*.

Jeong, S.; Sohn, J.; Jeon, J.; Shon, Y.; and Suk, H.-I. 2025. Frequency-Conditioned Diffusion Models for Time Series Generation.

Kingma, D. P.; and Welling, M. 2022. Auto-Encoding Variational Bayes. *stat*, 1050: 10.

Li, Y.; Meng, H.; Bi, Z.; Urnes, I. T.; and Chen, H. 2025. Population Aware Diffusion for Time Series Generation. *arXiv preprint arXiv:2501.00910*.

Mogren, O. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Rangapuram, S. S.; Seeger, M. W.; Gasthaus, J.; Stella, L.; Wang, Y.; and Januschowski, T. 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Węglarczyk, S. 2018. Kernel density estimation and its application. In *ITM web of conferences*, volume 23, 00037. EDP Sciences.

Xiao, C.; Li, M.; Zhang, Z.; Meng, D.; and Zhang, L. 2024. Spatial-Mamba: Effective Visual State Space Models via Structure-Aware State Fusion. *arXiv preprint arXiv:2410.15091*.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Yuan, X.; and Qiao, Y. 2024. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhu, Y.; Ye, Y.; Zhang, S.; Zhao, X.; and Yu, J. 2023. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems*, 36: 65168–65188.